Adaptive Rational Fractal Interpolation Function for Image Super-Resolution via Local Fractal Analysis

Xunxiang Yao[a], Qiang Wu[a], Peng Zhang[a], Fangxun Bao[b]

[a]*Faculty of Engineering and Information Technology, University of Technology Sydney, Australia*
[b]*School of Mathematics, Shandong University, China*

Abstract

Image super-resolution aims to generate high-resolution image based on the given low-resolution image and to recover the details of image. The common approaches include reconstruction-based methods and interpolation-based methods. However, these existing methods show difficulty in processing the regions of image with complicated texture. To tackle such problems, fractal geometry is applied on image super-resolution, which demonstrates its advantages when describing the complicated details in image. The common fractal-based method regards the whole image as a single fractal set. That is, it does not distinguish the complexity difference of texture across all regions of image regardless of smooth regions or texture rich regions. Due to such strong presumption, it causes artificial errors while recovering smooth area and texture blurring at the regions with rich texture. In this paper, the proposed method produces rational fractal interpolation model with various setting at different regions to adapt to the local texture complexity. In order to facilitate such mechanism, the proposed method is able to segment the image region according to its complexity which is determined by its local fractal dimension. Thus, the image super-resolution process is cast to an optimization problem where local fractal dimension in each region is further optimized until the optimization convergence is reached. During the optimization (i.e. super-resolution), the overall image complexity (determined by local fractal dimension) is maintained. Compared with state-of-the-art method, the proposed method shows promising performance according to qualitative evaluation and quantitative evaluation.

Keywords: image super-resolution, texture detail, fractal function, vertical scaling factor, fractal dimension

1. Introduction

Image super-resolution (SR) mainly aims to obtain a high-quality image with higher resolution from low-resolution image. The essential of this technique is to estimate unknown pixels via the known information of the low-resolution image. Image super-resolution technique has been used in various fields, such as military, medical, remote sensing satellite, television, etc.

The approaches of super-resolution can be roughly grouped into four categories: interpolation based methods, reconstruction-based methods, learning-based methods, and fractal-based methods. Interpolation-based methods refer to recover a continuous signal by estimating image data from a set of discrete image data samples. Reconstruction-based methods model degradation of images through point diffusion function. Thus, the degraded image can be represented as $L = BH + N$, where H is the unknown high-resolution image, B represents blurring kernel, and N is the added noise. Learning-based methods were developed to capture complex relationship between high-resolution image and low-resolution image on image patches. Please note that fractal-based methods normally can be included in the category of interpolation based methods i.e. fractal interpolation. To highlight the uniqueness of fractal-based methods and their advantages, this paper reviews such typical super-resolution methods using fractal theory separately.

The simplest type of interpolation-based methods is classic interpolation functions under images 'smoothness' assumption, such as bilinear [1], bicubic [2] and spline function[3]. Besides these classic methods, many other interpolation methods were presented, such as the edge-guided interpolation method [4] and ICBI [5]. They are very simple and fast. However, they usually generate blurring edge and artificial texture details.

Due to strong discontinuities contained in images, the smoothness assumption of interpolation based method leads to blurring edges and texture disorder. The reconstruction-based method solves this problem through prior knowledge (gradient prior [6], edge prior [7, 8], similarity prior [9] and texture prior [10]). These approaches are quite effective in keeping edge structure and suppress some artifacts. However, for the larger up-sampling factors, the prior will be invalid, and the quality of the reconstructed image drops dramatically.

Learning-based methods can be further categorized into two types: example-based methods, self-example-based methods. Example-based super-resolution methods learn mapping function from low-resolution patches to its corresponding high-resolution patches with external training datasets. Representative methods include k-nearest neighbor (k-NN) learning [11], manifold learning [12], sparse coding [13] and regression-based [14, 15] methods. Self-example-based super-resolution algorithms [16, 17, 18], which only need the information of image itself, are based on the observation that image often has certain numbers of structures that they repeat in/ across scales at other positions. Although the learning-based method has 'photo-realistic' result, they have the following disadvantages: (1) Longer processing time. (2) Lack of flexibility: It cannot magnify image (i.e. super-resolution) with arbitrary scale. For the different scaling factors, the model needs to be trained separately. (3)Unexpected artifacts in complicated texture region.

In order to tackle problems on the artifacts in rich texture regions exposed in all methods presented above, the fractal theory is introduced into the research on image super-resolution, and it provides an easier way to describe the complex natural surface. In fractal geometry, the image can be reproduced with fractal function. The image intensity of a natural object can be regarded as a kind of fractal [19]. Based on the previous research, we categorized the fractal-based super-resolution method into three categories: Iterated Fractal System (IFS), Fractal Brownian Function, fractal theory combined with another method. AE Jacquin [20] proposed image compression via fractal theory without human interactive. This kind of method is based on the assumption that image redundancy can be exploited through self-transform ability on a block basis. The IFS based method [21, 22] includes two main steps: First, the image is divided into non-overlapping range blocks. For each range block, the image is searched for a domain block that is very similar to range block. This kind of method has limitations that it requires image must be a 'strictly' fractal image. It means that every single part is a copy of the whole image. Actually, natural image present self-similarity on the whole but not in such a strict way. For the Fractal Brownian Function based method, authors [23, 24] investigated the potential of fractal interpolation. In order to improve the efficiency of the fractal coding method, Wee and Shin [25] proposed a novel fractal super-resolution algorithm where contrast scaling factor and block determined the fractal affine transform jointly. This method has lower computation complexity, but, it lacks flexibility and adaptability. Later, the fractal technique is combined with another method, Xu et

al. [26] established a fractal model for an image that employed the gradient as a fractal measure for the point set of an image. By using the assumption of scale invariance of local fractal dimension and length, the problem of image super-resolution and enhancement can be solved jointly. Then, Yu et al. [27] presented a super-resolution algorithm that combines fractal with the example-based method. The above mentioned fractal-based methods preserve vivid texture details than other non-fractal based methods. One of the disadvantages is that the local fractal feature (e.g. fractal scaling factors and fractal dimension) and underlying characteristic coherence of the image are not well jointly considered. Some unexpected artifacts still occurred since they ignore the importance of the vertical scaling factor that reflects self-similarity.

In this paper, we develop a novel rational fractal model based on rational spline function [28]. Based on such model, we propose a new adaptive rational fractal super-resolution (ARFSR) algorithm which applies local fractal feature rather than uniform fractal features across the whole image [25]. We split the whole image into patches and classify those patches into different categories according to its complexity. For every class, the different fractal interpolation function (with different setting) is chosen adaptively. In the 'smooth' area, the rational fractal function deduces to rational function. In the edge and rich texture area, we exploit the relationship between vertical scaling factors and local fractal dimension to maintain texture details. The complexity of image keeps invariance by adjusting vertical scaling factors (actually, the complexity is decreased). With the help of optimizing vertical scaling factor, rational fractal function is utilized to maintain the self-similarity characteristic. The proposed method provides much flexibility in upsampling scaling factor. Our method can magnify image with continuous integer ratio while learning-based must be re-trained for every scaling factor. At the same time, it preserves texture details efficiently.

To obtain 'photo-realistic' high-resolution image with vivid texture details and sharp edges, we construct an adaptive rational fractal super-resolution method that exploits the relationship between local fractal dimension and vertical scaling factor. The proposed super-resolution method has the following unique characteristics: 1) To address the problems mentioned above in the previous fractal-based method, we treat the whole image as a multifractal set. According to the texture complexity of local region, the rational fractal interpolation model (with different setting) is chosen adaptively; 2) We cast super-resolution process as an optimization process during which the

overall image complexity (determined by local fractal dimension) is maintained. We assume that the complexity of upscaled image is invariance (in fact, the complexity of image decreased for the loss of high-frequency information). The optimization technique is applied via the vertical scaling factor.

The remainder of this paper is organized as follows: Section 2 describes the related work. In section 3, we present the AFSR algorithm which focuses on the determination of vertical scaling factor. Experiments and discussions are given to evaluate the effectiveness of the algorithm in section 4. Finally, section 5 concludes this paper.

2. Related work

2.1. Rational Fractal Interpolation for SR

Previously, earlier methods use interpolation function based on 'smoothness' assumption [29] which converts the digital image as a continuous smooth surface. However, those methods show limitations in blurred edges and artificial texture details. The fractal theory is introduced into the research on image super-resolution, and it provides an easier way to describe the complex natural surface. In fractal geometry, the image can be reproduced with fractal function. The image intensity of a natural object can be regarded as a kind of fractal [19].

Barnsley first proposed the fractal interpolation by using a certain Iterated Function Systems (IFS) [30]. The method was extended to enable interpolation on 2D and 3D space [31, 32]. Let image I as the plane region, $(x, y) \in I = [x_1, x_N; y_1, y_M]$, z is the intensity value of pixel (x, y) $(x^i, y^j) \in I_{i,j} = (x_i, x_{i+1}; y_j, y_{j+1})$, $z^{i,j}$ is the intensity value of pixel (x^i, y^j), (x, y) and (x_N, y_M) are coordinates of up-left pixel and bottom-right pixel of the original image. Let $\phi_i(x)$ be contractive homeomorphisms [30]: $I \to I_i$:

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

where $0 \leq \mu < 1$.

Let $I_j(y)$ be contractive homeomorphisms: $I \to I_j$:

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

where $0 \leq \lambda < 1$. The most recent and representative fractal interpolation function generated by IFS has the following equation [28]:

$$\begin{cases} \phi_i(x) = a_i x + b_i, \\ \varphi_j(y) = c_j y + d_j, \\ F_{i,j}(x, y, z) = s_{i,j} z + P_{i,j}(\phi_i(x), \varphi_j(y)) - s_{i,j} B_{i,j}(x, y). \end{cases} \tag{1}$$

where

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

where $s_{i,j}$ called vertical scaling factors, $P_{i,j}(\phi_i(x), \varphi_j(y))$ is a polynomial function which represents rotating or translating, and $B_{i,j}(x, y)$ is a perturbation function which act as regulation term to further modify how single part similar to global one.

Remark The interpolation model is identified uniquely by the values of scaling factor $s_{i,j}$. If vertical scaling factor $s_{i,j} = 0$, then the rational fractal function deduces to the bivariate rational interpolation function. If scaling factor $s_{i,j} \neq 0$, then it is a rational fractal function. Details see in reference [28].

For the sake of problem re-formulation, we rewrite the equation (1) for the super-resolution problem as following:

$$\begin{cases} \phi(x_i) = \dfrac{x_{i+1} - x_i}{x_N - x_1} * x_i + \dfrac{x_N x_i - x_1 x_{i+1}}{x_N - x_1} = x^i, \\ \varphi(y_j) = \dfrac{y_{j+1} - y_j}{y_M - y_1} * y_j + \dfrac{y_M y_j - y_1 y_{j+1}}{y_M - y_1} = y^j, \\ \begin{aligned} I_{i,j}(x^i, y^j) = {} & s_{i,j} z + \omega_{0,0}(\theta, \alpha_{i,j}) * \omega_{0,0}(\eta, \beta_{i,j}) * (z_{i,j} - s_{i,j} z_{1,1}) \\ & + \omega_{0,1}(\theta, \alpha_{i,j}) * \omega_{0,0}(\eta, \beta_{i,j}) * (z_{i+1,j} - s_{i,j} z_{N,1}) \\ & + \omega_{0,0}(\theta, \alpha_{i,j}) * \omega_{0,1}(\eta, \beta_{i,j}) * (z_{i,j+1} - s_{i,j} z_{1,M}) \\ & + \omega_{0,1}(\theta, \alpha_{i,j}) * \omega_{0,1}(\eta, \beta_{i,j}) * (z_{i+1,j+1} - s_{i,j} z_{N,M}) \\ & + R(x, y) \end{aligned} \end{cases} \tag{2}$$

where (x, y) is the coordinate of a pixel in the original image, $z = I(x, y)$ is the pixel value of the original image. (x^i, y^j) is the coordinate of pixel

in corresponding high-resolution image, $z^{i,j} = I_{i,j}(x^i, y^j)$ is the pixel value of corresponding high-resolution image. $\omega_{0,0}$ and $\omega_{0,1}$ are base function, and $R(x, y)$ is remainder term, details see in [28].

2.2. Local fractal dimension

The definition of 'fractal' was first proposed by Benoit Mandelbrot because the complex geometry of objects cannot be described by an integral dimension. Fractal geometry can be used to characterize irregular shape or other complex objects that traditional geometry is unable to do. The fractal dimension is the key point to quantify how irregular an object is.

The fractal dimension (FD) measures the complexity of image texture which corresponds to the human perception of image roughness. Several definitions of fractal dimensions have been proposed in recent years, such as Hausdorff-Besicovitch dimension [33]. The most widely used fractal dimension method is the box-counting method [34]. A fractal set A can be divided into number of subset is determined by scale ε, the scale ε and fractal set A must generally follow the power law:

$$D = \lim_{\varepsilon \to 0} \frac{logN(A, \varepsilon)}{log(1/\varepsilon)} \tag{3}$$

In this equation, D is the fractal dimension. $N(A, \varepsilon)$ is the number of subsets.

The fractal dimension has a strong correlation with the vertical scaling factor [35]. As for fractal spline interpolation, the fractal dimension and vertical scaling factor have the following equation:

$$D = 1 + \frac{log \sum_{i=1}^{N} s_i}{logN}; \tag{4}$$

where s_i is the vertical scaling factor, and N is the number of interpolation interval.

In this paper, fractal dimension will be carefully investigated on each image region because every region where demonstrates different complexity. We believe that it should be a local fractal dimension rather than a global uniform global fractal dimension. In the proposed method, the image region will be classified into different categories according to its complexity (reflected by the local fractal dimension). According to the local region characteristics, different interpolation functions are utilized.

3. Proposed Method

In this section, we will discuss how to use rational fractal function to get a high-resolution (HR) image from the low-resolution (LR) image. First, we split the whole image into overlapping patches with a size of 3×3. For every single patch, the local fractal dimension is calculated. Second, we segment these patches into different categories adaptively by analyzing the distribution of local fractal dimension. Third, the rational fractal interpolation function is determined through fractal analysis on each region. Finally, the optimization process is conducted under the assumption that the fractal dimension is invariance in the process of upsampling. Especially, unlike the previous method that one patch has only one vertical scaling factor, every adjacent pixels' vertical scaling factor is determined. The diagram of the proposed method is shown in Fig.1.

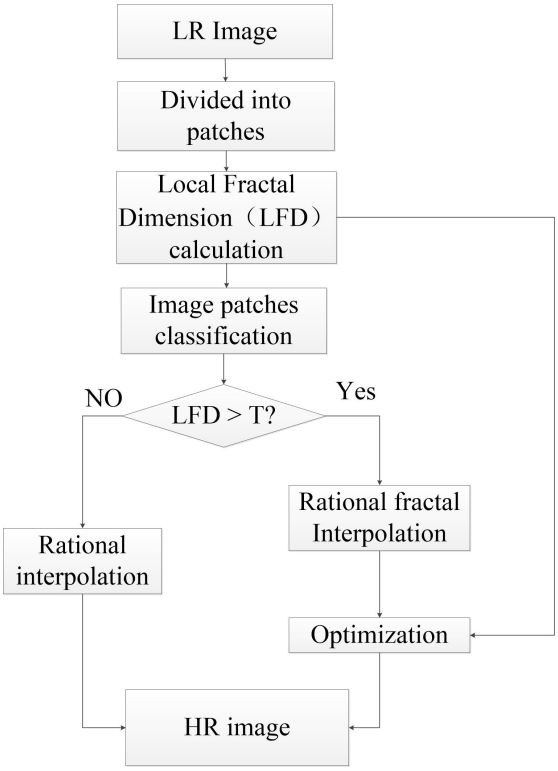
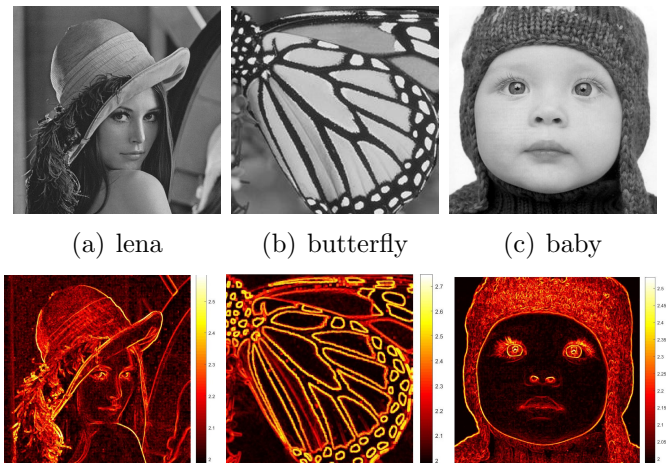


Figure 1: Illustration of super-resolution process

3.1. Image Region Segmentation

In fractal geometry, fractal dimension is a ratio measuring the complexity of image pattern (strictly speaking, a fractal pattern). Here, we segmented patches according to its local fractal dimension. As for the image which contains smooth area and texture area, the fractal dimension in each area is different. Texture area is more complicated than smooth area and usually presents self-similar characteristic. Usually, the local fractal dimension in the texture area is higher than in smooth area. The previous fractal-based method usually treats the whole image as a single fractal set while it usually generates a large error. It is because smooth areas in the image do not present self-similarity in the image. The first step of the proposed method is to divide the whole image into patches of $3 * 3$. For every single patch, the local fractal dimension is calculated by 'box-counting' method. By analyzing the distribution of the local fractal dimension, the Otsu's method [36] is applied for selecting suitable threshold which is used for patch classification. Then, these patches are categorized into two classes: fractal self-similarity patch (texture area) and non-fractal self-similarity patch (smooth area). The result of segmentation is shown in Fig.2.

As shown in Fig.2, the whole image is divided into different area effectively. We regard the area with small local fractal dimension as smooth area and the area with larger local fractal dimension as texture area.



(a) lena (b) butterfly (c) baby

(d) lena (e) butterfly (f) baby

Figure 2: Image regions classified into texture area and smooth area

3.2. Single Image Super-Resolution

According to the result of patch segmentation based on the distribution of local fractal dimension, the different interpolation function is chosen on the basis patches' feature.

In the 'smooth' area, we choose $s_{i,j} = 0, \alpha_{i,j} = 10, \beta_{i,j} = 1$, then the FIF coincides with the bivariate rational interpolation function [29]:

$$\begin{aligned} P_{i,j}(\phi_i(x), \varphi_j(y)) = \sum_{s=0}^{1} \sum_{r=0}^{1} [a_{r,s}(\theta, \eta) z_{i+r,j+s} \\ + b_{r,s}(\theta, \eta) h_i d_{i+r,j+s} \\ + c_{r,s}(\theta, \eta) l_j d_{i+r,j+s}], \end{aligned} \tag{5}$$

In the texture area, we use the rational fractal interpolation as shown in formula (2). The shape parameters are set as: $\alpha_{i,j} = 10, \beta_{i,j} = 1$. As for the vertical scaling factor, the initial value is $(LFD - 2)$. From the equation (2), we can see that the vertical scaling factors have great influence on the final result. So, it is important to select suitable vertical scaling factor for image super-resolution.

3.3. Optimization

Apparently, it is meaningless to choose the value of vertical scaling factor in random. In the fractal geometry, the vertical scaling factors reflect how the local area similar to the global area. However, the existing fractal-based super-resolution methods rarely discuss how the vertical scaling factors should be determined to achieve better super-resolution performance. In [25], Wee and Shin proposed a novel fractal super-resolution algorithm by using a fixed vertical scaling factor in the whole image. In our previous work [37], we proposed a vertical scaling factor calculation method by exploiting the relationship of global fractal dimension and local fractal dimension. This method calculates the vertical scaling factors for each patch. Actually, every two adjacent pixels should have different scaling factors. The more suitable value of $s_{i,j}$, the more accurate of fractal functions we get. In this work, we propose calculating the vertical scaling factor based on the relation between local fractal dimension and vertical scaling factor. Such relationship provides more fine-grained information for calculating the vertical scaling factor for every two adjacent pixels.

To assist local vertical scaling factor calculation on the fine-grained level, intuitively, it is better to use patches which have uniform local fractal feature. Thus, it may calculate a stable and unique local vertical scaling factor.

According to equation (4), we extend it to 2-D surface interpolation. The equation (4) can be rewritten as follows:

$$D = 2 + \frac{log \sum_{i=1}^{M} \sum_{j=1}^{M} s_{i,j}}{logM}; \tag{6}$$

where D is the fractal dimension, $s_{i,j}$ is the vertical scaling factor, and M is the size of input patch minus one.

Based on the assumption that the complexity of the image does not change in the process of super-resolution, so the fractal dimension is fixed. We optimize the vertical scaling factor by using the thought of Particle Swarm Optimization (PSO)[38]. After the initialization of vertical scaling factor, there are only $2 * 2$ variables to estimate for patches with the size of $3 * 3$. By minimizing the difference between the HR patch and input LR patch, we have:

$$min|LFD_{HR} - LFD_{LR}|; \tag{7}$$

According to Eq.(6), Eq.(7) can be rewritten as the following:

$$min|\frac{log \sum_{i=1}^{M} \sum_{j=1}^{M} s_{i,j}}{logM} + 2 - LFD_{LR}|; \tag{8}$$

$s_{i,j}$ updated the value with the following equation:

$$s_{i,j}^{t+1} = s_{i,j}^{t} + \delta * (LFD_t - LFD_{t-1}) * \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \tag{9}$$

where δ is inertia weight. Detail of the method is shown in the scheme of Algorithm 1.

4. Experiments

To demonstrate the effectiveness of the proposed method, the quality and quantity comparison is conducted on different up-sampling factors. During the evaluation, we use two benchmark datasets Set 5 and Set 14. They are often used as the benchmark in other works [39, 40, 41] since it contains some

Algorithm 1 Optimization

1. Given the patch local fractal dimension LFD_{LR} and initial value of vertical scaling factor S.
2. Get the high-resolution patch via rational fractal interpolation function.
3. Calculate local fractal dimension of high-resolution patch LFD_{HR}.
4. If the $|LFD_{HR} - LFD_{LR}| < \varepsilon$.
   This is the final result
   else
   Update vertical scaling factor by $S + \delta * (LFD^{t}_{HR} - LFD^{t-1}_{HR})$.
   Go to 2.
5. Get final result.

challenging images for the current existed methods. We directly downsampled the original image as the low-resolution input image. We compare the proposed ARFSR method with five state-of-the-art methods: reconstruct-based (Tai's [8]), Learning-based (ANR [42], SelfexSR [18], SRCNN [43]), fractal-based (Xu's [26]). Since human eyes are more sensitive intensity than color, we convert the image from RGB color space to YCbCr color space. Like other state-of-the-art methods, the luminance channel is up-sampling by the SR methods. Another channel is upscaled by bicubic. Quantitative evaluations of the SR results are assessed by the Peak Signal-to-Noise Ratio (PSNR), the Structural Similarity (SSIM) index and Feature Similarity (FSIM) Index. Upscaling factors 2, 3 and 4 are considered in this paper.

4.1. Analysis of the Algorithm

As for this algorithm, the fundamental of this method is lie in two aspects: patch classification and invariance of fractal dimension between corresponding LR patch and HR patch. Firstly, we split the whole image into patches, then the local fractal dimension is calculated for every single patch. By analyzing the local fractal dimension, image patches are classified into two different categories. Then, a different fractal function is applied according to image features. To maintain vivid texture details and edge structure, we assume the fractal is invariance in the upsampling process. Finally, vertical scaling factors are calculated under this assumption. Especially, in this paper, every two adjacent pixels' scaling factors are calculated. Unlike other methods only have one vertical scaling factors in one patch. The proposed

Table 1: Comparison of different methods on runtime (s) (Set14)

Image	Tai's	ANR	SelfExSR	SRCNN	Xu's	Proposed
Baboon	0.98	8.98	8.3	177.34	4.9	209.32
Barbara	1.76	15.38	13.99	457.19	8.1	361.83
Bridge	1.03	9.91	8.15	151.87	5.22	231.66
Coastguard	0.40	3.77	1.89	99.44	2.16	231.66
Comic	0.35	3.38	1.96	73.30	1.97	82.22
Face	0.30	2.86	1.50	58.27	1.21	72.7
Flowers	0.73	6.7	5.32	162.99	3.69	170.54
Foreman	0.39	3.74	1.88	86.54	2.07	96.34
Lenna	1.03	9.69	8.01	247.77	4.82	243.06
Man	1.02	9.68	8.08	246.26	5.21	232.85
Monarch	1.54	14.48	13.25	398.53	7.69	368.11
Pepper	0.13	9.65	8.03	229.47	5.23	235.34
Ppt3	1.38	12.79	11.94	387.6	6.55	302.38
Zebra	0.90	8.46	6.98	193.62	4.61	202.87
Average	0.92	8.53	7.09	212.11	4.53	207.07

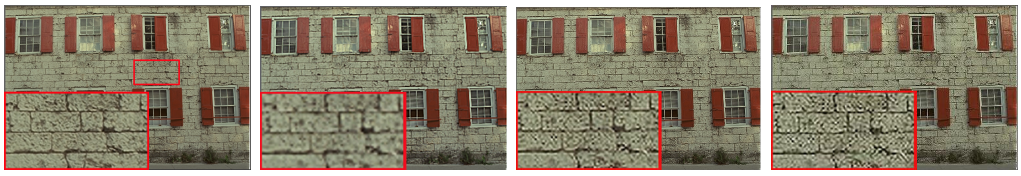
vertical scaling factors calculation method can adaptive local features well. To obtain vertical scaling factors, optimize technique is applied. In this case, we will compare the running time of the proposed method with five state-of-art methods. The results are presented in the following table. As shown in table 1, the runtime of the proposed method is slightly higher than the others. The proposed method is an interpolation-based method, and it should have a lower running time. This situation lies in the following two reasons: the calculation of the fractal dimension and the optimizing vertical scaling factor. The calculation of the fractal dimension is very time-consuming. Since the local fractal calculated for every single image patch, this is the one reason for the longer processing time. The other one is optimizing vertical scaling factors, and we estimate the vertical scaling factors for every two adjacent pixels. However, the runtime of the proposed method is similar to the reconstruction-based method and self-example based method. Our method shows a promising result, especially for larger upscale factor.

As for the upscale factor, the proposed method could enlarge images with a continuous integer ratio. In essence, the proposed method is an interpolation-based method. It can add an arbitrary number of points in two

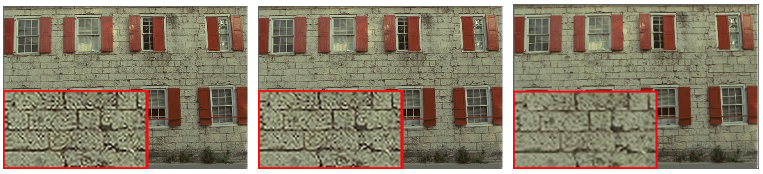
adjacent pixels. Besides, the specified vertical scaling factors could maintain fine texture details under larger upscale factor.

4.2. Comparison with State-of-Art Methods

We compare our method with Tai's [8], ANR [42], SelfexSR [18], SRCNN [43] and Xu's [26] on all the image in Set5 and Set14. As for quality assessment, we conduct experiments on various natural images with texture and edge. In order to demonstrate effectiveness of the proposed method, we upscale image with different scales ($\times 2, \times 3, \times 4$). The Peak Signal to Noise Ratio (PSNR), Structural SIMilarity Index (SSIM) and Feature SIMilarity Index (FSIM) are utilized to evaluate the result in different aspects. As shown in table 2, 3 and 4, our method achieves the best PSNR, SSIM and FSIM measures for the test images. Compared with fractal-based method, the proposed method improves PSNR from 3.7 db to 3.8 db on average.



(a) Original (b) TAI (c) ANR (d) SelfExSR



(e) SRCNN (f) Xu (g) Proposed

Figure 3: Comparison of results ($\times 2$) on Wall image.

We perform tests on natural images to show the improvements of the proposed method over the visual effects. As shown in Fig.3, we test $Wall$ for texture preserving and artificial effects. The proposed algorithm exhibits visually appealing appearance compared to the comparison method. As shown in Fig.3, in the result of SelfExSR, SRCNN, and ANR, the texture is distortion, and artifacts appeared along the edges. The images upscaled by TAI exhibit blurred artifacts. The texture details are not preserved efficiently in the Tai. The fractal-based method also suffers from texture distortion due to failed in capturing suitable vertical scaling factor. Xu's method also uses the

Table 2: Comparison of different methods on PSNRs,SSIMs, FSIMs (Set5)

Methods	Criterion	Scale	Baby	Bird	Butterfly	Head	Woman	Average
	PSNR		30.736	28.172	22.445	28.377	27.495	25.450
Tai's	SSIM	×2	0.937	0.885	0.820	0.729	0.875	0.850
	FSIM		0.954	0.909	0.821	0.885	0.898	0.893
	PSNR		31.917	31.069	23.452	30.493	26.823	22.875
ANR	SSIM	×2	0.951	0.870	0.852	0.754	0.896	0.876
	FSIM		0.971	0.938	0.833	0.885	0.909	0.907
	PSNR		30.564	29.565	21.836	30.081	25.429	27.495
SelfExSR	SSIM	×2	0.926	0.977	0.714	0.723	0.822	0.811
	FSIM		0.962	0.932	0.721	0.878	0.650	0.865
	PSNR		31.376	30.433	22.459	29.783	24.749	27.995
SRCNN	SSIM	×2	0.945	0.922	0.828	0.732	0.949	0.862
	FSIM		0.971	0.926	0.826	0.874	0.901	0.900
	PSNR		34.261	34.614	26.225	29.573	30.301	30.995
Xu's	SSIM	×2	0.981	0.950	0.927	0.829	0.947	0.912
	FSIM		0.988	0.975	0.906	0.921	0.954	0.949
	PSNR		37.896	37.510	29.942	33.647	35.032	**34.805**
Proposed	SSIM	×2	0.984	0.986	0.907	0.837	0.949	**0.929**
	FSIM		0.987	0.971	0.906	0.925	0.954	**0.946**
	PSNR		27.488	24.941	19.015	27.234	23.724	24.480
Tai's	SSIM	×3	0.858	0.774	0.682	0.682	0.771	0.753
	FSIM		0.910	0.849	0.740	0.852	0.834	0.837
	PSNR		27.735	26.084	19.237	27.682	22.488	24.645
ANR	SSIM	×3	0.867	0.812	0.651	0.651	0.778	0.759
	FSIM		0.919	0.865	0.711	0.843	0.839	0.835
	PSNR		26.890	25.147	18.054	27.568	21.406	23.813
SelfExSR	SSIM	×3	0.833	0.729	0.637	0.637	0.680	0.686
	FSIM		0.904	0.824	0.630	0.835	0.785	0.795
	PSNR		27.221	25.304	18.499	26.884	21.591	23.899
SRCNN	SSIM	×3	0.958	0.795	0.621	0.621	0.762	0.740
	FSIM		0.917	0.856	0.708	0.826	0.827	0.826
	PSNR		-	-	-	-	-	-
Xu's	SSIM	×3	-	-	-	-	-	-
	FSIM		-	-	-	-	-	-
	PSNR		37.262	30.918	24.628	30.643	28.601	**29.510**
Proposed	SSIM	×3	0.923	0.871	0.749	0.741	0.835	**0.824**
	FSIM		0.946	0.890	0.771	0.866	0.862	**0.867**
	PSNR		25.339	22.861	17.301	25.552	21.405	22.437
Tai's	SSIM	×4	0.784	0.683	0.583	0.632	0.691	0.674
	FSIM		0.870	0.806	0.685	0.817	0.789	0.793
	PSNR		25.375	23.590	17.240	26.169	20.253	22.525
ANR	SSIM	×4	0.790	0.712	0.579	0.603	0.687	0.674
	FSIM		0.879	0.820	0.645	0.827	0.795	0.739
	PSNR		24.511	22.590	16.013	25.615	18.996	21.545
SelfExSR	SSIM	×4	0.747	0.625	0.443	0.583	0.573	0.594
	FSIM		0.854	0.781	0.578	0.806	0.730	0.749
	PSNR		24.798	22.944	16.391	25.569	19.366	21.813
SRCNN	SSIM	×4	0.783	0.697	0.555	0.589	0.674	0.659
	FSIM		0.876	0.811	0.646	0.816	0.780	0.785
	PSNR		28.290	26.415	19.078	26.524	22.975	24.650
Xu's	SSIM	×4	0.905	0.874	0.742	0.712	0.823	0.811
	FSIM		0.940	0.897	0.769	0.856	0.863	0.865
	PSNR		30.488	29.830	29.432	24.020	28.259	**28.406**
Proposed	SSIM	×4	0.888	0.851	0.724	0.724	0.816	**0.807**
	FSIM		0.925	0.872	0.846	0.754	0.846	**0.849**

(a) Origin (b) TAI (c) ANR (d) SelfExSR

(e) SRCNN (f) XU (g) Proposed

Figure 4: Comparison of results (×3) on Baby image

invariance of fractal dimension between LR image and HR image. However, Xu's method maintains the whole image fractal dimension invariant while our method keeps every local patch fractal dimension unchanged. Therefore, the proposed method is more localized than Xu's. Further, Xu's method uses ICBI algorithm upsampling LR image firstly and then use optimizing technique latterly. The first step's result has huge influence on the final result, and it also limits the upscaled factor. While, the proposed method shows the promising result in texture preserving.

And also, edge structure and detail preservation are tested in Fig.4 and Fig.5. As shown red rectangle in Figure.4 and Figure.5, the images generated by the proposed method in Fig.4 (g) and Fig.5 (g) show an outstanding improvement over the other magnified images. The image obtained by Tai's method present smoothed edges. In the SRCNN, SelfExSR and ANR magnified images, artifacts appear, and the texture twisted heavily. In our method, the magnified images have smooth edges and little noisy artifacts. Based on the above experiment results, our method could reduce ringing, blocking and blurring effectively, and have better results in image details.

Table 3: Comparison of different methods on PSNRs,SSIMs, FSIMs (Set14-1)

Methods	Criterion	Scale	Baboon	Barbara	Bridge	Coastguard	Comic	Face	Flowers	Foreman
	PSNR		22.027	24.769	24.159	24.963	21.721	27.487	24.957	26.247
Tai's	SSIM	×2	0.724	0.833	0.806	0.563	0.635	0.739	0.752	0.878
	FSIM		0.886	0.920	0.906	0.756	0.756	0.866	0.814	0.890
	PSNR		20.948	23.410	23.726	25.054	21.935	30.454	26.066	29.842
ANR	SSIM	×2	0.752	0.850	0.835	0.681	0.759	0.754	0.817	0.910
	FSIM		0.893	0.925	0.924	0.828	0.829	0.885	0.869	0.924
	PSNR		17.916	20.568	18.471	21.267	17.342	27.309	21.048	25.231
SelfExSR	SSIM	×2	0.486	0.668	0.567	0.417	0.441	0.624	0.558	0.688
	FSIM		0.765	0.810	0.796	0.742	0.680	0.835	0.725	0.840
	PSNR		20.070	22.612	18.572	24.216	20.914	29.743	25.151	28.884
SRCNN	SSIM	×2	0.717	0.839	0.528	0.656	0.729	0.732	0.796	0.900
	FSIM		0.890	0.924	0.792	0.818	0.817	0.874	0.859	0.915
	PSNR		20.199	23.435	18.772	26.810	23.448	29.551	27.742	29.168
Xu's	SSIM	×2	0.821	0.888	0.549	0.794	0.843	0.828	0.896	0.945
	FSIM		0.897	0.946	0.802	0.900	0.901	0.921	0.925	0.951
	PSNR		25.845	30.095	25.513	32.336	28.416	33.226	31.817	**33.687**
Proposed	SSIM	×2	0.857	0.916	0.921	0.816	0.854	0.847	0.895	**0.947**
	FSIM		0.922	0.962	0.956	0.892	0.902	0.927	0.926	**0.952**
	PSNR		20.119	22.687	20.652	22.635	18.897	26.990	22.400	25.209
Tai's	SSIM	×3	0.585	0.732	0.670	0.477	0.536	0.678	0.656	0.799
	FSIM		0.825	0.849	0.844	0.748	0.732	0.856	0.783	0.852
	PSNR		19.019	21.356	21.096	22.354	18.513	27.677	22.216	26.288
ANR	SSIM	×3	0.540	0.707	0.653	0.472	0.539	0.651	0.652	0.819
	FSIM		0.795	0.834	0.836	0.763	0.733	0.845	0.780	0.861
	PSNR		17.916	20.568	18.471	21.267	17.342	27.309	21.048	25.231
SelfExSR	SSIM	×3	0.486	0.668	0.567	0.417	0.441	0.624	0.558	0.688
	FSIM		0.765	0.810	0.796	0.742	0.680	0.835	0.725	0.840
	PSNR		18.264	20.621	15.664	21.638	17.618	26.917	21.368	25.251
SRCNN	SSIM	×3	0.511	0.689	0.316	0.446	0.504	0.623	0.623	0.803
	FSIM		0.784	0.827	0.720	0.748	0.717	0.829	0.765	0.845
	PSNR		-	-	-	-	-	-	-	-
Xu's	SSIM	×3	-	-	-	-	-	-	-	-
	FSIM		-	-	-	-	-	-	-	-
	PSNR		23.899	26.510	21.802	27.560	23.946	31.190	27.260	**29.764**
Proposed	SSIM	×3	0.658	0.790	0.752	0.567	0.631	0.745	0.737	**0.859**
	FSIM		0.844	0.888	0.882	0.770	0.773	0.869	0.815	**0.876**
	PSNR		20.476	22.544	21.577	22.490	18.644	25.674	21.715	23.974
Tai's	SSIM	×4	0.487	0.677	0.560	0.427	0.394	0.649	0.578	0.780
	FSIM		0.784	0.813	0.789	0.596	0.628	0.785	0.707	0.818
	PSNR		18.892	21.363	21.139	22.153	18.491	28.361	22.601	26.337
ANR	SSIM	×4	0.519	0.704	0.649	0.457	0.529	0.675	0.659	0.811
	FSIM		0.778	0.827	0.821	0.755	0.733	0.859	0.788	0.856
	PSNR		17.916	20.568	18.471	21.267	17.342	27.309	21.048	25.231
SelfExSR	SSIM	×4	0.486	0.668	0.567	0.417	0.441	0.624	0.558	0.688
	FSIM		0.765	0.810	0.796	0.742	0.680	0.835	0.725	0.840
	PSNR		18.264	20.621	15.664	21.638	17.618	26.917	21.368	25.251
SRCNN	SSIM	×4	0.511	0.689	0.316	0.446	0.504	0.623	0.623	0.803
	FSIM		0.784	0.827	0.720	0.748	0.717	0.829	0.765	0.845
	PSNR		17.097	20.211	15.068	20.825	17.437	26.541	21.494	24.434
Xu's	SSIM	×4	0.553	0.737	0.356	0.474	0.546	0.710	0.675	0.841
	FSIM		0.785	0.840	0.735	0.725	0.744	0.856	0.803	0.864
	PSNR		23.549	26.587	21.802	26.922	23.591	29.292	26.528	**28.808**
Proposed	SSIM	×4	0.591	0.774	0.752	0.525	0.587	0.726	0.710	**0.841**
	FSIM		0.829	0.885	0.882	0.729	0.749	0.843	0.792	**0.868**

Table 4: Comparison of different methods on PSNRs,SSIMs, FSIMs (Set14-2)

Methods	Criterion	Scale	Lenna	Man	Monarch	Pepper	Ppt3	Zebra	Average
	PSNR		28.628	24.912	27.435	28.183	21.186	24.151	25.059
Tai's	SSIM	×2	0.919	0.845	0.951	0.934	0.902	0.851	0.809
	FSIM		0.948	0.920	0.947	0.951	0.875	0.890	0.877
	PSNR		29.540	25.078	28.615	30.240	23.018	25.360	25.949
ANR	SSIM	×2	0.909	0.866	0.959	0.922	0.943	0.908	0.847
	FSIM		0.954	0.932	0.965	0.959	0.927	0.927	0.910
	PSNR		28.613	23.710	27.047	28.237	22.151	24.153	24.500
SelfExSR	SSIM	×2	0.898	0.823	0.940	0.909	0.917	0.865	0.797
	FSIM		0.947	0.919	0.947	0.947	0.910	0.918	0.887
	PSNR		28.837	25.100	27.701	29.543	22.082	24.326	24.839
SRCNN	SSIM	×2	0.898	0.845	0.953	0.913	0.932	0.895	0.809
	FSIM		0.954	0.929	0.964	0.957	0.925	0.924	0.896
	PSNR		30.908	26.272	31.524	29.321	26.045	28.834	26.573
Xu's	SSIM	×2	0.957	0.926	0.986	0.951	0.967	0.967	0.880
	FSIM		0.974	0.953	0.985	0.968	0.965	0.974	0.933
	PSNR		36.469	31.722	35.649	33.448	29.942	33.252	**31.531**
Proposed	SSIM	×2	0.968	0.940	0.987	0.965	0.977	0.967	**0.918**
	FSIM		0.981	0.964	0.986	0.977	0.964	0.975	**0.949**
	PSNR		24.892	22.667	24.502	25.587	18.679	20.472	22.682
Tai's	SSIM	×3	0.830	0.726	0.899	0.854	0.806	0.713	0.711
	FSIM		0.882	0.850	0.886	0.886	0.788	0.825	0.828
	PSNR		25.219	22.148	24.243	26.471	19.157	20.747	22.636
ANR	SSIM	×3	0.814	0.721	0.897	0.849	0.834	0.743	0.706
	FSIM		0.892	0.849	0.903	0.900	0.826	0.831	0.832
	PSNR		24.750	20.582	23.094	24.558	18.223	19.599	24.425
SelfExSR	SSIM	×3	0.694	0.508	0.862	0.826	0.785	0.679	0.628
	FSIM		0.833	0.754	0.866	0.879	0.786	0.811	0.794
	PSNR		24.892	21.296	23.547	25.736	18.248	19.757	21.486
SRCNN	SSIM	×3	0.798	0.697	0.888	0.835	0.818	0.720	0.662
	FSIM		0.888	0.843	0.902	0.895	0.820	0.823	0.814
	PSNR		-	-	-	-	-	-	-
Xu's	SSIM	×3	-	-	-	-	-	-	-
	FSIM		-	-	-	-	-	-	-
	PSNR		32.292	27.717	29.774	31.345	24.893	36.333	**27.464**
Proposed	SSIM	×3	0.896	0.815	0.935	0.915	0.888	0.979	**0.734**
	FSIM		0.932	0.897	0.934	0.931	0.876	0.980	**0.808**
	PSNR		24.961	22.188	23.446	24.415	18.076	19.818	22.143
Tai's	SSIM	×4	0.787	0.647	0.841	0.833	0.729	0.576	0.641
	FSIM		0.861	0.808	0.852	0.864	0.708	0.734	0.768
	PSNR		26.787	22.625	24.796	27.492	19.909	21.769	23.051
ANR	SSIM	×4	0.832	0.729	0.902	0.856	0.844	0.639	0.700
	FSIM		0.902	0.850	0.897	0.902	0.817	0.797	0.827
	PSNR		22.673	19.417	21.146	22.309	16.562	17.104	19.612
SelfExSR	SSIM	×4	0.693	0.531	0.784	0.737	0.665	0.516	0.544
	FSIM		0.822	0.752	0.806	0.817	0.709	0.733	0.745
	PSNR		22.937	19.949	21.489	23.551	16.542	17.271	19.805
SRCNN	SSIM	×4	0.725	0.595	0.823	0.774	0.723	0.446	0.566
	FSIM		0.845	0.791	0.850	0.850	0.751	0.696	0.766
	PSNR		25.856	21.766	24.456	25.034	19.577	20.890	21.478
Xu's	SSIM	×4	0.867	0.768	0.925	0.883	0.851	0.782	0.712
	FSIM		0.918	0.868	0.918	0.908	0.855	0.877	0.835
	PSNR		31.256	27.629	29.165	29.254	24.659	26.387	**26.814**
Proposed	SSIM	×4	0.886	0.797	0.925	0.910	0.871	0.795	**0.764**
	FSIM		0.931	0.889	0.929	0.922	0.855	0.873	**0.855**

(a) Original (b) TAI (c) ANR (d) SelfExSR

(e) SRCNN (f) Xu (g) Proposed

Figure 5: Comparison of results (×4) on Head image

5. Conclusions

This work is a contribution to image super-resolution based on fractal interpolation. At first, we reformulate fractal iterated function for image super-resolution problem. The constructed fractal iterated function has different forms of expression with varied values of scaling factor and shape parameters, which coincides with the diversity of image features, and it can describe complex geometric structure of image accurately. In an image large smooth area and texture area contained, we split the image into local regions which have uniform local fractal feature to select suitable interpolation function. In the texture, we select fractal iterated function which contains vertical scaling factors. In order to get 'photo-realistic' texture details, vertical scaling factors calculation method is proposed. This method under the assumption the local fractal dimension keeps invariance in the process of upsampling. Experimental results on test images demonstrate that the proposed method achieved very competitive performance not only subjective but objective.

References

[1] K. T. Gribbon, D. G. Bailey, A novel approach to real-time bilinear interpolation, Proceedings of the IEEE International Workshop on Electronic Design, Test and Applications (2004) 126–131.

[2] C. F. Zhang CM, Zhang X, Cubic spline surface fitting to image with edges as constraints, IEEE International Conference on Image Processing (2013) 1046–1050.

[3] M. R. Matsumoto S, Kamada M, Adaptive image interpolation by cardinal splines in piecewise constant tension, Optimization Letters 6 (7) (2012) 1265–1280.

[4] X. Li, M. T. Orchard, New edge-directed interpolation, IEEE transactions on image processing 10 (10) (2001) 1521–1527.

[5] A. Giachetti, N. Asuni, Real-time artifact-free image upscaling, IEEE Transactions on Image Processing 20 (10) (2011) 2760–2768.

[6] J. Sun, Z. Xu, H. Shum, Gradient profile prior and its applications in image super-resolution and enhancement, IEEE Transactions on Image Processing 20 (6) (2011) 1529–1542.

[7] R. Fattal, Image upsampling via imposed edge statistics, ACM transactions on graphics 26 (3) (2007) 95–8.

[8] Y. Tai, S. Liu, M. S. Brown, S. Lin, Super resolution using edge prior and single image detail synthesis, in: IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 2400–2407.

[9] H. Zhang, J. Yang, Y. Zhang, T. S. Huang, Image and video restorations via nonlocal kernel regression, IEEE Transactions on cybernetics 43 (3) (2013) 1035–1046.

[10] L. C. Pickup, S. J. Roberts, A. Zisserman, A sampled texture prior for image super-resolution, Advances in neural information processing systems (2004) 1587–1594.

[11] Q. Wang, X. Tang, H. Shum, Patch based blind image super resolution, in: IEEE International Conference on Computer Vision, Vol. 1, 2005, pp. 709–716.

[12] H. Chang, D.-Y. Yeung, Y. Xiong, Super-resolution through neighbor embedding, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 1, 2004, pp. I–I.

[13] J. Yang, J. Wright, T. Huang, Y. Ma, Image super-resolution via sparse representation, IEEE transactions on image processing 19 (11) (2010) 2861–2873.

[14] K. Zhang, X. Gao, D. Tao, X. Li, Single image super-resolution with non-local means and steering kernel regression, IEEE Transactions on Image Processing 21 (11) (2012) 4544–4556.

[15] H. He, W. Siu, Single image super-resolution using gaussian process regression, IEEE Conference on Computer Vision and Pattern Recognition (2011) 449–456.

[16] Y. Tian, F. Zhou, W. Yang, X. Shang, Q. Liao, Anchored neighborhood regression based single image super-resolution from self-examples, Proceedings of the International Conference on Image Processing (2016) 2827–2831.

[17] X. Wei, P. L. Dragotti, Freshfri-based single-image super-resolution algorithm, IEEE Transactions on Image Processing 25 (8) (2016) 3723–3735.

[18] J.-B. Huang, A. Singh, N. Ahuja, Single image super-resolution from transformed self-exemplars, in: IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 5197–5206.

[19] A. P. Pentland, Fractal-based description of natural scenes, IEEE transactions on pattern analysis and machine intelligence PAMI-6 (6) (1984) 661–674.

[20] A. E. Jacquin, Image coding based on a fractal theory of iterated contractive image transformations, IEEE Transactions on image processing 1 (1) (1992) 18–30.

[21] G. Tu, C. Zhang, J. Wu, X. Liu, Remote sensing image processing using wavelet fractal interpolation, in: Proceedings International Conference on Communications, Circuits and Systems, Vol. 2, 2005, pp. 701–706.

[22] D. D. Giusto, M. Murroni, G. Soro, Slow motion replay of video sequences using fractal zooming, IEEE Transactions on Consumer Electronics 51 (1) (2005) 103–111.

[23] Z. Han, T. Denney, Interpolation of 2-d fractional brownian motion using first order increments, in: Proceedings International Conference onImage Processing, 1998, pp. 222–226.

[24] Z. Han, T. Denney, Incremental fourier interpolation of 2-d fractional brownian motion, IEEE Transactions on Industrial Electronics 48 (5) (2001) 920–925.

[25] Y. C. Wee, H. J. Shin, A novel fast fractal super resolution technique, IEEE Transactions on Consumer Electronics 56 (3) (2010) 1537–1541.

[26] H. Xu, G. Zhai, X. Yang, Single image super-resolution with detail enhancement based on local fractal analysis of gradient, IEEE Transactions on circuits and systems for video technology 23 (10) (2013) 1740–1754.

[27] L. Yu, Y. Xu, H. Xu, X. Yang, Self-example based super-resolution with fractal-based gradient enhancement, IEEE International Conference on Multimedia and Expo Workshops (2013) 1–6.

[28] F. Bao, X. Yao, Q. Sun, Y. Zhang, C. Zhang, Smooth fractal surfaces derived from bicubic rational fractal interpolation functions, Science China Information Sciences 61 (9) (2018) 099104.

[29] Y. Liu, Y. Zhang, Q. Guo, C. Zhang, Image interpolation based on weighted and blended rational function, in: Asian Conference on Computer Vision, 2014, pp. 78–88.

[30] M. F. Barnsley, Fractal functions and interpolation, Constructive approximation 2 (1) (1986) 303–329.

[31] C. M. Wittenbrink, Ifs fractal interpolation for 2d and 3d visualization, in: Proceedings of the 6th conference on Visualization, 1995, p. 77.

[32] H. Xie, H. Sun, The study on bivariate fractal interpolation functions and creation of fractal interpolated surfaces, Fractals 5 (04) (1997) 625–634.

[33] H. P. McKean Jr, et al., Hausdorff-besicovitch dimension of brownian motion paths, Duke Mathematical Journal 22 (2) (1955) 229–234.

[34] N. Sarkar, B. Chaudhuri, An efficient differential box-counting approach to compute fractal dimension of image, IEEE Transactions on systems, man, and cybernetics 24 (1) (1994) 115–120.

[35] H. Millan, M. González-Posada, A. Morilla, E. Pérez, Self-similar organization of vertisol microstructure: A pore–solid fractal interpretation, Geoderma 138 (3-4) (2007) 185–190.

[36] N. Otsu, A threshold selection method from gray-level histograms, IEEE transactions on systems, man, and cybernetics 9 (1) (1979) 62–66.

[37] X. Yao, Y. Zhang, F. Bao, Y. Liu, C. Zhang, The blending interpolation algorithm based on image features, Multimedia Tools and Applications 77 (2) (2018) 1971–1995.

[38] Y. Shi, et al., Particle swarm optimization: developments, applications and resources, in: evolutionary computation, 2001. Proceedings of the 2001 Congress on, Vol. 1, 2001, pp. 81–86.

[39] J. Huang, T. Liu, P. L. Dragotti, T. Stathaki, Srhrf+: Self-example enhanced single image super-resolution using hierarchical random forests., in: CVPR Workshops, 2017, pp. 1067–1075.

[40] H. Ren, M. Elkhamy, J. Lee, Image super resolution based on fusing multiple convolution neural networks, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), IEEE, 2017, pp. 1050–1057.

[41] Z. Wang, D. Liu, J. Yang, W. Han, T. Huang, Deep networks for image super-resolution with sparse prior, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 370–378.

[42] R. Timofte, V. De Smet, L. Van Gool, Anchored neighborhood regression for fast example-based super-resolution, in: Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 1920–1927.

[43] C. Dong, C. C. Loy, K. He, X. Tang, Learning a deep convolutional network for image super-resolution, in: European conference on computer vision, 2014, pp. 184–199.