Hindawi
Discrete Dynamics in Nature and Society
Volume 2018, Article ID 9801361, 16 pages
https://doi.org/10.1155/2018/9801361

Research Article

An Adaptive Image Inpainting Method Based on Continued Fractions Interpolation

Lei He,[1] **Yan Xing**,[1] **Kangxiong Xia**,[2] **and Jieqing Tan**[1]

[1]*School of Mathematics, Hefei University of Technology, Hefei 230009, China*
[2]*Beijing Institute of Technology, Beijing 100081, China*

Correspondence should be addressed to Yan Xing; xy1128@126.com, Kangxiong Xia; 2917111511@qq.com, and Jieqing Tan; jieqingtan@hfut.edu.cn

Received 25 January 2018; Accepted 11 March 2018; Published 26 June 2018

Academic Editor: Francisco R. Villatoro

In view of the drawback of most image inpainting algorithms by which texture was not prominent, an adaptive inpainting algorithm based on continued fractions was proposed in this paper. In order to restore every damaged point, the information of known pixel points around the damaged point was used to interpolate the intensity of the damaged point. The proposed method included two steps; firstly, Thiele's rational interpolation combined with the mask image was used to interpolate adaptively the intensities of damaged points to get an initial repaired image, and then Newton-Thiele's rational interpolation was used to refine the initial repaired image to get a final result. In order to show the superiority of the proposed algorithm, plenty of experiments were tested on damaged images. Subjective evaluation and objective evaluation were used to evaluate the quality of repaired images, and the objective evaluation was comparison of Peak Signal to Noise Ratios (PSNRs). The experimental results showed that the proposed algorithm had better visual effect and higher Peak Signal to Noise Ratio compared with the state-of-the-art methods.

1. Introduction

Image inpainting is an important branch of image processing which studies how to restore the damaged part based on human visual mechanism. Currently, there are many image inpainting methods, including methods based on partial differential equation (PDE) [1–4], methods based on texture synthesis [5, 6], methods based on sparse representation [7–10], and other methods [11–15].

PDE is introduced to propagate local structures from the exterior to the interior of the damaged regions. Zhou and Gao [16] studied the pricing problem of the American options with fractal transmission system under two-state regime switching models, which could be formulated as a free boundary problem of time-fractional PDE system. Su et al. [17] solved the problem of the partial differential equations by using local polynomial regression. Bertalmio et al. [1] proposed an image inpainting method based on PDE for the first time in the conference of SIGGRAPH, where the damaged parts were repaired by the iteration along the direction of isophotes of surrounding pixel points. And then Bertalmio et al. [1] proposed an image inpainting model based on high-order partial differential equations of transmission theory. Chan and Shen [2, 3] proposed an image inpainting model based on total variation and a new model based on curvature diffusion. Pascal et al. [4] proposed a new compression framework with homogeneous, biharmonic, and edge-enhancing diffusion which supported different strategies for data selection and storage and gave a detailed analysis of the advantages and disadvantages of the three partial differential equations (PDEs). These methods based on PDE are suitable for straight lines, curves, and small regions, and they are not suitable for texture details of large areas.

The texture synthesis scheme is another effective method for image inpainting [5, 6]. Efros and Leung [5] proposed a nonparametric sampling method based on single pixel synthesis, and for the first time the Markov Random Field model was adopted in texture synthesis. Criminisi et al. [6] proposed an image inpainting method based on texture

synthesis by using priority match. The damaged images can be repaired well by using these texture synthesis methods; nevertheless, the problem of boundaries blur still exists.

With the advent of sparse representation, sparse priors have also been used to solve the inpainting problem. Hu and Xiong [7] proposed an image inpainting scheme by combining the Criminisi algorithm with sparse representation, where the sparse representation inpainting method was used to replace the best matching patch search in Criminisi et al.'s algorithm. Shen et al. [8] proposed an example-based image inpainting method by using the sparse representation based patch inpainting model and isophote-based priority. A patch based image inpainting method for multiview images was proposed [9], which was executed in two phases. Rao et al. [10] presented the concept of group based sparse representation instead of traditional patch based sparse representation, which was used as the basic unit. The sparse representation based methods are more suitable than the PDE-based and texture synthesis methods for large texture areas.

Huo et al. [11] proposed an automatic video scratch removal method based on Thiele type continued fraction, which adopted Thiele's rational interpolation to interpolate the intensity of every damaged pixel point. Bornemann and Marz [12] proposed a fast noniterative method for image inpainting, which traversed the inpainting domain by the fast marching method just once while transporting, along the way, image values in a coherence direction robustly estimated by means of the structure tensor. The distribution of training data was learned to predict missing content in a damaged image [13]. Pathak et al. [14] presented an unsupervised visual feature learning algorithm driven by context-based pixel prediction, and by the context encoder, the appearance and the semantics of visual structures were captured. In [15], an image inpainting algorithm based on Kriging interpolation technique was proposed, where the Kriging interpolation technique automatically could fill the damaged region and scratched regions.

These methods above have good repaired effect, however, by which texture details cannot be well processed. Considering that the reconstruction images have better visual effect and prominent texture by the continued fractions [11, 18–20], in this paper, we propose an adaptive image inpainting method based on continued fractions.

The main contributions of this paper are as follows: the adaptive inpainting scheme based on Thiele's rational interpolation is proposed; the novel inpainting model by Thiele's rational interpolation combined with Newton-Thiele's rational interpolation is proposed.

2. Overview of the Proposed Method

In this section, we mainly summarize the proposed method. Our method consists of two phases, namely, adaptive inpainting phase and refining inpainting phase. In the adaptive inpainting phase, we adopt Thiele's rational interpolation function and the corresponding mask image to interpolate the intensities of damaged pixel points. According to the mask image, we can judge the overall direction of scratches. If the overall direction of scratches is horizontal, then the damaged pixel points will be processed in columns; otherwise, they are processed in rows. For example, the overall direction of scratches is vertical in Figure 1, and the intensities of damaged pixel points will be interpolated in rows. We scan the mask image line by line and find the positions of damaged pixel points, and, for every damaged pixel point, we adopt the information of known pixel points near it and Thiele's rational interpolation function to interpolate its intensity. In order to get correct intensity, we adaptively select interpolation sampling points that are closer to the damaged point. That is, the known pixel points are selected as interpolation sampling points by bilateral symmetry way centering on the damaged point. When all damaged pixel points are processed, we can get an initial repaired image.

In the refining inpainting phase, we adopt Newton-Thiele's rational interpolation function to update the intensity of every damaged point. After the first phase, the damaged points have been repaired, and, in the second phase, we will refine the previous result in order to be closer to the original image. We use the mask image again to find the position of every damaged point corresponding to the initial repaired image. Different from previous continued fractions inpainting method [11], except for the damaged point being interpolated, the other damaged points are used as the known pixel points and the intensities are those of the corresponding points in the initial repaired image. As shown in Figure 1, by using Newton-Thiele's rational interpolation function and information of 16 known pixel points around the damaged point being interpolated, the intensity of the damaged point can be interpolated and then the intensity of this point in initial repaired image will be replaced by it. Finally, we can get final repaired result.

3. Adaptive Inpainting via Thiele's Rational Interpolation

3.1. Thiele's Interpolating Continued Fractions. The continued fractions are an ancient branch of mathematics, and the theory appeared for the first time in 1948 [21]. The most useful form of continued fractions is the Thiele type continued fractions which can be defined as follows.

Suppose a set of real or complex points $X = \{x_0, x_1, \ldots, x_m\}$ a function $f(x)$ defined in this domain $D \supset X$, where $x_0, x_1, \ldots, x_m$ don't have to be distinct from one another. One basic approach to approximate function $f(x)$ is Thiele's interpolating continued fraction which can be expressed as [22–25]

$$T_m(x) = \phi[x_0] + \frac{x - x_0}{\phi[x_0, x_1]} + \frac{x - x_1}{\phi[x_0, x_1, x_2]} + \cdots + \frac{x - x_{m-1}}{\phi[x_0, x_1, \ldots, x_m]}, \quad (1)$$

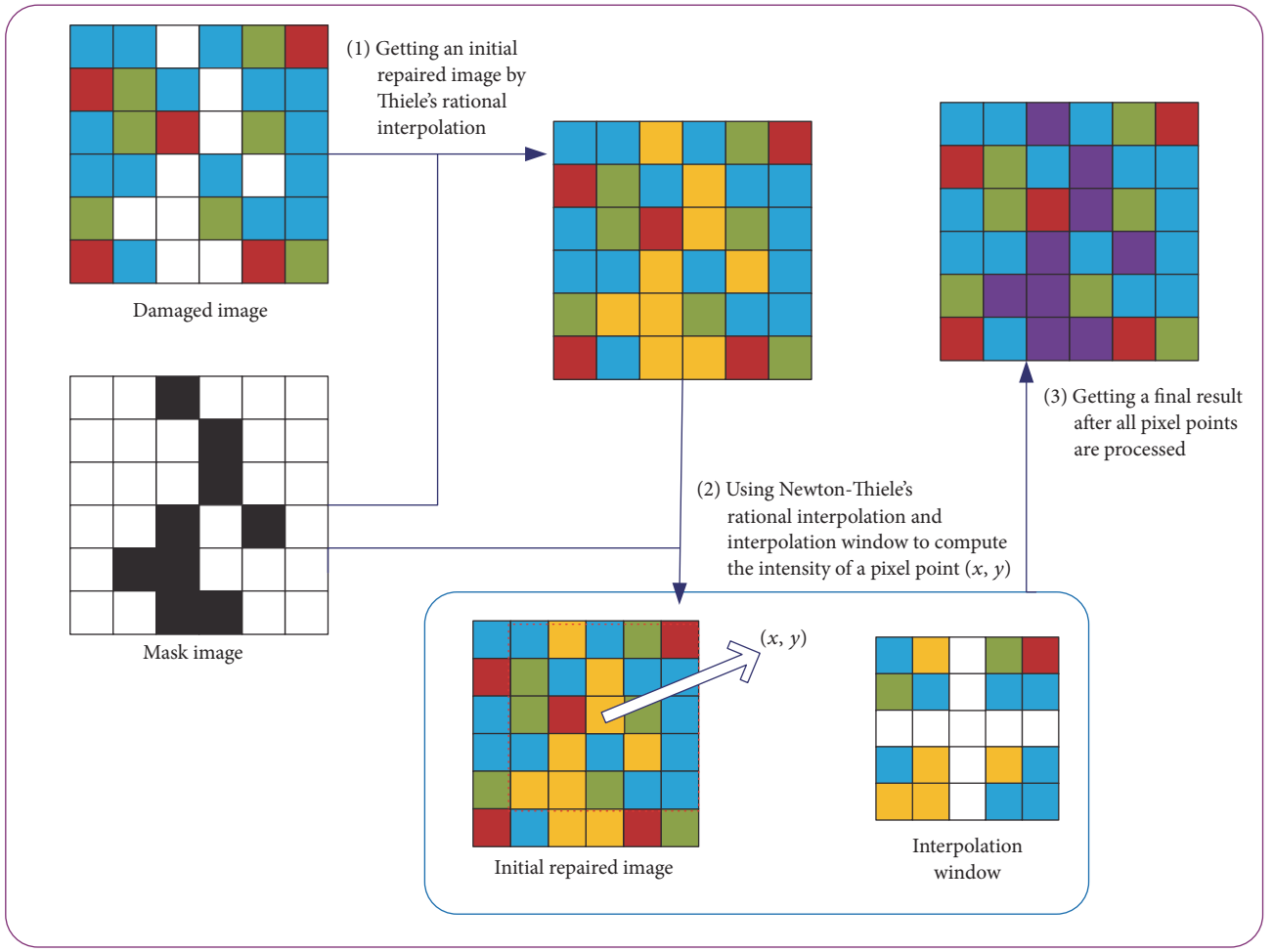


Figure 1: Flowchart of the proposed method. For a damaged image, we use Thiele's rational interpolation function and the mask image to get an initial repaired image. And then, by using Newton-Thiele's rational interpolation function, interpolation window, and the mask image, the initial repaired image is refined to get a final result.

where $\phi[x_0, x_1, \ldots, x_i]$ $(i = 0, \ldots, m)$ are the inverse differences of the function $f(x)$ at points $x_0, x_1, \ldots, x_i$, which can be defined as

$$\begin{aligned} &\phi\left[x_i\right] = f\left(x_i\right), \quad i = 0, 1, 2, \ldots, m, \\ &\phi\left[x_p, x_q\right] = \frac{x_q - x_p}{\phi\left[x_q\right] - \phi\left[x_p\right]}, \\ &\phi\left[x_i, \ldots, x_j, x_k, x_l\right] \\ &\quad = \frac{x_l - x_k}{\phi\left[x_i, \ldots, x_j, x_l\right] - \phi\left[x_i, \ldots, x_j, x_k\right]}. \end{aligned} \tag{2}$$

It is not difficult to show that $T_m(x)$ is a rational function with the degrees of numerator polynomial and denominator polynomial not exceeding $[(m+1)/2]$ and $[m/2]$, respectively, satisfying $T_m(x_i) = f(x_i)$, $i = 0, 1, \ldots, m$.

3.2. Newton-Type Approximation of Thiele Type Continued Fractions. From the above section, we find that when the denominator is zero and the inverse differences b_i cannot be calculated, Thiele's interpolating continued fraction $T_m(x)$ should be replaced by the following Newton-type polynomial:

$$\begin{aligned} T_m(x) = {} & p\left(x_0\right) + p\left(x_0, x_1\right)\left(x - x_0\right) + \cdots \\ & + p\left(x_0, x_1, \ldots, x_m\right)\left(x - x_0\right)\left(x - x_1\right) \\ & \cdots\left(x - x_{m-1}\right), \end{aligned} \tag{3}$$

where

$$\begin{aligned} &p\left(x_i\right) = f\left(x_i\right), \\ &p\left(x_i, x_j\right) = \frac{p\left(x_j\right) - p\left(x_i\right)}{x_j - x_i}, \\ &p\left(x_0, \ldots, x_{i-2}, x_{i-1}, x_i\right) \\ &\quad = \frac{p\left(x_0, \ldots, x_{i-2}, x_i\right) - p\left(x_0, \ldots, x_{i-2}, x_{i-1}\right)}{x_i - x_{i-1}}. \end{aligned} \tag{4}$$

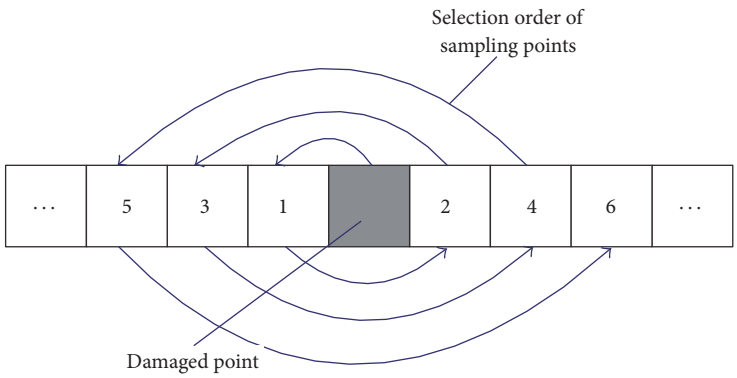


Figure 2: Selection of sampling points.

3.3. Selection of Interpolation Sampling Points. For image inpainting by Thiele's rational interpolation, the key is the selection of interpolation sampling points. Considering the computational complexity of continued fractions, we adaptively select 4 sampling points for interpolation. These sampling points are all known pixel points that are close enough to the damaged point being interpolated. The selection order of sampling points is as shown in Figure 2, where the black solid point is damaged pixel point and the numbers are the selection order of interpolation sampling points. The sampling points are selected by the way of bilateral symmetry, and the selection is finished when the number of sampling points is 4. Sometimes there are several damaged pixel points in one row (column); when we select interpolation sampling points of one damaged pixel point, we skip the other damaged pixel points.

3.4. Inpainting Algorithm by Thiele's Rational Interpolation. The proposed inpainting algorithm by Thiele's rational interpolation can work only when there is consistent scratching direction in damaged domain. As shown in Figure 3, it is a scratching simulated diagram, where the black solid points are the damaged pixel points. From Figure 3, we find that the scratching overall direction is vertical, so we select interpolation sampling points in the horizontal direction, which can be done by using the method of the previous section. From left to right, we scan every pixel point of one row. If the pixel point is the damaged point, we use the information of interpolation sampling points around it and Thiele's rational interpolation function to get its intensity.

Now we describe the adaptive inpainting algorithm in detail. From Figure 3, we select some pixel points in the dotted box, which are displayed in Figure 4. In Figure 4, the black solid points are damaged points and the dot points are the damaged points being interpolated, and the numbers are the selection order of interpolation sampling points. In order to better describe the selection order of interpolation sampling points for every damaged point, for continuous damaged points in the same row, we display them, respectively, in different rows. They have different selection orders of interpolation sampling points for those damaged points near the damaged point being interpolated. In general, the selection rule is principle of proximity and the interpolation sampling points are selected by the way of bilateral symmetry.

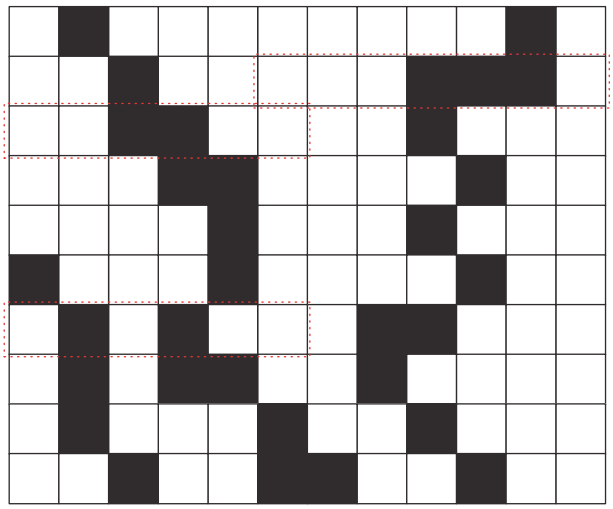


Figure 3: A scratching simulated diagram.

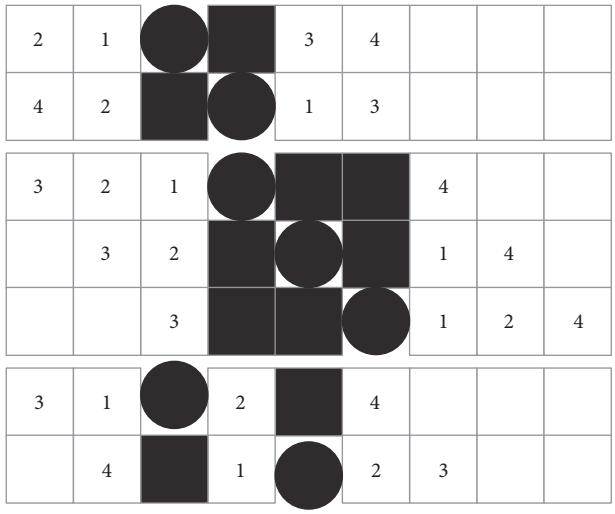


Figure 4: Inpainting schematic diagram.

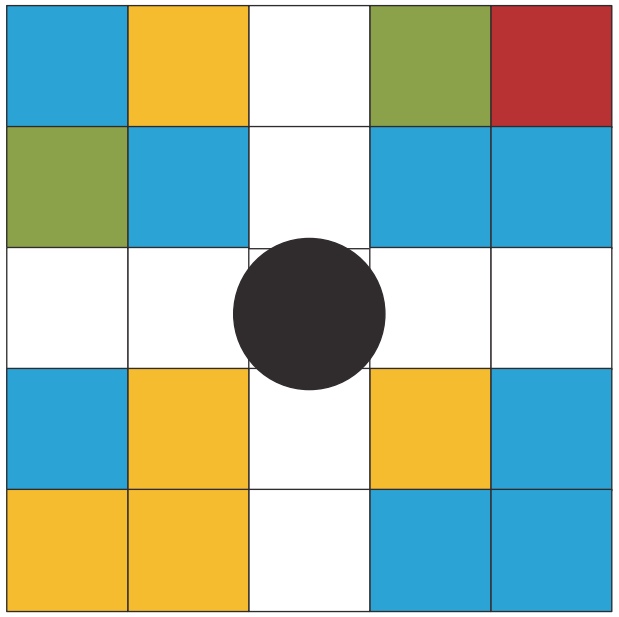


Figure 5: 4 ∗ 4 interpolation window.

If one pixel point is a damaged pixel point in searching process, then we skip this pixel point and continue to search for the next pixel point. After 4 interpolation sampling points are all selected, we use the information of these pixel points and Thiele's rational interpolation function to get the intensity of the damaged point. When the intensities of all damaged points are got, the initial repaired image can be got.

4. Refining Inpainting via Newton-Thiele's Rational Interpolation

4.1. Newton-Thiele's Rational Interpolation. Newton-Thiele's rational interpolation is formed jointly by Newton's polynomial in x and Thiele's continued fraction in y, which can be defined as follows [26, 27].

Let

$$\prod_{x}^{m} = \{x_i \mid i = 0, 1, \ldots, m\} \subset (a, b),$$
$$\prod_{y}^{n} = \{y_j \mid j = 0, 1, \ldots, n\} \subset (c, d), \tag{5}$$

where a, b, c, d, m, and n are natural numbers.

$$\prod_{x,y}^{m,n} = \prod_{x}^{m} \times \prod_{y}^{n} = \left\{\left(x_i, y_j\right) \mid i = 0, 1, \ldots, m;\ j = 0, 1, \ldots, n\right\}, \tag{6}$$

$$R_{m,n}^{NT}(x, y) = A_0(y) + (x - x_0) A_1(y) + \cdots + (x - x_0)(x - x_1) \cdots (x - x_{m-1}) A_m(y), \tag{7}$$

where $A_i(y)$, $i = 0, 1, \ldots, m$, are Thiele's interpolants based on continued fractions defined as follows:

$$A_i(y) = \phi_{NT}[x_0, \ldots, x_i; y_0] + \frac{y - y_0}{\phi_{NT}[x_0, \ldots, x_i; y_0, y_1]} + \frac{y - y_1}{\phi_{NT}[x_0, \ldots, x_i; y_0, y_1, y_2]} + \cdots + \frac{y - y_{n-1}}{\phi_{NT}[x_0, \ldots, x_i; y_0, y_1, \ldots, y_n]}, \tag{8}$$

where $\phi_{NT}[x_0, \ldots, x_i; y_0, \ldots, y_j]$ ($i = 0, 1, \ldots, m$; $j = 0, 1, \ldots, n$) are the blending differences, which can be calculated recursively as follows:

$$\phi_{NT}\left[x_i; y_j\right] = f\left(x_i, y_j\right), \quad \forall \left(x_i, y_j\right) \in \prod_{x,y}^{m,n},$$

$$\phi_{NT}\left[x_i, x_j; y_k\right] = \frac{\phi_{NT}\left[x_j; y_k\right] - \phi_{NT}\left[x_i; y_k\right]}{x_j - x_i},$$

$$\phi_{NT}\left[x_p, \ldots, x_q, x_i, x_j; y_k\right] = \frac{\phi_{NT}\left[x_p, \ldots, x_q, x_j; y_k\right] - \phi_{NT}\left[x_p, \ldots, x_q, x_i; y_k\right]}{x_j - x_i},$$

$$\phi_{NT}\left[x_p, \ldots, x_q; y_k, y_l\right] = \frac{y_l - y_k}{\phi_{NT}\left[x_p, \ldots, x_q; y_l\right] - \phi_{NT}\left[x_p, \ldots, x_q; y_k\right]},$$

$$\phi_{NT}\left[x_p,\ldots,x_q;y_r,\ldots,y_s,y_k,y_l\right]=\frac{y_l-y_k}{\phi_{NT}\left[x_p,\ldots,x_q;y_r,\ldots,y_s,y_l\right]-\phi_{NT}\left[x_p,\ldots,x_q;y_r,\ldots,y_s,y_k\right]}. \tag{9}$$

Then it is not difficult to show that $R_{m,n}^{NT}(x,y)$ determined by (7) and (8) satisfies

$$R_{m,n}^{NT}\left(x_i,y_j\right)=f\left(x_i,y_j\right),\quad i=0,1,\ldots,m;\ \ j=0,1,\ldots,n. \tag{10}$$

4.2. Newton-Type Approximation Formula of Newton-Thiele's Rational Interpolation. We find that if $\phi_{NT}[x_p,\ldots,x_q;y_r,\ldots,y_s,y_l]=\phi_{NT}[x_p,\ldots,x_q;y_r,\ldots,y_s,y_k]$ from the above equation, the denominator is zero, and the equation cannot be calculated. Similar to the approximation method of Thiele type continued fractions in Section 3.2, $A_i(y)$ defined in (8) should be replaced by the following Newton-type polynomial:

$$\begin{aligned}A_i(y)&=p^*\left(x_0,\ldots,x_i;y_0\right)\\&\quad+p^*\left(x_0,\ldots,x_i;y_0,y_1\right)\left(y-y_0\right)+\cdots\\&\quad+p^*\left(x_0,\ldots,x_i;y_0,\ldots,y_j\right)\prod_{j=0}^{n-1}\left(y-y_j\right),\\&\qquad\qquad i=0,\ldots,m,\end{aligned} \tag{11}$$

where

$$\begin{aligned}p^*\left(x_0,\ldots,x_i;y_0,\ldots,y_j\right)&=\frac{p^*\left(x_0,\ldots,x_i;y_0,\ldots,y_{j-2},y_j\right)-p^*\left(x_0,\ldots,x_i;y_0,\ldots,y_{j-2},y_{j-1}\right)}{y_j-y_{j-1}},\\p^*\left(x_0,\ldots,x_i;y_k\right)&=\frac{p^*\left(x_0,\ldots,x_{i-2},x_i;y_k\right)-p^*\left(x_0,\ldots,x_{i-2},x_{i-1};y_k\right)}{x_i-x_{i-1}},\\p^*\left(x_i;y_j\right)&=f\left(x_i,y_j\right),\\&\qquad\left(i=0,1,\ldots,m;\ \ j=0,1,\ldots,n\right).\end{aligned} \tag{12}$$

4.3. Newton-Thiele's Rational Interpolation Window. Different from Thiele's rational interpolation, a $4*4$ rectangle interpolation window is used in Newton-Thiele's interpolation process. We use 16 pixel points around the interpolation point as sampling points, which are not in the same row (column) as the interpolation point. The details are as shown in Figure 5, where the black dot is the point being interpolated and 16 color solid points around it are sampling points.

4.4. Refining Inpainting Algorithm by Newton-Thiele's Rational Interpolation. By using Thiele's interpolation inpainting algorithm, we get an initial repaired image, and next we need to refine the intensity of every damaged pixel point. Because every damaged point has been repaired initially, in the refinement process, except for the damaged point being interpolated, the other damaged points are treated as known pixel points whose intensities are those of the initial repaired image. The mask image is used again to determine the position of every damaged pixel point, and the interpolation window in Figure 5 and Newton-Thiele's rational interpolation function are used to get intensity of every damaged point. And then we update the intensity of the damaged point instead of that in initial repaired image. When the intensities of all damaged points have been interpolated and updated, we can get a final repaired image.

5. Implementation and Experimental Analysis

5.1. Algorithm Implementation. In order to get a better repaired image, we use Thiele's rational interpolation to get an initial repaired image, and then Newton-Thiele's rational interpolation is used to refine the initial repaired image to get a final result. The whole inpainting algorithm includes two steps: the adaptive inpainting and the refining inpainting. In the adaptive inpainting phase, Thiele's rational interpolation function is used to interpolate intensity of every damaged point, and the detailed process is summarized in Algorithm 1.

In the refining inpainting phase, Newton-Thiele's rational interpolation function is used to refine the intensity of every damaged point, and the detailed process is summarized in Algorithm 2. The implementation of Algorithm 2 is based on the result of Algorithm 1.

Table 1: The comparison of PSNRs.

PSNR	BSCB	CT	EB	TV	Thiele	Ours
Figure 6	33.299020	33.196864	30.390751	29.572404	33.087954	**33.405554**
Figure 7	34.034243	33.838789	31.795258	28.643684	33.902315	**34.253171**
Figure 8	20.082217	20.419491	17.850016	20.41844	20.748204	**22.006812**
Figure 9	32.907370	35.794873	33.724928	28.595606	35.921904	**36.095908**
Figure 10	25.846801	27.711887	24.733383	26.424346	26.993011	**29.085404**
Figure 11	23.084520	24.303285	18.296882	16.128918	23.808367	**25.803948**

Input: I: a damaged image, M: a mask image.
Output: S_1: an initial repaired image
(1) $[m, n] = \text{size}(I)$, $S_1 = I$,
% Initialization
(2) **For** $i = 0 : m - 1$
(3) **For** $j = 0 : n - 1$
% Judge damaged points by the mask image
(4) **If** $M(i, j) == 0$
% The point (i, j) is a damaged point
(5) Compute $T_m(i, j)$ by using the formulas (1) and (3) which is just the intensity of the damaged point (i, j).
% Compute the intensity of the damaged point by Thiele's rational interpolation function
(6) $S_1(i, j) = T_m(i, j)$
% Get the intensity of point (i, j) in the initial repaired image
(7) **End if**
(8) **End for**
(9) **End for**
(10) **Return**

Algorithm 1: Adaptive inpainting via Thiele's rational interpolation.

Input: S_1: the initial repaired image, M: the mask image.
Output: S_2: the final repaired image
(1) $[m, n] = \text{size}(M)$, $S_2 = S_1$
% Initialization
(2) **For** $i = 0 : m - 1$
(3) **For** $j = 0 : n - 1$
% Judge damaged points by the mask image
(4) **If** $M(i, j) == 0$
% The point (i, j) is a damaged point
(5) Compute $R_{m,n}^{NT}(i, j)$ which is just the intensity of the damaged point (i, j) by using the formulas (7) (8) (10).
% Compute the intensity of the damaged point by Newton-Thiele's rational interpolation function
(6) $S_2(i, j) = R_{m,n}^{NT}(i, j)$.
% Update the intensity of the damaged point (i, j) instead of that in the initial repaired image
(7) **End if**
(8) **End for**
(9) **End for**
(10) **Return**

Algorithm 2: Refining inpainting via Newton-Thiele's rational interpolation.

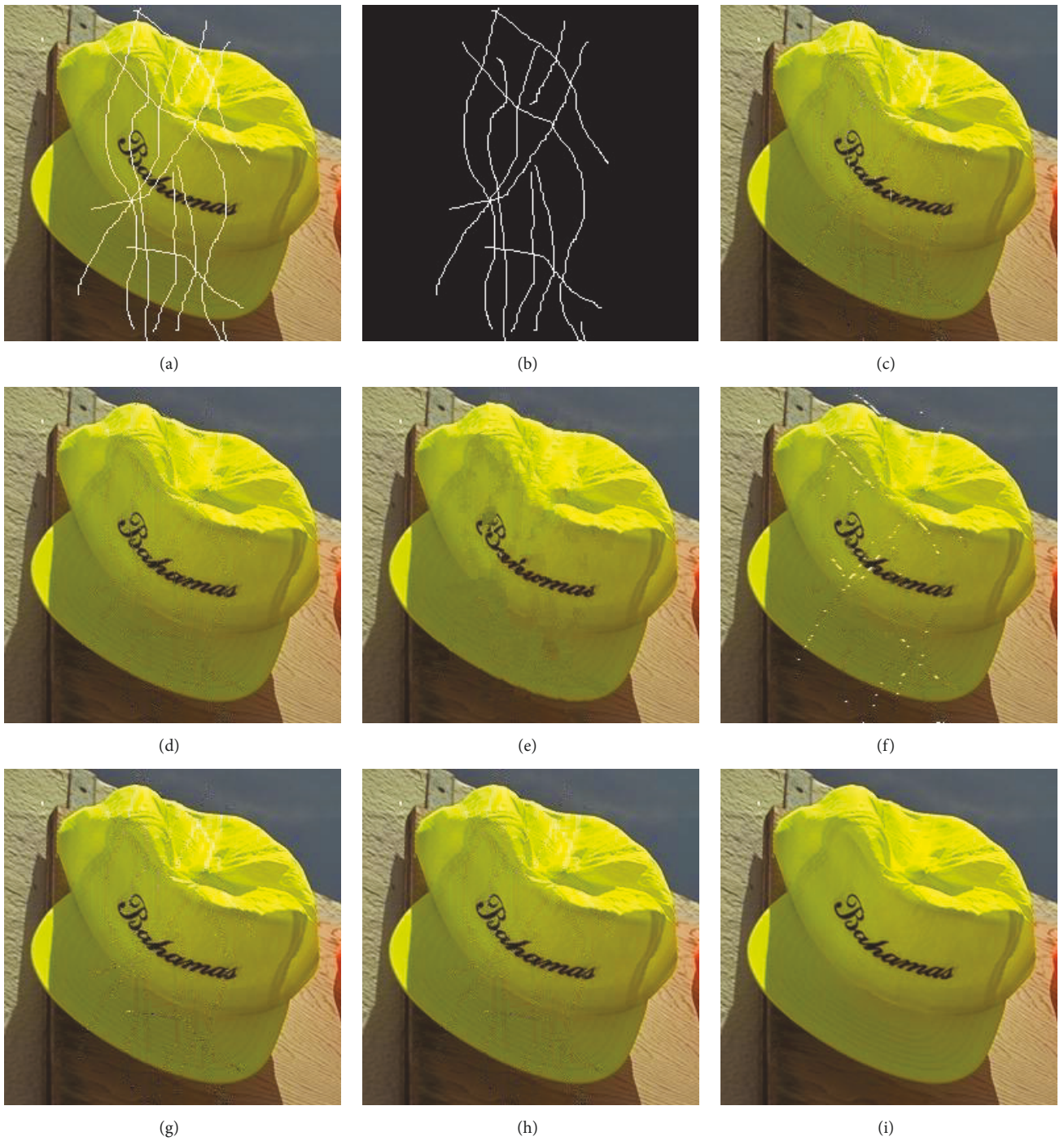


FIGURE 6: Repaired hat images by different inpainting methods. (a) The damaged image; (b) the mask image; (c) by BSCB; (d) by CT; (e) by EB; (f) by TV; (g) by Thiele; (h) by our method; (i) the original image.

Peak Signal to Noise Ratio (PSNR) is often used as a measure of signal reconstruction quality, which can be defined as follows:

$$\mathrm{MSE} = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\left\|O(i,j) - S_2(i,j)\right\|^2,$$

$$\mathrm{PSNR} = 10 \times \log_{10}\left(\frac{255^2}{\mathrm{MSE}}\right), \tag{13}$$

where m, n is the size of the damaged image, S_2 is the final repaired image, and O is the original image.

Figure 7: Repaired results by different inpainting methods. (a) The damaged image; (b) the mask image; (c) by BSCB; (d) by CT; (e) by EB; (f) by TV; (g) by Thiele; (h) by our method; (i) the original image.

5.2. Experimental Results and Analysis. In this section, we demonstrate the effectiveness and superiority of the proposed method through plenty of experiments. In our experiments, the original images are selected from standard datasets (set 5, set 14, and B100) and the original images with scratching are used as damaged images. The standard datasets are used for image processing, which can be downloaded from the Internet (http://vllab.ucmerced.edu/wlai24/LapSRN/). The intensities of repaired regions in the mask images are 255, and others are 0. For color images, in experiments, we apply our algorithm to each of the red, green, and blue

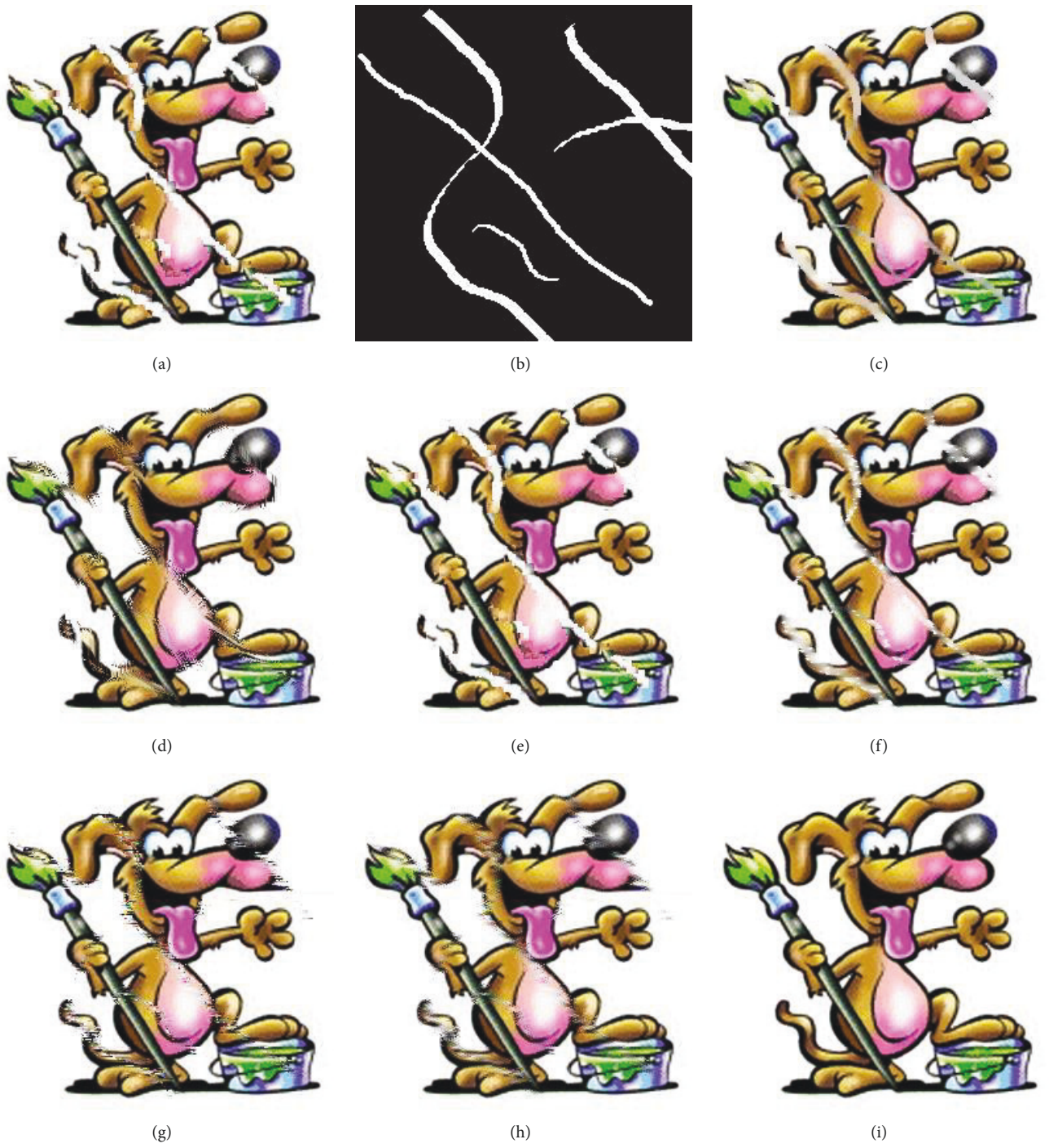


FIGURE 8: Repaired cartoon images by different inpainting methods. (a) The damaged image; (b) the mask image; (c) by BSCB; (d) by CT; (e) by EB; (f) by TV; (g) by Thiele; (h) by our method; (i) the original image.

channels, respectively. We use plenty of images for experiments and choose five algorithms for comparison with the proposed method. They are the schemes of papers [1] (BSCB for short), [12] (CT for short), [6] (EB for short), [3] (TV for short), and [11] (Thiele for short). All the experiments are tested on PC with Intel(R) core (TM) i3-4130, 3.4 GHz CPU, 8 GB RAM, NVIDIA 1 GB, and MATLAB 2010b.

From Figures 6–11, all images are with the size of 256×256. The PSNRs of the images repaired by above

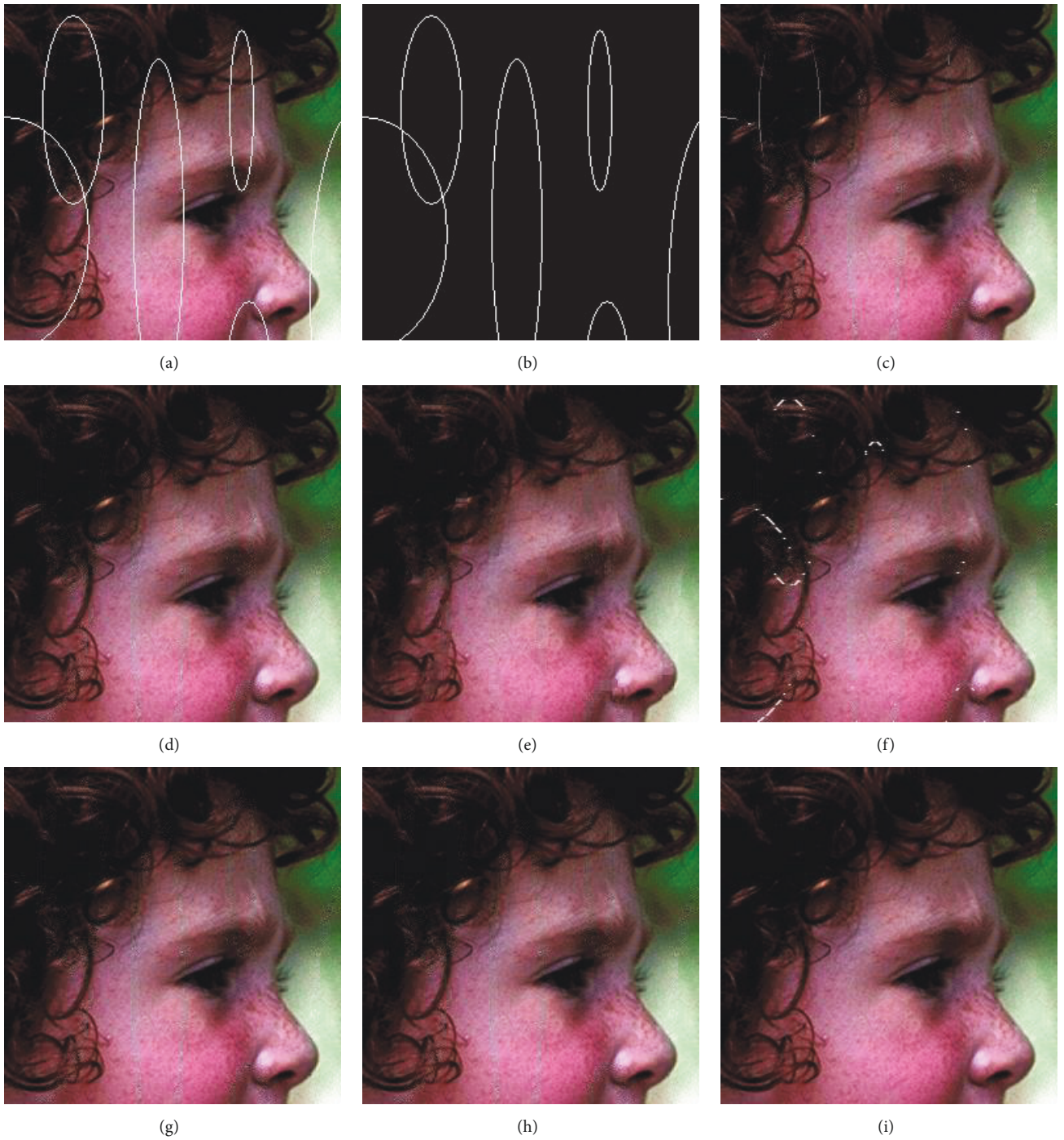
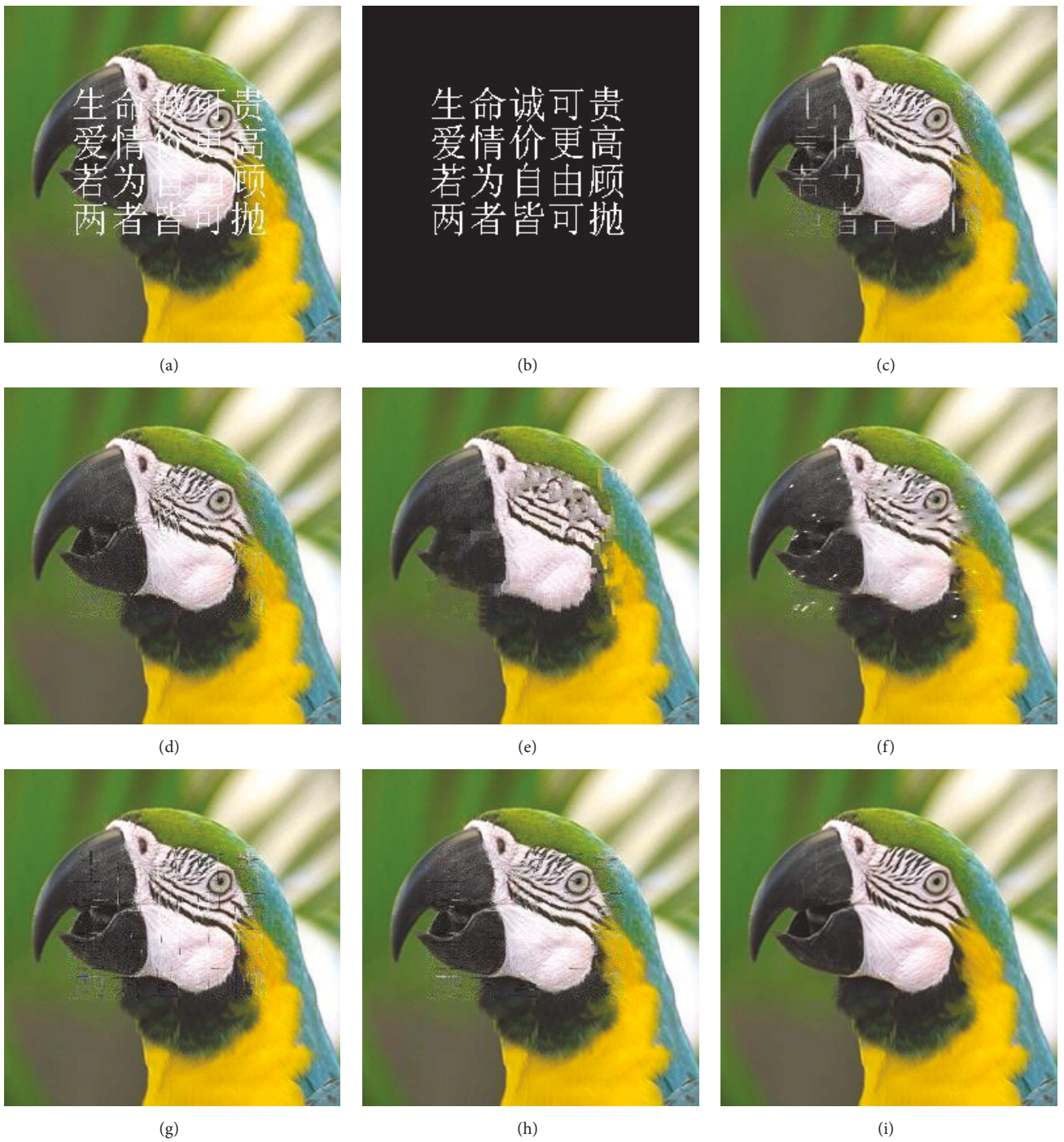


Figure 9: Repaired girl images by different inpainting methods. (a) The damaged image; (b) the mask image; (c) by BSCB; (d) by CT; (e) by EB; (f) by TV; (g) by Thiele; (h) by our method; (i) the original image.

different methods, respectively, are listed in Table 1, where the maximal PSNRs are highlighted with bold letters.

In order to illustrate the property of prominent texture details by our method, we select repaired image patches from Figure 11, which are shown in Figure 12. In order to show the superiority of our method, we select, respectively, one column from repaired image patches in Figure 12 and the intensities by different methods are compared, respectively, with that of original image patch, which are displayed in Figure 13. If intensities and frequency of intensity distribution are closer to those of original image patch, this means that



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

FIGURE 10: Repaired parrot images by different inpainting methods. (a) The damaged image; (b) the mask image; (c) by BSCB; (d) by CT; (e) by EB; (f) by TV; (g) by Thiele; (h) by our method; (i) the original image.

the repaired image by this method is closer to original image. From Figure 13, we find that the intensity distribution by our method is closer to that of the original image patch.

Through visual comparison of Figures 6–11, we find that the repaired results by TV [3] are not good due to some damaged domains not being repaired. When the damaged regions are large, the visual effects are not very good by BSCB [1] and EB [6] methods. By CT [12] and Thiele methods [11], the visual effects of repaired images are good; however, the details are not well repaired. The results by our method have better visual effect and prominent texture

Figure 11: Repaired raccoon images by different inpainting methods. (a) The damaged image; (b) the mask image; (c) by BSCB; (d) by CT; (e) by EB; (f) by TV; (g) by Thiele; (h) by our method; (i) the original image.

regions. From Figures 12-13, we find that the visual effect by our method is the best and the intensity distribution by our method is closer to that of the original image patch. Through objective comparisons of Table 1, we get that our PSNRs are all higher than those of other methods.

6. Discussions and Conclusions

A novel image inpainting method by using nonlinear rational interpolation was presented. Our approach was based on the observation that the texture details of repaired images by most image inpainting algorithms were not prominent.

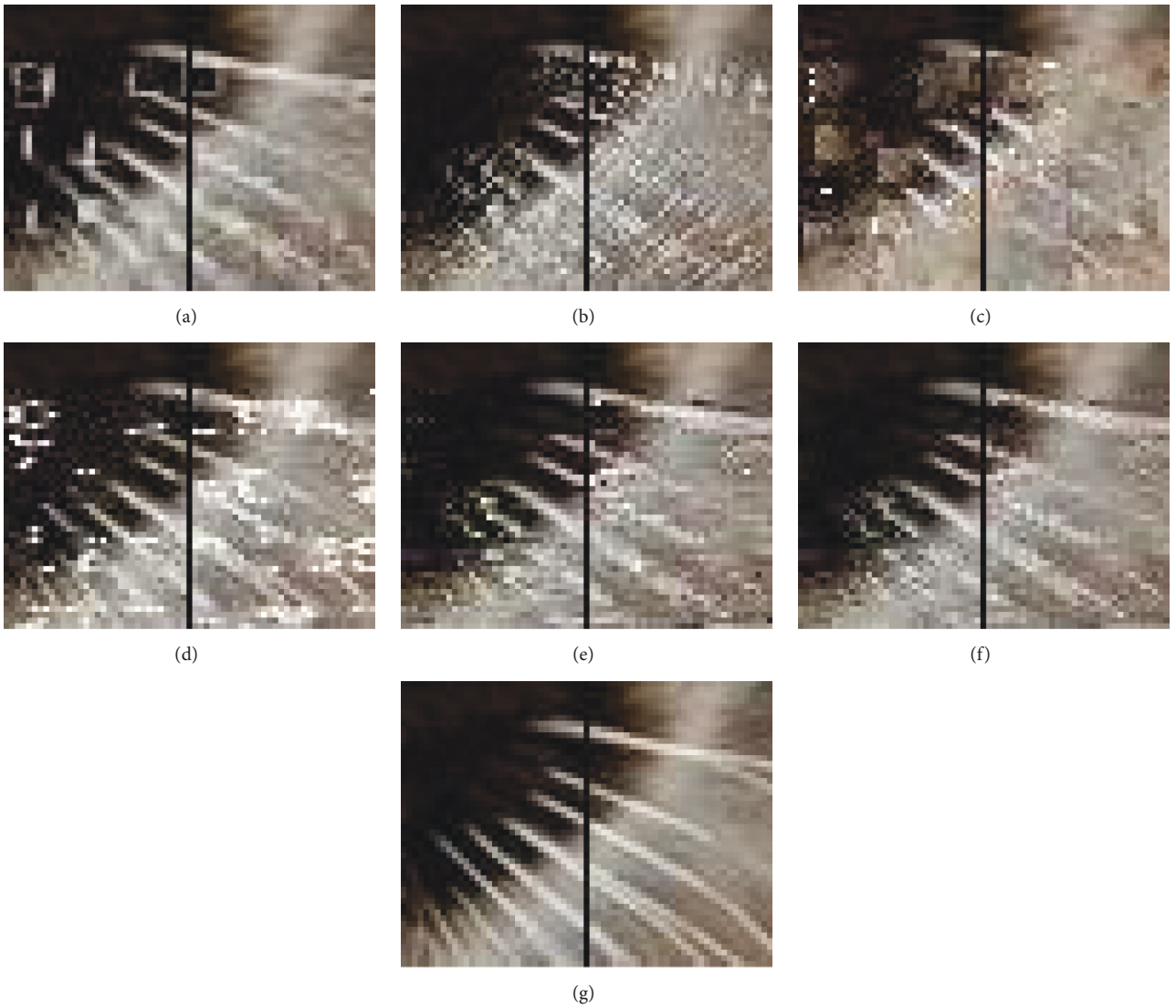


FIGURE 12: The texture patches display of repaired raccoon images by different inpainting methods. (a) By BSCB; (b) by CT; (c) by EB; (d) by TV; (e) by Thiele; (f) by our method; (g) the original image.

Inspired by the applications of continued fractions in image processing [11, 18–20], we proposed a novel image inpainting algorithm by using continued fractions rational interpolation. In order to obtain better repaired results, Thiele's rational interpolation was combined with Newton-Thiele's rational interpolation to repair damaged images. That is, the Thieles rational interpolation was used to interpolate adaptively initial repaired images, and Newton-Thiele's rational interpolation was used to refine the initial repaired images to get final results. Our method has been tested on a series of images, and results show that the proposed method has significantly better performance compared with the other inpainting approaches. The proposed method was suitable for the scratched images and small regions damaged images, and it was not suitable for the large area damaged images and badly scratched images. In future work, we will solve this problem.

Data Availability

Fully documented templates are available in the elsarticle package on CTAN.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grants nos. 61502141, 61070227, 61472466, and 11601115), the Anhui Provincial Natural Science Foundation (Grant no. 1508085QF128), and the Fundamental

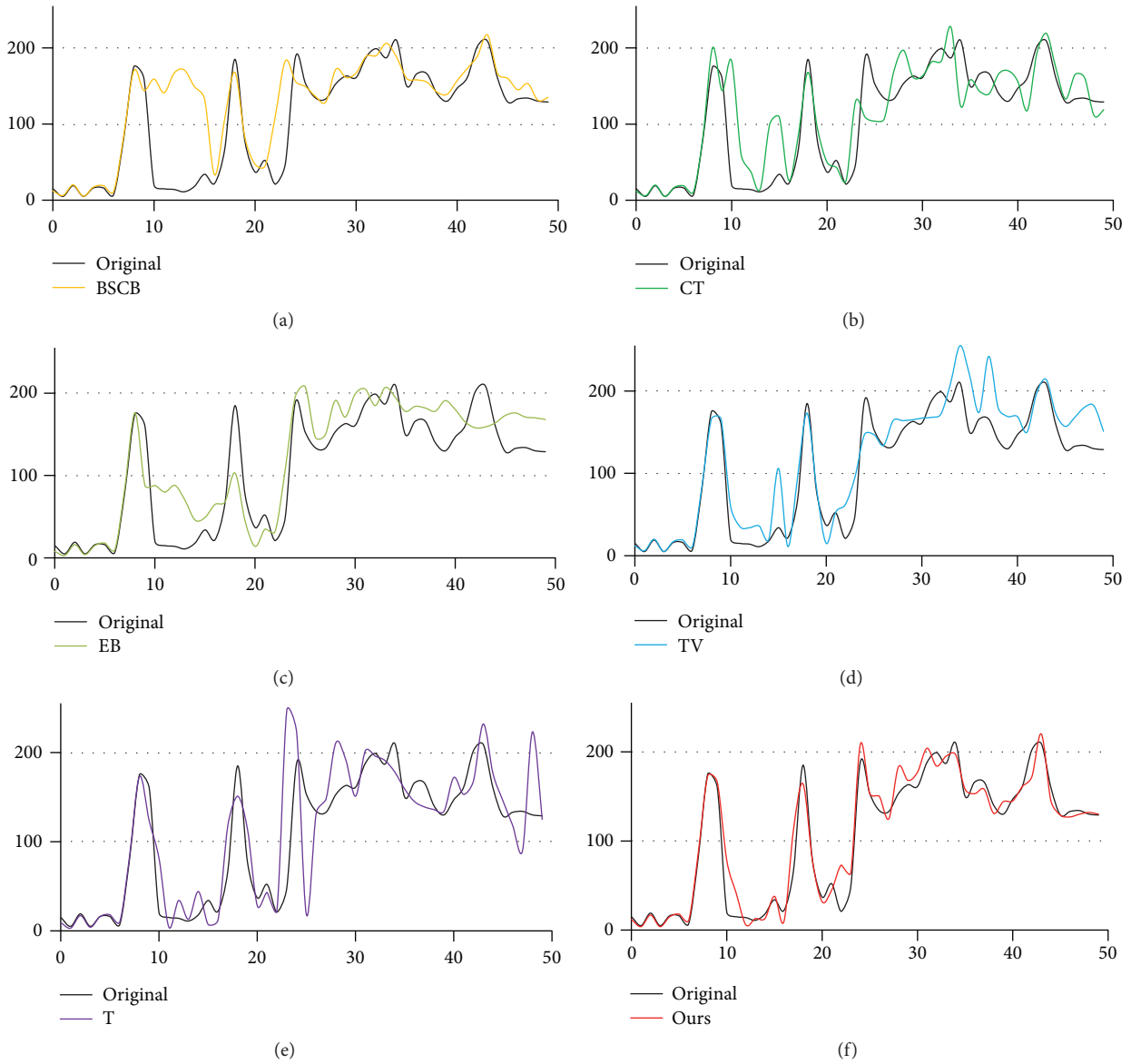


FIGURE 13: The intensity distribution of one column of repaired image patches. (a) By BSCB; (b) by CT; (c) by EB; (d) by TV; (e) by Thiele's interpolation; (f) by our method.

Research Funds for the Central Universities (Grants nos. JZ2015HGXJ0175 and JZ2016HGBZ1005).

References

[1] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques (SIGGRAPH '00)*, pp. 417–424, July 2000.

[2] T. F. Chan and J. Shen, "Nontexture inpainting by curvature-driven diffusions," *Journal of Visual Communication and Image Representation*, vol. 12, no. 4, pp. 436–449, 2001.

[3] T. F. Chan and J. Shen, "Mathematical models for local nontexture inpaintings," *SIAM Journal on Applied Mathematics*, vol. 62, no. 3, pp. 1019–1043, 2002.

[4] P. Pascal, S. Hoffmann, F. Nedwed, L. Hoeltgen, and J. Weickert, "Evaluating the true potential of diffusion-based inpainting in a compression context," *Signal Processing: Image Communication*, vol. 46, pp. 40–53, 2016.

[5] A. A. Efros and T. K. Leung, "Texture synthesis by non-parametric sampling," in *Proceedings of the 7th IEEE International Conference on Computer Vision (ICCV '99)*, vol. 2, pp. 1033–1038, IEEE, September 1999.

[6] A. Criminisi, P. Pérez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Transactions on Image Processing*, vol. 13, no. 9, pp. 1200–1212, 2004.

[7] G. Hu and L. Xiong, "Criminisi-Based Sparse Representation for Image Inpainting," in *Proceedings of the 3rd IEEE International Conference on Multimedia Big Data, BigMM 2017*, pp. 389–393, USA, April 2017.

[8] B. Shen, W. Hu, Y. Zhang, and Y.-J. Zhang, "Image inpainting via sparse representation," in *Proceedings of the 2009 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2009*, pp. 697–700, Taiwan, April 2009.

[9] S. Thaskani, S. Karande, and S. Lodha, "Multi-view image inpainting with sparse representations," in *Proceedings of the IEEE International Conference on Image Processing, ICIP 2015*, pp. 1414–1418, Canada, September 2015.

[10] T. J. V. S. Rao, M. V. G. Rao, and T. V. N. L. Aswini, "Image inpainting with group based sparse representation using self adaptive dictionary learning," in *Proceedings of the 4th International Conference on Signal Processing and Communication Engineering Systems, SPACES 2015 - In Association with IEEE*, pp. 301–305, India, January 2015.

[11] X. Huo, J. Tan, L. He, and M. Hu, "An automatic video scratch removal based on Thiele type continued fraction," *Multimedia Tools and Applications*, vol. 71, no. 2, pp. 451–467, 2014.

[12] F. Bornemann and T. Marz, "Fast image inpainting based on coherence transport," *Journal of Mathematical Imaging and Vision*, vol. 28, no. 3, pp. 259–278, 2007.

[13] R. A. Yeh, C. Chen, T. Y. Lim, A. G. Schwing, M. Hasegawa-Johnson, and M. N. Do, "Semantic image inpainting with perceptual and contextual losses," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–10, Honolulu, HI, USA, 2016.

[14] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context Encoders: Feature Learning by Inpainting," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2536–2544, Las Vegas, NV, USA, June 2016.

[15] M. S. Sapkal, P. K. Kadbe, and B. H. Deokate, "Image inpainting by Kriging interpolation technique for mask removal," in *Proceedings of the 2016 International Conference on Electrical, Electronics, and Optimization Techniques, ICEEOT 2016*, pp. 310–313, India, March 2016.

[16] Z. Zhou and X. Gao, "Laplace transform methods for a free boundary problem of time-fractional partial differential equation system," *Discrete Dynamics in Nature and Society*, Article ID 6917828, Art. ID 6917828, 9 pages, 2017.

[17] L. Su, T. Yan, Y. Zhao, and F. Li, "Local polynomial regression solution for partial differential equations with initial and boundary values," *Discrete Dynamics in Nature and Society*, vol. 2012, Article ID 201678, 2012.

[18] L. He, J. Tan, Z. Su, X. Luo, and C. Xie, "Corrigendum to "Super-resolution by polar Newton–Thiele's rational kernel in centralized sparsity paradigm" [Signal Processing: Image Communication, 31(2015), pp. 86–99]," *Signal Processing: Image Communication*, vol. 35, pp. 85-86, 2015.

[19] T. Bai, J. Tan, M. Hu, and Y. Wang, "A novel algorithm for removal of salt and pepper noise using continued fractions interpolation," *Signal Processing*, vol. 102, pp. 247–255, 2014.

[20] T. Bai and J. Tan, "Automatic detection and removal of high-density impulse noises," *IET Image Processing*, vol. 9, no. 2, pp. 162–172, 2015.

[21] H. S. Wall, *Analytic Theory of Continued Fractions*, D. Van Nostrand Company, New York, NY, USA, 1948.

[22] I. K. Khristina and W. Siemaszko, "Rational approximation and interpolation of functions by branched continued fractions," in *Rational approximation and Its Applications in Mathematics and Physic*, J. Gilewicz, M. Pindor, and W. Siemaszko, Eds., vol. 23 of *Lecture Notes in Mathematics*, pp. 24–40, Springer, Berlin, Germany, 1987.

[23] W. Siemaszko, "Thiele-type branched continued fractions for two-variable functions," *Journal of Computational and Applied Mathematics*, vol. 9, no. 2, pp. 137–153, 1983.

[24] A. Cuyt and B. Verdonk, "Multivariate reciprocal differences for branched Thiele continued fraction expansions," *Journal of Computational and Applied Mathematics*, vol. 21, no. 2, pp. 145–160, 1988.

[25] W. Siemaszko, "Branched continued fractions for double power series," *Journal of Computational and Applied Mathematics*, vol. 6, no. 2, pp. 121–125, 1980.

[26] J. Tan, "Computation of vector valued blending rational interpolants," *Numerical Mathematics: A Journal of Chinese Universities (English Series)*, vol. 12, no. 1, pp. 91–98, 2003.

[27] J. Tan and Y. Fang, "Newton-Thiele's rational interpolants," *Numerical Algorithms*, vol. 24, no. 1-2, pp. 141–157, 2000.