

Single-Image Super-Resolution Based on Rational Fractal Interpolation

Yunfeng Zhang, Qinglan Fan, Fangxun Bao, Yifang Liu, and Caiming Zhang

Abstract—This paper presents a novel single-image super-resolution (SR) procedure, which upscales a given low-resolution (LR) input image to a high-resolution image while preserving the textural and structural information. First, we construct a new type of bivariate rational fractal interpolation model and investigate its analytical properties. This model has different forms of expression with various values of the scaling factors and shape parameters; thus, it can be employed to better describe image features than current interpolation schemes. Furthermore, this model combines the advantages of rational interpolation and fractal interpolation, and its effectiveness is validated through theoretical analysis. Second, we develop a single-image SR algorithm based on the proposed model. The LR input image is divided into texture and non-texture regions, and then, the image is interpolated according to the characteristics of the local structure. Specifically, in the texture region, the scaling factor calculation is the critical step. We present a method to accurately calculate scaling factors based on local fractal analysis. Extensive experiments and comparisons with the other state-of-the-art methods show that our algorithm achieves competitive performance, with finer details and sharper edges.

Index Terms—Image super-resolution, rational fractal interpolation, image features, scaling factor, local fractal analysis.

I. INTRODUCTION

THE purpose of single-image super-resolution (SR) is to reconstruct a latent high-resolution (HR) image using a single low-resolution (LR) input. SR is a classic method of image processing which has value in both academic and industrial applications. SR has a wide range of practical applications, such as video surveillance, criminal investigation, remote sensing, medical image processing, and consumer electronics [1]. In general, existing SR methods can be broadly classified into two main categories: interpolation-based methods and learning-based methods.

Manuscript received May 9, 2017; revised November 3, 2017 and January 31, 2018; accepted April 8, 2018. Date of publication April 12, 2018; date of current version April 26, 2018. This work was supported in part by the National Natural Science Foundation of China under Grant 61373080, Grant 61672018, Grant 61402261, and Grant U1609218, and in part by the Fostering Project of Dominant Discipline and Talent Team of Shandong Province Higher Education Institutions. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Ce Zhu. (*Corresponding author: Fangxun Bao*)

Y. Zhang and Q. Fan are with the Department of Computer Science and Technology, Shandong University of Finance and Economics, Jinan 250014, China (e-mail: yfzhang@sdufe.edu.cn; qinglanfan@163.com).

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

Y. Liu is with the Department of Computer Science and Engineering, University at Buffalo, The State University of New York, Buffalo, NY 14260 USA (e-mail: yifangli@buffalo.edu).

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

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

Digital Object Identifier 10.1109/TIP.2018.2826139

Interpolation-based methods estimate the unknown pixels in the HR grid by employing their known neighbors. Traditional interpolation algorithms involving bi-linear and bi-cubic [2] interpolation are the most widely used methods in practice. However, the kernel functions used in the above methods are isotropic and cannot fully reflect the intrinsic structures of images. Thus, these interpolation approaches are prone to producing zigzagging artifacts along edges and blurring details in textures. In order to compensate for the deficiencies of traditional methods, edge-directed interpolation methods [3], [6] have recently been proposed. Although these types of methods can maintain the image edge structure, they often generate speckle noise or distortion around texture areas. The rational function that approximates the ideal kernel function [7] has been applied in image interpolation [8], [9]. Reconstructed images often have better visual results through rational function interpolation; however, the rational interpolation function results in an unsatisfactory level in preserving textural details.

Learning-based SR methods depend on the assumption that the high-frequency information missing in the LR image can be learned from a training set of HR and LR image pairs. Learning-based methods can be divided into two categories: the first depends on an external dictionary constructed from a set of external training images, and the second replaces the external training set with the LR image itself. For the former, a HR image is generated from a single LR image with the help of training pairs. The commonly used methods include regression-based methods [10], [11], sparse representation-based approaches [12], [13], patch-based SR methods [14], among others. Furthermore, anchored neighborhood regression (ANR) [15] and simple functions (SF) [20] provide state-of-the-art quality performance. Combining the best qualities of ANR and SF, an adjusted anchored neighborhood regression (A+) [17] method was proposed for fast SR. In recent years, deep learning technology has attracted considerable attention from numerous researchers. In image SR reconstruction, a deep learning approach (SRCNN) [16] was presented and directly learned an end-to-end mapping between the LR and HR images. Although such methods, which rely on an external dataset, typically perform well for some classes of images, they understandably have a considerable drawback: they are fixed and are thus not adapted to the input image [18]. For example, irregularities along the edges can be observed as in the results of [14], when the image content does not match the exemplary data set. In the latter case, the methods produce HR images by employing similarity redundancy information within both the same scale and across different scales. In [18], for each LR patch, similar self-examples were found, and it was necessary to learn a linear function to directly map

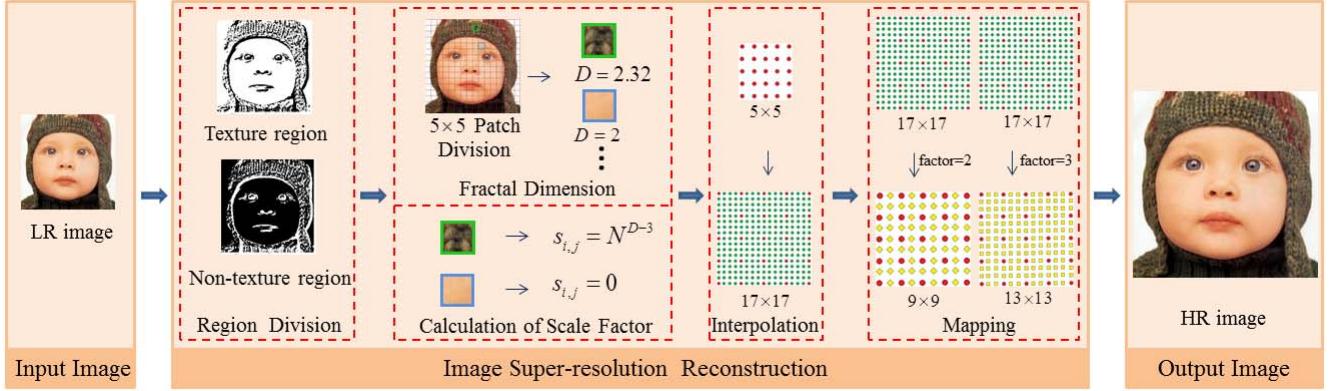


Fig. 1. Algorithm framework.

each LR patch into its HR version. However, in [18], the LR-HR patch pairs were found by searching only for “translated” versions of patches in the down-scaled images. A self-similarity-driven SR algorithm (selfExSR) [19] was proposed to expand the internal patch search space by allowing geometric variations, which improves the visual effect well. However, if the LR image does not contain sufficient repetitive patterns, then these algorithms tend to produce sharp edges rather than fine details. Considering the advantages of interpolation-based methods and learning-based methods, by applying the machine learning method in image interpolation, the study in [21] developed an image interpolation method in which nonlocal autoregressive modeling (NARM) was embedded in the sparse representation model. This method improves the visual effect well, but it tends to significantly degrade the quality of the reconstructed HR image when larger magnification ratios are performed, particularly image details and textural features.

Fractal is an efficient tool for describing the texture of an image, and is widely applied in texture description, classification, segmentation, among other functions. In [22], a texture multifractal spectrum was introduced. Based on multifractal analysis in wavelet pyramids of texture images, the study in [23] presented a textural descriptor that implicitly combines information from both spatial and frequency domains. A dynamic texture classification was proposed on the basis of dynamic fractal analysis in [24]. A method for HR optical image segmentation was proposed based on the multifractal characterization of an image in [25]. In addition, a fractal dimension-invariant filtering method [26] was presented for edge detection. In [27], a depth up-sampling algorithm which jointly uses local fractal analysis and boundary consistency analysis was proposed. However, only a limited amount of literature on applying fractal analysis in image SR has been available until now. For example, in [28], an image SR algorithm was proposed using a special type of orthogonal fractal coding method which can produce pleasing details, but fails to recover sharper edges. In [29], a texture enhancement algorithm which uses local fractal analysis for improving single-image SR performance was presented, and this can effectively enhance image details. However, this algorithm cannot provide satisfactory results in a stochastic texture region which does not follow the local fractal property.

Note that the above fractal analytical methods characterize the textural feature of the image primarily by using the fractal dimension. The fractal dimension is employed to describe the roughness of the texture, but it cannot accurately characterize textural details. Thus, it is necessary to construct a fractal function and apply related properties to accurately describe the image details.

In this paper, we construct a rational fractal interpolation model. On the one hand, compared with other polynomial interpolation kernel functions, the rational interpolation function is a more accurate approximation function for the ideal interpolation. It can preserve image edge structures well. On the other hand, the fractal function is an efficient model for describing image texture, and the reconstructed images have fine textural details and structural information when using rational fractal function interpolation. Based on this model, a novel single-image SR algorithm is proposed in which the fractal analysis method is applied in image interpolation. The major steps of the proposed algorithm are illustrated in Fig. 1. First, according to the image features, the image is divided into a texture region and a non-texture region. Second, we employ the rational fractal interpolation model in the texture region, and we use the rational interpolation model in the non-texture region. Finally, a HR image is obtained by pixel mapping.

By applying the rational fractal interpolation model in image SR reconstruction, the advantages of our method over other SR methods are as follows. First, it can recover more pleasing details than other interpolation methods. Second, unlike learning-based methods which depend on the source of training patches, our method can achieve competitive performance by only using LR image patch information.

The major contributions of the proposed method are summarized as follows:

- (1) We construct a rational fractal interpolation model based on previous research on rational splines [30], [33], and we investigate its analytical properties, such as error analysis, stability, quasi-locality, and fractal dimension.

- (2) Inspired by the constructed rational fractal interpolation model, we propose a single-image SR method by applying the fractal analysis method in the interpolation model. This method can accurately recover the spatial characteristic

information of the image, particularly image details and textural features.

(3) The extensions have two aspects: the first is consideration of the relationship between scaling factors and fractal dimension, and the second is consideration of the local characteristics of the image. Here, a method is presented to accurately calculate the scaling factors.

The remainder of this paper is organized as follows. Section II will develop a new type of bivariate rational fractal interpolation model, and it will present relevant theoretical analyses. Section III will present the proposed single-image SR reconstruction algorithm. The experimental results and analysis will be presented along with an evaluation of the effectiveness of the algorithm in Section IV. Section V will present some conclusions.

II. PROPOSED FRAMEWORK

A. Basic Idea

The fractal texture can describe the spatial characteristics information of the image correctly; moreover, the rational function is an approximation of the sinc function, which corresponds to ideal filtering. Based on the general characteristics described above, we construct a rational fractal interpolation model, and we provide the relevant theoretical analysis. The error analysis results prove that the proposed model is good in terms of the approximation effect, and it can control interpolation error. The stability analysis indicates that the proposed model has fairly strong adaptability in image interpolation. According to the property of quasi-locality, the model is suitable for describing the local features of an image. The fractal dimension reflects the complexity of the texture; furthermore, it has a strong correlation with the scaling factors. Here, we provide the method for calculating the fractal dimension.

Based on the above, the proposed interpolation model has greater flexibility and better practicality than previous models. In the following sections, the method of constructing the bivariate rational fractal interpolation function will be provided. In essence, our constructed rational function is embedded into an iterated function system. Then, the relevant theoretical analyses of the model will be presented.

B. Rational Fractal Interpolation Function

1) *Iterated Function System:* Fractal interpolation functions (FIFs) are types of continuous functions generated by iterated function systems (IFSs). FIFs corresponding to the following IFS form have been investigated the most extensively:

$$\begin{cases} \phi_i(x) = a_i x + b_i, \\ \varphi_j(y) = c_j y + d_j, \\ F_{i,j}(x, y, z) = s_{i,j} z + q_{i,j}(x, y), \end{cases} \quad (1)$$

where $|s_{i,j}| < 1$ and $s_{i,j}$ are called vertical scaling factors and $q_{i,j}(x, y)$ are continuous functions. The detailed process is shown in Appendix A.

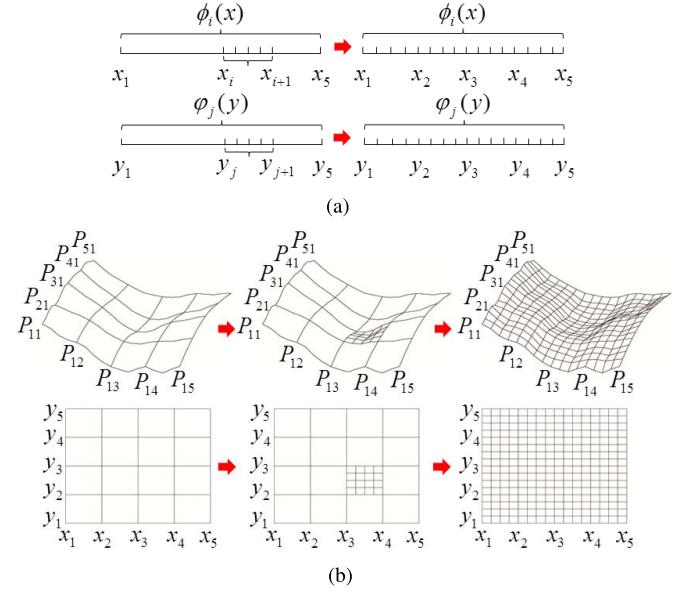


Fig. 2. Illustration of the iteration system.

2) *Construction of Rational FIFs:* Based on our previous work on bivariate rational interpolation, a novel method for constructing bivariate rational FIFs is proposed in this section. The core idea of the method is to treat continuous functions $q_{i,j}(x, y)$ in (1) as a “fractal perturbation” of bivariate rational interpolation functions $P_{i,j}(\phi_i(x), \varphi_j(y))$ obtained via rational perturbation base functions $B_{i,j}(x, y)$, namely,

$$q_{i,j}(x, y) = P_{i,j}(\phi_i(x), \varphi_j(y)) - s_{i,j} B_{i,j}(x, y).$$

Thus, the IFS (1) determines a continuous function $\Phi(x, y)$, which is called a bivariate rational FIF, and

$$\Phi(\phi_i(x), \varphi_j(y)) = s_{i,j} \Phi(x, y) + P_{i,j}(\phi_i(x), \varphi_j(y)) - s_{i,j} B_{i,j}(x, y). \quad (2)$$

Furthermore, if the shape parameters satisfy a suitable mild condition, the fractal function $\Phi(x, y)$ defined by (2) can be rewritten in the following simple matrix form:

$$\Phi(\phi_i(x), \varphi_j(y)) = s_{i,j} \Phi(x, y) + AEB, \quad (3)$$

where A is a 4-dimensional row vector, B is a 4-dimensional column vector, and E is a constant matrix of order 4. The specific construction of the bivariate rational FIF is shown in Appendix B.

Equation (2) shows that the bivariate rational FIF $\Phi(x, y)$ is given by an iterative scheme. In the following, we interpret the iterative process of the system via a data set of a 5×5 grid: $\{(x_i, y_j, f_{i,j}) : i, j = 1, 2, \dots, 5\}$. For each i and j ($i, j = 1, 2, 3, 4$), contractive homeomorphisms $\phi_i(x)$ and $\varphi_j(y)$ map intervals $[x_1, x_5]$ and $[y_1, y_5]$ to subintervals $[x_i, x_{i+1}]$ and $[y_j, y_{j+1}]$, and 5 points are obtained on $[x_i, x_{i+1}]$ and $[y_j, y_{j+1}]$ respectively, as shown in Fig. 2(a). Meanwhile, 5×5 interpolation points are provided on each subregion $[x_i, x_{i+1}] \times [y_j, y_{j+1}]$, and 5×5 points on the interpolation surface corresponding to this subregion can be acquired by the iterative scheme (2); take $[x_3, x_4] \times [y_2, y_3]$ as an example, as shown in Fig. 2(b), where the top and

TABLE I
DATASET TAKEN FROM FUNCTION (4)

$y_j \backslash f_{i,j} \backslash x_i$	0.1	0.2	0.3	0.4	0.5
0.1	1.1696	1.0959	1.0256	0.9592	0.8969
0.2	1.0959	1.0309	0.9685	0.9091	0.8529
0.3	1.0256	0.9685	0.9132	0.8602	0.8097
0.4	0.9592	0.9091	0.8602	0.8130	0.7678
0.5	0.8969	0.8529	0.8097	0.7678	0.7273

bottom parts of Fig. 2(b) represent interpolation surfaces and interpolation regions respectively. Thus, after the first iteration, we obtain 17×17 points on region $[x_1, x_5] \times [y_1, y_5]$ and the interpolation surface corresponding to the region. The same occurs with more iterations.

Remark 1: The construction of the rational FIF described here allows us to embed shape parameters within the structure of the fractal function, and the rational FIF $\Phi(x, y)$ defined by (2) is uniquely identified from the interpolation data and the values of the scaling factors and shape parameters. With different scaling factors and shape parameters, this fractal function can be expressed in different forms, specifically, if the scaling factors $s_{i,j} = 0$ for all i and j , $\Phi(x, y)$ reduces to the bivariate rational spline interpolant $P_{i,j}(x, y)$. This means that the presented bivariate rational fractal interpolation is a flexible and diverse form for the choice of suitable interpolant.

C. Theoretical Discussion

1) *Error Analysis:* In image interpolation, the accuracy of the interpolation model directly affects the quality of the interpolated image, which is generally assessed in terms of errors and is the key criterion for evaluating the interpolation method. The smaller the error is, the higher the accuracy is. In order to show the effectiveness with which the bivariate rational FIF $\Phi(x, y)$ defined by (2) approximates the original function $f(x, y)$, we provide the upper bound of the uniform error between $\Phi(x, y)$ and $f(x, y)$ as follows:

$$\begin{aligned} \|f - \Phi\|_\infty &\leq \frac{|s|_\infty}{1 - |s|_\infty} M + h^2 \left\| \frac{\partial^2 f}{\partial x^2} \right\|_\infty C \\ &\quad + l \left(\left\| \frac{\partial f}{\partial y} \right\|_\infty + \left\| \frac{\partial P_{i,j}}{\partial y} \right\|_\infty \right), \end{aligned}$$

where M is a constant associated with $f_{i,j}$, $d_{i,j}^*$ and $d_{i,j}$; C is an error constant associated with shape parameters: $|s|_\infty = \max\{|s_{i,j}| : i \in \mathcal{I}, j \in \mathcal{J}\}$; $h = \max\{h_i : i \in \mathcal{I}\}$; and $l = \max\{l_j : j \in \mathcal{J}\}$.

Example 1: Consider the interpolation data presented in Table I, which has been approximated from the following function by taking the values truncated to four decimal places:

$$f(x, y) = \frac{4}{(x+1)^2 + (y+1)^2 + 1}, \quad (x, y) \in [0.1, 0.5; 0.1, 0.5]. \quad (4)$$

The example will show the approximation effectiveness of the developed bivariate rational fractal interpolation.

Fig. 3 presents the graph of the error function $f(x, y) - \Phi(x, y)$, and the values of the error range from -4×10^{-3} to 4×10^{-3} . This graph clearly shows that the bivariate rational

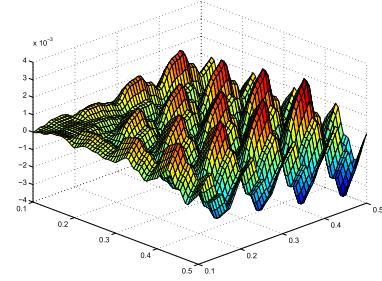


Fig. 3. Error surface $f(x, y) - \Phi(x, y)$.

TABLE II
THE PERTURBATION OF THE DATA IN TABLE I

$y_j \backslash f_{i,j} \backslash x_i$	0.1	0.2	0.3	0.4	0.5
0.1	1.167	1.096	1.026	0.959	0.897
0.2	1.096	1.031	0.969	0.909	0.853
0.3	1.026	0.969	0.913	0.860	0.810
0.4	0.959	0.909	0.860	0.813	0.768
0.5	0.897	0.853	0.810	0.768	0.727

spline FIFs provide a good approximation for the original function $f(x, y)$. This result confirms that the proposed rational fractal model can guarantee the quality of interpolated images well.

2) *Stability:* Stability is an important criterion for describing the quality of an interpolation function, and it measures the anti-interference ability of the perturbation of interpolation data. In this part, we provide this property of the developed bivariate rational FIF.

Let $\hat{\Delta} = \{(x_i, y_j, \hat{f}_{i,j}, \hat{d}_{i,j}^*, \hat{d}_{i,j}) : i \in \mathcal{I}; j \in \mathcal{J}\}$ be another given set of interpolation points, which are generated by perturbation of the vertical coordinates $f_{i,j}$ and first-order partial derivatives $d_{i,j}^*$ and $d_{i,j}$. Define an IFS $\{\hat{F}; (\varphi_i(x), \psi_j(y), \hat{F}_{i,j}(x, y, \hat{\Phi}(x, y))) : i \in \mathcal{I}; j \in \mathcal{J}\}$, where $\varphi_i(x)$ and $\psi_j(y)$ are defined as above, and

$$\begin{aligned} \hat{F}_{i,j}(x, y, \hat{\Phi}(x, y)) &= s_{i,j} \hat{\Phi}(x, y) + \hat{P}_{i,j}(\varphi_i(x), \psi_j(y)) \\ &\quad - s_{i,j} \hat{B}_{i,j}(x, y), \end{aligned}$$

$\hat{P}_{i,j}(\varphi_i(x), \psi_j(y))$ and $\hat{B}_{i,j}(x, y)$ satisfy the corresponding join-up conditions. $\hat{\Phi}(x, y)$ is the bivariate rational FIF generated by this IFS. We therefore have

$$\|\Phi - \hat{\Phi}\|_\infty \leq \frac{1 + |s|_\infty}{1 - |s|_\infty} \delta_f + \frac{2h}{1 - |s|_\infty} \delta_{d^*} + \frac{2l}{1 - |s|_\infty} \delta_d,$$

where $\delta_f = \max\{|f_{i,j} - \hat{f}_{i,j}| : i \in \mathcal{I}, j \in \mathcal{J}\}$, $\delta_{d^*} = \max\{|d_{i,j}^* - \hat{d}_{i,j}^*| : i \in \mathcal{I}, j \in \mathcal{J}\}$, and $\delta_d = \max\{|d_{i,j} - \hat{d}_{i,j}| : i \in \mathcal{I}, j \in \mathcal{J}\}$.

Example 2: Consider the interpolation data shown in Table II, which is a perturbation of the data in Table I.

Fig. 4 provides the surface of function $\Phi(x, y) - \hat{\Phi}(x, y)$. Fig. 4 shows that the presented bivariate rational fractal interpolation has good capacity for stability during the perturbation of interpolation data.

3) *Quasi-Locality:* In practical applications, it is often necessary to adjust the local shape of the interpolation surfaces without affecting the entire shape. Consequently, interpolation functions with local properties are highly competitive. Because FIF has some level of self-similarity, for the fractal surface

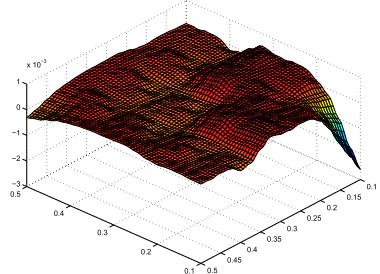
Fig. 4. Perturbation error surface $\Phi(x, y) - \hat{\Phi}(x, y)$.

TABLE III
INTERPOLATION DATASET FOR QUASI-LOCALITY ANALYSIS

$y_j \setminus f_{i,j}$	1	2	3	4	5
1	1.5	1.0	2.0	4.0	4.0
2	3.0	2.4	3.0	2.5	4.0
3	2.0	4.0	2.5	2.0	3.5
4	3.5	2.5	4.5	4.0	2.5
5	2.0	3.0	2.0	2.5	3.0

derived from FIF, each patch in a subregion will possess global properties inherited from the entire set of interpolation data. This implies that the shape of the entire interpolation surface may be affected if the IFS parameters are changed in a particular subregion. In this sense, it is difficult to constrain or adjust the local shape of the fractal surface. However, in the presented bivariate rational spline fractal interpolation scheme, the “height function” $P_{i,j}(x, y)$ with the shape parameters possesses the local shape property such that the bivariate rational FIF has good quasi-locality capacity on the parameters in the rational IFS. This finding will be interpreted using the following example.

Example 3: The interpolation data is shown in Table III.

Take $s_{i,j} = 0.02$ for all i and j ; the values of all shape parameters are 0.6. Fig. 5(a) shows the corresponding bivariate rational fractal surface $\Phi_4(x, y)$. Now, we consider the effects of the scaling factors and shape parameters on the shape of the fractal surface. First, we change $s_{i,j}$ to 0.15 for $i, j = 1, 2$, keeping the other scaling factors and all shape parameters as in Fig. 5(a), and Fig. 5(b) provides the corresponding rational fractal surface $\Phi_5(x, y)$. Second, take $\alpha_{i,j}^* = \beta_{i,j}^* = \gamma_{i,j}^* = 65$ for $i, j = 1, 2$, keeping all the scaling factors and other shape parameters as in Fig. 5(a), and Fig. 5(c) presents the graph of the corresponding rational fractal surface $\Phi_6(x, y)$. Third, take $\alpha_{i,j} = \beta_{i,j} = \gamma_{i,j} = 46$ for $i, j = 1, 2$, keeping all the scaling factors and other shape parameters as in Fig. 5(a), and Fig. 5(d) is the corresponding rational fractal surface $\Phi_7(x, y)$.

Fig. 5 shows that a significant change in the shape of the fractal surface occurs in the particular subregions in which changes of the scaling factors or shape parameters have been made, and the remainder pertaining to other subregions is not extremely sensitive. This result indicates that the developed bivariate rational fractal interpolation has a good property of quasi-locality on the IFS parameters; thus, the shape of the fractal surface can be adjusted locally without a significant influence in other regions. Furthermore, compared with the scaling factors, the shape parameters have a minor effect on the shape of the fractal surface.

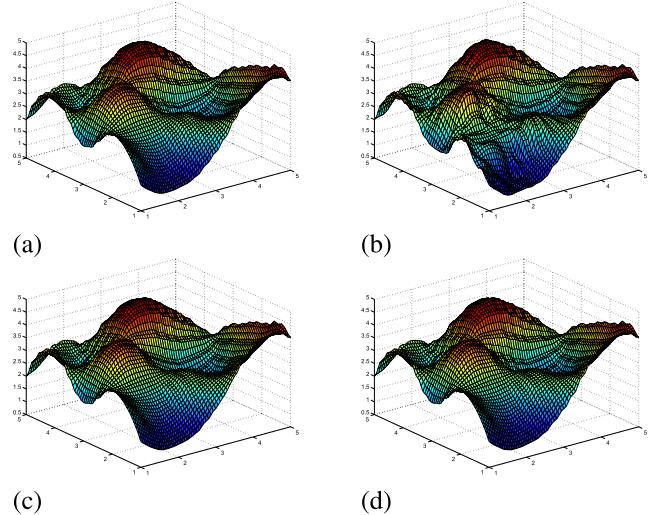


Fig. 5. Quasi-locality of the bivariate rational spline fractal interpolation: (a) Rational fractal surface Φ_4 , (b) Rational fractal surface Φ_5 with different $s_{i,j}$, (c) Rational fractal surface Φ_6 with different $\alpha_{i,j}^*, \beta_{i,j}^*, \gamma_{i,j}^*$, and (d) Rational fractal surface Φ_7 with different $\alpha_{i,j}, \beta_{i,j}, \gamma_{i,j}$.

4) Fractal Dimension: The fractal dimension of surfaces is a measure of the irregularity of surfaces. This describes the roughness of the surfaces. We provide the box-counting dimension of the rational fractal surface as follows.

Denote

$$S = \begin{pmatrix} |s_{\tau^{-1}(1)}| & |s_{\tau^{-1}(1)}| & \cdots & |s_{\tau^{-1}(1)}| \\ |s_{\tau^{-1}(2)}| & |s_{\tau^{-1}(2)}| & \cdots & |s_{\tau^{-1}(2)}| \\ \vdots & \vdots & & \vdots \\ |s_{\tau^{-1}(N^2)}| & |s_{\tau^{-1}(N^2)}| & \cdots & |s_{\tau^{-1}(N^2)}| \end{pmatrix}_{N^2 \times N^2},$$

where $\tau(i, j) = (i-1) \times N + j$ represents the enumeration of set $\{(i, j) : i, j = 1, 2, \dots, N\}$. Let the knots be equally spaced, namely, $h_i = l_j = \frac{b-a}{N}$, and let the interpolation points of every interval be neither x-collinear nor y-collinear. Then, the box-counting dimension D of the fractal surface defined by (13) is as follows:

$$D = \begin{cases} 1 + \log_N \lambda, & \lambda > N \\ 2, & \lambda \leq N \end{cases} \quad (5)$$

where $\lambda = \rho(S)$.

Furthermore, since

$$\lambda = \rho(S) = \sum_{k=1}^{N^2} \tau^{-1}(k) = \sum_{i=1}^N \sum_{j=1}^N s_{i,j},$$

equation (5) can be rewritten as follows:

$$D = \begin{cases} 1 + \log_N \sum_{i=1}^N \sum_{j=1}^N |s_{i,j}|, & \sum_{i=1}^N \sum_{j=1}^N |s_{i,j}| > N, \\ 2, & \text{others.} \end{cases} \quad (6)$$

Equation (6) shows that the fractal dimension of the surfaces is closely related to the values of the scaling factors in the corresponding bivariate rational FIFs. For the same interpolation data and values of the shape parameters, Fig. 6 provides the fractal interpolation surfaces corresponding to the different values of the scaling factors.

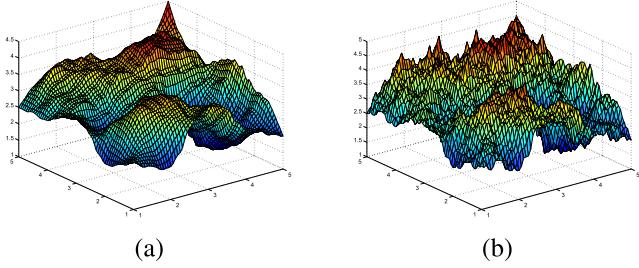


Fig. 6. Roughness of the surfaces for different scaling factors: (a) Smooth rational fractal surface with $s_{i,j} = 0.1$, and (b) Rational fractal surface with $s_{i,j} = 0.3$.

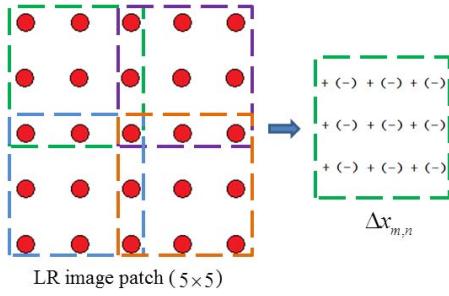


Fig. 7. Texture detection.

III. ADAPTIVE IMAGE SUPER-RESOLUTION RECONSTRUCTION ALGORITHM

A. Image Interpolation

1) *Regional Division*: In general, different regions in a single digital image have different structural characteristics. It is difficult for image interpolation to achieve better quality by using a single model. The proposed interpolation model has different forms of expression, which can be used to handle different regions. For images, regional division is a key step in the interpolation algorithm. The interpolation quality can be directly affected by the regional division results.

Unlike common texture detection methods, such as the Sobel and Canny operators, our objective is to detect more detailed textures such that the proposed SR algorithm can better play the dominant role of the rational fractal interpolation function. Here, we introduce the isoline method [34] to detect the textural and structural information of the image for regional division. Furthermore, we have conducted an experiment to choose an image patch of suitable size for texture detection (the smallest size of the interpolation unit is 3×3). As shown in Fig. 8, considering the robustness of the method, a 5×5 patch in a given LR input image is treated as a unit for detecting the roughness of texture because choosing a patch that is too small will result in noise sensitivity. Similarly, some texture regions will be considered as non-texture regions if the size of the image patch is too large.

Pixel value is the most intuitive expression of various structures in generic natural images, and it also satisfies the attributes of the image itself. Therefore, textural details can be detected according to differences between values of pixels. If the region detected includes an isopleth, then it is regarded as a texture region. Otherwise, it is regarded as a non-texture

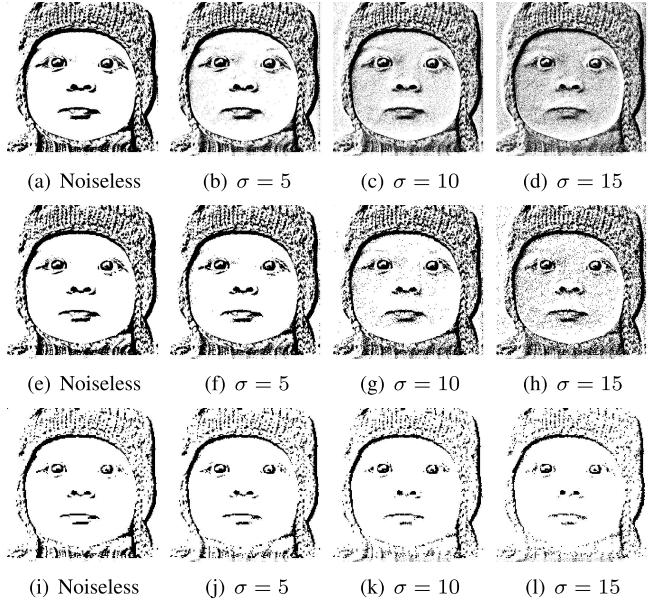


Fig. 8. Texture detection results of noisy images: (a)-(d) 3×3 patch, (e)-(h) 5×5 patch and (i)-(l) 7×7 patch.

region. As shown in Fig. 7, the main procedures of this method are as follows. First, each LR image patch (5×5) is divided into 4 sub-blocks (3×3). Second, using the first sub-block as an example, for LR image patch $f_{m,n}, m, n = 1, 2, \dots, 5$,

$$\lambda = \frac{\sum_{m=1}^5 \sum_{n=1}^5 f_{m,n}}{5 \times 5},$$

where $f_{m,n}$ is the pixel value at point (m, n) and λ is the average value of all pixels. Let $\Delta x_{m,n} = f_{m,n} - \lambda, m, n = 1, 2, 3$; the image can be divided into a texture region and a non-texture region using the following formula:

$$\begin{cases} \text{non-texture region}, & \Delta x_{m,n} \geq 0 \text{ or } \Delta x_{m,n} \leq 0, \forall m, n, \\ \text{texture region}, & \text{otherwise}. \end{cases} \quad (7)$$

Specifically, if all sub-blocks have a uniform sign, i.e., if a 5×5 LR image patch does not include an isopleth, then we regard it as a non-texture region. Otherwise, the LR image patch is regarded as a texture region. The binarization image is presented to evaluate the effectiveness of the proposed method in Fig. 9. As indicated by the experimental results, image texture details are effectively detected. The fractal can describe the spatial characteristics information of the image correctly; therefore, the proposed texture detection method is able to exert the advantages of the rational fractal interpolation function.

2) *Scaling Factor Calculation*: For image processing, the scaling factors reflect the complexity of the texture. According to the property analysis in Section II-C, the scaling factors play an important role in the rational fractal interpolation function. Thus, the quality of interpolation can be directly affected by the values of scaling factors. As the accuracy of scaling factors improves, greater interpolation quality is obtained. Generally, an image patch is interpolated with the

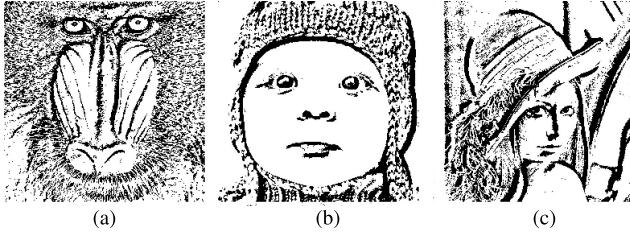


Fig. 9. Results of texture detection. (a) Baboon. (b) Child. (c) Lena.

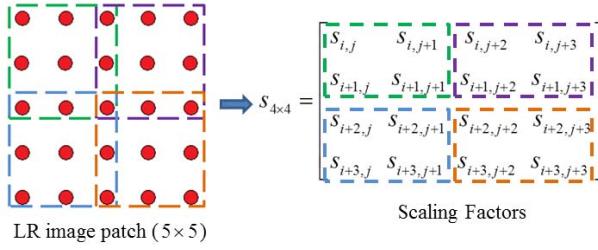


Fig. 10. Local scaling factors.

same scaling factors in the traditional fractal interpolation methods, which ignores the local variations (local geometric structure) of an image. Here, we present a method of calculating the scaling factors based on the local features of an image. This method involves two steps. First, the initial values of the scaling factors are obtained for each LR image patch. Second, the scaling factors are further obtained according to the image texture information.

As shown in Fig. 10, the concrete steps of the method are as follows. First, as in image processing, the fractal dimension reflects the complexity of the texture. Equation (6) shows that the fractal dimension has a strong correlation with the scaling factors. It is imperative to calculate the values of the scaling factors with the help of the fractal dimension. However, the fractal dimension D of formula (6) is the fractal dimension of the entire image, and the scaling factors $s_{i,j}$ of the LR image patch cannot be obtained by using it directly. Based on the property that the proposed bivariate rational FIF has a good capacity of quasi-locality on the parameters in the rational IFS, we present an efficient method of calculating the scaling factors with the help of the local fractal dimension [35]. Suppose that each LR image patch uses the same scaling factors, where the initial values of the scaling factors $S_{initial}$ are obtained by calculating the following formula:

$$S_{initial} = N^{D-3}. \quad (8)$$

Then, it is unsuitable to use the same scaling factor for LR image patches with complex geometric structures. Take the first sub-block ($f_{m,n}, m, n = 1, 2, 3$) as an example: if it is the texture region,

$$\begin{aligned} aver &= \frac{f_{1,1} + f_{1,2} + \dots + f_{3,3}}{9}, \\ sum &= |f_{1,1} - aver| + |f_{1,3} - aver| + |f_{3,1} - aver| \\ &\quad + |f_{3,3} - aver|, \\ s_{1,1} &= S_{initial} \times \frac{|f_{1,1} - aver|}{sum}, \end{aligned}$$

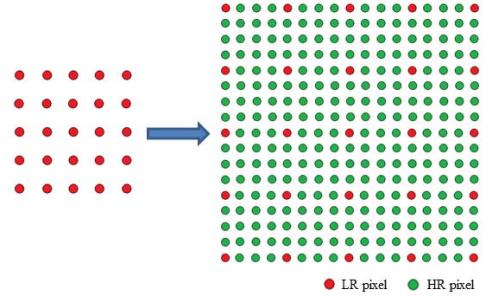


Fig. 11. Rational fractal interpolation model.

Algorithm 1 Image Interpolation Process

1. Divide the LR input image into many 5×5 patches.
2. For each 5×5 LR image patch, detect the texture by using the isoline method.
3. Calculate scaling factor $s_{i,j}$.
4. Select shape parameters α, β, γ .
5. The 17×17 HR image patch is obtained by using the bivariate rational fractal interpolation model.
6. Output the HR image by traversing each patch in the LR image.

$$\begin{aligned} s_{1,2} &= S_{initial} \times \frac{|f_{1,3} - aver|}{sum}, \\ s_{2,1} &= S_{initial} \times \frac{|f_{3,1} - aver|}{sum}, \\ s_{2,2} &= S_{initial} \times \frac{|f_{3,3} - aver|}{sum}; \end{aligned}$$

otherwise, $s_{1,1} = s_{1,2} = s_{2,1} = s_{2,2} = 0$.

3) Interpolation Algorithm: The interpolation model has different forms of expression, which can be used to address the texture region and non-texture region separately. In the texture region, the interpolation is rational fractal interpolation. In the non-texture region, we use the rational interpolation ($s_{i,j} = 0$).

Based on the above analysis for regional division, the LR input image is divided into many 5×5 patches. According to block characteristics, a 5×5 vector control grid is used to construct the interpolation surface, and we then use it and rational fractal interpolation to compute every point's intensity in the magnified image. As shown in Fig. 11, for the given 5×5 image patch, a 17×17 patch can be obtained by using the proposed model (see Section II-B). The red dots represent the LR image pixels, and the missing HR samples are the green dots. The main steps of the interpolation algorithm are summarized in **Algorithm 1**.

The LR image is divided into 5×5 patches, and the proposed interpolation model is used to estimate the missing pixels in the LR image patch. In order to guarantee a smooth connection between patches, there is an overlapping pixel point between the patches in the processing. The image interpolation process is shown in Fig. 12. The interpolation is extended to the entire image by traversing each patch in raster-scan order in the LR image, namely from left to right and from top to bottom.

B. Mapping

The proposed rational fractal interpolation function is an IFS. A large number of pixels are obtained with an increasing number of iterations, and more detailed information can be

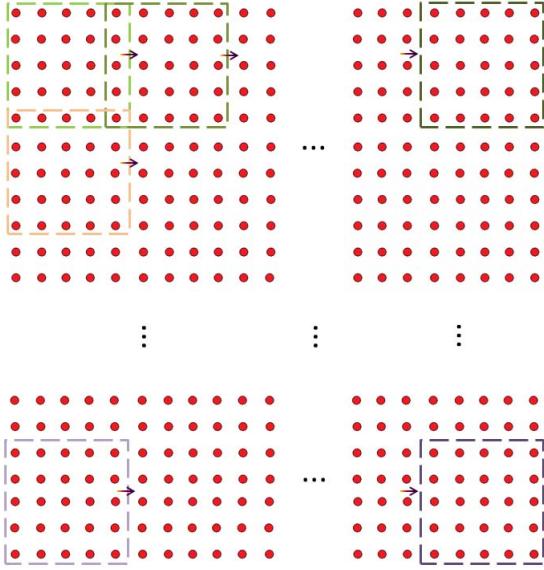
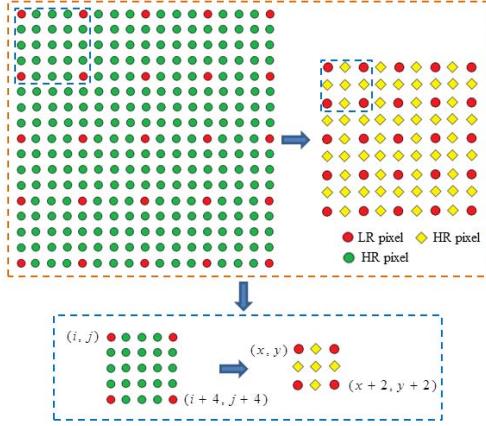


Fig. 12. Illustration of the image interpolation process.

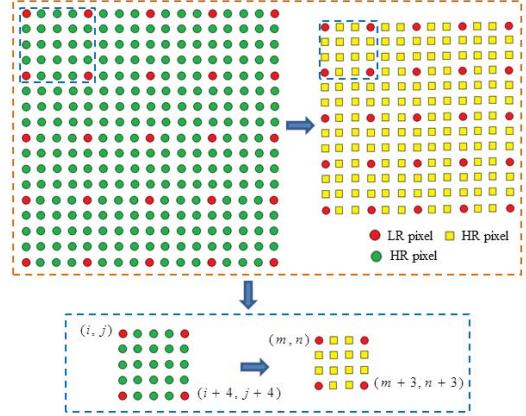
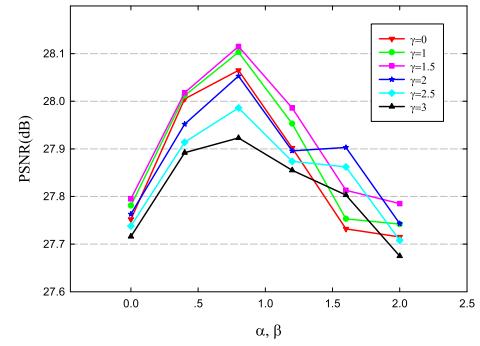
Fig. 13. The process of pixel mapping ($\times 2$).

captured. Thus, the image can be amplified at any integral multiple via pixel mapping. In fact, pixel mapping is a sampling process.

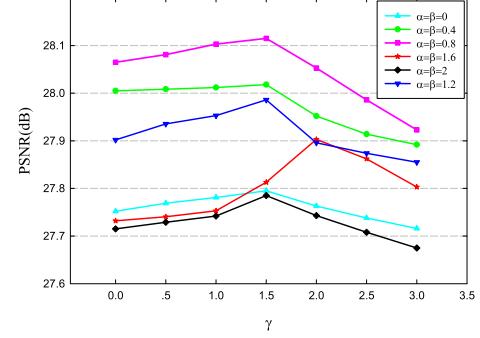
Specifically, after the first iteration, the image can be magnified with three different scale factors ($\times 2$, $\times 3$, and $\times 4$). As shown in Section II-B, for the given 5×5 patch, a 17×17 patch can be obtained by using the proposed model. Thus, the image with an upscaling factor of 4 can be obtained directly by using the proposed model. The image also can be magnified with two different scale factors (2 and 3) using the pixel mapping process which will be described in the following sections. As shown in Fig. 13 and Fig. 14, taking the first sub-block ($f_{i,j}$) as an example, the HR image sub-block ($g_{x,y}$ and $h_{m,n}$) can be obtained using the following: $f_{i,j}$, $g_{x,y}$, $h_{m,n}$, the pixel values at points (i, j) , (x, y) , (m, m) , respectively.

As shown in Fig. 13,

$$g_{x,y+1} = \sum_{t=j}^{j+4} \frac{f_{i,t}}{\sum_{t=j}^{j+4} f_{i,t}} \times f_{i,t}, \quad g_{x+1,y} = \sum_{s=i}^{i+4} \frac{f_{s,j}}{\sum_{s=i}^{i+4} f_{s,j}} \times f_{s,j},$$

Fig. 14. The process of pixel mapping ($\times 3$).

(a)



(b)

Fig. 15. Performance variation of the proposed model with different parameters α , β , γ on the training dataset.

$g_{i+1,j+1}$ is obtained from the following equations:

$$\begin{aligned} varHor &= var(f_{i+2,j+1}, f_{i+2,j+2}, f_{i+2,j+3}), \\ varVer &= var(f_{i+1,j+2}, f_{i+2,j+2}, f_{i+3,j+2}), \\ var45 &= var(f_{i+1,j+3}, f_{i+2,j+2}, f_{i+3,j+1}), \\ var135 &= var(f_{i+1,j+1}, f_{i+2,j+2}, f_{i+3,j+3}), \\ varMin &= \min(varHor, varVer, var45, var135), \end{aligned}$$

where $var()$ is a variance function and $\min()$ is a minimal function. Variance reflects the difference between pixel values. Texture direction can be obtained approximatively according to the minimum value of variance. Thus, $g_{i+1,j+1}$ is closest to these pixels whose texture direction is consistent with that

TABLE IV
OBJECTIVE QUALITY ASSESSMENT OF DIFFERENT METHODS (SET5)

Image	scale	bicubic	BMF	NEDI	ICBI	DCCI	Lanczos	ScSR	NARM	SRCCN	A+	SelfExSR	ANR	Tai's	Proposed
baby	$\times 2$	33.198	31.309	35.873	36.620	20.992	32.037	32.446	36.776	31.376	31.570	30.564	31.917	30.736	36.757
		0.967	0.922	0.982	0.984	0.978	0.944	0.960	0.949	0.945	0.946	0.926	0.951	0.937	0.984
		0.976	0.956	0.989	0.991	0.942	0.966	0.972	0.990	0.971	0.971	0.962	0.971	0.954	0.992
bird	$\times 2$	31.924	29.432	35.140	37.712	23.674	31.003	31.273	38.573	30.433	30.756	29.565	31.069	28.172	36.333
		0.941	0.888	0.967	0.977	0.965	0.900	0.934	0.977	0.922	0.927	0.870	0.930	0.885	0.979
		0.947	0.908	0.971	0.978	0.926	0.921	0.939	0.973	0.932	0.935	0.902	0.938	0.909	0.980
butterfly	$\times 2$	24.585	22.490	26.822	28.566	22.403	23.905	24.081	30.167	22.459	23.100	21.836	23.452	22.445	29.204
		0.880	0.815	0.924	0.935	0.930	0.798	0.878	0.954	0.828	0.851	0.714	0.852	0.820	0.950
		0.873	0.844	0.920	0.926	0.899	0.787	0.875	0.936	0.826	0.846	0.721	0.833	0.821	0.938
head	$\times 2$	32.081	30.705	33.814	33.710	23.252	31.218	31.467	34.423	29.783	30.126	30.081	30.493	28.377	34.451
		0.804	0.742	0.847	0.840	0.831	0.759	0.786	0.856	0.732	0.742	0.723	0.754	0.729	0.857
		0.918	0.859	0.928	0.929	0.885	0.895	0.906	0.931	0.874	0.878	0.878	0.885	0.885	0.938
women	$\times 2$	28.191	26.142	30.830	32.657	24.749	27.397	27.484	34.107	25.924	26.408	25.429	26.823	27.495	34.185
		0.915	0.847	0.952	0.952	0.949	0.866	0.908	0.961	0.883	0.892	0.822	0.896	0.875	0.963
		0.926	0.879	0.952	0.959	0.926	0.896	0.916	0.965	0.901	0.906	0.865	0.909	0.898	0.968
average	$\times 2$	29.995	28.015	32.495	33.853	23.014	29.112	29.350	34.809	27.995	28.392	27.495	28.750	25.450	34.186
		0.901	0.842	0.934	0.937	0.930	0.853	0.893	0.939	0.862	0.871	0.811	0.876	0.850	0.946
		0.928	0.889	0.952	0.956	0.915	0.893	0.921	0.959	0.900	0.907	0.865	0.907	0.893	0.963
baby	$\times 3$	29.035	28.190	-	-	-	28.383	28.620	28.458	27.221	27.369	26.890	27.735	27.488	32.950
		0.894	0.837	-	-	-	0.867	0.887	0.858	0.861	0.833	0.867	0.858	0.952	0.952
		0.934	0.902	-	-	-	0.919	0.931	0.962	0.917	0.918	0.904	0.919	0.910	0.973
bird	$\times 3$	27.232	25.808	-	-	-	26.663	26.799	31.801	25.304	25.585	25.147	26.084	24.941	31.622
		0.845	0.785	-	-	-	0.787	0.835	0.929	0.795	0.804	0.729	0.812	0.774	0.932
		0.885	0.837	-	-	-	0.858	0.878	0.932	0.856	0.860	0.824	0.865	0.849	0.933
butterfly	$\times 3$	20.472	19.016	-	-	-	20.004	20.038	24.773	18.499	18.934	18.054	19.237	19.015	23.229
		0.738	0.668	-	-	-	0.648	0.735	0.891	0.664	0.700	0.551	0.691	0.678	0.810
		0.764	0.742	-	-	-	0.696	0.766	0.877	0.708	0.727	0.630	0.711	0.740	0.873
head	$\times 3$	29.233	28.958	-	-	-	28.858	28.786	31.253	26.884	27.322	27.568	27.682	27.234	31.746
		0.711	0.687	-	-	-	0.686	0.693	0.779	0.621	0.639	0.637	0.651	0.682	0.815
		0.871	0.818	-	-	-	0.859	0.862	0.881	0.826	0.834	0.835	0.843	0.852	0.900
women	$\times 3$	23.964	23.036	-	-	-	23.430	23.516	28.275	21.591	22.089	21.406	22.488	23.724	28.317
		0.813	0.757	-	-	-	0.750	0.805	0.909	0.762	0.774	0.680	0.778	0.771	0.919
		0.862	0.813	-	-	-	0.833	0.856	0.917	0.827	0.836	0.785	0.839	0.834	0.918
average	$\times 3$	25.987	25.001	-	-	-	25.467	25.551	28.912	23.899	24.259	23.813	24.645	24.480	29.572
		0.800	0.746	-	-	-	0.748	0.791	0.879	0.740	0.755	0.686	0.759	0.753	0.885
		0.963	0.822	-	-	-	0.833	0.858	0.913	0.826	0.835	0.795	0.835	0.837	0.919
baby	$\times 4$	26.733	26.128	-	-	-	26.226	26.433	-	24.798	25.042	24.511	25.375	25.339	30.900
		0.828	0.764	-	-	-	0.798	0.821	-	0.783	0.782	0.747	0.790	0.784	0.917
		0.902	0.856	-	-	-	0.883	0.898	-	0.876	0.877	0.854	0.879	0.870	0.953
bird	$\times 4$	24.673	23.787	-	-	-	24.245	24.391	-	22.944	23.237	22.590	23.590	22.861	28.883
		0.755	0.710	-	-	-	0.702	0.745	-	0.697	0.704	0.625	0.712	0.683	0.872
		0.840	0.788	-	-	-	0.822	0.835	-	0.811	0.815	0.781	0.820	0.806	0.899
butterfly	$\times 4$	18.490	16.923	-	-	-	18.057	18.191	-	16.391	17.005	16.013	17.240	17.031	20.757
		0.639	0.561	-	-	-	0.550	0.633	-	0.555	0.592	0.443	0.579	0.583	0.728
		0.698	0.676	-	-	-	0.647	0.696	-	0.646	0.650	0.578	0.645	0.685	0.752
head	$\times 4$	27.682	27.197	-	-	-	27.238	27.373	-	25.569	25.849	25.615	26.169	25.552	30.354
		0.666	0.657	-	-	-	0.641	0.653	-	0.589	0.592	0.583	0.603	0.632	0.734
		0.847	0.789	-	-	-	0.833	0.843	-	0.816	0.819	0.806	0.827	0.817	0.875
women	$\times 4$	21.769	21.366	-	-	-	21.314	21.444	-	19.366	19.942	18.996	20.253	21.405	25.425
		0.735	0.699	-	-	-	0.668	0.727	-	0.674	0.685	0.573	0.687	0.691	0.832
		0.820	0.770	-	-	-	0.794	0.815	-	0.780	0.791	0.730	0.795	0.789	0.867
average	$\times 4$	23.869	23.080	-	-	-	23.416	23.566	-	21.813	22.215	21.545	22.525	22.437	27.263
		0.724	0.678	-	-	-	0.672	0.715	-	0.659	0.671	0.594	0.674	0.674	0.816
		0.821	0.775	-	-	-	0.796	0.817	-	0.785	0.790	0.749	0.739	0.793	0.869

of $g_{i+1,j+1}$. For example, if $\text{varMin} = \text{varHor}$, $g_{i+1,j+1} = \frac{f_{i+2,j+1} + f_{i+2,j+2} + f_{i+2,j+3}}{3}$, and so on.

As shown in Fig. 14,

$$\begin{aligned}
 h_{m,n+1} &= \sum_{t=j}^{j+3} \frac{f_{i,t}}{\sum_{t=j}^{j+3} f_{i,t}} \times f_{i,t}, \\
 h_{m,n+2} &= \sum_{t=j+1}^{j+4} \frac{f_{i,t}}{\sum_{t=j+1}^{j+4} f_{i,t}} \times f_{i,t}, \\
 h_{m+1,n} &= \sum_{s=i}^{i+3} \frac{f_{s,j}}{\sum_{s=i}^{i+3} f_{s,j}} \times f_{s,j}, \\
 h_{m+2,n} &= \sum_{s=i+1}^{i+4} \frac{f_{s,j}}{\sum_{s=i+1}^{i+4} f_{s,j}} \times f_{s,j},
 \end{aligned}$$

$h_{m+1,n+1}$ is determined by 9 pixels ($f_{s,t}, s = i, \dots, i+2, t = j, \dots, j+2$), $h_{m+1,n+2}$ is determined by 9 pixels ($f_{s,t}, s = i, \dots, i+2, t = j+2, \dots, j+4$), $h_{m+2,n+1}$ is determined by 9 pixels ($f_{s,t}, s = i+2, \dots, i+4, t = j, \dots, j+2$), $h_{m+2,n+2}$ is determined by 9 pixels ($f_{s,t}, s = i+2, \dots, i+4, t = j+2, \dots, j+4$), and the method of computation is the same as mentioned above.

Because FIF has some level of self-similarity, fine image structures can be recovered using this feature. Moreover, it is highly advantageous to improve the interpolation accuracy of images using these pixels.

IV. EXPERIMENTAL AND DISCUSSION

We conduct a variety of experiments to evaluate the effectiveness of the proposed SR algorithm. Thirteen classic and recent state-of-the-art methods, including interpolation, statistical and learning methods, are used to compare the SR

TABLE V
OBJECTIVE QUALITY ASSESSMENT OF DIFFERENT METHODS (SET14)

Image	scale	bicubic	BMF	Lanczos	ScSR	NARM	SRCCNN	A+	SelfExSR	ANR	Tai's	Proposed
baboon	$\times 3$	20.406	20.626	19.858	20.098	21.467	18.264	18.706	17.916	19.019	20.119	21.538
		0.610	0.509	0.567	0.597	0.539	0.511	0.527	0.486	0.540	0.585	0.710
		0.836	0.777	0.814	0.829	0.873	0.784	0.793	0.765	0.795	0.821	0.875
barbara	$\times 3$	23.004	23.151	22.476	22.633	24.572	20.621	20.743	20.568	21.356	22.687	24.737
		0.754	0.715	0.721	0.744	0.760	0.689	0.692	0.668	0.707	0.732	0.837
		0.873	0.858	0.848	0.866	0.923	0.827	0.831	0.810	0.834	0.849	0.923
bridge	$\times 3$	21.236	20.946	20.590	20.856	22.856	15.664	20.779	18.471	21.096	20.652	23.028
		0.700	0.584	0.656	0.686	0.640	0.316	0.639	0.567	0.653	0.670	0.809
		0.859	0.796	0.835	0.854	0.905	0.720	0.835	0.796	0.836	0.844	0.906
coastguard	$\times 3$	23.660	23.383	23.135	23.339	24.984	21.638	22.018	21.267	22.354	22.635	25.068
		0.519	0.431	0.478	0.512	0.622	0.446	0.458	0.417	0.472	0.477	0.599
		0.767	0.632	0.768	0.757	0.783	0.748	0.754	0.742	0.763	0.748	0.799
comic	$\times 3$	20.030	19.420	19.507	19.645	21.990	17.618	18.154	17.342	18.513	18.897	22.131
		0.585	0.455	0.531	0.575	0.730	0.504	0.527	0.441	0.539	0.536	0.733
		0.761	0.680	0.736	0.756	0.828	0.717	0.729	0.680	0.733	0.732	0.814
face	$\times 3$	29.200	28.546	28.720	28.765	31.247	26.917	27.322	27.309	27.677	26.990	31.638
		0.710	0.685	0.678	0.693	0.781	0.623	0.637	0.624	0.651	0.678	0.786
		0.874	0.818	0.861	0.865	0.883	0.829	0.836	0.835	0.845	0.856	0.908
flowers	$\times 3$	23.703	22.999	23.090	23.255	26.742	21.368	21.891	21.048	22.216	22.400	26.351
		0.704	0.624	0.641	0.692	0.809	0.623	0.644	0.558	0.652	0.656	0.812
		0.809	0.735	0.780	0.802	0.856	0.765	0.777	0.725	0.780	0.783	0.908
foreman	$\times 3$	26.304	25.676	25.807	25.842	29.247	25.251	25.703	25.231	26.228	25.209	28.265
		0.853	0.831	0.783	0.847	0.921	0.803	0.814	0.688	0.819	0.799	0.925
		0.884	0.859	0.842	0.879	0.927	0.845	0.859	0.840	0.861	0.852	0.929
lena	$\times 3$	27.218	26.785	26.669	26.785	30.272	24.892	25.219	24.750	25.681	26.052	30.803
		0.861	0.835	0.835	0.851	0.863	0.798	0.804	0.694	0.814	0.830	0.930
		0.914	0.894	0.897	0.909	0.959	0.888	0.891	0.833	0.892	0.882	0.963
man	$\times 3$	23.595	23.346	23.071	23.169	25.853	21.296	21.811	20.582	22.148	22.667	25.832
		0.774	0.716	0.731	0.762	0.750	0.697	0.711	0.508	0.721	0.726	0.865
		0.877	0.841	0.855	0.872	0.932	0.843	0.849	0.754	0.849	0.850	0.933
monarch	$\times 3$	25.600	24.286	25.048	25.163	29.575	23.547	23.979	23.094	24.243	24.502	28.643
		0.918	0.874	0.895	0.914	0.933	0.888	0.894	0.862	0.897	0.899	0.962
		0.920	0.891	0.891	0.918	0.965	0.902	0.906	0.866	0.903	0.886	0.965
pepper	$\times 3$	26.060	25.753	25.818	25.577	28.560	25.736	26.221	24.558	26.471	25.587	30.211
		0.891	0.884	0.865	0.880	0.870	0.835	0.844	0.826	0.849	0.854	0.941
		0.914	0.906	0.897	0.909	0.966	0.895	0.901	0.879	0.900	0.886	0.960
ppt3	$\times 3$	19.877	18.619	19.737	19.454	22.276	18.248	18.905	18.223	19.157	18.679	22.664
		0.855	0.778	0.826	0.850	0.892	0.818	0.834	0.785	0.834	0.806	0.926
		0.837	0.807	0.806	0.909	0.920	0.820	0.841	0.786	0.826	0.788	0.907
zebra	$\times 3$	21.983	20.885	21.623	21.565	26.022	19.757	20.368	19.599	20.747	20.472	25.711
		0.780	0.643	0.745	0.770	0.798	0.720	0.731	0.679	0.743	0.713	0.892
		0.851	0.793	0.839	0.848	0.942	0.823	0.830	0.811	0.831	0.825	0.943
average	$\times 3$	23.705	23.172	23.225	23.296	26.118	21.486	22.272	24.425	22.636	22.682	26.187
		0.751	0.683	0.711	0.740	0.779	0.662	0.696	0.628	0.706	0.711	0.837
		0.855	0.806	0.833	0.855	0.904	0.814	0.830	0.794	0.832	0.828	0.909

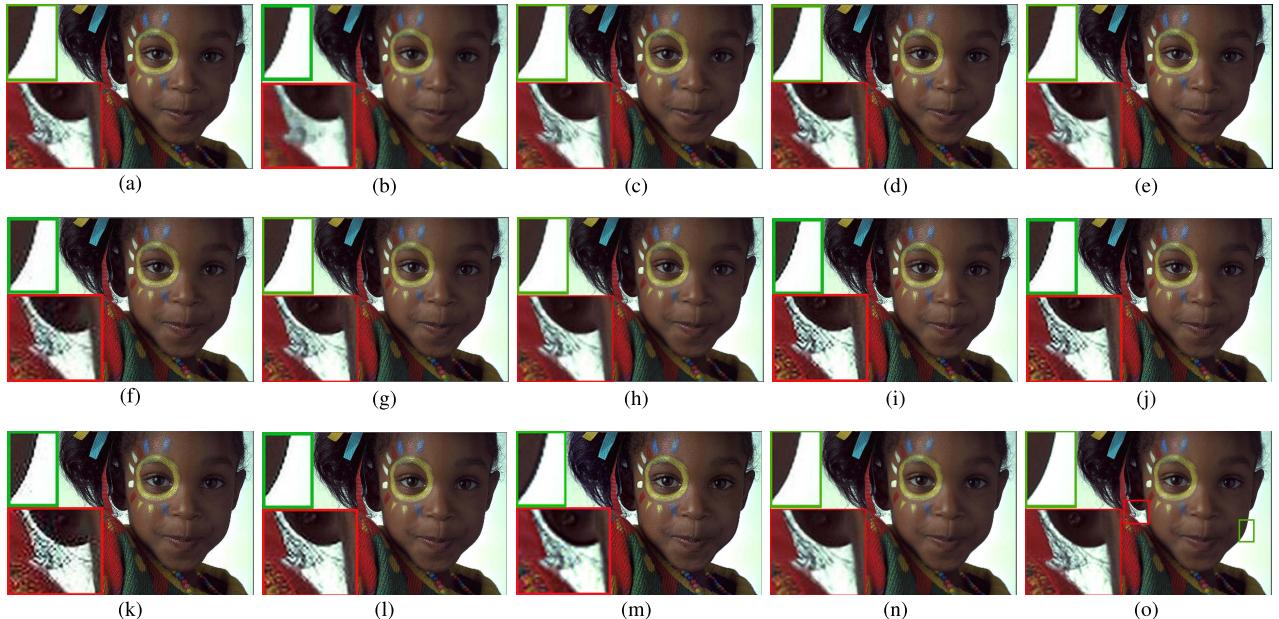


Fig. 16. Comparison of SR results ($\times 2$) on Girl image. (a) bicubic. (b) BMF. (c) NEDI. (d) ICBI. (e) DCCI. (f) Lanczos. (g) ScSR. (h) NARM. (i) SRCNN. (j) A+. (k) selfExSR. (l) ANR. (m) Tai's. (n) our method. (o) original image.

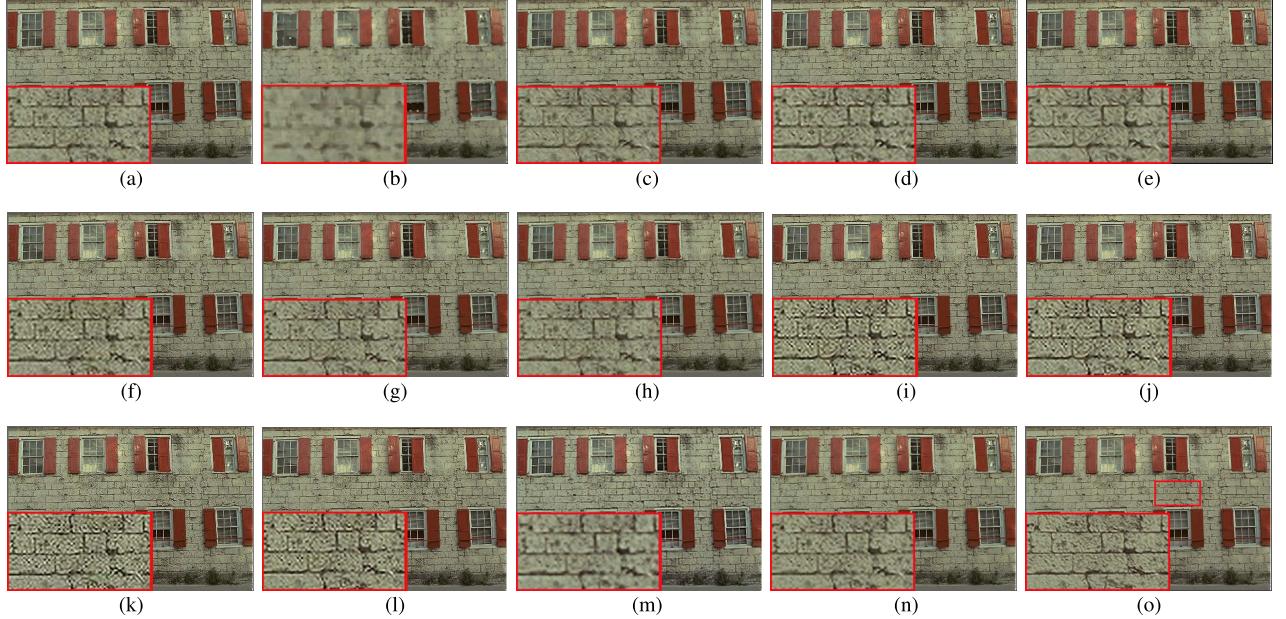


Fig. 17. Comparison of SR results ($\times 2$) on Wall image. (a) bicubic. (b) BMF. (c) NEDI. (d) ICBI. (e) DCCI. (f) Lanczos. (g) ScSR. (h) NARM. (i) SRCNN. (j) A+. (k) selfExSR. (l) ANR. (m) Tai's. (n) our method. (o) original image.

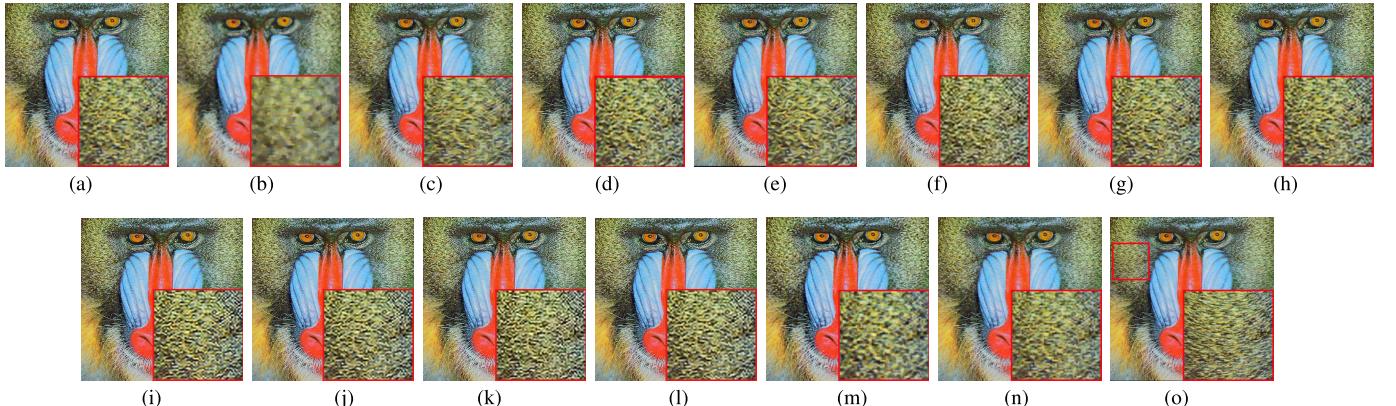


Fig. 18. Comparison of SR results ($\times 2$) on Baboon image. (a) bicubic. (b) BMF. (c) NEDI. (d) ICBI. (e) DCCI. (f) Lanczos. (g) ScSR. (h) NARM. (i) SRCNN. (j) A+. (k) selfExSR. (l) ANR. (m) Tai's. (n) our method. (o) original image.

performance. Interpolation-based SR methods include bicubic interpolation, BMF (applying median filtering for image preprocessing and then performing bicubic interpolation), NEDI [3], ICBI [5], DCCI [6], and Lanczos interpolation. Tai *et al.* [39] method is a statistical method. ScSR [13], NARM [21], SRCNN [16], A+ [17], SelfExSR [19], and ANR [15] are learning techniques. Moreover, ScSR, SRCNN, A+, and ANR rely on an external training set. Set5 [36], Set14 [37], and DIV2K [38] are employed to evaluate the performance of the single-image SR methods in this paper. In order to assess the SR performance obtained from different methods, three objective assessment indices of peak signal-to-noise ratio (PSNR), structural similarity (SSIM), and feature similarity (FSIM) are reported.

A. Experimental Configuration

In order to demonstrate the improvements of the proposed algorithm, we perform tests on the image datasets mentioned above. In our experiments, LR images are obtained by

down-sampling the HR images directly along both the horizontal and vertical directions by a factor of 2, 3, or 4. For instance, down-sampling with a factor of 2 means reducing a $N \times N$ image to a $\frac{N}{2} \times \frac{N}{2}$ image by throwing away every other row and column. On the other hand, we also test the performance of all SR methods in more realistic conditions in which LR images are obtained by unknown downscaling operators. The human visual system is more sensitive to the luminance channel than the chrominance channels. Thus, we transform the color RGB images into YCbCr color space and only conduct all compared methods in the Y channel. The remaining channels (Cb and Cr) are simply magnified using the bicubic interpolation algorithm. All experiments are performed using software provided by the authors of these methods.

B. Parameter Settings

The proposed rational fractal interpolation model contains shape parameters that can be employed to slightly adjust the shape of the surface. The error analysis in Section II-C shows

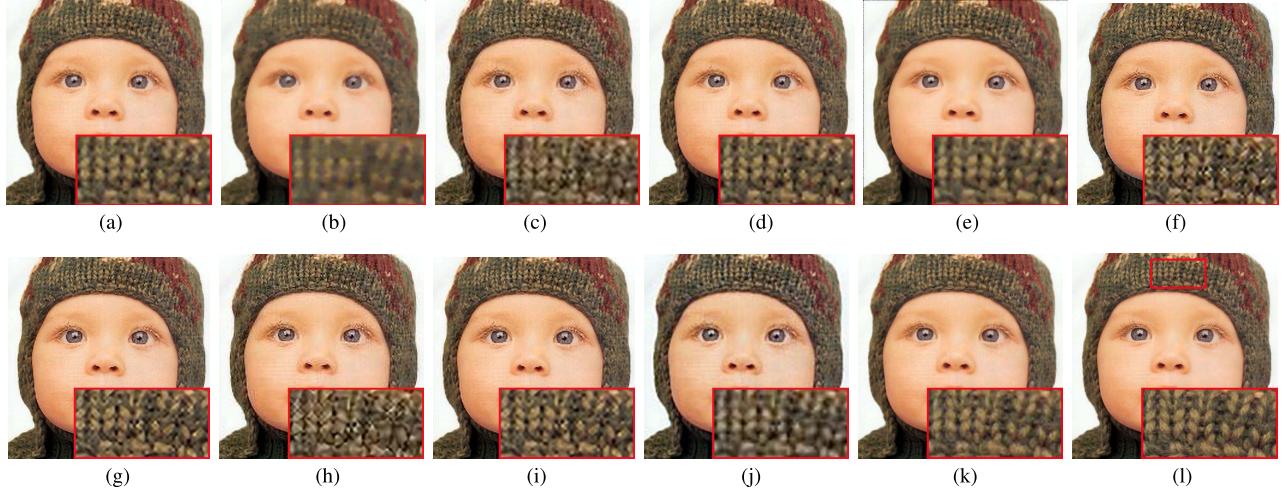


Fig. 19. Comparison of SR results ($\times 3$) on Baby image. (a) bicubic. (b) BMF. (c) Lanczos. (d) ScSR. (e) NARM. (f) SRCNN. (g) A+. (h) selfExSR. (i) ANR. (j) Tai's. (k) our method. (l) original image.

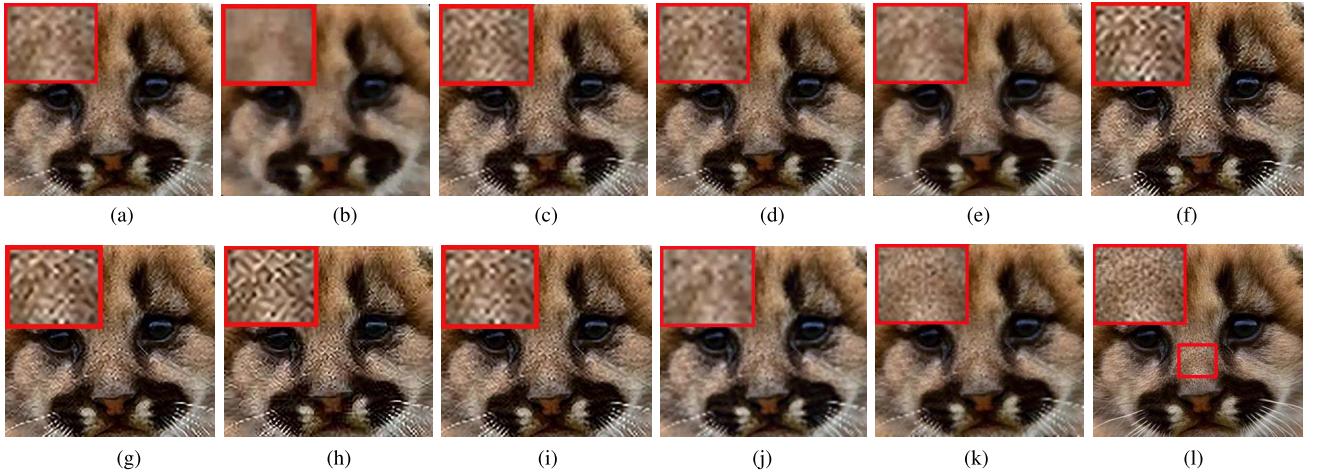


Fig. 20. Comparison of SR results ($\times 3$) on Raccoon image. (a) bicubic. (b) BMF. (c) Lanczos. (d) ScSR. (e) NARM. (f) SRCNN. (g) A+. (h) selfExSR. (i) ANR. (j) Tai's. (k) our method. (l) original image.

that the shape parameters have a minor effect in the rational fractal interpolation function compared with the scaling factors. For an interpolated image based on the model, the influence of the shape parameters is minor for local image pixels. Nevertheless, we obtain the suitable range of shape parameters using a number of training images. Fig. 15 plots the changing curved lines, which vary with different shape parameters on the training dataset. The difference is small between α and β ; for simplicity, let $\alpha = \beta$. Based on this figure, the shape parameters are determined from the numbers of experiments, and α, β, γ are selected in $[0.4, 1.2], [0.4, 1.2], [0.5, 2.5]$ at random, respectively. Furthermore, the experimental results demonstrate that the range of shape parameters that we have selected is reasonable.

C. Quality Assessment

In order to examine the performance of the proposed algorithm, we ran three versions of the algorithm for three different scale factors (2, 3, and 4).

Tables IV and V present the objective qualities generated by all methods for the test images. In Table IV, Set5 is used to evaluate the performance of upscaling factors of 2, 3, and 4, and Set14 is used to evaluate the upscaling factor of 3 in Table V. Each image corresponds to three lines in the tables, PSNR, SSIM, and FSIM from top to bottom, respectively. “-” indicates that related methods cannot achieve the function that the image is magnified with factors of 3 and 4. It is clear from these results that the proposed algorithm achieves the best results among the compared methods in terms of the three quantitative assessments. Specifically, when the image is magnified with higher scale factors, the objective data of our method are higher than is achieved using other methods. For example, our method yields the best quantitative results for test images with an upscaling factor of 4 in Table IV, 3.3–5.7 dB PSNR better than other methods on average. This result occurs because the proposed interpolation model is IFS, and more image information can be obtained with less data, compared with other methods. Moreover, the proposed method has clear

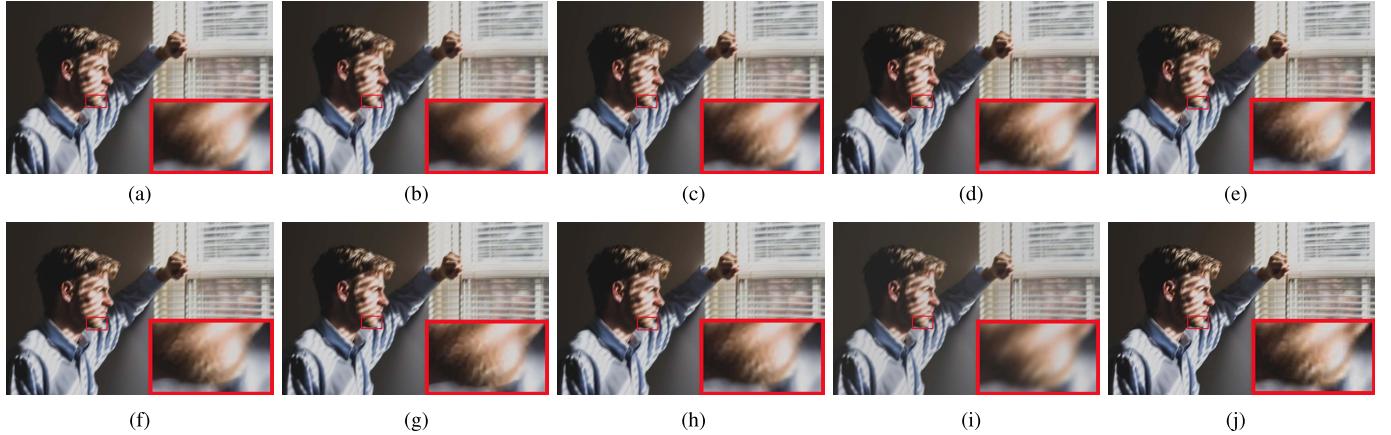


Fig. 21. Comparison of SR results ($\times 4$) on Man image. (a) bicubic. (b) BMF. (c) Lanczos. (d) ScSR. (e) SRCNN. (f) A+. (g) selfExSR. (h) ANR. (i) Tai's. (j) our method.

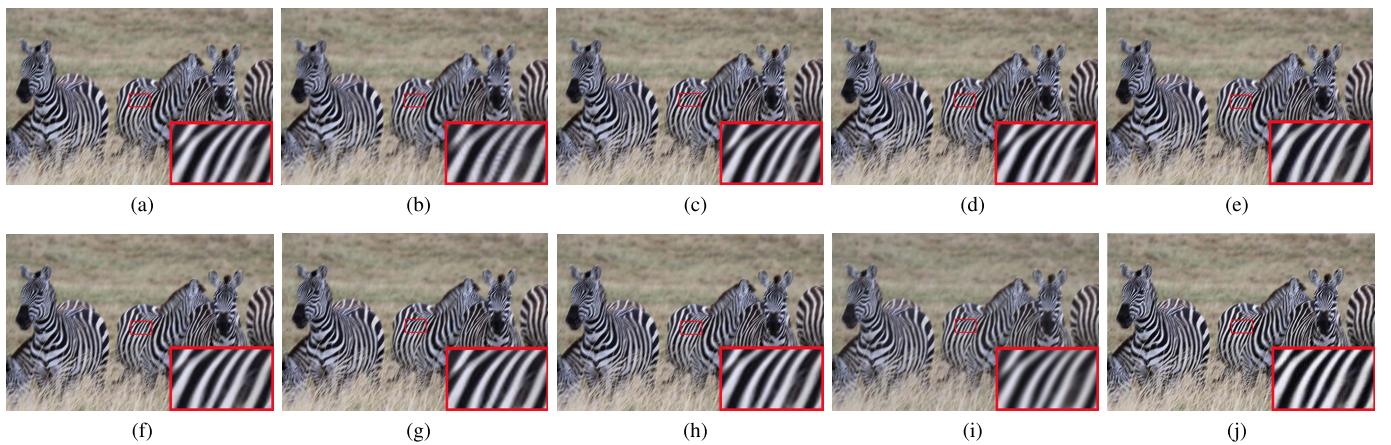


Fig. 22. Comparison of SR results ($\times 4$) on Zebra image. (a) bicubic. (b) BMF. (c) Lanczos. (d) ScSR. (e) SRCNN. (f) A+. (g) selfExSR. (h) ANR. (i) Tai's. (j) our method.

advantages compared with other methods in the SSIM and FSIM, which are used for measuring the structure similarity between two images. This result demonstrates that the fractal model is quite efficient for describing image textures.

For visual comparisons, Figs. 16-22 present the SR results of these methods. Figs. 16-18 show images with a resolution enhancement factor of 2, Figs. 19-20 show images with a resolution enhancement factor of 3, and Figs. 21-22 show the SR results of different approaches using an upscaling factor of 4, where LR images are obtained by unknown downscaling operators. Overall, compared with those of the other state-of-the-art methods, the results of the proposed scheme are more appealing. The primary advantage of our approach is its ability to obtain vivid texture regions.

As shown in Figs. 16-18, the proposed method can recover more pleasing details than the other compared methods. The images generated by NEDI, ICBI, and DCCI are fuzzy and distorted. Bicubic, BMF, and Tai's methods also suffer from blurred artifacts. The results achieved by Lanczos and ScSR show unnatural details. The texture details are not efficiently preserved in SRCNN, A+, selfExSR, and ANR. Furthermore, in Fig. 16, the improvement in edge preservation is significant

in the proposed algorithm's image compared to the other methods, where jaggy and blurry artifacts are observed along the edges in the magnified local regions. NARM, NEDI, and our proposed method can efficiently preserve the sharpness of edges.

As shown in Figs. 19-20, overall, with severe blurring artifacts, the performance of BMF is always inferior to that of the other methods. Fig. 19(c) and Fig. 20(c) show the results of the bicubic method, which fails to recover the structural information of images. As evidenced, some fine image structures are not recovered by Lanczos. ScSR suffers from aliasing artifacts in the texture region. The images obtained by NARM produce overly smooth edges, resulting in a loss of the fine textures. In the SRCNN, A+, selfExSR, and ANR magnified images, artifacts appear, and the texture is twisted heavily. Tai's method suffers from fuzzy artifacts. Our method succeeds in avoiding these undesired effects. Additionally, the images generated by the proposed method show an outstanding improvement on the compared methods.

Figs. 21-22 show the SR experiments on two LR benchmarks with unknown sampling kernels. A comparison of different SR approaches for restoring the texture details is

shown in Fig. 21. Bicubic, BMF, Lanczos, ScSR, and Tai's methods are prone to blurring the image details. Displeasing artifacts along edges are produced by SRCNN and selfExSR. selfExSR also suffers from blocking artifacts. ANR, A+, and our method achieve better results among the considered methods. As shown in Fig. 22, the proposed algorithm exhibits a visually appealing appearance compared to the comparison method.

To summarize, our method outperforms the other algorithms in terms of the objective quality of the SR images. Furthermore, from the visual results presented, the proposed algorithm can perform better in image detail areas than the other methods, and it can retain the edge region well. Therefore, the constructed interpolation model is quite efficient for describing the space characteristics information of images.

V. CONCLUSION

In this paper, we present a single-image SR algorithm based on the rational fractal interpolation model, which is more suitable for describing the structures of an image. First, for each LR image patch, the isoline method is employed to detect texture, such that more detailed textures can be obtained, and the LR image is divided into texture regions and non-texture regions. Second, in the interpolation model, the scaling factors play an important role, whereas the influence of the shape parameters is minor. Based on the relationship between scaling factors and the fractal dimension, the scaling factors are accurately calculated by using the image local structure feature. Nevertheless, a suitable range of shape parameters is obtained using a number of training images. Then, rational fractal interpolation and rational interpolation are used in the texture region and the non-texture region, respectively. Specifically, each LR image patch is first interpolated, and the interpolation is extended to the entire image by traversing each patch. Finally, an HR image is obtained by pixel mapping. Because the proposed rational fractal interpolation function is an IFS, the image can be amplified at any integral multiple by selecting a suitable mapping. The experimental results demonstrate that the proposed algorithm achieves competitive performance and generates high-quality SR images with sharp edges and rich texture.

APPENDIX

A. Iterated Function System (IFS)

Let $\{(x_i, y_j, z_{i,j}), i = 1, 2, \dots, N; j = 1, 2, \dots, M\}$ be a given set of data points, closed interval $I = [a, b] \subseteq \mathcal{R}$ contains $\{x_1, x_2, \dots, x_N\}$, and closed interval $J = [c, d] \subseteq \mathcal{R}$ contains $\{y_1, y_2, \dots, y_M\}$. Set $I_i = [x_i, x_{i+1}]$ for $i \in \mathcal{I} = \{1, 2, \dots, N-1\}$ and $J_j = [y_j, y_{j+1}]$ for $j \in \mathcal{J} = \{1, 2, \dots, M-1\}$. Denote $\mathcal{I} = \{1, 2, \dots, N\}$, $\mathcal{J} = \{1, 2, \dots, M\}$. Let $\phi_i(x)$ be contractive homeomorphisms: $I \rightarrow I_i$,

$$\begin{aligned}\phi_i(x_1) &= x_i, \quad \phi(x_N) = x_{i+1}, \\ |\phi_i(c_1) - \phi_i(c_2)| &\leq \lambda |c_1 - c_2|, \quad \forall c_1, c_2 \in I,\end{aligned}$$

where $0 \leq \lambda < 1$. Let $\varphi_j(y)$ be contractive homeomorphisms: $J \rightarrow J_j$

$$\begin{aligned}\varphi_j(y_1) &= y_j, \quad \varphi_j(y_M) = y_{j+1}, \\ |\varphi_j(d_1) - \varphi_j(d_2)| &\leq \mu |d_1 - d_2|, \quad \forall d_1, d_2 \in J,\end{aligned}$$

where $0 \leq \mu < 1$. Furthermore, let $F = I \times J \times \mathcal{R}$, and for $i \in \mathcal{I}, j \in \mathcal{J}$, the continuous mappings $F_{i,j} : F \rightarrow \mathcal{R}$ fulfill

$$\begin{aligned}F_{i,j}(x_1, y_1, z_{1,1}) &= z_{i,j}, \quad F_{i,j}(x_1, y_M, z_{1,M}) = z_{i,j+1}, \\ F_{i,j}(x_N, y_1, z_{N,1}) &= z_{i+1,j}, \quad F_{i,j}(x_N, y_M, z_{N,M}) = z_{i+1,j+1}.\end{aligned}$$

For $i \in \mathcal{I}, j \in \mathcal{J}$, define functions $w_{i,j} : F \rightarrow F$,

$$w_{i,j}(x, y, z) = (\phi_i(x), \varphi_j(y), F_{i,j}(x, y, z)), \quad (9)$$

then, IFS (9) generates a unique attractor which is a graph of a continuous function $\Phi : I \times J \rightarrow \mathcal{R}$ satisfying $\Phi(x_i, y_j) = z_{i,j}$, function Φ is called a FIF, and the following relation holds:

$$\Phi(\phi_i(x), \varphi_j(y)) = F_{i,j}(x, y, \Phi(x, y)), \quad i \in \mathcal{I}, j \in \mathcal{J},$$

or

$$\Phi(x, y) = F_{i,j}(\phi_i^{-1}(x), \varphi_j^{-1}(y), \Phi \circ (\phi_i^{-1}(x)).$$

FIFs corresponding to the following IFS form were investigated the most extensively:

$$\begin{cases} \phi_i(x) = a_i x + b_i, \\ \varphi_j(y) = c_j y + d_j, \\ F_{i,j}(x, y, z) = s_{i,j} z + q_{i,j}(x, y), \end{cases} \quad (10)$$

where

$$\begin{aligned}a_i &= \frac{x_{i+1} - x_i}{x_N - x_1}, \quad b_i = \frac{x_N x_i - x_1 x_{i+1}}{x_N - x_1} \\ c_j &= \frac{y_{j+1} - y_j}{y_M - y_1}, \quad d_j = \frac{y_M y_j - y_1 y_{j+1}}{y_M - y_1},\end{aligned}$$

with $-1 < s_{i,j} < 1$. The parameters $s_{i,j}$ are called the vertical scaling factors.

B. Construction of Bivariate Rational FIFs

Let $\Omega = [a, b; c, d]$ be the plane region, $\Delta = \{(x_i, y_j, f_{i,j}, d_{i,j}^*, d_{i,j}) : i \in \mathcal{I}; j \in \mathcal{J}\}$ a given set of data points, where $a = x_1 < x_2 < \dots < x_N = b$ and $c = y_1 < y_2 < \dots < y_M = d$ are the knot spacings, $f_{i,j}$ represents $f_{i,j}(x, y)$ at the point (x_i, y_j) , and $d_{i,j}^*$ and $d_{i,j}$ are the chosen first-order partial derivatives $\frac{\partial f(x,y)}{\partial x}$ and $\frac{\partial f(x,y)}{\partial y}$ at the knots (x_i, y_j) , respectively. Denote $h_i = x_{i+1} - x_i$, $l_j = y_{j+1} - y_j$, $H_N = x_N - x_1$, $L_M = y_M - y_1$, and for any point $(x, y) \in \Omega$, $\theta = \frac{x-x_1}{x_N-x_1}$, $\eta = \frac{y-y_1}{y_M-y_1}$.

Denote

$$\begin{aligned}w_{0,0}(\theta, \bullet_{i,j}^*) &= \frac{(1-\theta)^2(a_{i,j}^* + \theta\gamma_{i,j}^*)}{(1-\theta)^2a_{i,j}^* + \theta(1-\theta)\gamma_{i,j}^* + \theta^2\beta_{i,j}^*}, \\ w_{0,1}(\theta, \bullet_{i,j}^*) &= \frac{\theta^2(\beta_{i,j}^* + (1-\theta)\gamma_{i,j}^*)}{(1-\theta)^2a_{i,j}^* + \theta(1-\theta)\gamma_{i,j}^* + \theta^2\beta_{i,j}^*}, \\ w_{1,0}(\theta, \bullet_{i,j}^*) &= \frac{\theta(1-\theta)^2a_{i,j}^*}{(1-\theta)^2a_{i,j}^* + \theta(1-\theta)\gamma_{i,j}^* + \theta^2\beta_{i,j}^*}, \\ w_{1,1}(\theta, \bullet_{i,j}^*) &= \frac{\theta^2(1-\theta)\beta_{i,j}^*}{(1-\theta)^2a_{i,j}^* + \theta(1-\theta)\gamma_{i,j}^* + \theta^2\beta_{i,j}^*}.\end{aligned}$$

We construct bivariate rational spline interpolants as follows: first, the x-direction interpolating curve $P_{i,j}^*(\phi_i(x))$ in $[x_1, x_N]$ is defined by:

$$P_{i,j}^*(\phi_i(x)) = \sum_{r=0}^1 [\omega_{0,r}(\theta, \bullet_{i,j}^*) f_{i+r,j} + h_i \omega_{1,r}(\theta, \bullet_{i,j}^*) d_{i+r,j}^*],$$

where $\alpha_{i,j}^* > 0$, $\beta_{i,j}^* > 0$, and $\gamma_{i,j}^* > 0$. The interpolant $P_{i,j}^*(\phi_i(x))$ satisfies:

$$P_{i,j}^*(x_r) = f_{r,j}, \quad P_{i,j}^{*\prime}(x_r) = d_{r,j}^*, \quad r = i, i+1.$$

Then, for $i \in \mathcal{I}$, $j \in \mathcal{J}$ and $\alpha_{i,j} > 0$, $\beta_{i,j} > 0$, $\gamma_{i,j} > 0$, using the x-direction interpolation function $P_{i,j}^*(\phi_i(x))$ defines the bivariate rational interpolation function on Ω as follows:

$$P_{i,j}(\phi_i(x), \varphi_j(y)) = \sum_{s=0}^1 [\omega_{0,s}(\eta, \bullet_{i,j}) P_{i,j+s}^*(\phi_i(x)) + l_j \omega_{1,s}(\eta, \bullet_{i,j}) D_{i,j+s}(\phi_i(x))], \quad (11)$$

where

$$D_{i,j}(\phi_i(x)) = \omega_{0,0}(\theta, \bullet_{i,j}^*) d_{i,j} + \omega_{0,1}(\theta, \bullet_{i,j}^*) d_{i+1,j}.$$

The interpolant $P_{i,j}(x, y)$ satisfies:

$$\begin{aligned} P_{i,j}(x_r, y_s) &= f_{r,s}, \quad \frac{\partial P_{i,j}(x_r, y_s)}{\partial x} = d_{r,s}^*, \\ \frac{\partial P_{i,j}(x_r, y_s)}{\partial y} &= d_{r,s}, \quad r = i, i+1; s = j, j+1. \end{aligned}$$

Motivated by the construction method of bivariate rational spline interpolation, rational perturbation base functions are defined as follows:

$$B_{i,j}(x, y) = \sum_{s=0}^1 [\omega_{0,s}(\eta, \bullet_{i,j}) B_{s(M-1)+1,i,j+s}(x) + L_M \omega_{1,s}(\eta, \bullet_{i,j}) K_{s(M-1)+1,i,j+s}(x)],$$

where

$$\begin{aligned} B_{k,i,j}(x) &= \sum_{r=0}^1 [\omega_{0,r}(\theta, \bullet_{i,j}^*) f_{r(N-1)+1,k} \\ &\quad + H_N \omega_{1,r}(\theta, \bullet_{i,j}^*) d_{r(N-1)+1,k}^*], \\ K_{k,i,j}(x) &= \omega_{0,0}(\theta, \bullet_{i,j}^*) d_{1,k} + \omega_{0,1}(\theta, \bullet_{i,j}^*) d_{N,k}. \end{aligned}$$

Then, function $B_{i,j}(x, y)$ satisfies

$$\begin{aligned} B_{i,j}(x_r, y_s) &= f_{r,s}, \quad \frac{\partial B_{i,j}(x_r, y_s)}{\partial x} = d_{r,s}^*, \\ \frac{\partial B_{i,j}(x_r, y_s)}{\partial y} &= d_{r,s}, \quad r = 1, N; s = 1, M. \end{aligned}$$

Now, consider the following IFS:

$$\begin{cases} \phi_i(x) = a_i x + b_i, \\ \varphi_j(y) = c_j y + d_j, \\ F_{i,j}(x, y, z) = s_{i,j} z + P_{i,j}(\phi_i(x), \varphi_j(y)) \\ \quad - s_{i,j} B_{i,j}(x, y), \end{cases} \quad (12)$$

then, the IFS $\{\Omega \times \mathcal{R}; (\phi_i(x), \varphi_j(y), F_{i,j}(x, y, z))\}$ defined by (12) admits a unique attractor G , and G is the graph of a continuous function $\Phi(x, y)$, and

$$\begin{aligned} \Phi(\phi_i(x), \varphi_j(y)) &= s_{i,j} \Phi(x, y) + P_{i,j}(\phi_i(x), \varphi_j(y)) \\ &\quad - s_{i,j} B_{i,j}(x, y). \end{aligned} \quad (13)$$

Denote:

$$\begin{aligned} A &= (\omega_{0,0}(\theta, \bullet_{i,j}^*), \omega_{0,1}(\theta, \bullet_{i,j}^*), \omega_{1,0}(\theta, \bullet_{i,j}^*), \omega_{1,1}(\theta, \bullet_{i,j}^*)), \\ B &= (\omega_{0,0}(\eta, \bullet_{i,j}), \omega_{0,1}(\eta, \bullet_{i,j}), \omega_{1,0}(\eta, \bullet_{i,j}), \omega_{1,1}(\eta, \bullet_{i,j}))^T, \\ E &= \begin{pmatrix} E_{0,0} & E_{0,1} & D_{0,0} & D_{0,1} \\ E_{1,0} & E_{1,1} & D_{1,0} & D_{1,1} \\ D_{0,0}^* & D_{0,1}^* & 0 & 0 \\ D_{1,0}^* & D_{1,1}^* & 0 & 0 \end{pmatrix}, \end{aligned}$$

where for $r, s = 0, 1$,

$$\begin{aligned} E_{r,s} &= f_{i+r,j+s} - s_{i,j} f_{r(N-1)+1,s(M-1)+1}, \\ D_{r,s}^* &= h_i d_{i+r,j+s}^* - s_{i,j} H_N d_{r(N-1)+1,s(M-1)+1}^*, \\ D_{r,s} &= l_j d_{i+r,j+s} - s_{i,j} L_M d_{r(N-1)+1,s(M-1)+1}. \end{aligned}$$

When the shape parameters satisfy $\bullet_{i,j}^* = \bullet_{i,j+1}^*$, the fractal surface $\Phi(x, y)$ defined by (13) can be rewritten as the following simple matrix form:

$$\Phi(\phi_i(x), \varphi_j(y)) = s_{i,j} \Phi(x, y) + AEB. \quad (14)$$

REFERENCES

- [1] S. C. Park, M. K. Park, and M. G. Kang, "Super-resolution image reconstruction: A technical overview," *IEEE Signal Process. Mag.*, vol. 20, no. 3, pp. 21–36, May 2003.
- [2] H. S. Hou and H. Andrews, "Cubic splines for image interpolation and digital filtering," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 26, no. 6, pp. 508–517, Dec. 1978.
- [3] X. Li and M. T. Orchard, "New edge-directed interpolation," *IEEE Trans. Image Process.*, vol. 10, no. 10, pp. 1521–1527, Oct. 2001.
- [4] L. Zhang and X. Wu, "An edge-guided image interpolation algorithm via directional filtering and data fusion," *IEEE Trans. Image Process.*, vol. 15, no. 8, pp. 2226–2238, Aug. 2006.
- [5] A. Giachetti and N. Asuni, "Real-time artifact-free image upscaling," *IEEE Trans. Image Process.*, vol. 20, no. 10, pp. 2760–2768, Oct. 2011.
- [6] 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.
- [7] P. Thévenaz, T. Blu, and M. Unser, "Image interpolation and resampling," in *Proc. Handbook Med. Imag.*, 2000, pp. 393–420.
- [8] S. Carrato and L. Tenze, "A high quality 2× image interpolator," *IEEE Signal Process. Lett.*, vol. 7, no. 6, pp. 132–134, Jun. 2000.
- [9] Y. Liu, Y. Zhang, Q. Guo, and C. Zhang, "Image interpolation based on weighted and blended rational function," in *Proc. ACCV Workshops Comput. Vis.*, Apr. 2014, pp. 78–88.
- [10] K. I. Kim and Y. Kwon, "Example-based learning for single-image super-resolution," in *Pattern Recognition* (Lecture Notes in Computer Science), vol. 5096. Berlin, Germany: Springer, Jun. 2008, pp. 456–465.
- [11] Y. Tang, Y. Yuan, P. Yan, and X. Li, "Greedy regression in sparse coding space for single-image super-resolution," *J. Vis. Commun. Image Represent.*, vol. 24, no. 2, pp. 148–159, Feb. 2013.
- [12] J. Yang, J. Wright, T. Huang, and Y. Ma, "Image super-resolution as sparse representation of raw image patches," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.
- [13] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Trans. Image Process.*, vol. 19, no. 11, pp. 2861–2873, Nov. 2010.
- [14] Y. HaCohen, R. Fattal, and D. Lischinski, "Image upsampling via texture hallucination," in *Proc. IEEE Int. Conf. Comput. Photograph.*, Mar. 2010, pp. 1–8.

- [15] R. Timofte, V. De Smet, and L. Van Gool, "Anchored neighborhood regression for fast example-based super-resolution," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1920–1927.
- [16] 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. (ECCV)*, 2014, pp. 184–199.
- [17] R. Timofte, V. De Smet, and L. Van Gool, "A+: Adjusted anchored neighborhood regression for fast super-resolution," in *Proc. Asian Conf. Comput. Vis. (ACCV)*, Nov. 2015, pp. 111–126.
- [18] 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.
- [19] J.-B. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 5197–5206.
- [20] C.-Y. Yang and M.-H. Yang, "Fast direct super-resolution by simple functions," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2013, pp. 561–568.
- [21] 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.
- [22] Y. Xu, H. Ji, and C. Fermüller, "Viewpoint invariant texture description using fractal analysis," *Int. J. Comput. Vis.*, vol. 83, no. 1, pp. 85–100, 2009.
- [23] Y. Xu, X. Yang, H. Ling, and H. Ji, "A new texture descriptor using multifractal analysis in multi-orientation wavelet pyramid," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2010, pp. 161–168.
- [24] Y. Xu, Y. Quan, H. Ling, and H. Ji, "Dynamic texture classification using dynamic fractal analysis," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Nov. 2011, pp. 1219–1226.
- [25] M. Voorons, M. Germain, G. B. Benie, and K. Fung, "Segmentation of high resolution images based on the multifractal analysis," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2003, vol. 6, no. 6, pp. 3531–3533.
- [26] H. Xu, J. Yan, N. Persson, W. Lin, and H. Zha, "Fractal dimension invariant filtering and its CNN-based implementation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3825–3833.
- [27] M. Liu, Y. Zhao, J. Liang, C. Lin, H. Bai, and C. Yao, "Depth map up-sampling with fractal dimension and texture-depth boundary consistencies," *Neurocomputing*, vol. 257, no. 27, pp. 185–192, Sep. 2017.
- [28] Y. C. Wee and H. J. Shin, "A novel fast fractal super resolution technique," *IEEE Trans. Consum. Electron.*, vol. 56, no. 3, pp. 1537–1541, Aug. 2010.
- [29] 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.
- [30] Q. Duan, H. Zhang, Y. Zhang, and E. H. Twizell, "Bounded property and point control of a bivariate rational interpolating surface," *Comput. Math. Appl.*, vol. 52, nos. 6–7, pp. 975–984, Sep./Oct. 2006.
- [31] Y. Zhang, F. Bao, C. Zhang, and D. Qi, "Local shape control of a bivariate rational interpolating surface with mixing conditions," in *Proc. ISVD*, Jun. 2011, pp. 200–205.
- [32] Y. Zhang, F. Bao, C. Zhang, and Q. Duan, "A weighted bivariate blending rational interpolation function and visualization control," *J. Comput. Anal. Appl.*, vol. 14, no. 7, pp. 1303–1321, Nov. 2012.
- [33] Q. Sun, F. Bao, Y. Zhang, and Q. Duan, "A bivariate rational interpolation based on scattered data on parallel lines," *J. Vis. Commun. Image Represent.*, vol. 24, no. 1, pp. 75–80, Jan. 2013.
- [34] S. Osher and J. A. Sethian, "Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations," *J. Comput. Phys.*, vol. 79, no. 1, pp. 12–49, Nov. 1988.
- [35] S. Novianto, Y. Suzuki, and J. Maeda, "Near optimum estimation of local fractal dimension for image segmentation," *Pattern Recognit. Lett.*, vol. 24, nos. 1–3, pp. 365–374, Jan. 2003.
- [36] 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. (BMVC)*, Guildford, U.K., Sep. 2012, pp. 135.1–135.10.
- [37] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *Proc. Int. Conf. Curves Surfaces*, Jun. 2010, pp. 711–730.
- [38] E. Agustsson and R. Timofte, "NTIRE 2017 challenge on single image super-resolution: Dataset and study," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 1122–1131.
- [39] Y.-W. Tai, S. Liu, M. S. Brown, and S. Lin, "Super resolution using edge prior and single image detail synthesis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2010, pp. 2400–2407.



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, 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. His current research interests include computer-aided geometric design, digital image processing, computational geometry, and function approximation.



Qinglan Fan received the B.E. degree from the School of Information and Electrical Engineering, Ludong University, Yantai, China, in 2015. She is currently pursuing the M.S. degree with the Shandong Provincial Key Laboratory of Digital Media Technology, Shandong University of Finance and Economics. Her research interests include image/video process and fractals.



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, Xian, 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.



Yifang Liu received the bachelor's degree in electronic information science and technology and the master's degree in computer science from the Shandong University of Finance and Economics, Jinan, China, in 2012 and 2015, respectively. She is currently pursuing the Ph.D. degree with the University at Buffalo, Buffalo, NY, USA. Her research interests include robotics, computer vision, and image processing.



Caiming Zhang received the B.S. and M.E. degrees in computer science from Shandong University in 1982 and 1984, respectively, and the Dr. Eng. degree in computer science from the Tokyo Institute of Technology, Japan, in 1994. From 1997 to 2000, he held a visiting position with the University of Kentucky, Lexington, KY, USA. He is currently a Professor and the Doctoral Supervisor with the School of Computer Science and Technology, Shandong University. He is also the Dean and a Professor with the School of Computer Science and Technology, Shandong University of Finance and Economics. His research interests include CAGD, CG, information visualization, and medical image processing.