A Computationally Efficient Super-Resolution Reconstruction Algorithm Based On The Hybird Interpolation

Xiangguang Zhang
Information Institute, Qingdao University of Science and Technology
Email: xiangguang_zhang@163.com

Yun Liu
Information Institute, Qingdao University of Science and Technology
Email: lyun-1027@163.com

Abstract—**In order to release the limitations on the computational complexity and the detail missing of the traditional interpolation algorithm, this paper proposes a double channel hybrid interpolation algorithm for super-resolution reconstruction that can not only reduce the complexity effectively but also keep the details of the image sufficiently. The improved algorithm involves following methods: first, the parallel processing is used in this project; second, the cubic B-spline linear interpolation is introduced in the low frequency domain; third, the edge interpolation algorithm is applied in the high frequency details. The simulation and experiments both indicate that this algorithm is able to process the image in real time with details preserved.**

Index Terms—**Super Resolution Reconstruction, Linear Interpolation, Cubic B-Spline Interpolation, Texture Detail, Mahalanobis Distance**

I. INTRODUCTION

The interpolation is an important reconstruction technology for image processing. The main tasks of image processing technology are about to reduce noise and to simplify computational complexity, especially the computational efficiency[1] in the real application. Interpolation based on the asymptotic theory is a method of constructing new data points within the range of a discrete set of known data points, and choosing the right interpolatory function is very important. Many researchers devoted themselves to the optimal function, but the reward becomes less and less[2].

For a long time, the popular method of image interpolation is linear interpolation including polynomial interpolation and convolutional interpolation which can transform into each other[3]. In the mathematical subfield of numerical analysis, polynomial interpolation is the interpolation of a given data set by a polynomial. In other words, given some data points (such as obtained by sampling), the aim is to find a polynomial which goes exactly through these points. As the low resolution image can be obtained through the convolution between hyper-resolution image and low pass filter, the reconstructed image can also get from the same operation between the low resolution image and the suitable function. That is the convolution interpolation.

Although the classical image interpolations such as nearest-neighbor interpolation, bilinear interpolation, bicubic interpolation and so on have low computational complexity and can be actualized easily by software and hardware, the edge detail area of the magnified image inevitability has sawtooth phenomenon and smooth phenomenon because classical linear interpolations can not produce the high frequency of image. Generally speaking, human vision is always sensitive to the edge detail area. Consequently, it is impossible for linear interpolation to provide the better visual effect.

Aimed at the deficiency of the traditional linear interpolation, many researchers proposed some improved algorithms. An interpolation algorithm based on PDE is presented according to the shortcoming of the traditional methods[4]. Although this method is good at the reconstruction of edge area in some directions, it is inadequate to describe the whole edge directions because only the gradient of x positive axis and y positive axis are considered in the description of the known gradient, meanwhile, it sacrificed its computational complexity and not to get better reconstruction results than spline interpolation in the flat area. A local adaptive magnification interpolation algorithm is proposed[5], and an improved edge direction interpolation algorithm is brought forward[6]. The other authors also put forward an adaptive max-relativity interpolation algorithm[7]. Although the above algorithms can improve image quality degenerated by the edge blur, the computational complexity restricts practical applications in data communication and image processing. The fast self-adaptive edge-oriented interpolation algorithm[8] is proposed to get better image quality at low-bit rate, however, the large error is produced because the strategy leads to the error accumulation and the edge orient misjudgment which indirectly use the estimated neighbor pixels to estimate interpolated pixels.

doi:10.4304/jcp.5.6.885-892

II. METHODOLOGY

According to the computational complexity, relativity, error accumulation and the analysis of former work, a double-channel image reconstruction model based on the characteristics of edge area and smooth area is proposed as Figure. 1.

The main idea of the hybrid image reconstruction algorithm as follow: the edge area and smooth area of the low-resolution image can be classified by the edge detector, and the parallel processing is adopted in the reconstruction of the edge and smooth area. The interpolation algorithm of smooth area restricts the real-time application of SRR because 70%~90% of the whole image area is smooth area. Because of using same interpolation strategies in smooth and edge area, some algorithms[4]-[8] sacrificed their computational complexity and did not get better reconstruction results than the traditional linear interpolation in the flat area. Different form those algorithms, this paper applies the linear interpolation algorithm to smooth area and uses the max-relativity edge interpolation algorithm in edge area respectively.

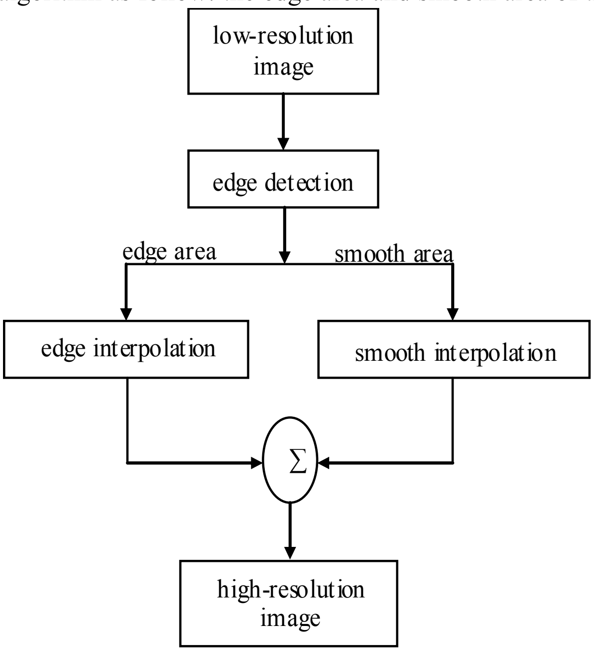


Figure 1. The sketch map of hybrid interpolation algorithm.

A. Interpolation Algorithm of Smooth Area

(1) The performance comparison of the smooth area interpolation algorithms

The value of all position can be computed by the interpolation function of the linear interpolation. The interpolation of the equal-interval discrete data can be described as follow:

$$f(x)=\sum_{k=0}^{K-1}C_k h(x-x_k) \quad (1)$$

Where h is the interpolation kernel function, and C_k denotes the weighted parameter. The value precision and computational complexity of the linear interpolation depend on the interpolation kernel, and the core work of interpolation algorithm is the design of interpolation kernel function. The traditional linear interpolation includes the nearest-neighbor interpolation, the bilinear interpolation, Lagrange interpolation, Newton interpolation and so on.

Although the nearest-neighbor interpolation has low computational complexity, the blocking artifact is noticeable because of the bad spectrum of the interpolation kernel when the image is magnified. The interpolation function of Lagrange interpolation with low computational complexity has not continuous differentiability because of the fixed order[9]. Though the reconstruction quality of the bilinear interpolation is better than the nearest-neighbor interpolation, its interpolation operation with many corresponding membership leads to several times computational complexity compare with the nearest-neighbor interpolation. According to the fast and smooth character of spine interpolation to smooth area because of the bigger support region than bilinear interpolation, this paper adopts the cubic B-spine interpolation in the smooth area.

(2) The theory of interpolation algorithm of smooth area--cubic B-spine interpolation

The cubic B-spline interpolation can be represented as follow:

$$\hat{f}(\xi)=\sum_{k=1}^{K}c_k s_k(\xi) \quad (2)$$

Where C_k denotes the parameter got from input, $s_k(\xi)$ is interpolation kernel function, and K denotes the amount of data. According to sampling theory, the Cardinal spine can be regarded as the kernel function:

$$s_k(\xi)=\frac{\sin 2\pi\Omega(\xi-\xi_k)}{2\pi(\xi-\xi_k)}=\sin c(\xi-\xi_k) \quad (3)$$

Where Ω denotes the single bandwidth of $f(x)$. We can use f_k instead of C_k to get a precise approximation when the sampling frequency is bigger than the Nyquist frequency.

Definition $\pi:\xi_0<\xi_1<\ldots<\xi_n<\xi_{n+1}$ is the point of the real axis in the interval $[\xi_0,\xi_{n+1}]$, the n-order B-spine in π can be expressed as follow:

$$B_n(\xi;\xi_0,\xi_1,\xi_2,\ldots,\xi_{n+1})=(n+1)\sum_{k=0}^{n+1}\frac{(\xi-\xi_k)^n U(\xi-\xi_k)}{\omega(\xi_k)} \quad (4)$$

where

$$\omega(\xi_k)=\prod_{\substack{j=0\\ j\neq k}}^{n+1}(\xi_k-\xi_j) \quad (5.a)$$

$$U(\xi-\xi_k)=\begin{cases}(\xi-\xi_k)^n & \text{for} \quad \xi>\xi_k \\ 0 & \text{for} \quad \xi\le\xi\end{cases} \quad (5.b)$$

$$n = 1,2,3,\cdots\cdots$$

The low order B-spine function of the uniform interval point includes rectangle function, triangle function, bell function, cubic spine function and so on.

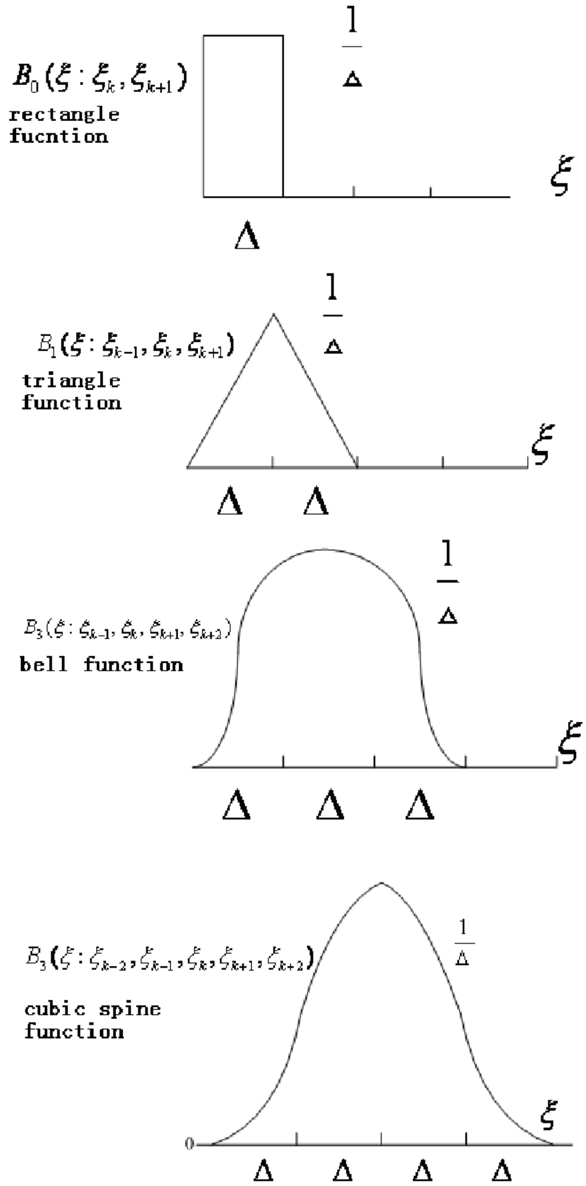


Figure 2. B-spine function.

All the figures of functions are listed in the Figure 2.

Where $\Delta = \xi_k - \xi_{k-1}$. To the equal interval point, we can get $B_1 = B_0 * B_0$, $B_2 = B_0 * B_0 * B_0$, and $B_3 = B_0 * B_0 * B_0 * B_0$ from equation (4), where * denotes convolution operation.

A standard cubic B-spine function is listed in Figure 3.

From Figure 2, it's easy to find that the interpolation kernel function of the B_0 and B_1 is not good. The span of spine function will be increased and many points will be ignored when the order of function is bigger than three. Considering the function smoothing and the low computational complexity, it is suitable to choose the cubic spine as the interpolation kernel function [10-11].

So, the cubic B-spine function can be obtained from

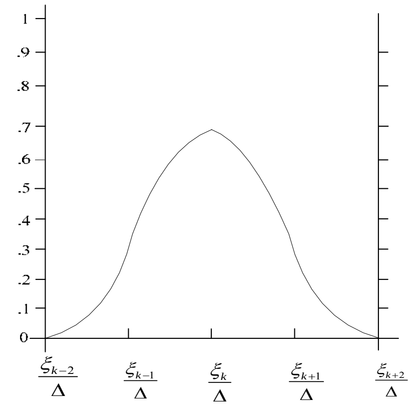


Figure 3. Standard cubic B-spine function.

equation (5):

$$\begin{aligned} s(\xi - \xi_k) &= B_3(\xi : \xi_{k-2}, \xi_{k-1}, \xi_k, \xi_{k+1}, \xi_{k+2}) \\ &= [(\xi - \xi_{k-2})^3 U(\xi - \xi_{k-2}) - 4(\xi - \xi_{k-1})^3 U(\xi - \xi_{k-1}) \\ &+ 6(\xi - \xi_k)^3 U(\xi - \xi_k) - 4(\xi - \xi_{k+1})^3 U(\xi - \xi_{k+1}) \\ &+ (\xi - \xi_{k+2})^3 U(\xi - \xi_{k+2})] \frac{1}{6\Delta^4} \end{aligned} \quad (6)$$

The interpolation function of the cubic B-spine interpolation can be gotten from the following equation:

$$\begin{aligned} \hat{f}(\xi_k, \eta_l) &= \frac{1}{36\Delta^2}[(c_{k-1,l-1} + 4c_{k,l-1} + c_{k+1,l-1}) + 4(c_{k-1,l} + 4c_{k,l} + c_{k+1,l}) + \\ &= (c_{k-1,l+1} + 4c_{k,l+1} + c_{k+1,l+1})] \end{aligned} \quad (7)$$

B. The Max-relativity Edge Interpolation Algorithm

The max-relativity edge interpolation algorithm insures the points for interpolation can close with the pixels of edge directions through comparing the relativity between the horizontal, vertical, clockwise 45 degree and anti-clockwise 45 degree based on the information of the six neighbor points. The whole description of the interpolation algorithm is listed as follow:

Let us consider a low resolution image I is $M \times N$, and the high resolution image Y we wanted is $2M \times 2N$. To estimate the edge pixel $Y_{i,j}$, we classify $Y_{i,j}$ two parts, where (i, j) denotes the line and column of pixel. The I part includes both pixels located even-line odd-column and even-column odd-line when $i + j$ is odd, and the II part includes both pixels located even-line even-column and odd-column odd-line when $i + j$ is even.

(1) The processing of I part

Let's set an example by the pixel Y_0 located odd-line even-column as Figure 4. The four directions correlations of the points for interpolation Y_0 in 3×3 area of high-resolution image can be defined as $R(k, Y_0)$, where Y_0 denotes the points for interpolation, $k \in \{h, v, d_1, d_2\}$, the four directions correlations can be represented as following: the correlation of horizontal direction can be defined as $R(h, Y_0)$, and the vertical is $R(v, Y_0)$, the clockwise 45 degree and anti-clockwise 45 degree are $R(t_1, Y_0)$ and $R(t_2, Y_0)$.

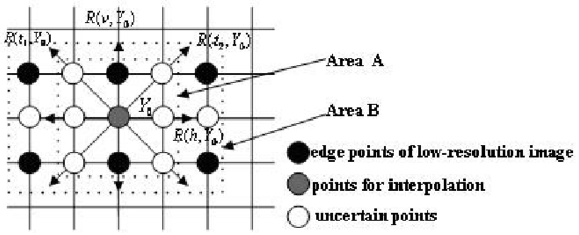


Figure 4. The interpolation of 3×3 and 3×2 area in high-resolution image.

The pixel for interpolation Y_0 can be calculated from the pixels which have the max-relativity with Y_0. The error will be introduced in the value of Y_0 and the edge direction of high-resolution image if we only use the vertical direction information to estimate Y_0 because the pixels of three directions have uncertainty in all four directions of 3×3 region A. Furthermore, the error will also be introduced in the other pixels for interpolation by using the mentioned Y_0 to estimate their values, the rest may be deduced by analogy, and the huge estimate error will appear in the final interpolation results.

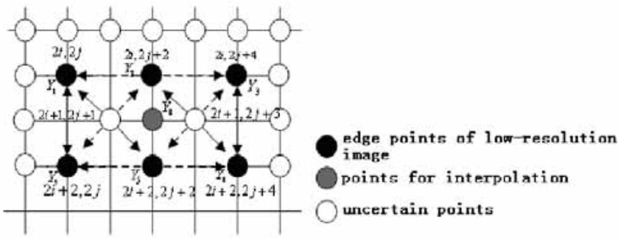


Figure 5. The multi-direction max-relativity interpolation of the pixel located odd-line even-column.

In order to reduce the error accumulation and the computational complexity, the neighbor B sized 3×2 made up of six pixels from the neighbor of Y_0 in low-resolution image is used directly to calculate the correlation $\hat{R}(k, Y_0)$ of four directions in area B of Y_0 according to the relativity of the same direction, and the mentioned $\hat{R}(k, Y_0)$ can be used to estimate $R(k, Y_0)$ of the area A in high-resolution image. The calculation method of $\hat{R}(k, Y_0)$ will be discussed.

Let $Y_1 \sim Y_6$ denote the six down-sample pixels, and Y_0 is the pixel for interpolation. The correlations of the four directions can be defined as following: the correlation of horizontal direction can be defined as $\hat{R}(h, Y_0)$, and the vertical is $\hat{R}(v, Y_0)$, the clockwise 45 degree and anti-clockwise 45 degree are $\hat{R}(t_1, Y_0)$ and $\hat{R}(t_2, Y_0)$.

Let $f(k)$ denote the pixel value, where $k \in \{Y_0, Y_1, Y_2, Y_3, Y_4, Y_5, Y_6\}$. In order to get the correlations of all directions, all kinds of pixels will be divided into four groups and each group comprises two two-dimension vectors. The division method can be described as following:

Horizontal direction:

$$\vec{X}_1 = (f(Y_1), f(Y_3)),\ \vec{Y}_1 = (f(Y_4), f(Y_6))$$

Vertical direction:

$$\vec{X}_2 = (f(Y_1), f(Y_4)),\ \vec{Y}_2 = (f(Y_3), f(Y_6))$$

Clockwise 45 degree:

$$\vec{X}_3 = (f(Y_2), f(Y_3)),\ \vec{Y}_3 = (f(Y_4), f(Y_5))$$

Anti-clockwise 45 degree:

$$\vec{X}_4 = (f(Y_1), f(Y_2)),\ \vec{Y}_4 = (f(Y_5), f(Y_6))$$

The Mahalanobis Distance is adopted in this paper to estimate the correlation of all direction vectors because it can not be affected by the dimension and it can enlarge the small change. The Mahalanobis Distance can be defined as (8) when the m-dimension vector X and Y are Gaussian distribution and the co-deviation matrix is Σ.

$$D(X,Y) = \left[(X-Y)\sum{}^{-1}(X-Y)^T\right]^{1/2} \quad (8)$$

So, the calculation method of the correlations in all directions can be described as (9)-(12):

$$\hat{R}(h, Y_0) = \left[(\vec{X}_1 - \vec{Y}_1)\sum{}^{-1}(\vec{X}_1 - \vec{Y}_1)^T\right]^{1/2} \quad (9)$$

$$\hat{R}(v, Y_0) = \left[(\vec{X}_2 - \vec{Y}_2)\sum{}^{-1}(\vec{X}_2 - \vec{Y}_2)^T\right]^{1/2} \quad (10)$$

$$\hat{R}(t_2, Y_0) = \left[(\vec{X}_3 - \vec{Y}_3)\sum{}^{-1}(\vec{X}_3 - \vec{Y}_3)^T\right]^{1/2} \quad (11)$$

$$\hat{R}(t_1, Y_0) = \left[(\vec{X}_4 - \vec{Y}_4)\sum{}^{-1}(\vec{X}_4 - \vec{Y}_4)^T\right]^{1/2} \quad (12)$$

Furthermore, we can get the max-relativity $R_{\max}$ from (13), and the max-relativity interpolation of the pixel for interpolation will be gotten from the edge direction estimated by $R_{\max}$.

$$R_{\max} = \min(\hat{R}(k, Y_0)), k \in \{h, v, t_1, t_2\} \quad (13)$$

It is seldom that the approximate pixels are used to estimate the new pixel when the mentioned max-interpolation is used to get the point for interpolation. Though the computational complexity of the method will be increased a little, the error accumulation will be avoided effectively.

Like the mentioned strategy, the pixels located even-line odd-column can be calculated from six pixels of the 2×3 neighbor in low-resolution image.

(2) The processing of II part

In Figure 6, the pixel X_0 for interpolation can be calculated by the neighbor pixels of four directions in high-resolution 3×3 area, where X_0 is $Y_{i,j}$ ($i+j=even$), and the neighbor of X_0 comprises the original pixels and the estimated pixels $Y_{i,j}$ from I.

The correlation functions of all four directions are: the correlation of horizontal direction can be defined as $\hat{R}(h,Y_0)$, and the vertical is $\hat{R}(v,Y_0)$, the clockwise 45 degree and anti-clockwise 45 degree are $\hat{R}(t_1,Y_0)$ and $\hat{R}(t_2,Y_0)$. Unlike the I part, the Euclid Distance is adopted to calculate the max-relativity from the eight points neighbor in the II part, and the calculation method of the correlation parameter $\hat{R}(k,X_0)$ can be described as (14)-(17):

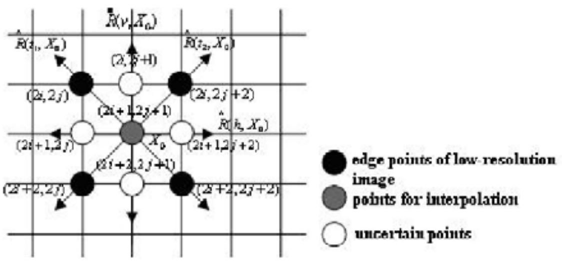


Figure 6. The interpolation of the pixel located even-line even-column.

$$\hat{R}(h,X_0)=\left|f(2i+1,2j)-f(2i+1,2j+2)\right| \quad (14)$$

$$\hat{R}(v,X_0)=\left|f(2i,2j+1)-f(2i+2,2j+1)\right| \quad (15)$$

$$\hat{R}(t_1,X_0)=\left|f(2i,2j)-f(2i+2,2j+2)\right| \quad (16)$$

$$\hat{R}(t_2,X_0)=\left|f(2i,2j+2)-f(2i+2,2j)\right| \quad (17)$$

Furthermore, we can get the max-relativity $R_{\max}$ from (18), and the max-relativity interpolation of the pixel for interpolation will be gotten from the edge direction estimated by $R_{\max}$.

$$R_{\max}=\min(\hat{R}(k,X_0)),k\in\{h,v,t_1,t_2\} \quad (18)$$

It is easy to estimate the points for interpolation in part II such as t_2 direction, and the other directions are same as t_2 . Let $R_{\max}=\hat{R}(t_2,X_0)$, and the pixels with the max-relativity are $Y_{2i,2j+2}$ and $Y_{2i+2,2j}$, so,

$$\text{If } \hat{s}=f(2i,2j+2)+f(2i,2j+2) \quad (19)$$

We can get

$$f(X_0)=\left(\frac{f^2(2i,2j+2)+f^2(2i,2j+2)}{\hat{s}}\right)/2 \quad (20)$$

Furthermore, the linear interpolation is used directly if the edge pixels situate on all sides of the image edge.

III. EXPERIMENTAL RESULTS AND ALGORITHM ANALYSIS

The images “Lena”, “Boat”, “Cameraman” and “Elaine” (256×256 8 bit) are used to test the performance of our hybrid interpolation reconstruction algorithm. The results of the simulation experiments are listed in the Figure 7, Figure 8, Figure 9 and Figure 10. The PSNR (Peak Signal-to-Noise Ratio), Mean and Square deviation of those filters are listed in the Table 1, Table 2, Table 3 and Table 4. The PSNR value of the new algorithm is much larger than that of the bicubic interpolation algorithm, and the square deviation of the new algorithm is much smaller than that of the bicubic interpolation algorithm. Moreover, hybird algorithm keeps the high-frequency details of the image much better than that of the bicubic interpolation algorithm.

IV. CONCLUSIONS

According to the problem in image reconstruction such as losing of high frequency details and the unsatisfied realtimeness of the linear interpolation algorithm, this paper proposes a double channel hybrid interpolation algorithm for super-resolution image reconstruction. This kind of super-resolution reconstruction algorithm is able to reduce the complexity effectively and to keep the details of the image sufficiently as well by the following methods: first, the parallel processing is used in the processing; second, the cubic B-spline linear interpolation is used in the low frequency area; third, the edge interpolation algorithm is used in the high frequency area. The performance of the hybrid interpolation reconstruction algorithm has been compared with that of the bicubic interpolation algorithm. Simulation experiments reveal that the proposed algorithm significantly outperforms bicubic interpolation algorithm by having much higher PSNR with real-time, consistent and stable performance. The practical photography experiments by different light illuminations also indicate that the algorithm can contribute to both real time applications and the detail preserving method.

ACKNOWLEDGMENT

The author would like to thank the anonymous reviewers for their valuable comments that helped to improve the paper. The National Natural Science Foundation of P.R China (60641010) supported the work.

(a) Original high resolution image

(b) Low resolution image

(c) The results of the bicubic interpolation reconstruction

(d) The results of proposed hybrid interpolation reconstruction

Figure 7. Simulation experiments of "Lena"

TABLE I.
COMPARISION OF PSNR, SQUARE DEVIATION, MEAN OF "LENA"

Image type	Bicubic interpolation algorithm	Proposed hybrid interpolation reconstruction algorithm
Square deviation	245.37	90.7
Mean	-1.5732	0.2569
PSNR	22.684	29.64

(a) Original high resolution image

(b) Low resolution image

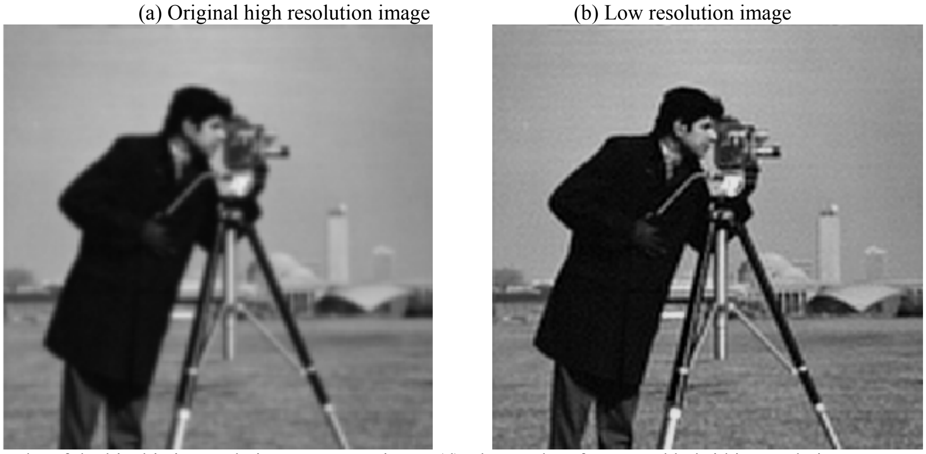
(c) The results of the bicubic interpolation reconstruction (d) The results of proposed hybrid interpolation reconstruction

Figure 8. Simulation experiments of “Boat”

TABLE II.
COMPARISION OF PSNR, SQUARE DEVIATION, MEAN OF “BOAT”

Image type	Bicubic interpolation algorithm	Proposed hybrid interpolation reconstruction algorithm
Square deviation	210.43	34.32
Mean	2.0572	-0.3905
PSNR	24.87	32.87

(a) Original high resolution image (b) Low resolution image



(c) The results of the bicubic interpolation reconstruction (d) The results of proposed hybrid interpolation reconstruction

Figure 9. Simulation experiments of “Cameraman”

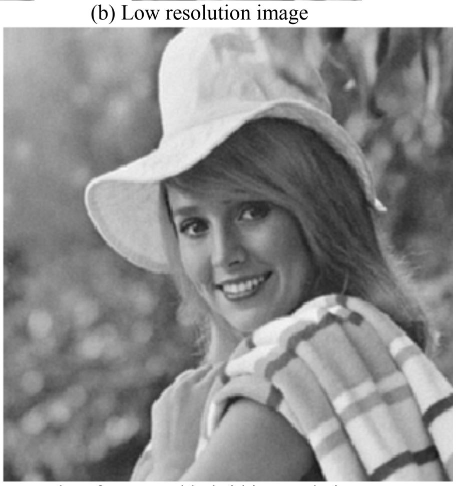
TABLE III.
COMPARISION OF PSNR, SQUARE DEVIATION, MEAN OF “CAMERAMAN”

Image type	Bicubic interpolation algorithm	Proposed hybrid interpolation reconstruction algorithm
Square deviation	232. 7	35.6
Mean	3.3357	0.3851
PSNR	23.29	32.68

(a) Original high resolution image

(b) Low resolution image

(c) The results of the bicubic interpolation reconstruction



(d) The results of proposed hybrid interpolation reconstruction

Figure 10. Simulation experiments of "Elaine"

TABLE IV.
COMPARISION OF PSNR, SQUARE DEVIATION, MEAN OF "ELAINE"

Image type	Bicubic interpolation algorithm	Proposed hybrid interpolation reconstruction algorithm
Square deviation	205.6	32.5
Mean	0.0774	1.5343
PSNR	25.063	33.21

REFERENCES

[1] Philippe, Interpolation Revisited, IEEE Transactions on medical imaging, 2000, 19(7).

[2] Thierry Blu, Linear Interpolation Revitalized, IEEE Transactions on medical imaging, 2004, 13(5).

[3] Thomas M. Lehmann, Claudia Gonner, Klaus Spitzer. Survey : Interpolation methods in Medical Image Processing.IEEE Trans.on Medical Imaging.1999,18(11):1049-1075

[4] Hao Jiang ,C.Moloney. A new direction adaptive scheme for image interpolation. IEEE Trans.on Image Processing.2002,6(3):369-372

[5] CHOI KN, CARCASSONIM, HANCOCK ER.Recovering Facial Pose with the EM Algorithm[J]. Pattern Recognition.2002,35(10):2073-2093

[6] YOSHINOBU EBISAWAL.Face Pose Estimation Based on 3D Detection of Pupils and Nostrils[A].VECIMS 2005 IEEE International Conference on Virtual Environments, Human2Computer Interfaces,and Measurement Systems GiardiniNaxos[C].Itay,2005

[7] YAO P,EVANS G,CALWAY A.Using Affine Correspondence to Estimate 3D Facial Pose[A].Proceedings of the IEEE International Conference on Image Proceeding[C]. Thessaloniki,2001,3:919-922

[8] U YX,CHEN LB, ZHOU Y, et al. Estimating Face Pose by Facial Asymmetry and Geometry[A].Proceedings of the Sixth IEEE International Conference on Automatic Face and Gesture Recognition FGR'04[C].IEEE,2004

[9] R.W.Schafer,L.R. Rabiner. A digital signal processing approach to interpolation. Proc.IEEE.1973(61):692-702

[10] Nebot J.Prades, A.Albiol,Bachiller C. Enhanced B-Spline interpolation of images. IEEE Trans.on Image Processing.1998(3): 289-293

[11] M.Unser,A.Aldroubi,M.Eden.Fast B-Spline Transforms for Continuous Image Representation and Interpolation.IEEE Trans.on Pattern Analysis and Machine Intelligence.1991,13(3):277-285