A Single-Image Super-Resolution Method Based on Progressive-Iterative Approximation

Yunfeng Zhang, Ping Wang, Fangxun Bao, Xunxiang Yao, Caiming Zhang, and Hongwei Lin

***Abstract*—In this paper, a novel single image super-resolution (SR) method based on progressive-iterative approximation is proposed. To preserve textures and clear edges, the image SR reconstruction is treated as an image progressive-iterative fitting procedure and achieved by iterative interpolation. Due to different features in different regions, we first employ the nonsubsampled contourlet transform (NSCT) to divide the image into smooth regions, texture regions, and edges. Then, a hybrid interpolation scheme based on curves and surfaces is proposed, which differs from the traditional surface interpolation methods. Specifically, smooth regions are interpolated by the non-uniform rational basis spline (NURBS) surface geometric iteration. To retain textures, control points are increased, and the progressive-iterative approximation of the NURBS surface is employed to interpolate the texture regions. By considering edges in an image as curve segments that are connected by pixels with dramatic changes, we use NURBS curve progressive-iterative approximation to interpolate the edges, which sharpens the edges and can maintain the image edge structure without jaggy and block artifacts. The experimental results demonstrate that the proposed method significantly outperforms the state-of-the-art methods in terms of both subjective and objective measures.**

***Index Terms*—Single image super-resolution, regional division, surface geometric iteration, curve geometric iteration.**

I. Introduction

SINGLE image super-resolution (SISR) aims to reconstruct a high-resolution (HR) image from a single low-resolution (LR) image. SISR is receiving increasing attention because of its extensive application in many fields, such as image-based medical analysis, satellite remote sensing imaging, video surveillance, and computer vision. Various SISR methods have been reported. Overall, the SISR approaches can be divided into three categories: reconstruction-based methods, interpolation-based methods, and learning-based methods.

For the reconstruction-based methods, the LR image is treated as an image obtained from several degradation factors such as down-sampling and blurring, which leads to the SISR being an ill-posed problem. Consequently, various priors are employed to reconstruct the corresponding HR image. The commonly used priors include smoothness prior [1]–[3], gradient prior [4]–[6], sparsity prior [7], [8], nonlocal self-similarity prior [10]–[12], and learned-based prior [9]–[11]. All of these priors have relevant advantages and disadvantages because their focuses are different. Methods that integrate two or more priors have also been presented [10]–[13]. The reconstruction-based method can effectively suppress unwanted artifacts by adding constraints. However, the high-frequency details can easily be regarded as artifacts, which may lead to being smoothed as the magnification increases.

For the interpolation-based methods [14]–[18], the core idea is to estimate the missing information from the LR neighbors. There are two popular interpolation schemes: polynomial-based schemes and edge-directed strategies. The bilinear interpolation and the bicubic interpolation are typical polynomial-based interpolation methods. Because of their low computational complexity, these approaches have been widely applied in many fields. However, the interpolated images often suffers from blurry and jaggy artifacts along the edges of objects. To solve this problem, edge-directed methods [14]–[17] are presented. Zhou *et al.* [15] proposed an edge-directed cubic convolution interpolation scheme and presented a method for estimating missing pixels on the strong edge. Such methods can maintain edge structures but typically incur noise or distortion in texture regions. To reconstruct a high-quality HR image, Zhu *et al.* [18] obtained the interpolated pixels by using the nonlocal geometric similarities and directional gradients of the LR image. The details of objects can be preserved, but the obtained HR image is not satisfactory, suffering from relevant artifacts such as jaggy artifacts. These interpolated-based approaches produce less jaggy edges.

Manuscript received May 4, 2019; revised August 18, 2019; accepted September 17, 2019. Date of publication September 25, 2019; date of current version May 21, 2020. This work was supported in part by the National Natural Science Foundation of China under Grants 61373080, 61672018, 61402261, and 61772309, in part by the National Natural Science Foundation of China-Zhejiang Two Integration Joint Fund Key Project under Grant U1609218, in part by the Primary Research and Development Plan of Shandong Province under Grants 2016GSF120013, 2017GGX10109, 2018GGX101013, and 2019GSF109112, in part by the Fostering Project of Dominant Discipline and Talent Team of Shandong Province Higher Education Institutions, in part by the Shandong Provincial Natural Science Foundation for Excellent Youth under Grant ZR2018JL022, and in part by Shandong Provincial Natural Science Foundation under Grant ZR2019MF051. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Marco Carli. *(Corresponding author: Fangxun Bao.)*

Y. Zhang is with the Computer Science and Technology, Shandong University of Finance and Economics, Jinan 250014, China (e-mail: yfzhang@sdufe.edu.cn).

P. Wang is with the Software and IT Department, École de Technologie Supérieure, Montreal, QC H3C1K3, Canada (e-mail: wp_scdedu@163.com).

F. Bao is with the Department of Mathematics, Shandong University, Jinan 250100, China (e-mail: fxbao@sdu.edu.cn).

X. Yao is with the Faculty of Engineering and Information Technology, University of Technology, Sydney, NSW 2007, Australia (e-mail: 12967776@student.uts.edu.au).

C. Zhang is with the Computer Science and Technology, Shandong University, Jinan, Shandong, 250101 China (e-mail: czhang@sdu.edu.cn).

H. Lin is with the Department of Mathematical Sciences, Zhejiang University, Hangzhou 310058, China (e-mail: hwlin@zju.edu.cn).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2019.2943750

However, the recovery of high-frequency components is limited due to the inadequate information. As the magnification increases, these types of methods are more prone to generate artifacts, such as distortion, aliasing, and blocks [19].

The learning-based methods recover the HR image by building the mapping relationship between LR and HR image pairs. According to how the mapping relationship is constructed, the learning-based methods can be classified as example-based algorithms [20]–[22], sparse-coding-based algorithms [23], [24], and regression-based algorithms [25]–[28]. For the example-based algorithms, Huang *et al.* [21] proposed a self-similarity-driven SR algorithm (selfExSR), which expanded the internal patch search space through geometric variations. This method can improve visual effects well, but it cannot produce fine details for the LR image without sufficient patterns. For sparse-coding-based algorithms, Yang *et al.* [23] learned LR and HR dictionaries independently and employed a consistent coding scheme to ensure the accuracy of HR coding coefficients. Kang *et al.* [24] achieved SR by using sparse representation and image decomposition, and the method can avoid block artifacts. The results of these methods are still not satisfactory when there are not enough textures present in the training sets. Regression-based algorithms obtained good quality. Timofte *et al.* [25] developed the anchored neighborhood regression (ANR) to reconstruct SR images, and then an adjusted anchored neighborhood regression (A+) method was developed for fast SR [26]. Recently, deep learning has attracted the attention of many researchers [29], [30]. Dong *et al.* [29] presented a SR method (SRCNN) based on a convolutional neural network. This method performs well for some types of images. However, it is prone to generate visual blur and aliasing artifacts [6].

The combination of different methods has recently been attracting more attention [31]–[33]. Dong *et al.* [31] presented an image interpolation method by combining a machine learning method with image interpolation. By embedding a nonlocal autoregressive modeling (NARM) into the sparse representation model, the method can reduce the correlation between the sample matrix and the sparse representation dictionary such that the sparse representation is more effective. The constructed image can maintain the edge structure and suppress the zigzag and ringing artifacts. However, when the image is processed with a larger scale factor, the image details are lost and the textures are blurred. Zhang *et al.* [32] presented a novel image restoration method that combines nonlocal selfsimilarity and global structure sparsity in a single efficient model. Combining the fractal method with rational interpolation, Zhang *et al.* [33] presented a SISR method based on the constructed rational fractal interpolation model. Because the rational interpolation function can preserve edge structures, and the fractal is an effective tool to describe image textures, the method can retain fine details and clear edges to some extent.

In practice, a natural scene is a continuous surface [34], also called a realistic surface, and the images are captured from natural scenes. Therefore, the image can be regarded as a discrete data point set from the realistic surface which can be obtained by $sinc$ kernel function. Based on this fact, interpolation methods are highly feasible. However, it is difficult for traditional polynomial interpolation and rational interpolation approaches to achieve the realistic surface. For example, polynomial interpolation function fails to approximate $sinc$ kernel function effectively, which could not get the realistic surface. Compared with the polynomial interpolation function, rational interpolation function can better approximate the $sinc$ kernel function. Nevertheless, the rational surface is generated by the interpolation data in one time so that there are errors between the generated rational surface and the realistic surface, leading to unpleasing results in image edges and textures. In comparison to traditional methods, geometric iterative method, also called progressive-iterative approximation (PIA), has a clear geometric meaning, and this approach was developed by Lin [35]. By adjusting the control points of the curve or surface iteratively, the limited curve or surface will interpolate the given data point set. In another word, geometric iterative method can approximate the $sinc$ kernel function by a series of rational function. In this paper, inspired by this, we treat the procedure of restoring the HR image as an iterative interpolation procedure that can gradually approximate the realistic surface. By embedding the interpolation into the geometric iteration, a novel SR method based on geometric iterative method is proposed. The framework of the proposed method is illustrated in Fig. 1.

Overall, the main contributions of the proposed method are as follows:

1) Treating the image SR reconstruction as an image progressive-iterative fitting procedure, we present a novel SISR method based on PIA. The intermediate result of the proposed SR method is explicit, which facilitates human-computer interactions.
2) Considering the features of regions and edges, a hybrid scheme for reconstructing HR images is proposed, which combines surface interpolation and curve interpolation. The scheme can not only preserve more high-frequency details on texture regions but also obtain sharp and clear image edges.
3) For edges, in contrast to patch interpolation, we present the curve interpolation using the pixels. The pixel-based curved interpolation method can avoid producing jaggy and block artifacts.

The remainder of this paper is arranged as follows. The proposed progressive-iterative approximation-based SR method is detailed in Section II. The Experimental results and discussions are presented in Section III. Section IV concludes this paper.

II. Proposed Interpolation Method Based on PIA

In this section, we propose a novel SISR method based on PIA. First, the image is divided into three regions, namely, smooth regions, texture regions, and edges, via the nonsubsampled contourlet transform (NSCT). Then, a hybrid interpolation scheme based on curves and surfaces is proposed to handle these different regions. The input LR image is considered as the initial control points. For the smooth regions, the geometric iteration of the non-uniform rational basis spline (NURBS) surface is employed, which can gradually interpolate the LR image. For

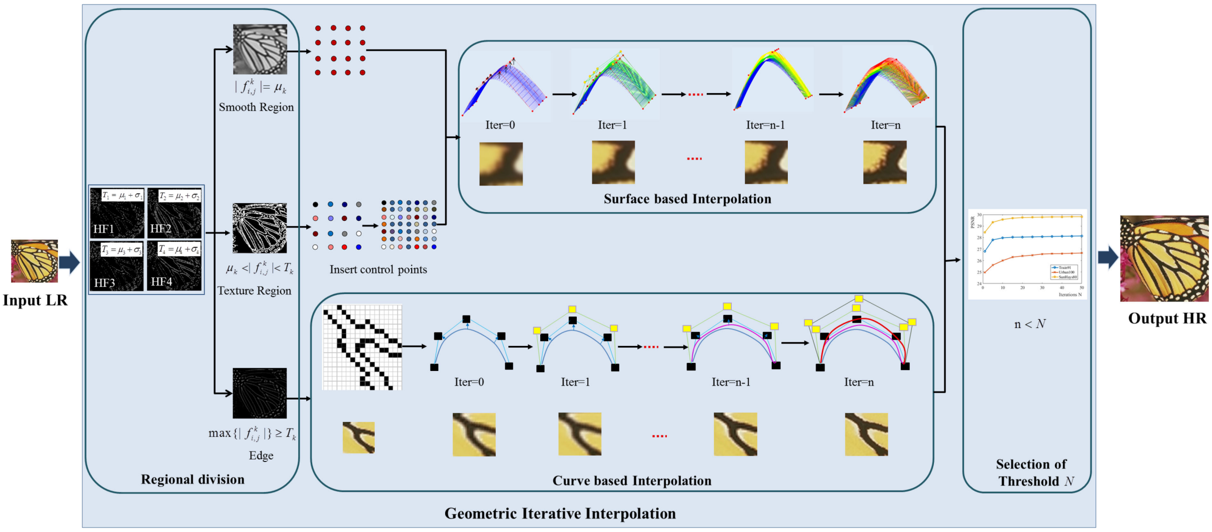


Fig. 1. Flowchart of the proposed method.

texture regions with more details, more control points are added, which contributes to the fitting precision. Then the geometric iterative method is employed to gradually approximate the limited surface. The constructed surface with more control points can better characterize the textures of the image. Edges can be regarded as some curve segments where the grey value dramatically changes. Interpolation geometric iteration of the NURBS curve is designed for edges. Finally, an experiment on training sets is conducted to determine the termination condition. The NURBS is part of numerous industry-wide standards, and it is well known that the NURBS function yields stable and robust interpolation results [36]. Using the same format of curve and surface can ensure that the reconstructed SR image is authentic without splice marks.

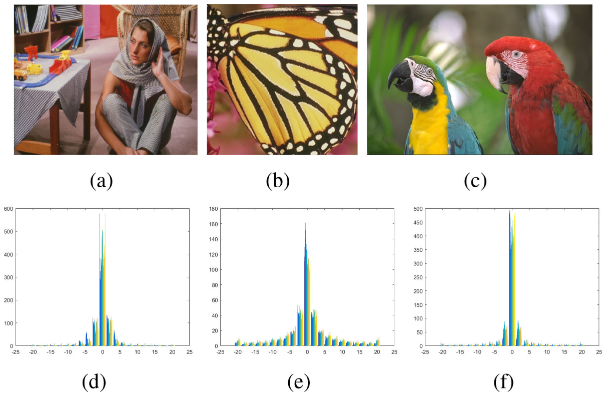


Fig. 2. Images and the high frequency sub-band coefficient histogram.

A. Regional Division

In an image, different regions have different structural features. It is necessary to handle the different regions with different methods. Therefore, regional division plays a vital role in the image interpolation. The results of the regional division directly affect the interpolation quality.

The NSCT can detect the smooth regions, the texture regions, and the edges [37]. Moreover, the NSCT is shift-invariant, which means that each pixel of the transform subbands corresponds to that of the input image in the same location. The NSCT is used to divide the image.

We can obtain the high frequency information of the image in different directions by using the NSCT. The high-frequency coefficients are closely related to the edges and details. A higher absolute value of the coefficient indicates clearer edges. Thus, according to the coefficients, the thresholds are estimated for guiding the regional division. From statistics, the high-frequency coefficients obey a normal distribution, which can be proven by K-S tests [38]. The images and their corresponding high-frequency coefficient histograms are shown in Fig. 2. Let μ be the mean value and σ be the standard deviation. Then, $T_{Hk\_1} = \mu + \sigma$ and $T_{Hk\_2} = \mu + 2\sigma$ are viewed as the thresholds, and $k = 1, 2, 3, 4$. If the absolute value of the coefficient is higher than T_{Hk_2}, then the corresponding pixel belongs to the edges. If the absolute value of the coefficient is between T_{Hk_1} and T_{Hk_2}, then the pixel belongs to the texture regions. Otherwise, it belongs to the smooth regions. Fig. 3 shows the procedure of the regional division.

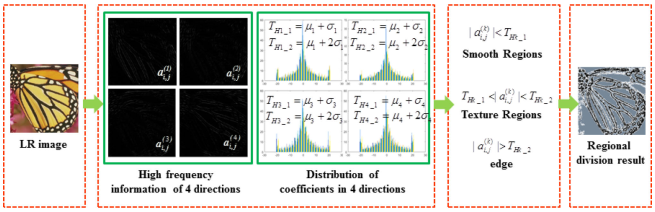
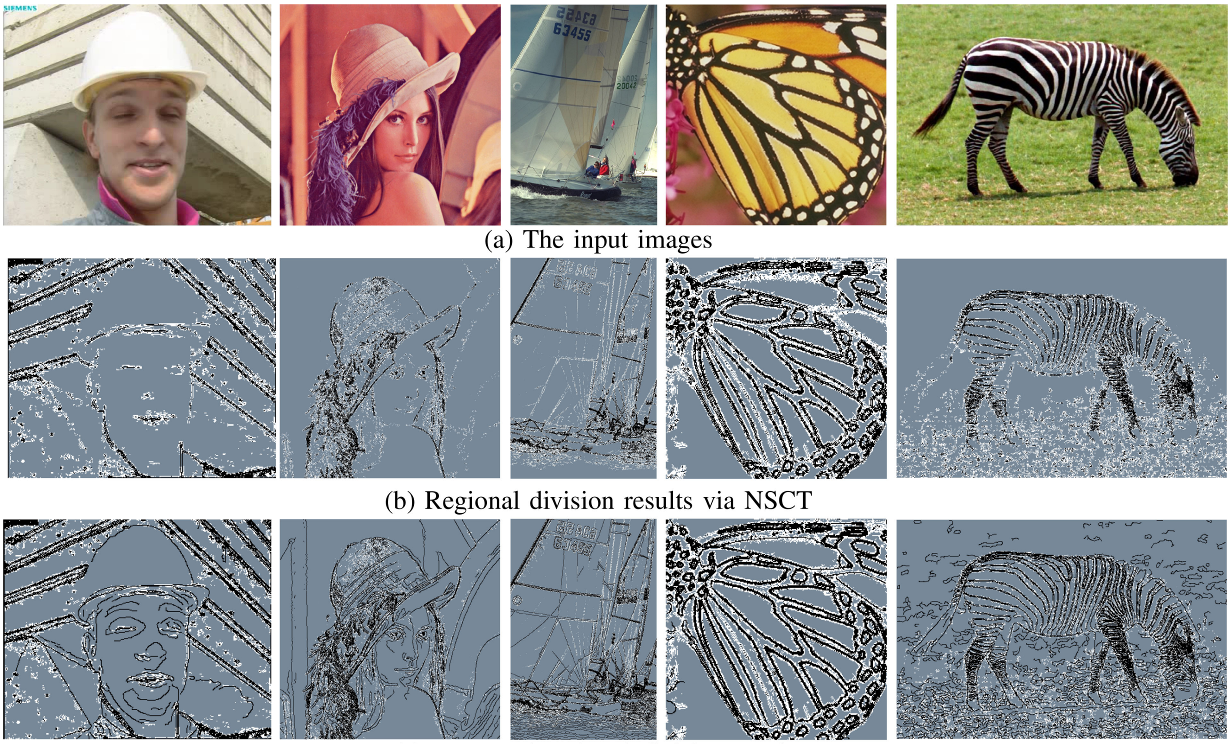

LR image
$a_{i,j}^{(1)}$
$a_{i,j}^{(2)}$
$a_{i,j}^{(3)}$
$a_{i,j}^{(4)}$
High frequency information of 4 directions
$T_{H1\_1} = \mu_1 + \sigma_1$
$T_{H1\_2} = \mu_1 + 2\sigma_1$
$T_{H2\_1} = \mu_2 + \sigma_2$
$T_{H2\_2} = \mu_2 + 2\sigma_2$
$T_{H3\_1} = \mu_3 + \sigma_3$
$T_{H3\_2} = \mu_3 + 2\sigma_3$
$T_{H4\_1} = \mu_4 + \sigma_4$
$T_{H4\_2} = \mu_4 + 2\sigma_4$
Distribution of coefficients in 4 directions
$|a_{i,j}^{(k)}| < T_{Hk\_1}$
Smooth Regions
$T_{Hk\_1} < |a_{i,j}^{(k)}| < T_{Hk\_2}$
Texture Regions
$|a_{i,j}^{(k)}| > T_{Hk\_2}$
edge
Regional division result

Fig. 3. Procedure of the regional division.

Fig. 4(b) shows the results of the regional division. The experimental results imply the effectiveness of the regional division.

Edge localization with high quality plays a key role in image SR since 80–90% of the image energy lies in edges. Though NSCT can detect smooth regions, texture regions and edges, it



(a) The input images

(b) Regional division results via NSCT

(c) Improved regional division results

Fig. 4. Results of the regional division (the blue regions are the smooth regions, the white regions are the texture regions, and the black curves are edges).

is pixel-based method, which the detected edges may suffer from discontinuity. Canny edge detector can detect complete contour. To ameliorate the edge localization, the stack of edges detected by NSCT and edges detected by Canny is regarded as the edges. Fig. 4(c) shows the improved results of regional division (the stack of edges detected by NSCT and edges detected by Canny is regarded as the edges, the texture regions detected by NSCT except edges detected by Canny is regarded as texture regions).

B. Interpolation of Smooth Regions

We apply the NURBS surface progressive-iterative approximation to interpolate the smooth regions.

For the given $m \times n$ LR image patch, first, let $P = \{P_{i,j} : i = 0, 1, \ldots, m-1; j = 0, 1, \ldots, n-1\}$ be the data set, which is also the initial control point set. Each $P_{i,j}$ corresponds to a blending basis function $N_{i,j}(u, v)$, and $u \in [0, 1]$, $v \in [0, 1]$ are parameters. Then, the initial NURBS surface is constructed as follows:

$$R^{(0)}(u, v) = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} R_{i,j}^{(0)} N_{i,j}(u, v), \qquad (1)$$

where $R_{i,j}^{(0)} = (\omega_{i,j} P_{i,j}^{(0)}, \omega_{i,j})$ is the control points in the projective space, and $\omega_{i,j}$ is the weight factor. $N_{i,j}(u, v) = B_{i,j}(u) B_{i,j}(v)$, and $B_{i,j}(u)$ and $B_{i,j}(v)$ are B-spline basis functions defined on the $u-$direction and $v-$direction knot vectors respectively. The initial control points and the constructed initial NURBS surface are shown in Fig. 5.

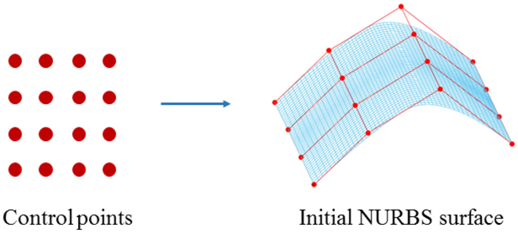


Fig. 5. The initial NURBS surface construction.

After constructing the initial patch, a NURBS surface progressive fitting procedure is applied to make the NURBS patch approximate the data set. The progressive fitting algorithm is a geometric iterative scheme that begins with an initial patch $R^{(0)}(u, v)$. The kth step of the iteration is taken as an example to explain the algorithm. Suppose that the NURBS surface patch after the kth iteration is $R^{(k)}(u, v)$ with control points $P_{i,j}^{(k)}$.

$$R^{(k)}(u, v) = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} R_{i,j}^{(k)} N_{i,j}(u, v). \qquad (2)$$

where $R_{i,j}^{(k)} = (\omega_{i,j} P_{i,j}^{(k)}, \omega_{i,j})$ is the control points in the projective space.

To obtain the $(k+1)$st NURBS surface patch $R^{(k+1)}(u, v)$, the difference vector $\delta_{i,j}^{(k)}$ for each data point $P_{i,j}$ is calculated.

$$\delta_{i,j}^{(k)} = P_{i,j} - R^{(k)}(u_i, v_j). \qquad (3)$$

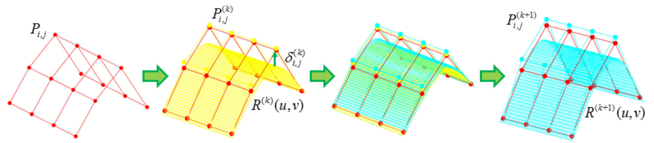


Fig. 6. The construction procedure of $(k+1)$st NURBS surface patch.

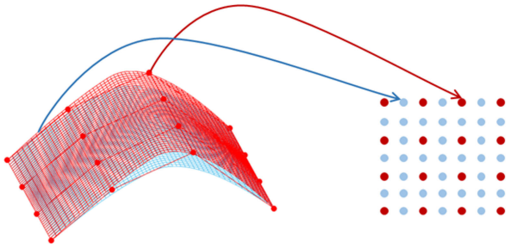


Fig. 7. Limit NURBS surface patch and the HR image patch.

Then, the new NURBS control points $P^{(k+1)}_{i,j}$ for the $(k+1)$st NURBS surface patch $R^{(k+1)}(u,v)$ are generated.

$$P^{(k+1)}_{i,j} = P^{(k)}_{i,j} + \delta^{(k)}_{i,j}. \tag{4}$$

According to the new control points, the $(k+1)$st NURBS surface patch is derived as follows. Fig. 6 shows the construction procedure of $(k+1)$st NURBS surface patch.

$$R^{(k+1)}(u,v) = \sum_{i=0}^{m-1}\sum_{j=0}^{n-1} R^{(k+1)}_{i,j} N_{i,j}(u,v). \tag{5}$$

Finally, the limited NURBS surface patch that approximates the data set is obtained, as shown in Fig. 7(a). According to the limited surface patch, we can acquire the corresponding HR image patch, as shown in Fig. 7(b). The red dots represent the LR image pixels, and the blue dots are the interpolated pixels.

C. Interpolation of Texture Regions

Compared with the smooth regions, the texture regions feature complicated details. The control points determine the shape of the surface, and adding additional control points allows a better approximation of a given data set, which means that more textures can be captured for HR images. Here, we employ bicubic interpolation to add more control points. In the experiments, we also evaluate the reconstructed HR images with different interpolation methods of increasing control points. According to the experimental results, different methods with increased control points have little influence on the reconstructed HR images. The initial NURBS surface patch is constructed with these control points for geometric iterative interpolation.

After increasing the number of control points, we construct the NURBS surface geometric iterative model to interpolate the texture regions. For the given $m \times n$ LR image patch, at the beginning of the iteration, $P_{i,j}$ $(i = 1, 2, \ldots, 2m-1; j = 1, 2, \ldots, 2n-1)$ are selected as the initial control point set, as shown in Fig. 8(a). The dots in the red box represent the LR image pixels, and the remaining dots represent the additional control points. According to the control points, the initial NURBS surface $R^{(0)}(u,v)$ is constructed, as shown in Fig. 8(b).

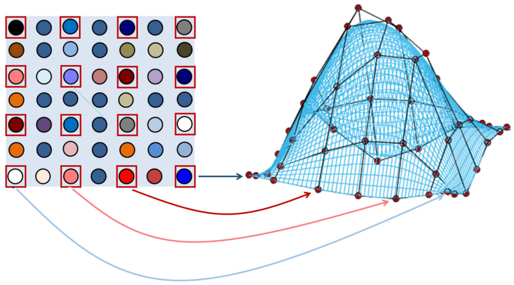


Fig. 8. The initial control points and the initial NURBS surface.

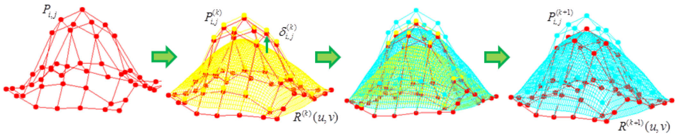


Fig. 9. The construction procedure of $(k+1)$st NURBS surface patch.

$$R^{(0)}(u,v) = \sum_{i=1}^{2m-1}\sum_{j=1}^{2n-1} R^{(0)}_{i,j} N_{i,j}(u,v) \tag{6}$$

The difference vector $\delta^{(1)}_{i,j}$ for each data point $P_{i,j}$ is calculated as follows.

$$\delta^{(1)}_{i,j} = P_{i,j} - R^{(0)}(u_i, v_j)$$

Although the difference vectors of all control points are calculated, we focus on the difference vector of LR pixels rather than that of the additional control points. Furthermore, the new control points are obtained.

$$P^{(1)}_{i,j} = P^{(0)}_{i,j} + \delta^{(0)}_{i,j},$$

and the new surface

$$R^{(1)}(u,v) = \sum_{i=1}^{2m-1}\sum_{j=1}^{2n-1} R^{(1)}_{i,j} N_{i,j}(u,v).$$

Similarly, suppose that the kth NURBS surface $R^{(k)}(u,v) = \sum_{i=1}^{2m-1}\sum_{j=1}^{2n-1} R^{(k)}_{i,j} N_{i,j}(u,v)$ is obtained, and then the difference vectors and the new control points are acquired.

$$\delta^{(k)}_{i,j} = P_{i,j} - R^{(k)}(u_i, v_j) \tag{7}$$

$$P^{(k+1)}_{i,j} = P^{(k)}_{i,j} + \delta^{(k)}_{i,j} \tag{8}$$

Then, the $(k+1)$st surface can be derived as

$$R^{(k+1)}(u,v) = \sum_{i=1}^{2m-1}\sum_{j=1}^{2n-1} R^{(k+1)}_{i,j} N_{i,j}(u,v). \tag{9}$$

where $R^{(k+1)}_{i,j} = (\omega_{i,j} P^{(k)}_{i,j}, \omega_{i,j})$. Fig. 9 shows the construction procedure of (k + 1)st NURBS surface patch. The limited surface is the result of fitting the data points $P_{r,l}(r = 1, 3, 5, \ldots; l = 1, 3, 5, \cdots)$, as shown in Fig. 10. The blue surface is the initial surface, and the red surface is the limited surface. Then, the SR image patch can be obtained based on the limited surface.

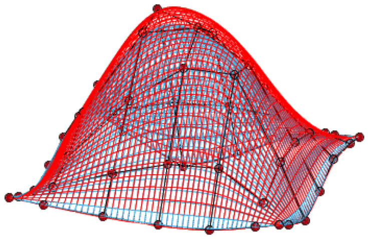
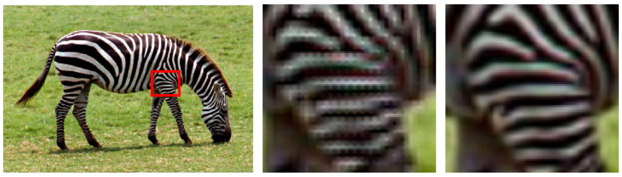
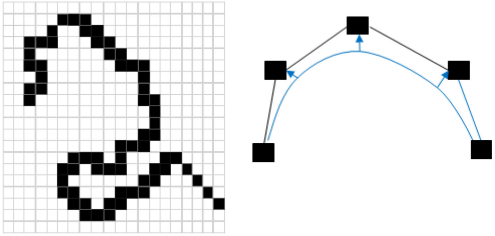


Fig. 10. The limited NURBS surface.



(a) LR (b) patch interpolation (c) curve interpolation

Fig. 11. Comparison of using the patch interpolation and the curve interpolation for SR.



(a) (b)

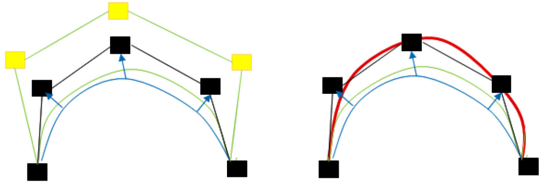
Fig. 12. The edge and the initial curve.

D. Interpolation of Edges

Edges are curve segments in an image. The curve segments are connected by pixels with dramatic changes in the intensity function or in the spatial derivatives of the intensity function. For edges, traditional interpolation methods are prone to produce jaggy and block artifacts. This occurs because the methods use patch interpolation to interpolate the roughness edges, which also considers the pixels in non-edge regions. Fig. 11 presents the SR reconstruction results obtained by patch interpolation and curve interpolation. Fig. 11(a) is the input LR image. Fig. 11(b) shows a local HR image obtained using patch interpolation. To solve this problem, we apply NURBS curve geometric iterative method to interpolate the edges, which only uses information provided by pixels on the edges. Fig. 11(c) shows a local HR image obtained using the curve. Compared with Fig. 11(b), it is clear that using the curve to interpolate the edges can preserve the sharp edges.

Fig. 12(a) is the LR patch, and the black block represents the edge pixels. We treat the pixels that are on the edges as initial control points $P_i^{(0)}(i = 0, 1, 2, \ldots, m-1)$, and we construct the initial NURBS curve $Q^{(0)}(t)$. Fig. 12(b) is the initial NURBS curve with five control points.

$$Q^{(0)}(t) = \sum_{i=0}^{m-1} Q_i^{(0)} B_i(t), \tag{10}$$



(a) (b)

Fig. 13. The curve $Q^{(1)}(t)$ and the limited curve.

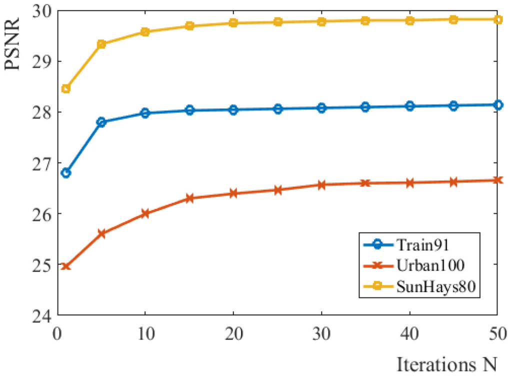


Fig. 14. PSNR changes with iterations.

where m is the number of edge pixels, and $Q_i^{(0)} = (\omega_i P_i^{(0)}, \omega_i)$. $\omega_{i,j}$ are the weights. $B_i(t)$ are B-spline basis functions defined on the knot vectors $\mathbf{t}$.

Then, the difference vector of each point is calculated as follows:

$$\delta_i^{(1)} = P_i^{(0)} - Q^{(0)}(t_i).$$

Furthermore, the new control points are obtained by

$$P_i^{(1)} = P_i^{(0)} + \delta_i^{(1)},$$

and the new curve $Q^{(1)}(t)$ can be constructed as follows.

$$Q^{(1)}(t) = \sum_{i=0}^{m-1} Q_i^{(1)} B_i(t),$$

where $Q_i^{(1)} = (\omega_i P_i^{(1)}, \omega_i)$. In Fig. 13(a), the black blocks are the initial control points, which are also LR pixels, and the yellow blocks are the new control points $P_i^{(1)}$. The green curve is $Q^{(1)}(t)$.

Similarly, for the kth iteration, we can obtain the difference vector $\delta_i^{(k)}$ and the new points $P_i^{(k)}$.

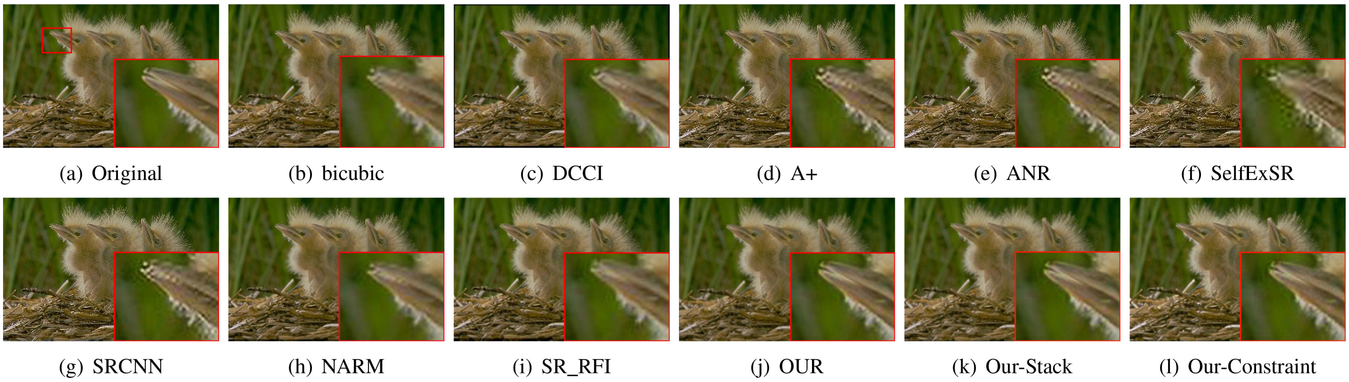
$$\delta_i^{(k)} = P_i^0 - Q^{(k)}(t_i) \tag{11}$$

$$P_i^{(k)} = P_i^{(k-1)} + \delta_i^{(k)} \tag{12}$$

Then, the kth curve is generated as

$$Q^{(k)}(t) = \sum_{i=0}^{m-1} Q_i^{(k)} N_i(t). \tag{13}$$

Fig. 13(b) shows the limited curve, the red curve, which is the result of fitting the data points $P_i^{(0)}$. Then, according to the


(a) Original (b) bicubic (c) DCCI (d) A+ (e) ANR (f) SelfExSR

(g) SRCNN (h) NARM (i) SR_RFI (j) OUR (k) Our-Stack (l) Our-Constraint

Fig. 15. Reconstructed results of Birds (×2).

limited curve, the HR pixels from the edges can be obtained, the edges are further reconstructed.

Since the iterative process of the geometric iterative method has a clear geometric meaning, geometrical constraint conditions can be added in each iterative process. In order to get more clearer edges, a gradient constraint term is given.

$$P_h^{(k)} = P_\nabla, P_\nabla = \frac{P_i^{(0)} + P_j^{(0)}}{2},$$

where $P_i^{(0)}$ and $P_j^{(0)}$ are two data points, namely initial control points, and P_∇ is the point in the gradient direction between edge points $P_i^{(0)}$ and $P_j^{(0)}$, which is regarded as the new control points $P_h^{(k)}$. i, j, h are the index of the control points, which represents the position of the control points. k represents the number of the iteration. The gradient constraint term is introduced to the edge interpolation iterative process. The edge interpolation iterative process with the gradient constraint can still run unattended.

III. Experimental Results

To demonstrate the efficiency of our proposed progressive-iterative approximation based SR method, we compare it with other state-of-the-art methods: bicubic, DCCI [15], ANR [25], A+ [26], SelfExSR [21], SRCNN [29], CCS [23], NARM [31], and SR_RFI [33]. Set5 [39], Set14 [40], Kodak [41], and B100 (consists 100 images randomly selected from BSDS500 [42]) are employed to evaluate the performance of the SISR methods in this paper. Moreover, the PSNR, structural similarity (SSIM), and feature similarity (FSIM) are employed to evaluate the objective quality of the SR results. Besides, the execution time of different SR algorithms are discussed. We also provide results of experimental cases with real and synthetic images in terms of interpolation performance.

A. Selection of Threshold N

Since the geometric iteration function converges for the blending curves and surfaces with normalized totally positive basis [35], we prescribe a threshold N to control whether to terminate the iteration procedure. When the number of iterations exceeds the prescribed threshold N, the progressive fitting iterations are terminated. Then, the limited curve or surface is obtained, and the HR image is reconstructed.

We derive the threshold N through testing. The peak signal-to-noise ratio (PSNR) values of three test image sets (Train91, Urban100, SunHays80) versus the iteration times are plotted. The LR counterparts are obtained by down-sampling the input images. The HR images were reconstructed from the LR images using the proposed method with different value of N. The changes in the average PSNR curves with the different value of N are shown in Fig. 14. It is clear that the higher PSNR values can be achieved when $N \in [30, 40]$. The PSNR values change slightly when $N > 40$. Thus, we set $N = 40$ in our implementation.

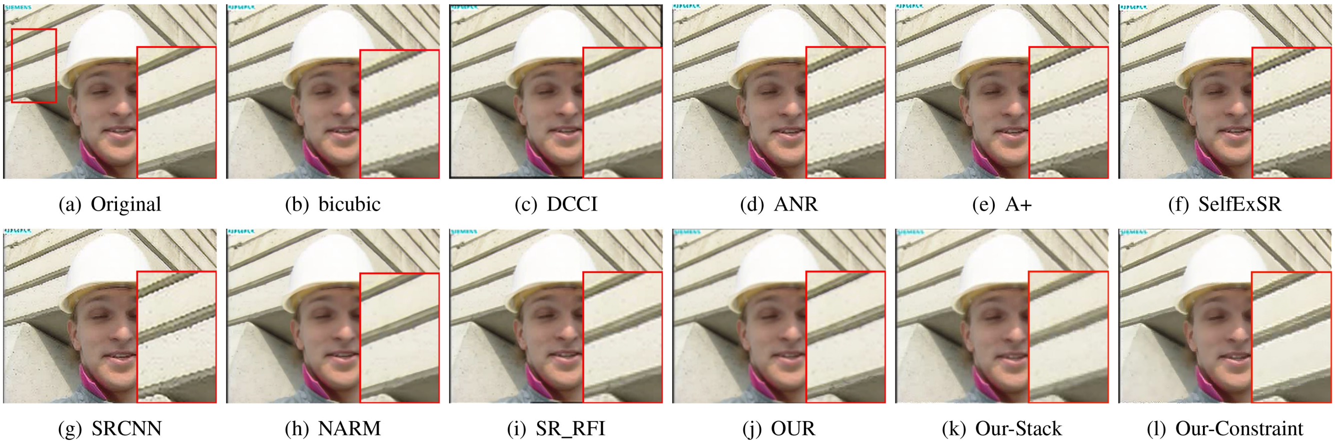
B. Experimental Configuration

The proposed SISR method is based on the geometric iterative model. The model can generate continuous limit surface, according to which the image can be amplified at any multiple via sampling process. In the experiments, HR images are down-sampled directly along both the horizontal and vertical directions for generating LR images by a factor of 2 and 3. Moreover, The human visual system is more sensitive to the luminance channel than the chrominance channel. We transform the color RGB images into YCbCr color space, and the SR process is only performed on the luminance channel (Y), and bicubic interpolation is used to interpolate the rest of the components (Cb and Cr which are usually low frequency channels and stores the chromatic information only). The three modified components are then combined and converted back to RGB to get the estimated color SR image displayed.

C. Subjective Evaluation

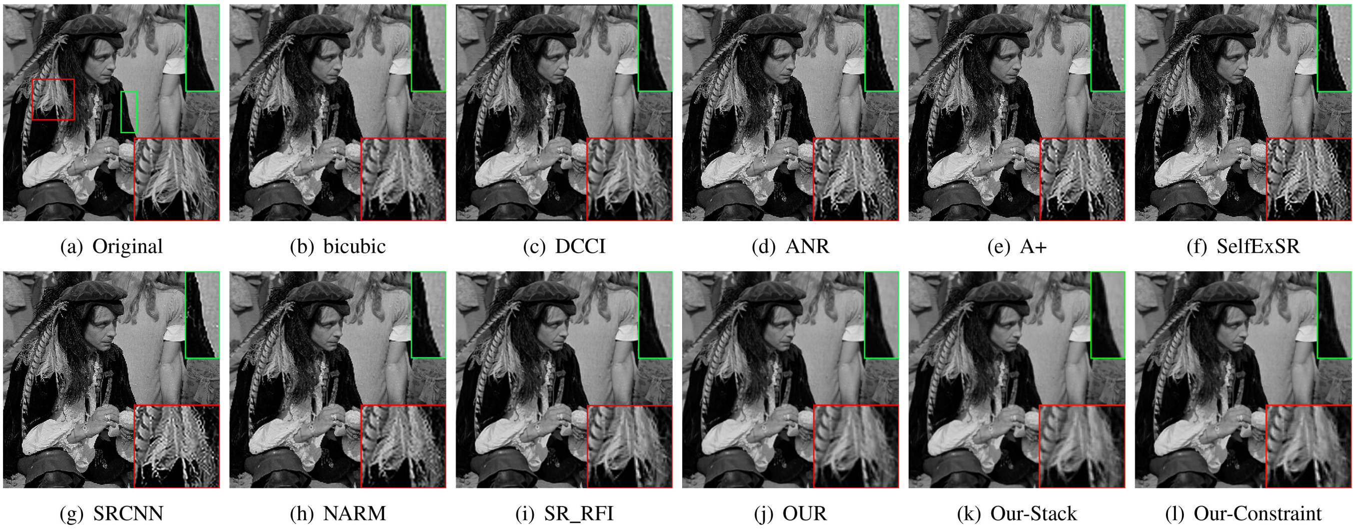
Fig. 15–Fig. 19 present the SR 2× results, and Fig. 20–Fig. 24 present the SR 3× results.

From Fig. 15–Fig. 18, we find that the bicubic, DCCI, ANR, A+, SelfExSR, and SRCNN methods produce unpleasing jaggy artifacts along the edges. In Fig. 17 and Fig. 18, the NARM also suffers from unpleasing jaggy artifacts along the edges. The NARM can generate clear edges in Fig. 15 and Fig. 16. However, NARM produces over-smoothed textures. The DCCI method produces unpleasant HR images with severe distortion


(a) Original (b) bicubic (c) DCCI (d) ANR (e) A+ (f) SelfExSR

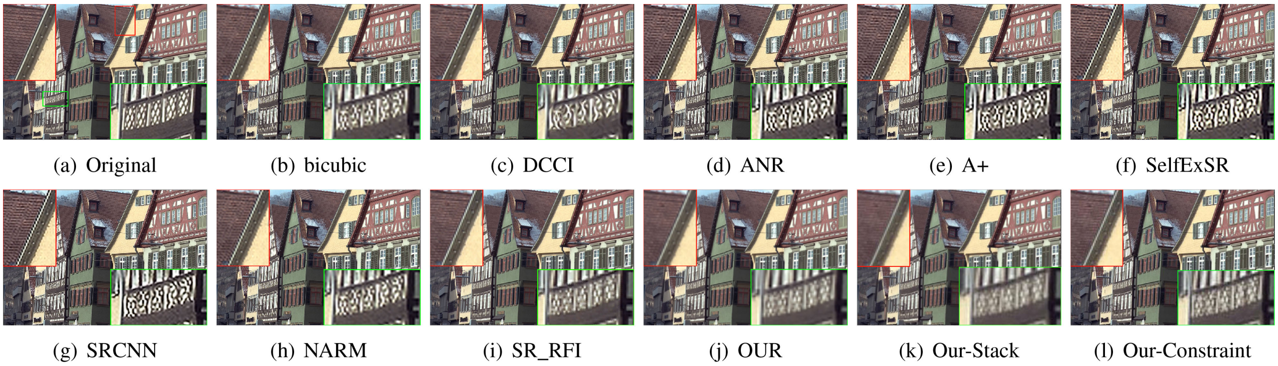
(g) SRCNN (h) NARM (i) SR_RFI (j) OUR (k) Our-Stack (l) Our-Constraint

Fig. 16. Reconstructed results of Foreman (×2).


(a) Original (b) bicubic (c) DCCI (d) ANR (e) A+ (f) SelfExSR

(g) SRCNN (h) NARM (i) SR_RFI (j) OUR (k) Our-Stack (l) Our-Constraint

Fig. 17. Reconstructed results of Man (×2).

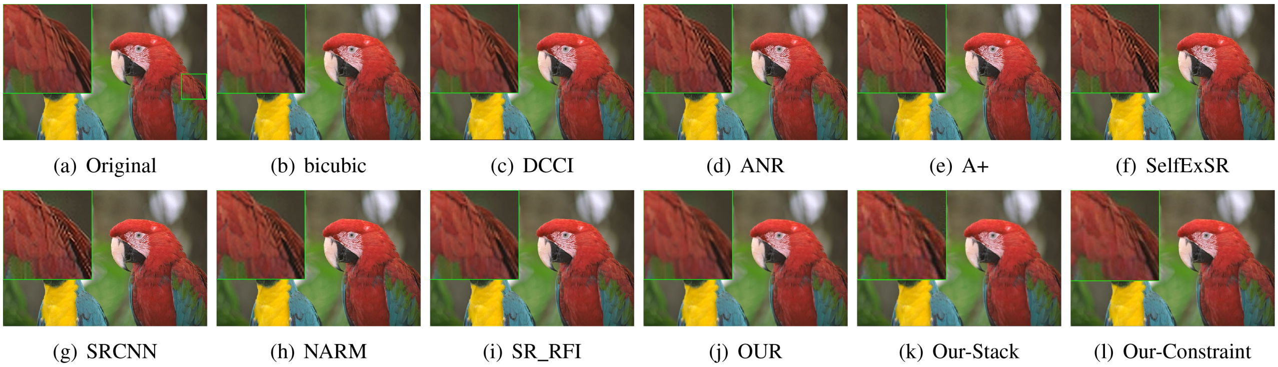

(a) Original (b) bicubic (c) DCCI (d) ANR (e) A+ (f) SelfExSR

(g) SRCNN (h) NARM (i) SR_RFI (j) OUR (k) Our-Stack (l) Our-Constraint

Fig. 18. Reconstructed results of Building (×2).

and unnatural textural details. In Fig. 18, bicubic, DCCI, ANR, A+, SelfExSR, SRCNN, and NARM also suffer from the distortion. SR_RFI generates more natural textures but yield jaggy edge artifacts. Our method produces clear edges and details with fewer artifacts.

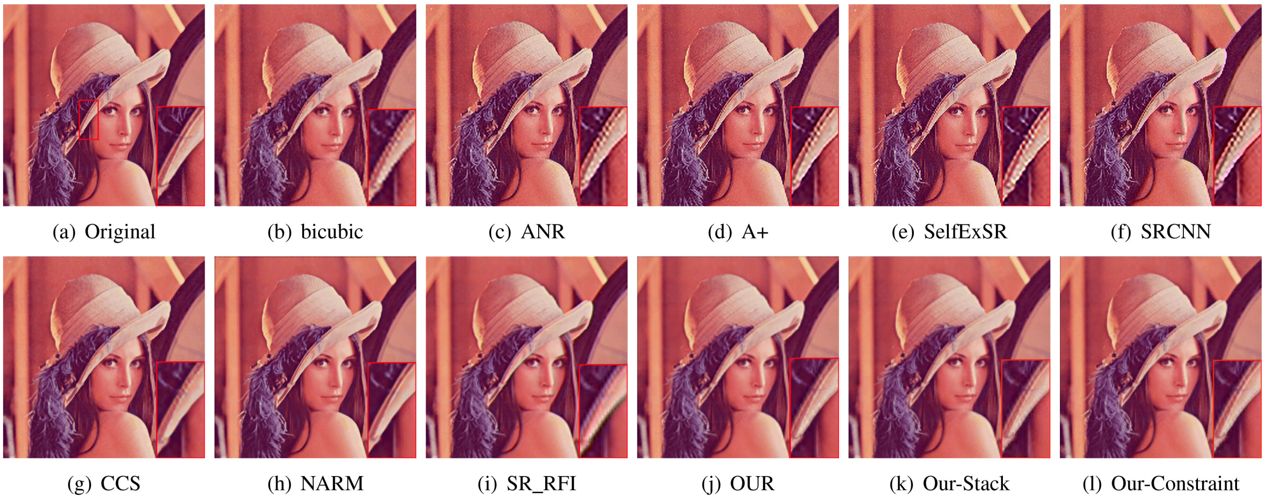
In Fig. 19, there are severe unpleasing artifacts in the HR images generated by bicubic, DCCI, ANR, A+, SelfExSR, SRCNN, NARM and SR_RFI. Though the HR image has slightly blur, the HR image generated by the proposed method has high quality without artifacts.



(a) Original (b) bicubic (c) DCCI (d) ANR (e) A+ (f) SelfExSR

(g) SRCNN (h) NARM (i) SR_RFI (j) OUR (k) Our-Stack (l) Our-Constraint

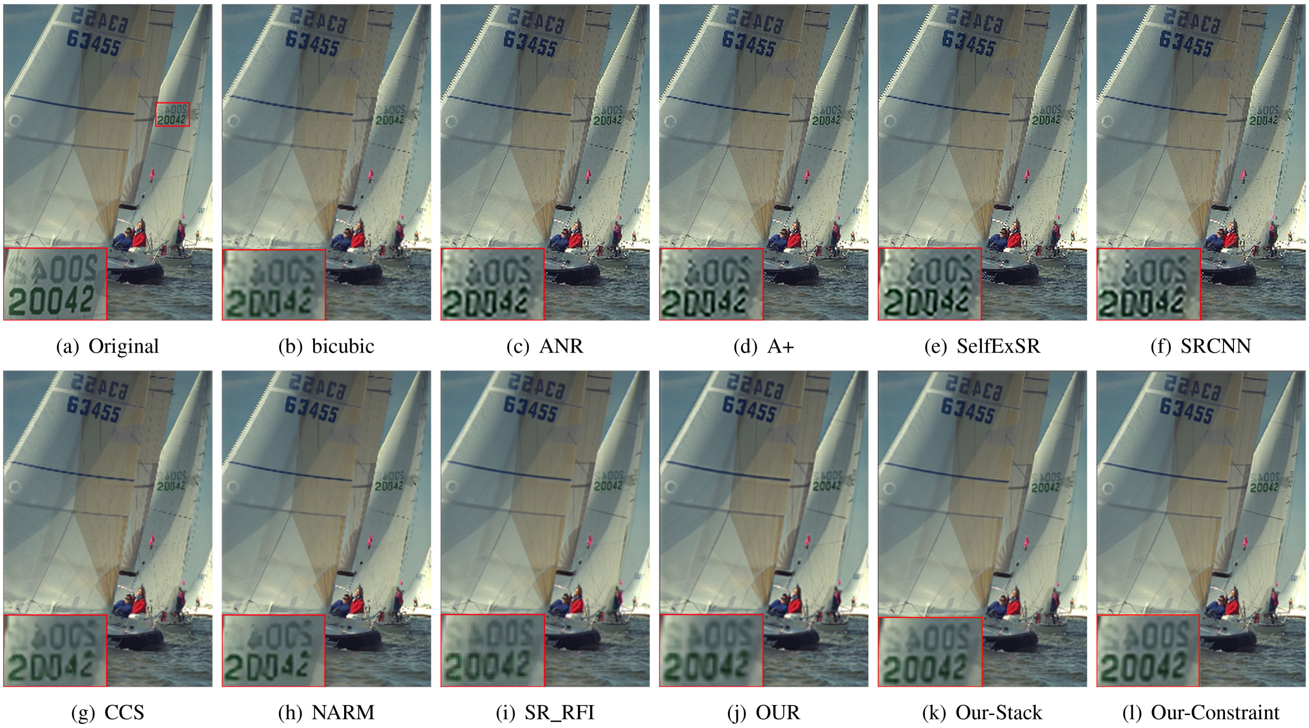
Fig. 19. Reconstructed results of Parrots (×2).



(a) Original (b) bicubic (c) ANR (d) A+ (e) SelfExSR (f) SRCNN

(g) CCS (h) NARM (i) SR_RFI (j) OUR (k) Our-Stack (l) Our-Constraint

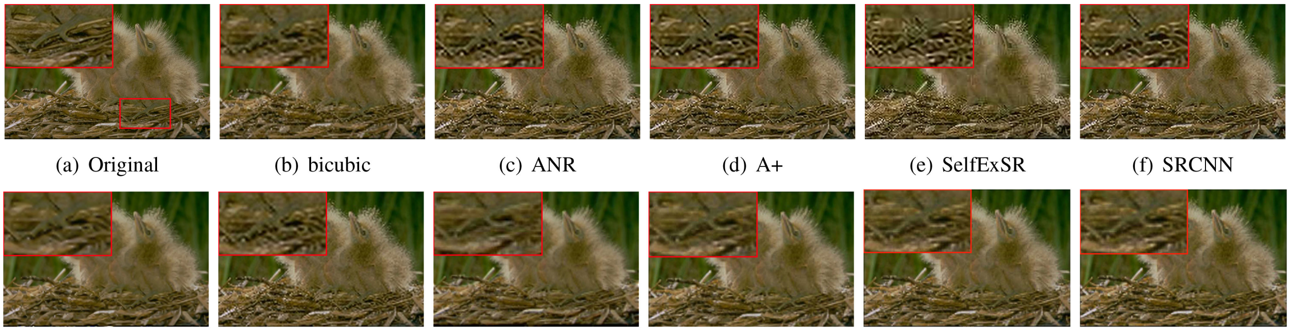
Fig. 20. Reconstructed results of Lena (×3).



(a) Original (b) bicubic (c) ANR (d) A+ (e) SelfExSR (f) SRCNN

(g) CCS (h) NARM (i) SR_RFI (j) OUR (k) Our-Stack (l) Our-Constraint

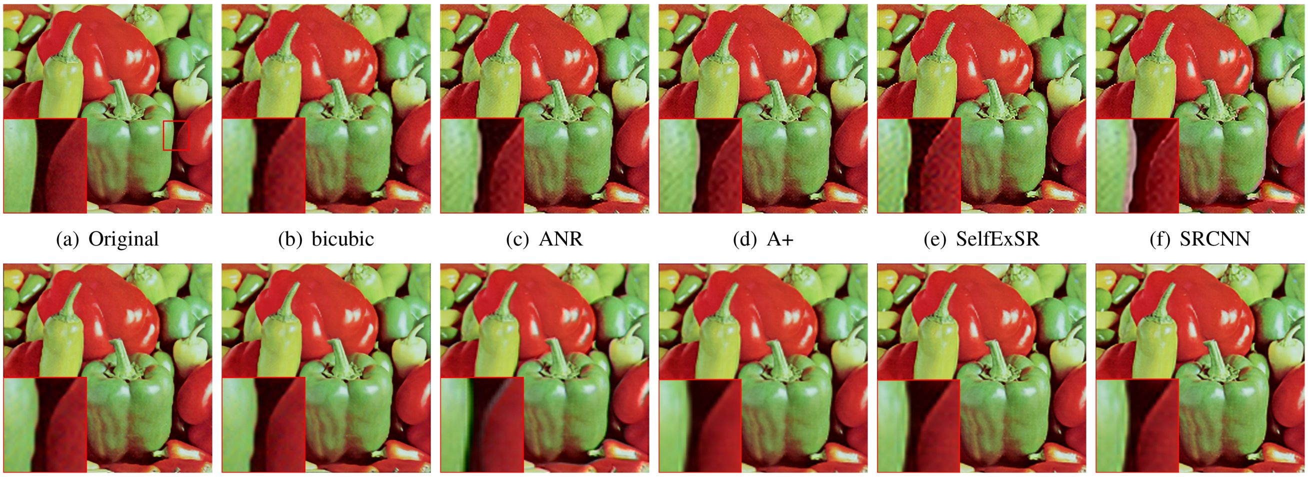
Fig. 21. Reconstructed results of Boats (×3).



(a) Original (b) bicubic (c) ANR (d) A+ (e) SelfExSR (f) SRCNN

(g) CCS (h) NARM (i) SR_RFI (j) OUR (k) Our-Stack (l) Our-Constraint

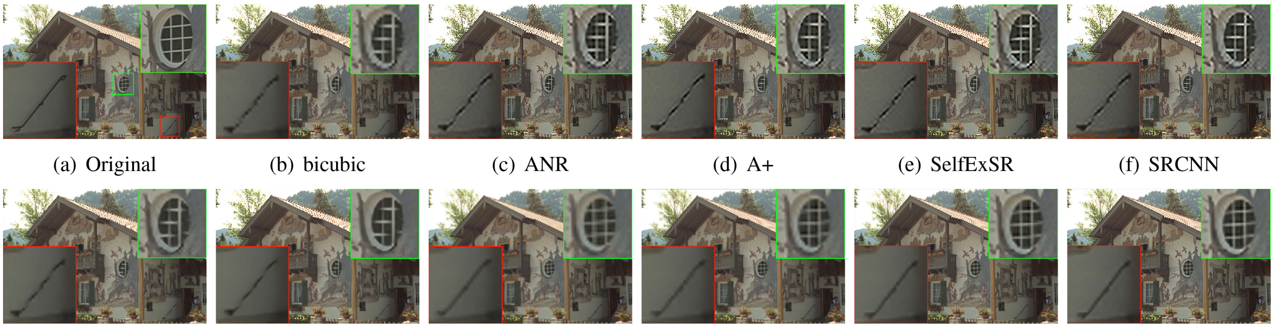
Fig. 22. Reconstructed results of Birds (×3).



(a) Original (b) bicubic (c) ANR (d) A+ (e) SelfExSR (f) SRCNN

(g) CCS (h) NARM (i) SR_RFI (j) OUR (k) Our-Stack (l) Our-Constraint

Fig. 23. Reconstructed results of Pepper (×3).



(a) Original (b) bicubic (c) ANR (d) A+ (e) SelfExSR (f) SRCNN

(g) CCS (h) NARM (i) SR_RFI (j) OUR (k) Our-Stack (l) Our-Constraint

Fig. 24. Reconstructed results of House (×3).

As shown in Figs. 20 and 23, the HR images generated by bicubic, ANR, A+, SelfExSR, and CCS suffer from severe jaggy artifacts along the edges. SRCNN and SR_RFI produce obvious aliasing artifacts. The results of NARM have clear and sharp edges, but the texture regions are over-smoothed, and some details are lost. The proposed method can recover satisfactory edges and more details.

As shown in Fig. 21, the HR images obtained with the bicubic, ANR, A+, SelfExSR, SRCNN, CCS, and NARM methods suffer from severe jaggy artifacts. Meanwhile, the images from the bicubic, ANR, A+, SelfExSR, and SRCNN methods also suffer from distortion. The results achieved with CCS and NARM are over-smoothed. SR_RFI can efficiently preserve the details, but the edges suffer from block artifacts. Additionally, the image constructed by the presented method shows an improvement in quality relative to the compared methods.

As shown in Fig. 22, the results achieved by the bicubic, ANR, A+, SelfExSR, SRCNN and NARM methods show artifacts, and

TABLE I
OBJECTIVE QUALITY ASSESSMENT OF DIFFERENT METHODS

Image	scale	bicubic	DCCI	ANR	A+	SelfExSR	SRCNN	CCS	NARM	SR_RFI	Proposed	Our-Stack	Our-Constraint
Birds	×2	29.6129	28.6399	27.7897	27.4140	27.1326	27.0015	-	**32.8522**	29.8006	31.4285	31.4139	31.4279
		0.8391	0.8978	0.8042	0.7942	0.7900	0.7855	-	**0.9063**	0.8454	0.8651	0.8526	0.8766
		0.9142	0.9247	0.8940	0.8887	0.8815	0.8839	-	**0.9489**	0.9075	0.9153	0.9015	0.9135
Foreman	×2	29.2572	20.2340	29.8420	29.4271	27.8294	28.9610	-	30.6151	29.4138	33.1244	33.0133	**33.1273**
		0.9277	0.9409	0.9105	0.9074	0.8935	0.9002	-	0.9479	0.9215	0.9735	0.9687	**0.9771**
		0.9397	0.9195	0.9246	0.9247	0.9113	0.9148	-	0.9582	0.9265	0.9837	0.9295	**0.9845**
Man	×2	26.5740	24.3282	25.0780	24.7288	24.2559	24.1750	-	28.2684	26.9288	**31.5599**	30.8984	30.9904
		0.9127	0.9412	0.8669	0.8546	0.8489	0.8459	-	0.9373	0.9580	**0.9845**	0.9594	0.9604
		0.9432	0.9407	0.9320	0.9309	0.9304	0.9294	-	0.9674	0.9612	**0.9821**	0.9768	0.9788
Building	×2	21.6645	22.8165	20.3631	20.1386	19.4931	19.6755	-	23.2743	22.1683	**23.6559**	23.6512	23.6558
		0.8732	0.9015	0.8191	0.8076	0.7995	0.7989	-	0.8970	**0.9392**	0.9007	0.8953	0.9111
		0.9064	0.9392	0.8957	0.8948	0.8923	0.8929	-	0.9404	0.9372	0.9528	0.9469	**0.9458**
Parrots	×2	31.6259	27.9518	31.7794	31.3462	29.8614	30.9877	-	33.3353	32.1882	**34.0856**	34.0791	34.0791
		0.9742	0.9834	0.9555	0.9498	0.9482	0.9495	-	0.9836	0.9846	0.9809	0.9841	**0.9860**
		0.9794	0.9629	0.9794	0.9790	0.9746	0.9790	-	0.9873	0.9853	**0.9878**	0.9870	0.9873
Average	×2	27.7469	24.7940	26.9704	26.6109	25.7144	26.1601	-	29.6690	28.0999	30.7708	30.2112	**30.8536**
		0.9053	0.9329	0.8712	0.8627	0.8560	0.8560	-	0.9344	0.9297	0.9409	0.9320	**0.9422**
		0.9365	0.9374	0.9251	0.9236	0.9180	0.9200	-	0.9604	0.9435	**0.9643**	0.9483	0.9620
Lena	×3	27.2151	-	25.681	25.2190	24.750	24.892	30.7723	30.2720	30.8031	30.9246	30.9286	**30.9388**
		0.8614	-	0.8142	0.8043	0.6940	0.7984	0.9395	0.8633	0.9304	0.9515	0.9432	**0.9532**
		0.9139	-	0.8925	0.8912	0.8333	0.8880	0.9491	0.9591	0.9637	**0.9679**	0.9637	0.9677
Boats	×3	26.8332	-	27.8603	27.6892	24.5558	27.2446	29.0872	28.1954	27.3381	29.9670	29.9950	**30.0950**
		0.8452	-	0.8763	0.8753	0.7817	0.8725	0.9164	0.9185	0.8743	0.9304	0.9315	**0.9338**
		0.8853	-	0.9130	0.9145	0.8481	0.9126	0.9385	0.9355	0.8919	0.9504	0.9530	**0.9561**
Birds	×3	26.4211	-	27.1391	26.7872	24.0606	26.3783	28.6614	28.4933	27.1032	29.4601	29.4155	**29.4615**
		0.7203	-	0.7702	0.7623	0.6515	0.7549	0.7977	0.8082	0.7291	0.8089	0.8005	**0.8095**
		0.8616	-	0.8768	0.8688	0.8217	0.8635	0.8835	0.8756	0.8463	0.8836	0.8704	**0.8995**
Pepper	×3	26.0731	-	26.4712	26.2210	24.5580	25.7366	31.4873	28.560	30.211	31.5055	31.5811	**31.7808**
		0.8915	-	0.8497	0.8444	0.8269	0.8354	0.9524	0.8705	0.9411	0.9666	0.9680	**0.9698**
		0.9150	-	0.9001	0.9012	0.8790	0.8955	0.9698	0.9670	0.9609	0.9748	0.9755	**0.9788**
House	×3	22.7526	-	22.7600	22.5406	20.2080	22.0734	24.4120	23.8204	23.8903	24.9487	24.9583	**24.9588**
		0.7520	-	0.8032	0.8003	0.6658	0.7919	0.8461	0.8444	0.7971	0.8773	0.8788	**0.8808**
		0.8671	-	0.8752	0.8765	0.8239	0.8722	0.9114	0.9027	0.8843	0.9284	0.9286	**0.9326**
Average	×3	25.8590	-	25.9823	25.6914	23.6264	25.2649	28.8840	27.8686	27.8691	29.3611	29.3757	**29.4469**
		0.8140	-	0.8227	0.8173	0.7239	0.8106	0.8904	0.8609	0.8544	0.9069	0.9044	**0.9094**
		0.8885	-	0.8915	0.8904	0.8412	0.8863	0.9304	0.9279	0.9094	0.9410	0.9382	**0.9469**

textures are heavily twisted. The CCS method is prone to blurring the details. SR_RFI and the presented method can efficiently preserve the vivid details.

In Fig. 24, the HR images generated by bicubic, ANR, A+, SelfExSR, SRCNN, CCS, and NARM are discontinuous at the edge and suffer from the information loss. SR_RFI produces block artifacts along the edge. Comparing with other methods, the proposed method gets a high quality HR image.

High quality edge localization can further improve the super-resolution reconstruction results. We improve edge localization by regarding the stack of edges detected by NSCT and edges detected by Canny as the final edges. Then, the improved HR images are obtained, shown as Fig. 15–Fig. 24(k). The edge reconstruction can be ameliorated through improving edge localization and fully utilizing the edge information. In the process of edges interpolation, based on the improved the edge localization, edge constraint conditions can be added to improve the reconstruction quality. We add the gradient constraint term to make the reconstructed edges more sharp. The reconstructed results are shown as Fig. 15–Fig. 24(l). From Fig. 15–Fig. 24(j)(k)(l), the visual effects are improved for specific images (such as Fig. 21, Fig. 23, and Fig. 24).

D. Objective Evaluation

In this subsection, Set5, Set14, Kodak, and B100 are employed to evaluate the performance of upscaling factors of 2 and 3. The PSNR, SSIM, and FSIM values achieved by these methods on part of the images from the four datasets are listed in Table I. Table II and Table III show the objective data of different methods on Set5 and Set14. From top to bottom, each line represents the PSNR, the SSIM, and the FSIM respectively. “-” indicates that related methods failed to magnify with the corresponding factor.

These results clearly demonstrate that the developed algorithm is notably better compared with the other methods in terms of the three quantitative assessments. Specifically, the proposed method yields the best quantitative results for most of the test images. When the image is magnified with higher magnification, the objective data of the proposed method are higher than the state-of-the-art, 0.46–5.98 dB PSNR better than other methods on average. Thus, the proposed SR method achieves the best performance among all the compared methods. For the proposed method with the improved edge localization and the improved edge reconstruction, the quality of generated HR results is ameliorated in terms of the objective evaluation. In fact, in the proposed method, the image can be regarded as a discrete data point set from the realistic surface, and the geometric iterative model can generate the continuous limit surface of realistic surface based on the given LR image. Thus, the loss of the high frequency information is restored. Specifically, when the image is magnified with higher scale factors, the proposed method has obvious superiority. The higher values in SSIM and FSIM, compared with other methods, illustrate that the proposed method can preserve the structure of the original image. Moreover, comparing with other methods, the higher PSNR illustrates

TABLE II
OBJECTIVE QUALITY ASSESSMENT OF DIFFERENT METHODS ON SET5

Image	scale	bicubic	DCCI	ANR	A+	SelfExSR	SRCNN	CCS	NARM	SR_RFI	Proposed	Our-Stack	Our-Constraint
		33.198	20.992	31.917	31.570	30.564	31.376	-	36.776	36.757	**36.779**	36.665	36.722
baby	×2	0.967	0.978	0.951	0.946	0.926	0.945	-	0.949	0.984	**0.984**	0.950	0.979
		0.976	0.942	0.971	0.971	0.962	0.971	-	0.990	0.992	**0.992**	0.976	0.980
		31.924	23.674	31.069	31.756	29.565	30.433	-	**38.573**	36.333	36.485	36.490	36.512
bird	×2	0.941	0.965	0.930	0.927	0.870	0.922	-	0.977	0.979	**0.979**	0.966	0.968
		0.947	0.926	0.938	0.935	0.902	0.932	-	0.973	0.980	**0.980**	0.964	0.976
		24.584	22.403	23.454	23.100	21.836	22.459	-	**30.167**	29.204	29.460	29.667	29.690
butterfly	×2	0.880	0.930	0.852	0.851	0.714	0.828	-	0.954	0.950	**0.954**	0.945	0.948
		0.873	0.899	0.833	0.846	0.721	0.826	-	0.936	0.938	**0.939**	0.921	0.929
		32.081	23.252	30.493	30.126	30.081	29.783	-	34.423	34.451	**34.847**	34.788	34.804
head	×2	0.804	0.831	0.754	0.742	0.723	0.732	-	0.856	0.857	**0.857**	0.843	0.850
		0.918	0.885	0.885	0.878	0.878	0.874	-	0.931	0.938	**0.939**	0.927	0.933
		28.191	24.749	26.823	26.408	25.429	25.924	-	34.107	**34.185**	32.076	32.264	32.555
women	×2	0.915	0.949	0.896	0.892	0.822	0.883	-	0.961	0.963	0.951	0.957	**0.963**
		0.926	0.926	0.909	0.906	0.865	0.901	-	0.965	**0.968**	0.958	0.962	0.966
		29.995	23.014	28.750	28.392	27.495	27.995	-	**34.809**	34.186	33.924	33.975	34.056
Average	×2	0.901	0.930	0.876	0.871	0.811	0.862	-	0.939	**0.946**	0.944	0.932	0.942
		0.928	0.915	0.907	0.907	0.865	0.900	-	0.959	0.963	**0.963**	0.950	0.957
		29.035	-	27.735	27.369	26.890	27.221	32.962	28.458	32.950	**33.001**	32.875	32.877
baby	×3	0.894	-	0.867	0.861	0.833	0.858	0.953	0.887	0.952	**0.953**	0.946	0.949
		0.934	-	0.919	0.918	0.904	0.917	0.976	0.962	0.973	**0.976**	0.967	0.972
		27.232	-	26.084	25.585	25.147	25.304	31.659	31.801	31.622	31.893	**31.896**	31.886
bird	×3	0.845	-	0.812	0.804	0.729	0.795	0.937	0.929	0.932	0.943	**0.944**	0.940
		0.885	-	0.865	0.860	0.824	0.856	0.935	0.932	0.933	0.935	**0.937**	0.929
		20.472	-	19.237	18.934	18.054	18.499	24.371	**24.773**	23.229	23.320	23.442	23.460
butterfly	×3	0.738	-	0.691	0.700	0.551	0.664	0.870	0.891	0.810	**0.892**	0.876	0.879
		0.764	-	0.711	0.727	0.630	0.708	0.874	**0.877**	0.873	0.872	0.869	0.872
		29.233	-	27.682	27.322	27.568	26.884	32.075	31.253	31.746	**32.512**	32.499	32.508
head	×3	0.711	-	0.651	0.639	0.637	0.621	0.785	0.779	0.815	**0.817**	0.800	0.812
		0.871	-	0.843	0.834	0.835	0.826	0.900	0.881	0.900	**0.900**	0.896	0.899
		23.964	-	22.488	22.089	21.406	21.591	**28.700**	28.275	28.317	27.823	27.829	27.834
women	×3	0.813	-	0.778	0.774	0.680	0.762	0.902	0.909	**0.919**	0.884	0.887	0.888
		0.862	-	0.839	0.836	0.785	0.827	**0.921**	0.917	0.918	0.906	0.908	0.911
		25.987	-	24.645	24.259	23.813	23.899	**29.953**	28.912	29.572	29.710	29.708	29.713
Average	×3	0.800	-	0.759	0.755	0.686	0.740	0.889	0.879	0.885	**0.898**	0.891	0.894
		0.963	-	0.835	0.835	0.795	0.826	**0.921**	0.913	0.919	0.918	0.915	0.917

that rich information can be restored. This result demonstrates that the proposed SR reconstruction method based on the geometric iterative model is quiet efficient for preserving the image information and the structure.

E. Execution Time of the Algorithm

We also provide execution time of the algorithm, compared against the others. Given 256×256 images, for factors 2,3,4, the average execution time of different SR methods are shown in Table IV. The bicubic and the DCCI are simple interpolation-based methods. This two kinds of methods have lower execution time, but the visual effects are not satisfying. The NARM takes the most running time. Learning-based methods cost lots of time in the training stage because they rely on data sets and much exercises, but they spend less time in the test stage. In the Table IV, the execution time of ANR, A+, SRCNN refer to the test time. Compare to the ANR, A+, and SRCNN, the SelfExSR is self-learning method so that its execution time is long. Comparatively, the SR_RFI and the proposed method are competitive in terms of the execution time.

F. Results on Synthetic and Real Images

To further demonstrate the interpolation performance of the proposed method, we also do experiments on synthetic and real images. Fig. 26 and Fig. 27 are synthetic images. Fig. 28 is a real image. In Fig. 26, the proposed method can obtain the satisfying HR image without distortion and discontinuity on edges. As shown in Fig. 27, the proposed method can get better textures without distortion. From Fig. 28, different from the state-of-the-art methods, our method can obtain high quality image with clear and undistorted edges. Table V shows the objective data of different methods. Compared to other methods, the proposed method achieves the best performance on synthetic and real images.

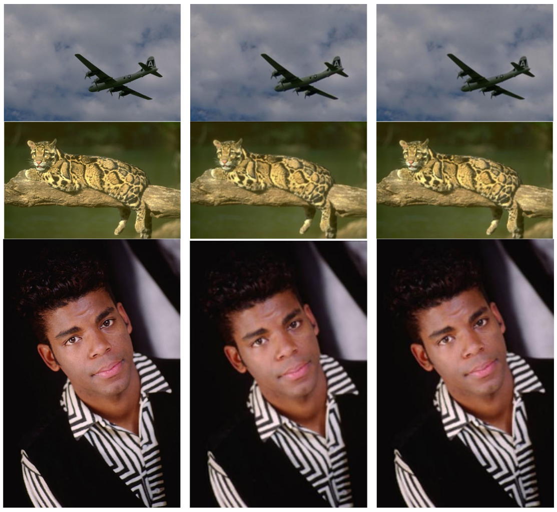

(a) HR (b) Bicubic (c) Fractal

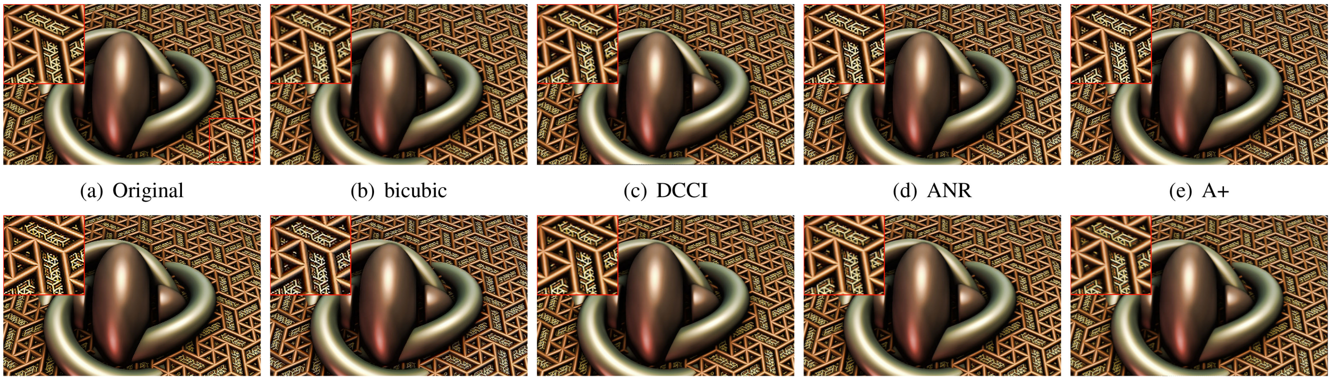
Fig. 25. Results generated by different control points added methods.

TABLE III
OBJECTIVE QUALITY ASSESSMENT OF DIFFERENT METHODS ON SET14

Image	scale	bicubic	DCCI	ANR	A+	SelfExSR	SRCNN	CCS	NARM	SR_RFI	Proposed	Our-Stack	Our-Constraint
baboon	×3	20.406	-	19.019	18.706	17.016	18.264	21.726	21.467	21.538	**22.907**	22.826	22.848
		0.610	-	0.540	0.572	0.486	0.511	0.721	0.539	0.710	0.781	0.785	**0.788**
		0.836	-	0.795	0.793	0.765	0.784	0.877	0.873	0.875	0.912	**0.914**	0.913
barbara	×3	23.004	-	21.356	20.743	20.568	20.621	24.791	24.572	24.737	**25.876**	25.802	25.822
		0.754	-	0.707	0.692	0.668	0.689	0.841	0.760	0.837	0.865	0.868	**0.880**
		0.873	-	0.834	0.831	0.810	0.827	0.928	0.923	0.923	**0.944**	**0.944**	0.943
bridge	×3	21.236	-	21.096	20.779	18.471	15.664	24.473	22.856	23.028	**25.150**	25.001	25.142
		0.700	-	0.653	0.639	0.567	0.316	0.821	0.640	0.809	0.855	0.854	**0.857**
		0.859	-	0.836	0.835	0.796	0.720	0.914	0.905	0.906	**0.929**	0.925	0.928
coastguard	×3	23.660	-	22.354	22.018	21.267	21.638	25.406	24.984	25.068	**26.032**	25.905	26.002
		0.519	-	0.472	0.458	0.417	0.446	0.622	0.622	0.599	0.605	0.608	**0.622**
		0.767	-	0.763	0.754	0.742	0.748	0.781	0.783	**0.799**	0.755	0.752	0.755
comic	×3	20.030	-	18.513	18.154	17.342	17.618	22.433	21.990	22.131	**22.637**	22.035	22.308
		0.585	-	0.539	0.527	0.441	0.504	0.721	0.730	0.733	**0.786**	0.689	0.699
		0.761	-	0.733	0.729	0.680	0.717	**0.830**	0.828	0.814	0.788	0.787	0.780
face	×3	29.200	-	27.677	27.322	27.309	26.917	32.075	31.247	31.638	**32.427**	32.212	32.248
		0.710	-	0.651	0.637	0.624	0.623	0.785	0.781	0.786	**0.793**	0.792	0.788
		0.874	-	0.845	0.836	0.835	0.829	0.902	0.883	**0.908**	0.885	0.874	0.877
flowers	×3	23.703	-	22.216	21.891	21.048	21.368	**26.993**	26.742	26.351	26.748	26.798	26.813
		0.704	-	0.652	0.644	0.558	0.623	0.810	0.809	0.812	**0.819**	0.792	0.811
		0.809	-	0.780	0.777	0.725	0.765	0.867	0.856	0.908	**0.909**	0.840	0.866
foreman	×3	26.304	-	26.228	25.703	25.231	25.251	29.587	29.247	28.265	**29.595**	29.414	29.500
		0.853	-	0.819	0.814	0.688	0.803	0.918	0.921	0.925	**0.927**	0.896	0.914
		0.884	-	0.861	0.859	0.840	0.845	0.931	0.927	0.929	**0.932**	0.898	0.899
lena	×3	27.215	-	25.681	25.219	24.750	24.892	30.772	30.272	30.803	30.925	30.929	**30.939**
		0.861	-	0.814	0.804	0.694	0.798	0.939	0.863	0.930	0.952	0.943	**0.953**
		0.914	-	0.892	0.892	0.833	0.888	0.949	0.959	0.963	0.968	0.963	**0.968**
man	×3	23.595	-	22.148	21.811	20.582	21.296	26.385	25.853	25.832	26.526	**26.557**	26.553
		0.774	-	0.721	0.711	0.508	0.697	0.877	0.750	0.865	0.899	0.900	**0.908**
		0.877	-	0.849	0.849	0.754	0.843	0.937	0.932	0.933	0.942	0.943	**0.948**
monarch	×3	25.600	-	24.243	23.979	23.094	23.547	**29.795**	29.575	28.643	28.693	28.379	28.553
		0.918	-	0.897	0.894	0.862	0.888	0.968	0.933	0.962	0.968	0.966	**0.968**
		0.920	-	0.903	0.906	0.866	0.902	**0.970**	0.965	0.965	0.958	0.958	0.960
pepper	×3	26.060	-	26.471	26.221	24.558	25.736	31.487	28.560	30.211	31.506	31.581	**31.781**
		0.891	-	0.849	0.844	0.826	0.835	0.952	0.870	0.941	0.967	0.968	**0.970**
		0.914	-	0.900	0.901	0.879	0.895	0.970	0.966	0.960	0.975	0.976	**0.979**
ppt3	×3	19.877	-	19.157	18.905	18.223	18.248	23.146	22.276	22.664	**23.194**	23.136	23.189
		0.855	-	0.834	0.834	0.785	0.818	0.935	0.892	0.926	0.935	0.935	**0.937**
		0.837	-	0.826	0.841	0.786	0.820	0.250	0.920	0.907	**0.920**	0.901	0.915
zebra	×3	21.983	-	20.747	20.368	19.599	19.757	26.661	26.022	25.711	26.669	26.773	**26.933**
		0.780	-	0.743	0.731	0.679	0.720	0.896	0.798	0.892	0.898	**0.899**	0.900
		0.851	-	0.831	0.830	0.811	0.823	0.945	0.942	0.943	0.923	0.923	**0.945**
average	×3	23.705	-	22.636	22.272	24.425	21.486	26.762	26.118	26.187	**27.063**	26.953	27.045
		0.751	-	0.706	0.696	0.628	0.662	0.843	0.779	0.837	**0.861**	0.850	0.857
		0.855	-	0.832	0.830	0.794	0.814	0.910	0.904	0.909	**0.910**	0.890	0.905

TABLE IV
EXECUTION TIME COMPARISON OF DIFFERENT ALGORITHMS

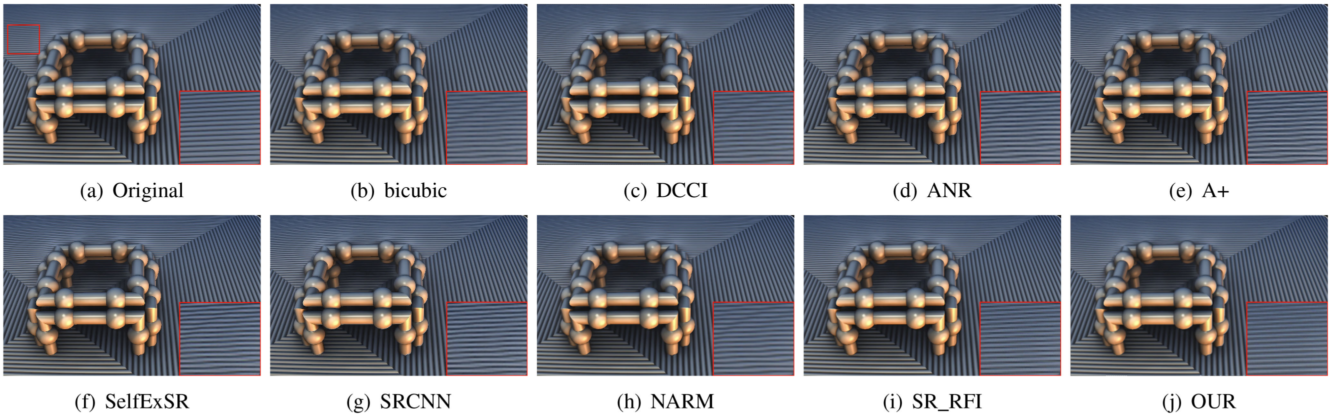
Image	bicubic	DCCI	ANR	A+	SelfExSR	SRCNN	NARM	SR_RFI	Proposed	Our-Stack	Our-Constraint
512 × 512	0.1097	1.4592	1.2225	1.1590	116.4637	1.9231	255.7483	166.2677	62.0030	62.3450	65.5469
768 × 768	0.1135	-	1.8003	1.6122	207.1107	1.9302	333.8700	167.6868	136.6943	136.6554	139.2221
1024 × 1024	0.1197	-	2.2540	2.2493	326.9340	1.9303	458.0184	169.7162	170.637	171.2215	180.2688



(a) Original (b) bicubic (c) DCCI (d) ANR (e) A+

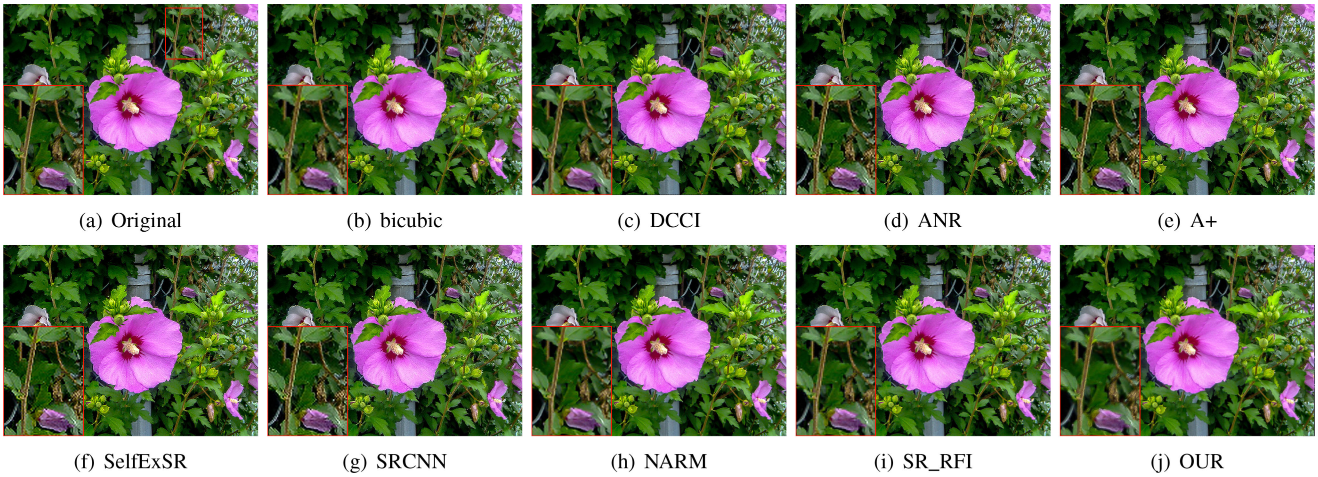
(f) SelfExSR (g) SRCNN (h) NARM (i) SR_RFI (j) OUR

Fig. 26. Reconstructed results (×2).


(a) Original (b) bicubic (c) DCCI (d) ANR (e) A+

(f) SelfExSR (g) SRCNN (h) NARM (i) SR_RFI (j) OUR

Fig. 27. Reconstructed results (×2).


(a) Original (b) bicubic (c) DCCI (d) ANR (e) A+

(f) SelfExSR (g) SRCNN (h) NARM (i) SR_RFI (j) OUR

Fig. 28. Reconstructed results (×2).

TABLE V

OBJECTIVE DATA COMPARISON OF DIFFERENT ALGORITHMS ON SYNTHETIC AND REAL IMAGES

Image	scale	bicubic	DCCI	ANR	A+	SelfExSR	SRCNN	NARM	SR_RFI	Proposed
Synthetic1	×2	22.7042	24.7688	21.3979	21.1484	20.5506	20.5270	**26.8252**	22.4226	25.1597
		0.9562	0.9844	0.9396	0.9354	0.9308	0.9332	0.9859	0.9607	**0.9862**
		0.9499	0.9862	0.9456	0.9464	0.9463	0.9459	0.9875	0.9503	**0.9880**
Synthetic2	×2	30.5004	29.5299	29.3094	29.0000	28.9212	28.8910	33.6619	29.5923	**33.6624**
		0.9446	0.9720	0.9312	0.9288	0.9263	0.9272	0.9716	0.9740	**0.9740**
		0.9566	0.9901	0.9527	0.9326	0.9521	0.9516	**0.9953**	0.9436	0.9925
Real1	×2	26.6454	24.1342	25.0187	24.6110	24.2761	23.9989	25.3616	26.9425	**27.7312**
		0.9264	0.8653	0.8801	0.8667	0.8752	0.8590	0.8709	**0.9690**	0.9463
		0.9534	0.9179	0.9428	0.9420	0.9398	0.9402	0.9267	0.9476	**0.9679**

G. Effects of Different Methods of Increased Control Points

In this section, we conduct an experiment to illustrate the impact that is exerted by different methods of increased control points. Here, we take the bicubic and fractal interpolation algorithms as examples. Fig. 25(b) shows the HR results generated by our method after increasing control points via the bicubic, and Fig. 25(c) shows the HR results by the proposed method after increasing control points via fractal interpolation method.

As shown in Fig. 25, the HR results achieved by the proposed method with bicubic and fractal control points added approaches both preserve textures and clear edges. There are few differences between the two reconstructed images. Thus, we employ the bicubic algorithm with simple form to insert control points in this paper.

IV. CONCLUSION

In this paper, a novel single image SR method based on progressive-iterative approximation is presented. By adjusting the control points of the curve and surface iteratively, the limited curve and surface that interpolate the given data point set are obtained. According to the limited curve and surface, we can obtain the required HR image. First, the NSCT algorithm is employed to detect textures and edges such that more textures and edge structures can be acquired, and the LR image is divided into a smooth region, a texture region and edges. Second, considering the features in different regions, a hybrid interpolation scheme based on curves and surfaces is proposed to interpolate the LR image for maintaining the textural details and clear edges. The NURBS surface geometric iteration and the NURBS curve geometric iteration are used for the smooth and texture regions and the edges, respectively. Specifically, in order to preserve more details in the texture region, we insert new control points before performing the NURBS surface geometric iteration procedure. Finally, based on the limited curve and surface, the HR image is obtained. The experimental results show that the proposed single-image SR can achieve competitive results and produce high-quality SR images with rich textures and clear edges.

References

[1] H. Takeda, S. Farsiu, and P. Milanfar, "Kernel regression for image processing and reconstruction," *IEEE Trans. Image Process.*, vol. 16, no. 2, pp. 349–366, Feb. 2007.

[2] A. Marquina and S. J. Osher, "Image super-resolution by TV-regularization and Bregman iteration," *J. Sci. Comput.*, vol. 37, no. 3. pp. 367–382, 2008.

[3] F. Shi, J. Cheng, L. Wang, and P.-T. Yap, "LRTV: MR image super-resolution with low-rank and total variation regularizations," *IEEE Trans. Med. Imag.*, vol. 34, no. 12, pp. 2459–2466, Dec. 2015.

[4] L. Wang, S. Xiang, G. Meng, H. Wu, and C. Pan, "Edge-directed single-image super-resolution via adaptive gradient magnitude selfinterpolation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 8, pp. 1289–1299, Aug. 2013.

[5] Q. Yan, Y. Xu, X. Yang, and T. Q. Nguyen, "Single image super-resolution based on gradient profile sharpness," *IEEE Trans. Image Process.*, vol. 25, no. 10, pp. 3187–3202, Oct. 2015.

[6] Q. Song *et al.*, "Fast image super-resolution via local adaptive gradient field sharpening transform," *IEEE Trans. Image Process.*, vol. 27, no. 4, pp. 1966–1980, Apr. 2018.

[7] W. Dong, L. Zhang, G. Shi, and X. Li, "Nonlocally centralized sparse representation for image restoration," *IEEE Trans. Image Process.*, vol. 22, no. 4, pp. 1620–1630, Apr. 2013.

[8] J. Liu, W. Yang, X. Zhang, and Z. Guo, "Retrieval compensated group structured sparsity for image super-resolution," *IEEE Trans. Multimedia*, vol. 19, no. 2, pp. 302–316, Feb. 2017.

[9] K. Zhang, D. Tao, X. Gao, X. Li, and Z. Xiong, "Learning multiple linear mappings for efficient single image super-resolution," *IEEE Trans. Image Process.*, vol. 24, no. 3, pp. 846–861, Mar. 2015.

[10] H. Chen, X. He, L. Qing, and Q. Teng, "Single image super-resolution via adaptive transform-based nonlocal self-similarity modeling and learning-based gradient regularization," *IEEE Trans. Multimedia*, vol. 19, no. 8, pp. 1702–1717, Aug. 2017.

[11] J. Jiang *et al.*, "Single image super-resolution via locally regularized anchored neighborhood regression and nonlocal means," *IEEE Trans. Multimedia*, vol. 19, no. 1, pp. 15–26, Jan. 2017.

[12] C. Ren, X. He, Q. Teng, Y. Wu, and T. Q. Nguyen, "Single image superresolution using local geometric duality and non-local similarity," *IEEE Trans. Image Process.*, vol. 25, no. 5, pp. 2168–2183, May 2016.

[13] T. Li, X. He, L. Qing, Q. Teng, and H. Chen, "An iterative framework of cascaded deblocking and superresolution for compressed images," *IEEE Trans. Multimedia*, vol. 20, no. 6, pp. 1305–1320, Jun. 2018.

[14] A. Giachetti and N. Asuni, "Real-time artifact-free image upscaling," *IEEE Trans. Image Process.*, vol. 20, no. 10, pp. 2760–2768, Oct. 2011.

[15] D. Zhou, X. Shen, and W. Dong, "Image zooming using directional cubic convolution interpolation," *IET Image Process.*, vol. 6, no. 6, pp. 627–634, Aug. 2012.

[16] H. Xu, G. Zhai, and X. Yang, "Single image super-resolution with detail enhancement based on local fractal analysis of gradient," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 10, pp. 1740–1754, Oct. 2013.

[17] M. Li, J. Liu, J. Ren, and Z. Guo, "Adaptive general scale interpolation based on weighted autoregressive models," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 2, pp. 200–211, Feb. 2015.

[18] S. Zhu, B. Zeng, L. Zen, and M. Gabbouj, "Image interpolation based on non-local geometric similarities and directional Gradients," *IEEE Trans. Multimedia*, vol. 18, no. 9, pp. 1707–1719, Sep. 2016.

[19] P. Thévenaz, T. Blu, and M. Unser, "Image interpolation and resampling," *Handbook of Medical Imaging, Processing and Analysis*, New York, NY, USA: Academic, 2000, pp. 393–420.

[20] T. Li, X. He, Q. Teng, and X. Wu, "Rotation expanded dictionary-based single image super-resolution," *Neurocomputing*, vol. 216, pp. 1–17, 2016.

[21] J. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed selfexemplars," in *Proc. IEEE Conf. Comp. Vis. Pattern Recogn.*, 2015, pp. 5197–5206.

[22] M. Bevilacqua, A. Roumy, C. Guillemot, and M.-L. A. Morel, "Single-image super-resolution via linear mapping of interpolated self-examples," *IEEE Trans. Image Process.*, vol. 23, no. 12, pp. 5334–5347, Dec. 2014.

[23] W. Yang *et al.*, "Consistent coding scheme for single-image super-resolution via independent dictionaries," *IEEE Trans. Multimedia*, vol. 18, no. 3, pp. 313–325, Mar. 2016.

[24] L. Kang, C.-C. Hsu, B. Zhuang, C. Lin, and C.-H. Yeh, "Learning-based joint super-resolution and deblocking for a highly compressed image," *IEEE Trans. Multimedia*, vol. 17, no. 7, pp. 921–934, Jul. 2015.

[25] R. Timofte, V. De Smet, and L. Van Gool, "Anchored neighborhood regression for fast example-based super-resolution," in *Proc. IEEE Int. Conf. Comp. Vis.*, 2013, pp. 1920–1927.

[26] R. Timofte, V. De Smet, and L. Van Gool, "A+: Adjusted anchored neighborhood regression for fast super-resolution," in *Proc. Asian Conf. Comp. Vis.*, 2015, pp. 111–126.

[27] Y. Zhang, Y. Zhang, J. Zhang, and Q. Dai, "CCR: Clustering and collaborative representation for fast single image super-resolution," *IEEE Trans. Multimedia*, vol. 18, no. 3, pp. 405–417, Mar. 2016.

[28] Y. Tang, Y. Yuan, P. Yan, and X. Li, "Greedy regression in sparse coding space for single-image super-resolution," *J. Vis. Commun. Image Representation*, vol. 24, no. 2, pp. 148–159, 2013.

[29] C. Dong, C.-C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 184–199.

[30] N. Kumar and A. Sethi, "Fast learning-based single image super-resolution," *IEEE Trans. Multimedia*, vol. 18, no. 8, pp. 1504–1515, Aug. 2016.

[31] W. Dong, L. Zhang, R. Lukac, and G. Shi, "Sparse representation based image interpolation with nonlocal autoregressive modeling," *IEEE Trans. Image Process.*, vol. 22, no. 4, pp. 1382–1394, Apr. 2013.

[32] M. Zhang and C. Desrosiers, "High-quality image restoration using low-rank patch regularization and global structure sparsity," *IEEE Trans. Image Process.*, vol. 28, no. 2, pp. 868–879, Feb. 2019.

[33] Y. Zhang, Q. Fan, F. Bao, Y. Liu, and C Zhang, "Single image super-resolution based on rational fractal interpolation, *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 3782–3797, Aug. 2018.

[34] S. M. Hu, "Research progress of digital geometry processing," *Chin. J. Comput.*, vol. 32, no. 8, pp. 1451–1469, 2009.

[35] H. W. Lin, H. J. Bao, and G. J. Wang, "Totally positive bases and progressive iteration approximation," *Comput. Math. Appl.*, vol. 50, no. 3, pp. 575–586, 2005.

[36] L. Piegl and W. Tiller, *The NURBS Book*. Berlin, Germany: Springer, 1995.

[37] A. L. Da Cunha, J. Zhou, and M. N. Do, "The nonsubsampled contourlet transform: Theory, design, and applications," *IEEE Trans. Image Process.*, vol. 15, no. 10, pp. 3089–3101, Oct. 2006.

[38] J. Millard and L. Kurz, "The Kolmogorov-Smirnov tests in signal detection," *IEEE Trans. Inf. Theory*, vol. 13, no. 2, pp. 341–342, 1967.

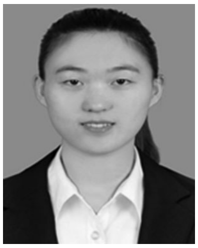
[39] M. Bevilacqua, A. Roumy, C. Guillemot, and M.-L. A. Morel, "Low-complexity single-image super-resolution based on nonnegative neighbor embedding," in *Proc. Brit. Mach. Vis. Conf.*, 2012, pp. 135.1–135.10.

[40] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *Prof. Int. Conf. Curves Surf.*, pp. 711–730, 2010.

[41] R. Franzen, "Kodak lossless true color image suite." [Online]. Available: http://r0k.us/graphics/kodak/

[42] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, May 2011.

Yunfeng Zhang received the B.E. degree in computational mathematics and application software from the Shandong University of Technology, Jinan, China, in 2000, and the M.S. degree in applied mathematics and the Ph.D. degree in computational geometry from Shandong University, Jinan, China, in 2003 and 2007, respectively. He is currently a Professor with the Shandong Provincial Key Laboratory of Digital Media Technology, Shandong University of Finance and Economics, Jinan, China. His current research interests include computer-aided geometric design, digital image processing, computational geometry, and function approximation.



Ping Wang received the M.S. degree from the Department of Computer Science and Technology, Shandong University of Finance and Economics, Jinan, China, in 2019. She is currently working toward the Ph.D. degree with the Software and IT Department, École de Technologie Supérieure, Montreal, QC, Canada. Her research interests include image processing and computer vision.

Fangxun Bao received the M.Sc. degree from the Department of Mathematics, Qufu Normal University, Qufu, China, in 1994 and the Ph.D. degree from the Department of Mathematics, Northwest University, Xi'an, China, in 1997. He is currently a Full Professor with the Department of Mathematics, Shandong University, Jinan, China. His research interests include computer-aided geometric design and computation, computational geometry, and function approximation.

Xunxiang Yao received the M.S. degree from the Shandong Provincial Key Laboratory of Digital Media Technology, Shandong University of Finance and Economics, Jinan, China. He is currently working toward the Ph.D. degree with the Faculty of Engineering and Information Technology, University of Technology, Sydney, NSW, Australia. His research interests include image processing.

Caiming Zhang received the B.S. and M.E. degrees in computer science from Shandong University, Jinan, China, in 1982 and 1984, respectively, and the Ph.D. in Engineering degree in computer science from the Tokyo Institute of Technology, Tokyo, Japan, in 1994. He is currently a Professor and Doctoral Supervisor with the School of Computer Science and Technology, Shandong University. He is also a Dean and Professor with the School of Computer Science and Technology, Shandong Economic University, Jinan, China. His research interests include computer-aided geometric design, computational geometry, information visualization, and medical image processing.

Hongwei Lin received the B.E. degree from the Department of Applied Mathematics and the Ph.D. degree from the Department of Mathematics, Zhejiang University, Hangzhou, China, in 1996 and 2004, respectively. He is currently a Professor with the State Key Laboratory of CAD and CG, Zhejiang University. His current research interests include computer-aided geometric design, reverse engineering, curve and surface fitting, and application of interval arithmetic and affine arithmetic in graphics.