

Iterative Method for Image Compression by Using LSPIA

Lijuan Hu, Yeqing Yi, Chengzhi Liu, and Juncheng Li

Abstract—In this paper, the curve fitting by cubic B-spline curves is used to compress gray-level images. The fitting tool is the progressive and iterative approximation method for least square fitting (LSPIA), which is used to approximate scanned image data. Different from the existing methods by using piecewise curves, the image data are fitted by a single curve. Hence it can well preserve the relative information between neighborhood pixels. In particular, to reduce the compression ratio, we further exploit some techniques to save storage space. Numerical implements show that the proposed method outperforms the existing methods by using fitting curves.

Index Terms—image compression, LSPIA, cubic B-spline, Hilbert scan, curve fitting.

I. INTRODUCTION

IN digital image processing, many image files with large amount of data are generated. These files usually need to be stored or transmitted. Therefore, an effective image compression method is often required to store and transmit these files. Digital images can be compressed because they often contain redundant and unrelated information. The task of image compression is to eliminate these redundant and unrelated information. Despite the fact that there exist a large amount of image compression techniques, further research is needed to meet the continuous improvements for more efficient compression methods. Therefore, the image compression remains a hot topic in image processing, on which various compression methods have been presented, see [1], [2], [3].

Roughly speaking, there exist two kinds of methods for image compression [3]. One is lossless compression, which only removes the redundant information of the image, such as medical images and fingerprint images, and so on. We can accurately restore the original image after decompression without any distortion [3], e.g., Huffman coding, arithmetic coding. However, it is known that the compression ratio (CR) is limited by the theory of statistical redundancy. Another is lossy compression, which can obtain high CR by reducing the quality of images, e.g., singular value decomposition [3], transform coding [3], predictive coding [3], curve fitting [4], [5] and so on. For more detailed research on image compression techniques, we refer the reader to read recent surveys ([1], [2], [3]).

Manuscript received June 21, 2020; revised August 28, 2020. This work was supported in part by the Natural Science Foundation of Hunan Province (Grant No. 2017JJ3124 and 2020JJ5267) and the Scientific Research Funds of Hunan Provincial Education Department (Grant No. 18C877).

L. Hu and Y. Yi are with School of Information, Hunan University of Humanities, Science and Technology, Loudi 417000, P.R. China, e-mail: yeqingyi@126.com(Yi) and 1023236794@qq.com(Hu).

C. Liu and J. Li are with School of Mathematics and Finance, Hunan University of Humanities, Science and Technology, Loudi 417000, P.R. China, e-mail:162101002@csu.edu.cn(Liu, Corresponding Author) and lijuncheng82@126.com(Li).

Curve fitting plays an important role in computer aided geometry design (CAGD), image processing, shape modeling and data mining. There are some works involving with curve fitting in image compression [4], [5], [6], [7], [8], [9], [11]. It is known that the data of a scanned image are very large, it may bring a large amount of computational cost and cause computational instability when fitting by higher order polynomials or non-polynomials. Hence we should find suitable fitting curves and efficient solvers to fit these data. Due to these reasons, piecewise curves are suggested in fitting these scanned data. For example, piecewise Bernstein polynomials of degree 2 [8], [9]; piecewise quasi cubic rational Bézier curve [10]; trigonometric Bézier curve [11] and so on. Despite the fact that fitting by piecewise curves has good fitting performance and is easy to operate, there are also drawbacks. For instance, the continuity condition between two adjacent curve segments can not be guaranteed, hence it may loss the relative information of the neighborhood pixels.

Very often, it is necessary to solve a linear system in traditional curve fitting methods. The system can be solved by direct solvers [12], e.g., Cholesky factorization, Gauss elimination with partial pivot and QR factorization. However, it is known that the solutions solved by direct solvers may be loss of accuracy and instability when the condition number of the coefficient matrices are very large [12]. We note in [13] that for large scale problems, the iterative methods would be good choices. In 2014, an efficient and robust iterative fitting method, named LSPIA, was proposed by Deng et al. [14]. It has the property of shape preserving and can deal with large data. Moreover, we can adjust the control points until the fitting error satisfies a given tolerance. These elegant merits make convenience in using the LSPIA in curve fitting. This naturally arises a question that whether we can apply the LSPIA to fit the image data and thus compress the image. In this paper, we study the application of LSPIA in image compression by using cubic B-spline curves.

This paper is organized as follows. In section II we review the least square fitting method and the methodology of LSPIA. The image compression algorithm is presented in Section III. Some image compression examples are given to illustrate the effectiveness of the proposed method in Section IV. Some improvement are presented in Section V. A brief conclusion is drawn in Section VI.

II. CURVE FITTING BY USING CUBIC B-SPLINE

A. Least square fitting curve

Let us first recall some details for the conventional curve-fitting method in CAGD.

Given a data points $\{\mathbf{Q}_j\}_{j=0}^m$ to be fitted, and t_j be the parameters of \mathbf{Q}_j such that $0 = t_0 < t_1 < \dots < t_m < 1$.

The aim of curve fitting is to find a cubic B-spline curve

$$\mathbf{r}(t) = \sum_{i=0}^n \mathbf{P}_i B_{i,3}(t), \quad (1)$$

that approximates the points $\{\mathbf{Q}_j\}_{j=0}^m$ best. Here the $\{\mathbf{P}_i\}_{i=0}^n$ in (1) are the control points to be determined, $B_{i,3}(t)$ are the cubic B-spline basic functions defined at the knot vector $\{0 = u_0 = u_1 = u_2 = u_3 < u_4 < u_5 < \dots < u_n < u_{n+1} = u_{n+2} = u_{n+3} = u_{n+4} = 1\}$, in details

$$B_{i,0}(t) = \begin{cases} 0 & \text{if } u_i \leq t \leq u_{i+1}; \\ 1 & \text{otherwise.} \end{cases}$$

$$B_{i,p}(t) = \frac{t - u_i}{u_{p+1} - u_i} B_{i,p-1}(t) + \frac{u_{i+p+1} - t}{u_{i+p+1} - u_{i+1}} B_{i+1,p-1}(t),$$

$$p = 1, 2, 3.$$

For simplicity, we denote the cubic B-spline basic functions $B_{i,3}(t)$ by $B_i(t)$ in this paper. Oftentimes, the number of the control points is less than that of the points to be fitted, i.e., $n < m$.

The main idea of the least square fitting (LSF) method is to find an optimal control polygon $\{\mathbf{P}_i\}_{i=0}^n$ that minimizes the distances between $\mathbf{r}(t)$ and $\{\mathbf{Q}_j\}_{j=0}^m$, i.e.,

$$\min_{\mathbf{P}_0, \mathbf{P}_1, \dots, \mathbf{P}_n} f(\mathbf{P}_0, \mathbf{P}_1, \dots, \mathbf{P}_n) = \min_{\mathbf{P}_i} \sum_{j=0}^m \|\mathbf{Q}_j - \mathbf{r}(t_j)\|^2$$

$$= \min_{\mathbf{P}_i} \sum_{j=0}^m \left\| \mathbf{Q}_j - \sum_{i=0}^n \mathbf{P}_i B_i(t_j) \right\|^2. \quad (2)$$

The norm in (2) is the Euclidean norm.

The optimal curve $\mathbf{r}(t)$ obtained by solving (2) is said to be the LSF curve of $\{\mathbf{Q}_j\}_{j=0}^m$. To minimize $f(\mathbf{P}_0, \mathbf{P}_1, \dots, \mathbf{P}_n)$, set the gradient of f to zero, i.e.,

$$\frac{\partial f}{\partial \mathbf{P}_i} = -2 \sum_{j=0}^m B_i(t_j) \|\mathbf{Q}_j - \sum_{i=0}^n \mathbf{P}_i B_i(t_j)\| = 0,$$

$$i = 0, 1, \dots, n.$$

Hence, we have

$$\mathbf{Q}_j - \sum_{i=0}^n \mathbf{P}_i B_i(t_j) = 0, j = 0, 1, \dots, m. \quad (3)$$

Let $\mathbf{P} = [\mathbf{P}_0, \mathbf{P}_1, \dots, \mathbf{P}_n]^T$ and $\mathbf{Q} = [\mathbf{Q}_0, \mathbf{Q}_1, \dots, \mathbf{Q}_m]^T$. Then the equations (3) can be written in the matrix form

$$\mathbf{B}\mathbf{P} = \mathbf{Q}, \quad (4)$$

where the matrix $B = (B_i(t_j))_{i=0,1,\dots,n}^{j=0,1,\dots,m}$ is the so-called collocation matrix resulting from the cubic B-spline basis. Therefore the control polygon $\{\mathbf{P}_i\}_{i=0}^n$ can be obtained by solving the linear system (4). Since $n < m$, the system (4) is over-determined and can be solved by solving the related system of normal equations, i.e.,

$$\mathbf{B}^T \mathbf{B} \mathbf{P} = \mathbf{B}^T \mathbf{Q}. \quad (5)$$

As mentioned earlier, the system (5) can be solved by direct solvers or iterative methods. In the following subsection, we will introduce an iterative method for curve fitting with clear geometric meaning.

In curve fitting, we need to measure the fitting error. Let $\{\mathbf{Q}_j\}_{j=0}^m$ be the points to be fitted and t_j be their corresponding parameters. Then we use

$$\varepsilon = \sqrt{\frac{1}{m+1} \sum_{j=0}^m \|\mathbf{Q}_j - \mathbf{r}(t_j)\|^2}. \quad (6)$$

to represent the fitting error of the fitting curve $\mathbf{r}(t)$.

B. LSPIA by using cubic B-spline

Given an ordered points set $\{\mathbf{Q}_j\}_{j=0}^m$ to be fitted, and t_j be the parameters of \mathbf{Q}_j such that $0 = t_0 < t_1 < \dots < t_m < 1$.

Firstly, we select $\{\mathbf{P}_i^{(0)}\}_{i=0}^n$ from $\{\mathbf{Q}_j\}_{j=0}^m$ as the initial control points and construct the initial approximate fitting curve

$$\mathbf{r}^{(0)}(t) = \sum_{i=0}^n \mathbf{P}_i^{(0)} B_i(t).$$

Let $\delta_j^{(0)} = \mathbf{Q}_j - \mathbf{r}^{(0)}(t_j)$, $j = 0, 1, \dots, m$. Then the first adjusting vector for the i -th ($i = 0, 1, \dots, n$) control point is given by

$$\Delta_i^{(0)} = \mu \sum_{j=0}^m B_i(t_j) \delta_j^{(0)},$$

where $\mu \in (0, 2/\lambda_0)$ is a constant, λ_0 represents the largest eigenvalue of $B^T B$.

Next, we can generate a new approximate fitting curve

$$\mathbf{r}^{(1)}(t) = \sum_{i=0}^n \mathbf{P}_i^{(1)} B_i(t),$$

where $\mathbf{P}_i^{(1)} = \mathbf{P}_i^{(0)} + \Delta_i^{(0)}$, $i = 0, 1, \dots, n$.

Suppose that we have obtained $(k-1)$ -th ($k = 1, 2, \dots$) curve $\mathbf{r}^{(k-1)}(t)$, then the k -th approximate fitting curve can be generated by

$$\mathbf{r}^{(k)}(t) = \sum_{i=0}^n \mathbf{P}_i^{(k)} B_i(t), \quad (7)$$

where

$$\begin{cases} \mathbf{P}_i^{(k)} = \mathbf{P}_i^{(k-1)} + \Delta_i^{(k-1)}, \\ \Delta_i^{(k-1)} = \mu \sum_{j=0}^m B_i(t_j) \delta_j^{(k-1)}, \\ \delta_j^{(k-1)} = \mathbf{Q}_j - \mathbf{r}^{(k-1)}(t_j). \end{cases} \quad (8)$$

Therefore, we get a sequence of curves $\mathbf{r}^{(k)}(t)$, $k = 0, 1, \dots$. The initial curve is said to have the LSPIA property if $\mathbf{r}^{(k)}(t)$ is convergent. The limit curve of $\mathbf{r}^{(k)}(t)$ is the LSF curve of $\{\mathbf{Q}_j\}_{j=0}^m$. Deng et al. proved that the B-spline curves have the LSPIA property [14].

Let $\mathbf{P}^{(k)} = [\mathbf{P}_0^{(k)}, \mathbf{P}_1^{(k)}, \dots, \mathbf{P}_n^{(k)}]^T$. Then the equations (8) can be written in the matrix form

$$\mathbf{P}^{(k+1)} = B^T \mathbf{Q} + (I - \mu B^T B) \mathbf{P}^{(k)}. \quad (9)$$

The LSPIA property means that the sequence of control polygons $\{\mathbf{P}_i^{(k)}\}_{i=0}^n$ converges to the control polygon of the LSF curve.

For LSPIA, the fitting error of the k -th approximate fitting curve $\mathbf{r}^{(k)}(t)$ is given by

$$\varepsilon^{(k)} = \sqrt{\frac{1}{m+1} \sum_{j=0}^m \|\mathbf{Q}_j - \mathbf{r}^{(k)}(t_j)\|^2}. \quad (10)$$

III. IMAGE COMPRESSION ALGORITHM

In this section, we employ the LSPIA for cubic B-spline curves to fit the image data. Curve fitting means to replace the original data points with less control points. Thus, we can reach the aim of data compression by LSPIA, and we illustrate in Fig. 1 the main process of image compression.

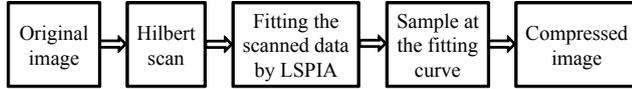


Fig. 1. The main process of image compression.

A. Construction of Hilbert curve

To compress an image, one has to convert the 2-dimensional image data to a 1-dimensional sequence by scanning the image. There are many methods to scan images [9], e.g., raster scan, Z-scan and Hilbert scan, etc. Recently, researchers are more likely to use Hilbert scan in the area of image processing because the Hilbert scan can preserve the relationship of the neighborhoods pixels as much as possible. In [9], Biswas compared the raster scan with Hilbert scan and concluded that Hilbert scanned images can provide better CR than raster scanned images. Therefore, we use Hilbert scan in this paper. For more details about Hilbert scan, readers can refer to [9]. We show in Fig. 3 two Hilbert curves of Lena and Girl given in Fig. 2.



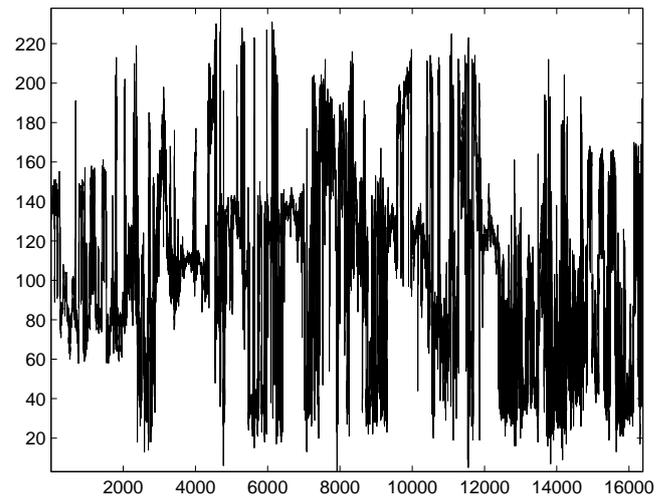
Fig. 2. Original 128×128 'Lena' and 'Girl'.

B. Fitting the Hilbert curve by LSPIA

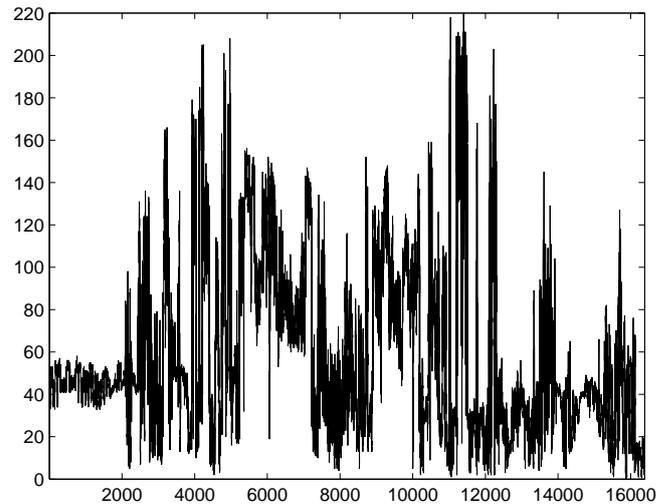
Suppose that we have obtained the 1-dimensional sequence $\{\mathbf{Q}_j\}_{j=0}^m$ by scanning the image. Then the LSPIA for B-spline is employed to fit $\{\mathbf{Q}_j\}_{j=0}^m$. Before giving the image compression algorithm, we discuss some implementation details of LSPIA.

1) *Parametrization*: Since the abscissas of \mathbf{Q}_j ($j = 0, 1, \dots, m$) are distributed uniformly, we can employ the uniform parametrization in which we assign a parameter $t_j = j/m$ for \mathbf{Q}_j .

Remark 3.1: The parametrization of data points is a difficult hotspot and plays important role in curves representation. In general, the accumulated chord parametrization is superior to the uniform one in curves representation. While in image compression, the uniform parametrization will yield smaller CR, which will be discussed later.



(a) Lena



(b) Girl

Fig. 3. The corresponding Hilbert curves of Lena and Girl in Fig. 2.

For comparison, the accumulated chord parametrization is also used in numerical examples, and is given by

$$\begin{aligned} t_0 &= 0, \quad t_m = 1, \\ t_j &= t_{j-1} + \frac{\|\mathbf{Q}_j - \mathbf{Q}_{j-1}\|}{D}, \quad j = 1, 2, \dots, m-1, \end{aligned} \quad (11)$$

where $D = \sum_{j=0}^m \|\mathbf{Q}_j - \mathbf{Q}_{j-1}\|$ is the totally chord length [14].

2) *Computation of knot vector*: We use the same knot vector as in [14], i.e.,

$$\begin{aligned} u_0 &= u_1 = u_2 = u_3 = 0, \\ u_{j+3} &= (1 - \alpha)t_{i-1} + \alpha t_i, \quad j = 1, 2, \dots, n-3, \\ u_{n+1} &= u_{n+2} = u_{n+3} = u_{n+4} = 1, \end{aligned} \quad (12)$$

where $i = \lceil jd \rceil$, $\alpha = jd - i$, $d = \frac{m+1}{n-2}$ and $\lceil x \rceil$ is the largest integer not larger than x .

3) *Selection of initial control points*: We note in [14] that the initial points $\{\mathbf{P}_i^{(0)}\}_{i=0}^n$ can be set arbitrary in theory. In this paper, the selection of the initial control points $\{\mathbf{P}_i\}_{i=0}^n$ is the same as the method used in [14], i.e.,

$$\begin{aligned} \mathbf{P}_0^{(0)} &= \mathbf{Q}_0, \quad \mathbf{P}_n^{(0)} = \mathbf{Q}_m, \\ \mathbf{P}_i^{(0)} &= \mathbf{Q}_{f(i)}, \quad i = 1, 2, \dots, n-1, \end{aligned} \quad (13)$$

where $f(i) = \left\lceil \frac{(m+1)i}{n} \right\rceil$.

C. Compression quality metrics

Very often, the peak signal to noise ratio (PSNR) and the CR are used to measure the quality of the compressed image. We review the definitions of CR and PSNR [1].

For the original data $\mathbf{Q}_j (j = 0, 1, \dots, m)$ and the reconstructed data $\tilde{\mathbf{Q}}_j = \mathbf{r}^{(k)}(t_j)$, the PSNR is defined as

$$\text{PSNR} = 10 \times \log_{10} \frac{255^2}{\text{MSE}}, \quad (14)$$

where $\text{MSE} = \frac{1}{m+1} \sum_{i=0}^m [\mathbf{Q}_j - \mathbf{r}^{(k)}(t_j)]^2$. The PSNR is usually termed as bit per pixel (bpp). Generally speaking, the bigger the PSNR, the higher the quality of the image, and vice versa.

The CR is defined as

$$\text{CR} = \frac{B_c}{B_{uc}}, \quad (15)$$

where B_c represents the number of bits in compressed data, B_{uc} represents the number of bits in uncompressed data. The CR is termed as bit per pixel (bpp) in image compression.

Next we discuss the CR of the proposed method. Given an image of size $M \times M$ pixels with gray levels $\{0, 1, \dots, 255\}$. Since the corresponding Hilbert curve is approximated by a cubic B-spline curve $\mathbf{r}^{(k)}(t)$, which can be stored by saving the control points $\mathbf{P}_i^{(k)} (i = 0, 1, \dots, n)$ as well as the knot vector $u_i (i = 0, 1, \dots, n)$. Besides, having obtained the fitting curve $\mathbf{r}^{(k)}(t)$, we need to save the parameters t_j when regenerating the approximate image data by sampling at t_j , i.e., $\mathbf{r}^{(k)}(t_j) (j = 0, 1, \dots, m)$. Clearly, it will bloats the repository unnecessarily, making it difficult to compress image efficiently. It should be pointed out that when the uniform parametrization is used, we never need to worry about saving t_j . In addition, we can obtain the knot vector u_i according to (12). This means that we can save massive storage space.

According to (7), we remark that the coordinates of the control points $\mathbf{P}_i^{(k)}$ do not always to be integers. It is well known that the decimals require larger storage space than the integers. To save storage space, we only take the integer parts of the coordinates as the control points of the cubic B-spline curve. This can be desirable in image compression for the following two reasons. On one hand, since $B_i(t) \leq 1 (t \in [0, 1])$, the decimal parts of the coordinates have a little influence on the results. More exactly, the impact of the ignored decimal parts on the recovered data is no more than 1. On the other hand, the image data are in the gray levels $\{0, 1, \dots, 255\}$. We have to round off the decimal parts of the recovered data. The rationality of integration of the control points is also verified by numerical experiments.

D. Image compression algorithm

To ensure the computational efficiency, the iteration (9) can also be terminated, if the following stopping criterion

$$\varepsilon^{(k)} \leq \theta \varepsilon^{(k-1)}, \theta \in (0, 1), \quad (16)$$

is satisfied.

Finally, we summarize the image compression algorithm into the following Algorithm 3.2.

Algorithm 3.2: (Image compression algorithm)

Input: a gray-level image.

Output: a compressed image.

- 1) Scan the image to obtain the Hilbert sequence $\{\mathbf{Q}_j\}_{j=0}^m$.
 - 2) Parameterize \mathbf{Q}_j with t_j and select the initial control points $\{\mathbf{P}_i^{(0)}\}_{i=0}^n$ according to (13).
 - 3) Compute the knot vector for cubic B-spline according to (12).
 - 4) Compute the collocation matrix B and the optimal value of μ .
 - 5) For $k = 1, 2, \dots, k_{max}$
 - (a) Update $\mathbf{P}^{(k)} = \mu B^T \mathbf{Q} + (I - \mu B^T B) \mathbf{P}^{(k-1)}$.
 - (b) Compute the fitting error $\varepsilon^{(k)}$ according to (10).
 - (c) If $\varepsilon^{(k)} \leq \theta \varepsilon^{(k-1)}$, break for.
- End for.
- 6) Compute the approximate fitting curve $\mathbf{r}^{(k)}(t)$ according to (7).
 - 7) Sample at t_j and obtain the recovered data $\tilde{\mathbf{Q}}_j = \mathbf{r}^{(k)}(t_j), j = 0, 1, \dots, m$.
 - 8) Calculate the PSNR and the CR according to (14) and (15), respectively.
 - 9) Display the compressed image.

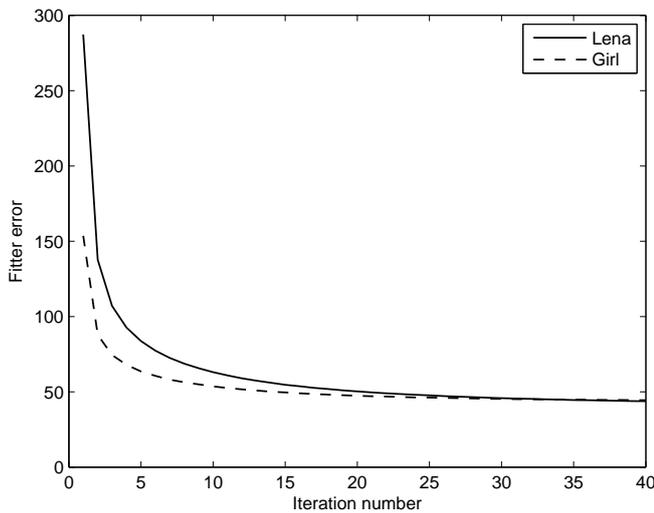
IV. IMAGE COMPRESSION EXAMPLES

In this section we employed Algorithm 3.2 to test the following two well-known images, which are often used to illustrate the effectiveness of image compression methods. All the numerical experiments were done by Matlab R2012b on a PC with Intel(R) Core(TM) i5-5200U CPU @2.20 GHz and RAM 6GB.

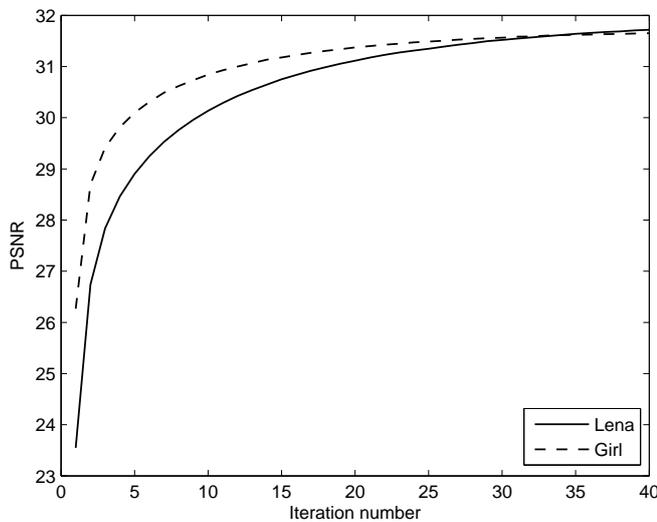
Firstly, we select $n = 6200$ control points when we compress the Lena image and select $n = 4745$ control points when we compress the Girl image. The fitting errors of the approximate fitting curves $\mathbf{r}^{(k)}(t)$ and the PSNR of the compressed images are shown in Fig. 4(a) and Fig. 4(b), respectively. We observe that the fitting error decreases fast in the first several iterations, and then decreases slowly. Similarly, the PSNR increases fast in the first several iterations, and then slows down. Therefore, there is no need to iterate many times because it is time-consuming. Hence we add a terminal condition (16) and we set $\theta = 0.98$ in our tests.

By using two different parametrization methods, we list in Table I the number of iterations required, the CR and the PSNR of the compressed images obtained by Algorithm 3.2. We denote by UP the uniform parametrization, by ACP the accumulated chord parametrization, by n the number of control points of cubic B-spline curve, and by k the iteration number required, respectively. In our tests, we test different n .

From Table I, we observe that the quality of the compressed images is improved as the number of control points increases. Consequently, the CR increases. Besides, despite the fact that the accumulated chord parametrization can provide good results for compression of images. As stated in Section III-C, it is not advisable in practice, because it will take up lots of storage space to save the knot vectors and parameters.



(a) Fitting error vs iteration



(b) PSNR vs iteration

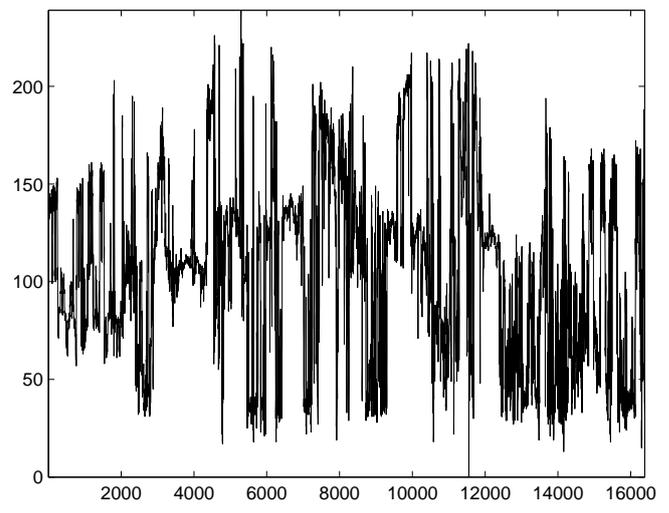
Fig. 4. The fitting error and PSNR of the compressed image versus the iteration.

TABLE I
NUMERICAL RESULTS OF COMPRESSED 128×128 IMAGES

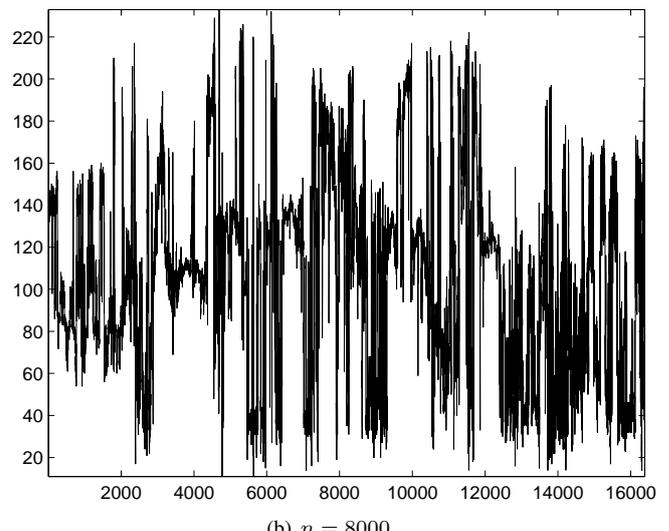
Image	n	CR	UP		ACP	
			k	PSNR	k	PSNR
Lena	4000	0.4883	8	26.3166	8	27.4178
	5000	0.6104	7	27.2148	7	28.3812
	6000	0.7324	9	28.1842	9	29.8114
	7000	0.8545	10	29.1810	10	31.4707
	8000	0.9766	10	30.0596	10	32.8733
	9000	1.0986	12	31.1216	12	34.0487
Girl	4000	0.4883	7	29.2893	7	29.9472
	5000	0.6104	7	30.0564	7	30.7359
	6000	0.7324	9	31.0910	9	32.4577
	7000	0.8545	10	32.0776	10	34.1655
	8000	0.9766	9	33.0083	9	35.7128
	9000	1.0986	11	34.0185	11	36.6352

In Fig. 5 and 6, we show the cubic B-spline curves with different n when fitting the scanned Hilbert curves in Fig. 3. In Fig. 7, we show the compressed images obtained by Algorithm 3.2. All the numerical results demonstrate that the proposed method achieves a good performance in image compression.

Secondly, we employ Algorithm 3.2 to compress $256 \times$



(a) $n = 5000$



(b) $n = 8000$

Fig. 5. Fitting the scanned Hilbert curves of Lena in Fig. 3(a).

256 gray images. In Table II, we list the numerical results obtained by Algorithm 3.2. The numerical results obtained by Biswas's methods [9] are also listed for comparison. We can observe that with the same CR, the PSNR of Algorithm 3.2 is bigger than those of Biswas's methods. This means that our method performs much better than Biswas's methods.

The compressed Lena and Girl images are shown in Fig. 8 and 9, respectively. We can see that the compressed images obtained by Biswas's methods suffer from visible blocking artifacts, while the compressed images obtained by Algorithm 3.2 could avoid blocking artifacts satisfactorily. Furthermore, it can be found from the detailed figures in Fig. 8 and 9 that there exists sawtooth effect at the edges, no matter whether Algorithm 3.2 or Biswas's methods are used. But the images compressed by Algorithm 3.2 have the weaker sawtooth effect than those by Biswas's methods. These results indicate that the proposed method not only reduces blocking artifacts and sawtooth effect but also has more compression effect than Biswas's methods.

V. DISCUSSION AND IMPROVEMENT

Oftentimes, compressed images are polluted by many kinds of noises during the process of image compression, especially in lossy compression. Consequently, it will bring

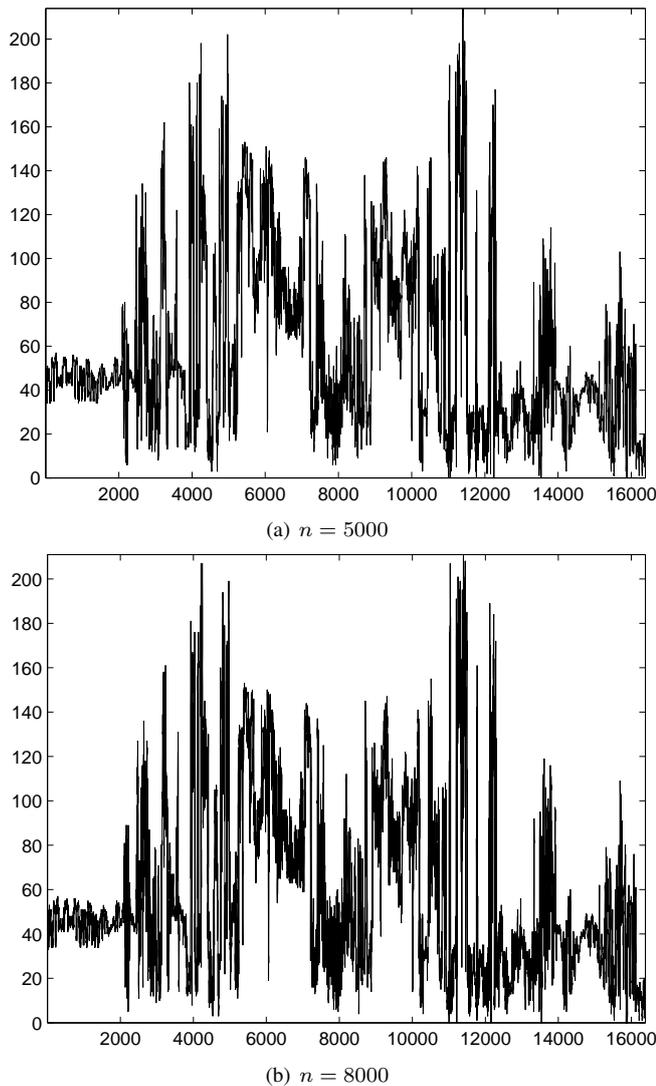


Fig. 6. Fitting the scanned Hilbert curves of Girl in Fig. 3(b).

TABLE II
COMPARISON OF ALGORITHM 3.2 WITH BISWAS'S METHODS [9]

Image	Algorithm 1 [9]		Algorithm 2 [9]		Algorithm 3.2	
	CR	PSNR	CR	PSNR	CR	PSNR
Lena	1.44	30.577	1.44	31.222	1.44(11796)	33.641
	1.28	29.757	1.34	30.821	1.28(10485)	33.495
Girl	1.09	30.577	1.07	30.436	1.07(8765)	36.010
	0.67	27.692	0.68	28.811	0.67(5488)	32.872

unbearable block artifacts, sawtooth effect at the edges and other defects. At this time, we can improve the quality of the compressed images by employing some pre-processing or post-processing techniques, such as transform coding [3], image filtering [15] and so on.

Here we use some image filtering algorithms to reduce the sawtooth effect at the edges. The median filtering [15] and the Gauss filtering [16] algorithms are employed to filtering the compressed image in Fig. 7(c). The filtered images are illustrated in Fig. 10. From these two examples we conclude that the image filtering algorithms can reduce the sawtooth effect of the compressed images but can not eliminate it entirely.



Fig. 7. Compressed 128×128 images.

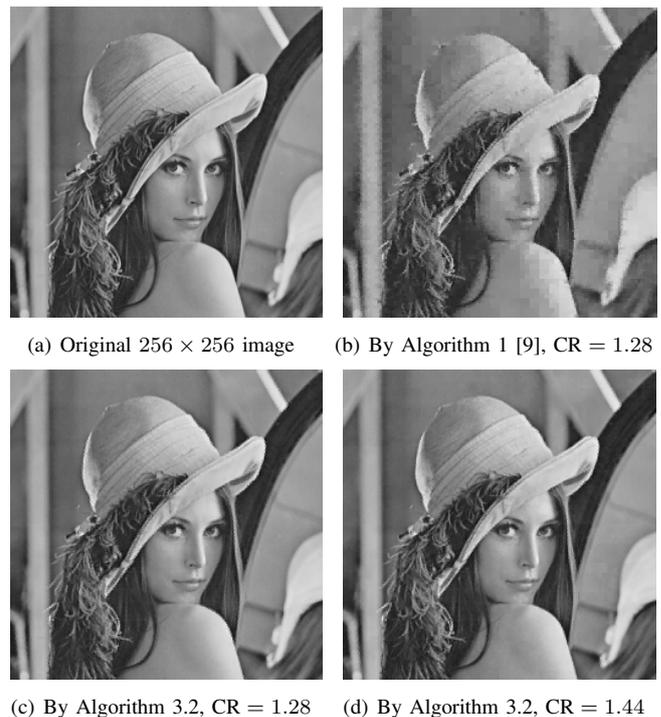


Fig. 8. Comparison of compressed Lena images.

VI. CONCLUSIONS

In this paper, we have exploited an image compression algorithm by using LSPIA due to its efficient and reliable performance in data fitting. Compared with the other curve fitting methods, the proposed method can well preserve the relative information between neighborhood pixels. Numerical experiments also show that the proposed method outperforms the similar image compression methods, in terms of the CR, the PSNR and the blocking artifacts of the compressed images.



Fig. 9. Comparison of compressed Girl images.



Fig. 10. Post-processing the compressed image in Fig. 8(c) with image filter.

As a shortcoming of this paper, we use a single curve to approximate the image data. This technique increases the quality of the compressed image at the cost of computational time.

REFERENCES

[1] J. Uthayakumar, T. Vengattaraman and P. Dhavachelvan, "A survey on data compression techniques: From the perspective of data quality, coding schemes, data type and applications," *Journal of King Saud University Computer & Information Sciences*, in press, 2018.

[2] M. A. Rahman and M. Hamada, "Lossless image compression techniques: A state-of-the-art survey," *Symmetry*, vol. 11, no. 10, pp. 1274-1295, 2019.

[3] A. J. Hussain, A. Al-Fayadhb and N. Radi, "Image compression techniques: A survey in lossless and lossy algorithms," *Neurocomputing*, vol. 300, no. 26, pp. 44-69, 2018.

[4] R. Panda and B. N. Chatterji, "Least squares generalized B-spline signal and image processing," *Signal Processing*, vol. 81, no. 10, pp. 2005-2017, 2001.

[5] K. S. Sim, F. F. Ting, J. W. Leong, and C. P. Tso, "Signal-to-noise Ratio estimation for SEM single image using cubic spline interpolation with linear least square regression," *Engineering Letters*, vol. 27, no. 1, pp. 151-165, 2019.

[6] S. M. Othman, A. E. Mohamed, Z. Nossair and M. I. El-Adawy, "Image compression using polynomial fitting," *International Conference on Electronics, Communication and Aerospace Technology 2019, 12-14 June*, 2019, Coimbatore, India, pp. 344-349.

[7] R. Banerjee and B. S. Das, "An energy efficient image compression scheme for wireless multimedia sensor network using curve fitting technique," *Wireless Networks*, vol. 25, no. 1, pp. 167-183, 2019.

[8] S. Biswas and S. K. Pal, "Image coding using modified Bézier-Bernstein approximation," *Information Sciences*, vol. 83, no. 3, pp. 175-197, 1995.

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

[10] J. C. Li and D. B. Zhao, "Image compression based on quadratic rational Bézier curve approximation," *Journal of Huazhong University of Science and Technology(Natural Science Edition)*, vol. 40, no. 1, pp. 21-25, 2012.

[11] J. C. Li and D. B. Zhao, "An investigation on image compression using the trigonometric Bézier curve with a shape parameter," *Mathematical Problems in Engineering*, vol. 2013, pp. 1-8, 2013.

[12] J. Demmel, "Applied numeric linear algebra," *Society for Industrial and Applied Mathematics*, 1997.

[13] Yousef Saad, "Iterative methods for sparse linear systems," *IEEE Computational Science & Engineering*, 1996.

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

[15] C. M. Bangug, A. C. Fajardo and R. P. Medina, "A modified median filtering algorithm (MMFA)," *Lecture Notes in Engineering and Computer Science: Proceedings of The International MultiConference of Engineers and Computer Scientists 2019, IMECS 2019, 13-15 March, 2019, Hong Kong*, pp. 377-380.

[16] L. Cadena, F. Cadena and A. Legalov, "Improve Method for Processing Dental Images with Fast Spatial Filter and Shearlet Transform," *Lecture Notes in Engineering and Computer Science: Proceedings of The International MultiConference of Engineers and Computer Scientists 2019, IMECS 2019, 13-15 March, 2019, Hong Kong*, pp. 371-376.