A New Algorithm for Image Resizing Based on Bivariate Rational Interpolation*

Shanshan Gao[1,2], Caiming Zhang[1,2], and Yunfeng Zhang[2]

[1] School of Computer Science and Technology, Shandong University, China
[2] School of Computer Science and Technology, Shandong Economic University, China
gsszxy@yahoo.com.cn

Abstract. A new method for image resizing by bivariate rational interpolation based on function values and partial derivative value is presented. When an original image is resized in an arbitrary ratio, the first step of the method is constructing the rational interpolation fitting the original surface where the given image data points are sampled from. The resized image can be obtained just by re-sampling on the interpolation surface. The algorithm presents how to estimate the partial derivative value of image data point needed for rational interpolation, and at same time considers the adjustment of tangent vector of the edge point to keep edges well defined. Various experiments are presented to show efficiency of the proposed method and that the resized images can preserve clear and sharp borders and hence offer more detail information in real application.

Keywords: Rational Interpolating Spline, Image Resizing, Shape Preserving.

1 Introduction

The problem of resizing images is fundamental and important in the fields such as medical imaging, remote sensing and some software of image processing. This problem arises frequently whenever a user wishes to get better view of a given image, then image resizing methods which can obtain the resized image with higher precision and good quality are required.

Usually, if we can get the interpolating surface fitting the original surface where the given image data points are sampled from, the resized image can be obtained just by re-sampling on the interpolation surface. The interpolation methods can be commonly divided into two types: polynomial interpolation and rational interpolation. At present many polynomial interpolations are popular in image processing application because of their simple implementation. Such as pixel replication or bilinear[1] interpolation. But the visual results of these interpolation methods all suffer from unacceptable effect (e.g. mosaics, aliasing, blocking) to some extent, especially on image edges. In order to

* Project supported by the National Nature Science Foundation of China (No. 60703081, 60673003), the National Nature Science Foundation of Shandong Province (No. Y2007G59), and the Visiting Scholar Foundation for excellent young teachers from colleges of Shandong Province.

G. Allen et al. (Eds.): ICCS 2009, Part II, LNCS 5545, pp. 770–779, 2009.

avoid these undesirable defects and obtain good smoothness, a lot of other interpolations are used, such as Bézier[2] or B-spline[3] interpolation, a locally adaptive resizing algorithm[4], resizing method based on vector quantization approximation for magnifying image by a factor of 2[5]. But unfortunately, the results gotten by these methods usually have fuzzy effect in the sharp region, zigzag stripes or blocking effect. Then the visual sharpness of the enlarged image can't match the quality of the original image. This is because most of these methods are based on a simple polynomial model with certain continuity, while preserving the low frequencies content of the source image, but not being able to enhance high frequencies, many details are lost.

However, rational function approximation is a typical non-linear approach, which can reflect the characteristics of the interpolated image surface better, especially it can reflect the mutation between the adjacent image pixels, and hence can keep clearness of edges, describe more complex shapes. References[6-7] use mixed form of polynomial and rational function: using Newton-Thiele as interpolation function to zoom the image and achieved good results. But the interpolation must be used on the whole block with complex computation.

Therefore, in this paper, bivariate rational interpolation model based on function values and partial derivative values is used to fit the original surface of the given discrete image. The zoomed image is obtained just by resampling on the interpolation surface. The partial derivative value of image data point needed for rational interpolation is computed and adjusted to keep shape preserving. In order to maintain the edge property, the adjustment of tangent vector of the edge is presented at same time.

2 Bivariate Rational Interpolation Based on Function Values and Partial Derivative Value

Digital image is a set of color data points on 2D plane. These data are not random, and not complete structural either, because these data are affected by many factors during the acquisition process such as the material of object, light intensity and angle and so on. So the value of adjacent data may have gradual change or have abrupt change. If polynomial is used to interpolate the image data points, it can reflect the gradual change very well because of its form and continuity, but it is inability for the abrupt change. Considering the property of the rational interpolation, we choose bivariate rational interpolation as interpolation function[8], which has the following advantages:1) pass through the known data points; 2) simple and explicit expression; 3) The expression is piece wised, each piece has its parameters, so it is easy for adjusting locally. The concrete form is described as follows:

Let Ω :[*a*,*b*;*c*,*d*]be the plane region, and $\{(x_i, y_i, P_{i,j}, \frac{\partial P_{i,j}}{\partial x}, \frac{\partial P_{i,j}}{\partial y}), i = 1,2,\cdots n; j = 1,2,\cdots m\}$ be a given set of data points, where $a = x_1 < x_2 < \cdots < x_n = b$ and $c = y_1 < y_2 < \cdots < y_n = d$ are the knot spacings, $P_{i,j}, \frac{\partial P_{i,j}}{\partial x}, \frac{\partial P_{i,j}}{\partial y}$ represent $P(x_i, y_j), \frac{\partial P(x,y)}{\partial x}, \frac{\partial P(x,y)}{\partial y}$ at the point (x_i, y_j) respectively. Let $h_i = x_{i+1} - x_i$, $l_j = y_{j+1} - y_j$, and for any point $(x, y) \in [x_i, x_{i+1}; y_j, y_{j+1}]$ in the (*x*,*y*)-plane, and let $\theta = \frac{x - x_i}{h_i}$ and $\eta = \frac{y - y_j}{l_j}$. First, for each $y = y_j, j = 1, 2 \cdots m$, construct the x-direct interpolating curve $P^*_{i,j}(x)$ in $[x_i, x_{i+1}]$;this is given by

$$P^*_{i,j}(x) = \frac{p^*_{i,j}(x)}{q^*_{i,j}(x)}, i = 1, 2 \cdots n-1 \tag{1}$$

Where $p^*_{i,j}(x) = (1-\theta)^3 \alpha^*_{i,j} P_{i,j} + \theta(1-\theta)^2 V^*_{i,j} + \theta^2(1-\theta) W^*_{i,j} + \theta^3 \beta^*_{i,j} P_{i+1,j}$

$q^*_{i,j}(x) = (1-\theta)\alpha^*_{i,j} + \theta\beta^*_{i,j}$

and $V^*_{i,j}(x) = (2\alpha^*_{i,j} + \beta^*_{i,j}) P_{i,j} + h_i \alpha^*_{i,j} \frac{\partial P_{i,j}}{\partial x}$,

$W^*_{i,j}(x) = (\alpha^*_{i,j} + 2\beta^*_{i,j}) P_{i+1,j} - h_i \beta^*_{i,j} \frac{\partial P_{i+1,j}}{\partial x}$.

with $\alpha^*_{i,j} > 0, \beta^*_{i,j} > 0$.This interpolation is called the rational cubic interpolation based on function values and derivatives which satisfies

$$P^*_{i,j}(x_i) = P_{i,j}, P^*_{i,j}(x_{i+1}) = P_{i+1,j}, P^{*\,\prime}_{i,j}(x_i) = \frac{\partial P_{i,j}}{\partial x}, P^{*\,\prime}_{i,j}(x_{i+1}) = \frac{\partial P_{i+1,j}}{\partial x}$$

Obviously, the interpolating function $P^*_{i,j}(x)$ on $[x_i, x_{i+1}]$ is unique for the given data $(x_r, P_{r,j}, \frac{\partial P_{r,j}}{\partial x}, r = i, i+1)$ and positive parameters $\alpha^*_{i,j}, \beta^*_{i,j}$.

Using the *x*-direction interpolation function, $P^*_{i,j}(x), i = 1, 2, \cdots n-1; j = 1, 2, \cdots m$ defines the bivariate rational interpolating function in $[x_1, x_n; y_1, y_m]$ .For each pair(*i*,*j*), $i = 1, 2, \cdots n-1$ and $j = 1, 2, \cdots m-1$, let $\alpha_{i,j} > 0, \beta_{i,j} > 0$, define the bivariate interpolating function $P_{i,j}(x, y)$ on $(x, y) \in [x_i, x_{i+1}; y_j, y_{j+1}]$ as follows

$$P_{i,j}(x, y) = \frac{p_{i,j}(x, y)}{q_{i,j}(y)}, i = 1, 2, \cdots n-1; j = 1, 2 \cdots m-1 \tag{2}$$

Where

$$p_{i,j}(x, y) = (1-\eta)^3 \alpha_{i,j} P^*_{i,j}(x) + \eta(1-\eta)^2 V_{i,j} + \eta^2(1-\eta) W_{i,j} + \eta^3 \beta_{i,j} P^*_{i,j+1}(x)$$

$$q_{i,j}(y) = (1-\eta)\alpha_{i,j} + \eta\beta_{i,j}$$

and

$$V_{i,j} = (2\alpha_{i,j} + \beta_{i,j}) P^*_{i,j}(x) + l_j \alpha_{i,j} f^*_{i,j}(x, y_j)$$

$$W_{i,j} = (\alpha_{i,j} + 2\beta_{i,j}) P^*_{i,j+1}(x) - l_j \beta_{i,j} f^*_{i,j+1}(x, y_{j+1})$$

with $f^*_{i,s}(x,y_s)=(1-\theta)\frac{\partial P_{i,s}}{\partial y}+\theta\frac{\partial P_{i+1,s}}{\partial y},\theta\in[0,1],s=j,j+1$ .It is obvious that $f^*_{i,s}(x,y_s)$ satisfy $f^*_{i,s}(x,y_s)=\frac{\partial P_{r,s}}{\partial y},r=i,i+1,s=j,j+1\cdot$

The term $P_{i,j}(x,y)$ is called the bivariate rational interpolation based on function values and partial derivative value which satisfies

$$P_{i,j}(x_r,y_s)=P(x_r,y_s),\frac{\partial P_{i,j}(x_r,y_s)}{\partial x}=\frac{\partial P_{r,s}}{\partial x},\frac{\partial P_{i,j}(x_r,y_s)}{\partial y}=\frac{\partial P_{r,s}}{\partial y},r=i,i+1,s=j,j+1$$

It is easy to understand that the interpolating function $P_{i,j}(x,y)$ on $[x_i,x_{i+1};y_j,y_{j+1}]$ is unique for the given data $(x_r,y_s,P_{r,s},\frac{\partial P_{r,s}}{\partial x},\frac{\partial P_{r,s}}{\partial y},r=i,i+1,s=j,j+1)$ and parameters $\alpha^*_{i,j},\beta^*_{i,j},\alpha^*_{i,j+1},\beta^*_{i,j+1},\alpha_{i,j},\beta_{i,j}$.

3 Image Resizing

Suppose that $I_{m,n}$ is a original image with $m\times n$ image data points $P_{i,j}$, $i=0,1,...,m-1,j=0,1,...,n-1$,. These image data are sampling values being regarded as taken from a surface $P(x,y)$. The best way to get a resized image with good quality is to resample from $P(x,y)$. Hence the problem of resizing an original image becomes a problem of constructing fitting surface $P(x,y)$ using the sampling points $P_{i,j}$, $i=0,1,...,m-1,j=0,1,...,n-1$.

Firstly, matrix $P_1=\{P_{i,j},0\le i\le m-1,0\le j\le n-1\}$ is expanded to $P_2=\{P_{i,j},0\le i\le m,0\le j\le n\}$, so the values of $P_{ij}(i=m\ or\ j=n)$ are added. Outer-interpolation method is used, then let $P_{m,j}=2P_{m-1,j}-P_{m-2,j}(0\le j\le n-1)$, $P_{i,n}=2P_{i,n-1}-P_{i,n-2}(0\le i\le m-1)$, $P_{m,n}=P_{m,n-1}+P_{m-1,n}-P_{m-1,n-1}$. At last interpolation surface $P(x,y)$ are constructed based on P_2 .On each interval [x_i,x_{i+1}]×[y_i,y_{i+1}], $i=0,1,...,m-1,j=0,1,...,n-1$, a bivariate rational interpolating spline surface $P_{i,j}(x,y)$ is constructed. To construct $P_{i,j}(x,y)$, see Fig1, the function value $P_{k,l}$, first partial derivatives $(P'_{k,l})_x$ and $(P'_{k,l})_y$ $k=i,i+1,l=j,j+1$, and the parameters $\alpha^*_{i,j},\beta^*_{i,j}$, $\alpha^*_{i,j+1},\beta^*_{i,j+1}$ and $\alpha_{i,j},\beta_{i,j}$ need to be known. The function value

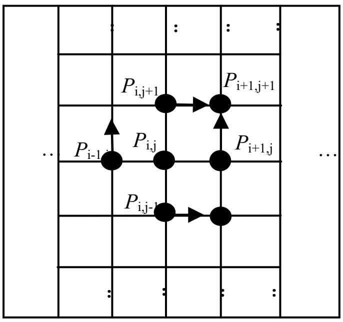


Fig. 1. construction of $P_{ij}(x,y)$

$P_{k,l}$ is the pixel value of original image, $\alpha^*_{i,j}, \beta^*_{i,j}, \alpha^*_{i,j+1}, \beta^*_{i,j+1}$ and $\alpha_{i,j}, \beta_{i,j}$ can be determined according to the condition mentioned above and the practical need. By massive experiments, we find that if $1 \le \alpha_{i,j}, \alpha^*_{i,j}, \alpha^*_{i,j+1} \le 1.5$ and $0.1 \le \beta_{i,j}, \beta^*_{i,j}, \beta^*_{i,j+1} \le 1.0$, the result of the method will be better. Now we discuss the computation of $(P'_{k,l})_x$ and $(P'_{k,l})_y$.

In order to estimate the first order partial derivative $(P'_{i,j})_x$ of $P_{i,j}$, $P_{i-1,j}$, $P_{i,j}$ and $P_{i+1,j}$ are generally used to construct fitting curve in *x*-direction. In similar, points $P_{i,j-1}$, $P_{i,j}$ and $P_{i,j+1}$ can be used to estimate $(P'_{i,j})_y$. Here we take $(P'_{i,j})_x$ as an example to introduce the estimating method for the first order partial derivative of $P_{i,j}$. A quadratic polynomial interpolating curve is constructed by $P_{i-1,j}$, $P_{i,j}$ and $P_{i+1,j}$ and the first order partial derivative is gotten from the curve to approximate $(P'_{i,j})_x$.

$$(P'_{i,j})_x = \frac{\Delta x_i \Delta P_{i-1,j} + \Delta x_{i-1} \Delta P_{i,j}}{\Delta x_{i-1} + \Delta x_i} = \frac{\Delta x_i}{\Delta x_{i-1} + \Delta x_i} \Delta P_{i-1,j} + \frac{\Delta x_{i-1}}{\Delta x_{i-1} + \Delta x_i} \Delta P_{i,j}$$

Where $\Delta x_i = x_{i+1,j} - x_{i,j}$, $\Delta P_{i,j} = \dfrac{P_{i+1,j} - P_{i,j}}{\Delta x_i}$.

Because digital image can be denote as regular data field, intervals between two neighbor points are equal, so:

$$(P'_{i,j})_x = (P_{i+1,j} - P_{i-1,j})/2 \tag{3}$$

For boundary data points $P_{1,j}$ and $P_{n,j}$, we have

$$P'_i(x_{i-1,j}) = \frac{x_{i-1,j} - x_{i+1,j} + x_{i-1,j} - x_{i,j}}{x_{i-1,j} - x_{i+1,j}} \Delta P_{i-1,j} + \frac{x_{i-1,j} - x_{i,j}}{x_{i+1,j} - x_{i-1,j}} \Delta P_{i,j} = 2\Delta P_{i-1,j} - (P'_{i,j})_x$$

then $(P'_{1,j})_x = 2\Delta P_{1,j} - (P'_{2,j})_x$. In similar: $(P'_{n,j})_x = 2\Delta P_{n-1,j} - (P'_{n-1,j})_x$

Rational spline curve based on function and partial derivative constructed directly with (3) sometimes can't get the shape that given data points advised, which is not acceptable in real application. For example, if the link lines of $P_{i-2,j}$, $P_{i-1,j}$, $P_{i,j}$ and $P_{i+1,j}$ is a planar convex polygon, see Fig2, In order to keep shape preserving, the curve P interpolated to $P_{i,j}$, $P_{i,j}$ $(P'_{i,j})_x$ and Q interpolated to $P_{i-1,j}$, $P_{i,j}$ $(P'_{i-1,j})_x$ must be in the convex region of points $P_{i-1,j}, P_{imid,j}$ and $P_{i,j}$. If *P* and *Q* are out of the region we need adjust the derivative computed by (3). We can process as follows:

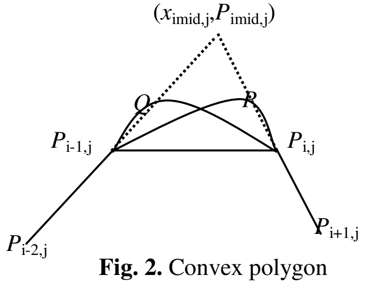


Fig. 2. Convex polygon

$$P = a_1 x^2 + b_1 x + c_1$$

Where $a_1 = \dfrac{P_{i-1,j} - P_{i,j} - (P'_{i,j})_x}{2x_{i,j} - 1}$, $b_1 = (P'_{i,j})_x - 2a_1 x_{i,j} \bullet c_1 = P_{i,j} - a_1 x_{i,j}^2 - b_1 x_{i,j}$

In similar: $Q = a_2 x^2 + b_2 x + c_2$.

The line between $(x_{i,j}, P_{i,j})$ and $(x_{i+1,j}, P_{i+1,j})$ is:

$$P_1 = (P_{i+1,j} - P_{i,j})(x - x_{i,j}) + P_{i,j} = A_1 x + B_1$$

A line between $(x_{i-2,j}, P_{i-2,j})$ and $(x_{i-1,j}, P_{i-1,j})$ is:

$P_2 = (P_{i-1,j} - P_{i-2,j})(x - x_{i-1,j}) + P_{i-1,j} = A_2 x + B_2$ Let $(x_{imid,j}, P_{imid,j})$ denote the intersection point of these two lines, see Fig2. Adjusting $(P'_{i,j})_x$ according to the cross point of interpolation curve *P* and line P_1. The cross point can be computed by the following formula:

$$P - P_1 = a_1 x^2 + b_1 x + c_1 - (P_{i,j} - P_{i-1,j})(x - x_{i-1,j}) - P_{i-1,j} = 0 \qquad (4)$$

Formula (4) has at least one root, namely the interpolating curve and line P_1 will intersect at the end point $(P_{i,j}, x_{i,j})$. If there is one root named x_p which is not end point and satisfies $x_{imid,j} < x_p < x_{i,j}$, it shows that curve *P* is not in the convex region of points $P_{i-1,j}$, $P_{imid,j}$ and $P_{i,j}$, see Fig2. As well known, if *P* is tangent with line P_1, *P* is in the convex region mentioned above. So the shape preserving can be satisfied. Resolving system of equations:

$$\begin{cases} A_1 x + B_1 = a_1 x^2 + b_1 x + c_1 \\ 2a_1 x + b_1 = A_1 \end{cases}$$ then $(P'_{i,j})_x$ is gotten.

In similar, we can get $(P'_{i-1,j})_x$ to make interpolation curve *Q* tangent with line P_2, and then make *Q* in the convex region of three points $P_{i-1,j}$, $P_{imid,j}$ and $P_{i,j}$.

During image interpolation process, keeping clearness of edges is necessary. If interpolate with (3), although rational spline can avoid the blurred phenomenon at a certain extent, the edges may be also blur in some case. When a data point P_i is on the edges of digital image, the gray values of its adjacent data points P_{i-1} and P_{i+1} have high difference, we call it a "saltation". Although rational spline can process these "saltation" phenomenon in certain degree, but for some cases that the difference of gray values is comparatively large (such as from white to black), interpolation error will also occur using rational spline. So in this paper, we present an adjusting method for derivative of data points. (3) can determine the derivative of data points. When the gray values of adjacent points have high difference, the derivative of edge points is also high. So we use gradient method to estimate a pixel point is on edge or not. In discrete case, we denote gradient of a pixel as follows:

$$G(x_i, y_i) = |df(x_i)| + |df(y_i)|$$

where $\begin{array}{l} df(x_i)=f(x_{i+1})-f(x_i) \\ df(y_i)=f(y_{i+1})-f(y_i) \end{array}$.

In this paper, we give a threshold value to decide a pixel is on edge or not. Threshold value can be taken as $th\in[30,40]$. If the gradient of a pixel is more than given threshold value, we denote it an edge pixel.

Tangent vector can affect the convex degree, so adjusting direction and length of tangent vector on edges to adjust interpolating curve and shape of surface. The blurred phenomenon on the edges of image can be improved greatly. For rational spline interpolation, derivatives of data points are expected to change smoothly. For the edge of image has close relation with the "inner pixels", the derivatives of edge points can be adjusted by weighted of the derivatives of boundary points and the derivatives of points in the around adjacent region. If a pixel point is a boundary point, we use the following formula to adjust the tagent of data points which are on the boundary of image:

$$P_{i,j}'(x_i)=\alpha_1 P_{i-1,j}'(x_i)+\alpha_2 P_{i,j}'(x_i)$$

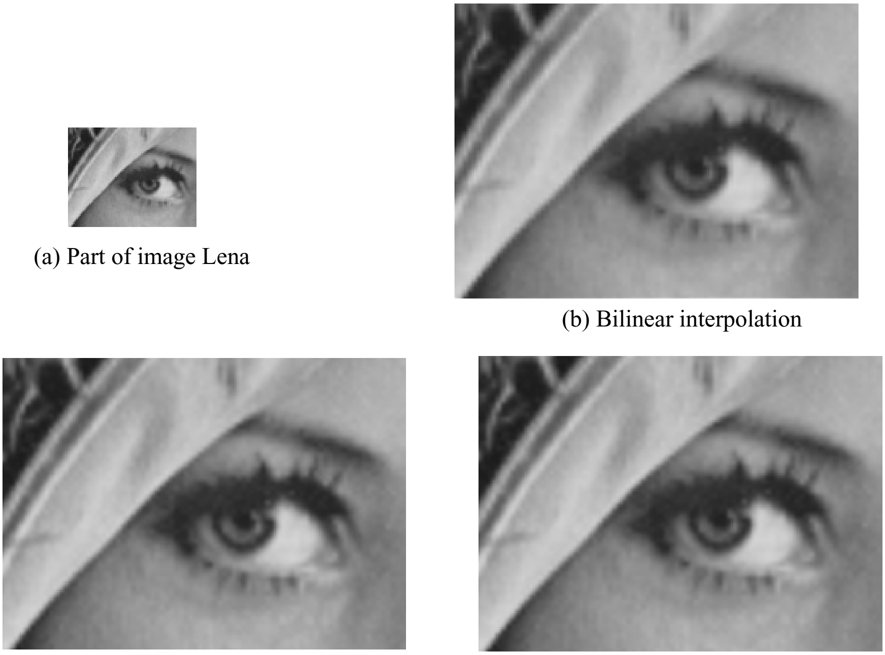
Where α_1 and α_2 are both tunable parameters. For simplification, denote $\alpha_1=\alpha_2=1/2$.

Then on the interval $[x_i,x_{i+1}]\times[y_i,y_{i+1}]$, the bivariate rational interpolating spline surface $P_{i,j}(x,y)$ based on function value and partial derivative value can be defined as (2). All the surfaces $P_{i,j}(x,y)$ are put together to form the fitting surface $P(x,y)$.

In order to get a resized $m'\times n'$ CT image $I'(x',y')$ from the original $m\times n$ CT image $I(x,y)$,where $m'=m\times s$, $n'=n\times s$, we just need to get more samples by increasing the sample density in accordance with the interval 1/s in x-direction and y-direction of original image respectively to get the image zoomed.

4 Experiments

In this section, the efficiency of the new method is compared with bilinear and bi-cubic interpolation methods, respectively. The new method has been generalized to work with some classical images, such as Lena, Girl, House, GoldHill and peppers and so on, and the results of resizing different part of image Lena with the magnification factor 3*3 and 5*5 respectively, see Fig3 and Fig4, will be taken as examples to show the comparison between the different methods, where image in (a) is original part of image Lena, image in (b) is created by bilinear interpolation to the original image data points, the one in (c) is created by bi-cubic interpolation, the one in (d) is created by the new method. From the images (b)-(d) show that image in (d) has better quality than images in (b)-(c). Fig5 shows another comparison working on the bird image. It is obvious that new method is implemented without producing the so-called mosaics or blocky effect, and the results maintain clearness of the image, including edges, hence offers more detail information. When the image is enlarged by a larger factor, the new method can still present better visual effect.



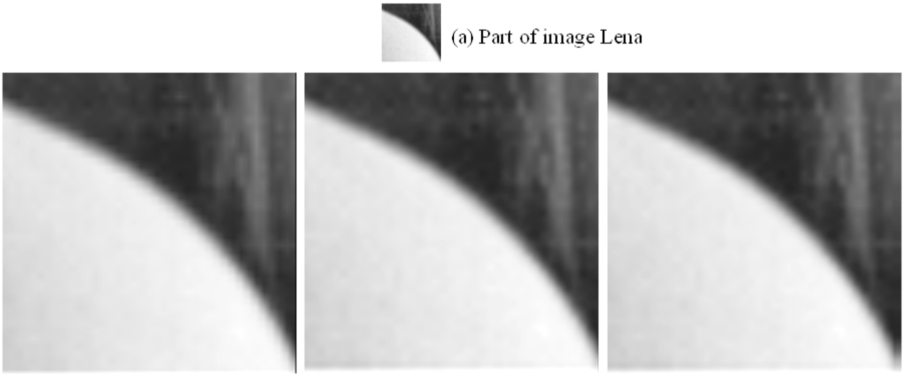
(a) Part of image Lena

(b) Bilinear interpolation

(c) Bicubic interpolation

(d) New method

Fig. 3. Enlarging for part of image Lena by 3*3



(a) Part of image Lena

(b) Bilinear intepolation

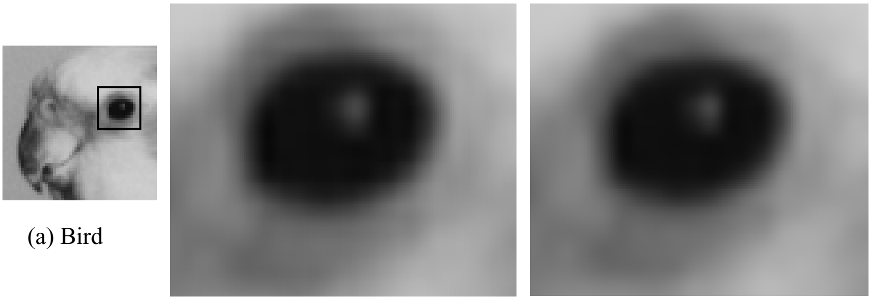
(c) Bicubic interpolation

(d) New method

Fig. 4. Enlarging for part of image Lena by 5*5

For a more systematic analysis for the new method, the peak signal-to-noise ratio (PSNR) is used for comparison. PSNR is defined as

$$PSNR = 10log_{10}\left(\frac{\sum_{ij} 255^2}{\sum_{ij}(I_{ij}^0 - I_{ij})^2}\right)\text{dB} \quad .$$



(a) Bird

(b) Bicubic interpolation

(c) New method

Fig. 5. Enlarging for part of image Bird by 10*10

Where $I^0$is the original image and I denotes the recovered image. Firstly, we shrink a selected set of images by 2* 2 using the bilinear interpolation, then magnify by the same factors using the new method and bicubic interpolation respectively. So the original image $I^0$is just the given image, and I is the image gotten after magnifying. Then the PSNR is computed and shown in Table1.

Table 1. PSNR Analysis

Algorithms	Images				
	Girl	Goldhill	House	Lena	peppers
Bicubic	25.472	24.910	23.285	24.503	25.131
New method	27.175	25.673	25.529	26.039	26.871

Besides, we undergo another test. Shrink the selected set of images above by 2* 2, magnify by the same factor, and shrinking and magnifying used the same method. The PSNR is computed and shown in Table2.

Table 2. PSNR Analysis

Algorithms	Images				
	Girl	Goldhill	House	Lena	peppers
Bilinear	25.883	27.311	27.532	28.293	27.367
Bicubic	27.490	27.499	25.038	28.293	26.413
New method	25.560	27.910	27.188	29.866	30.936

As shown in the Tables, the new algorithm improves the PSNR for all cases. We believe that it is due to a denoising feature of the algorithm. The form of rational interpolation, the property of shape preserving and the adjustment on the edgy area are all the reasons for the high quality of the resized image.

The basic algorithm for gray scale pictures can be easily generalized to the case of RGB colored digital images. Three color value R, G, B can be denoted as interpolation points for rational interpolation respectively, the zoomed image can be obtained by using the interpolation three times.

5 Conclusion

Image resizing by via standard polynomial interpolation methods usually lose more image details, which produces the so-called mosaics or blocky effect, and the edges of zoomed image is blurring. When the magnification factor is bigger, this effect is more evidence. As an important non-linear numerical analysis tool, rational interpolation method can describe the value relation of adjacent pixels better. In this paper, bivariant rational spline function being of advantage to preserving the mutation of the image is used as interpolation function to fit the original surface of the given discrete image. High frequency part of image on the region edge is preserved by adjusting the partial derivative of image data points. Comparing with traditional algorithms, the quality of resized image can be improved greatly with simple computation and arbitrary magnification factor including integer and fractional factor.

References

1. Castleman Kenneth, R.: Digital image processing. Tsinghua University Press, Beijing (1998)
2. Qingjie, S., Xiaopeng, Z., Enhua, W.: A method of image zooming in based on Bézier surface interpolation. Journal of Software 10(6), 570–574 (1999)
3. Zhaoxia, Y., Feng, L.: Guan Lutai1 Image enlargement and reduction with arbitrary accuracy through scaling relation of B-spline. Journal of Computer Aided Design & Computer Graphics 13(9), 824–827 (2001)
4. Battiato, S., Gallo, G., Stanco, F.: A locally adaptive zooming algorithm for digital images. Image and Vision Comput. 20(11), 805–812 (2002)
5. Chang, C., et al.: An image zooming technique based on vector quantization approximation. Image and Vision Computing 23, 1214–1225 (2005)
6. Min, H., Yousheng1, Z.: Image zooming based on Thiele's rational interpolation. Journal of Computer Aided Design &Computer Graphics 15(8), 1004–1007 (2003)
7. Min, H., Jieqing, T., Xiaoping1, L.: Method of image zooming based on bivariate vector valued rational interpolation. Journal of Computer Aided Design & Computer Graphics 16(11), 1496–1500 (2004)
8. Duan, Q., Zhang, H., Twizell, E.H.: A Bivariate Rational Interpolation with Symmetric Bases. Journal of Applied Mathematics and Computation 179(1), 190–199 (2006)
9. Zhang, Y., Duan, Q., Twizell, E.H.: Convexity control of a bivariate rational interpolating spline surfaces. Computers & Graphics 31, 679–687 (2007)