Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/MBE

MBE, 17(6): 6573–6600.
DOI: 10.3934/mbe.2020343
Received: 30 July 2020
Accepted: 14 September 2020
Published: 28 September 2020

Research article

Spatial general autoregressive model-based image interpolation accommodates arbitrary scale factors

Yuntao Hu[1], Fei Hao[2,*], Chao Meng[3], Lili Sun[1], Dashuai Xu[1] and Tianqi Zhang[1]

[1] Nanjing Institute of Technology, Kangni Institute of Industrial Science and Technology, Nanjing 211167, China
[2] School of Mechanical Engineering, Nanjing Institute of Technology, Nanjing 211167, China
[3] School of Mechanical Engineering, Southeast University, Nanjing 211189, China

* **Correspondence:** Email: feehao2012@163.com; Tel: +8618905152622.

Abstract: This paper proposed a novel image interpolation algorithm with an arbitrary upscaling factor based on the spatial general autoregressive model. First, to accommodate arbitrary scale factors, a non-integer mapping method was modulated into the spatial general autoregressive model, which was employed to model the piecewise stationary pattern with a higher description capacity than autoregressive models. A gradient angle guided extension method was utilized to extend the spatial general autoregressive model, and more pixels in the neighborhood were included to estimate the parameters of the spatial general autoregressive model. To realize the high-accuracy estimation of the model parameters, a regularization method via an elastic network was adopted to maintain the complexity of the object function in a reasonable state and address the overfitting problem. We also introduced an iterative curvature method to refine the interpolation result of those image blocks with large variances of gray intensities. Experiments on 25 images were conducted with integer and non-integer magnification factors to systematically verify the objective and subjective measures of the proposed method. The visual artifacts were effectively suppressed by the proposed method, and a flexible interpolation method for arbitrary scale factors was implemented.

Keywords: image processing; arbitrary scale interpolation; autoregressive; gradient adaptive extension; elastic network; iterative curvature.

1. Introduction

With the rapid development of multimedia technologies, the collection and transmission of multimedia data have become greatly convenient and easy. Meanwhile, multimedia data quality of imaging equipment limits the wider application of machine vision inspection system. The method to improve the image resolution through algorithms—namely, image interpolation, has become an active research field in machine vision. To a certain extent, image interpolation can effectively increase the image resolution to meet various applications, such as motion tracking and pose estimation. In addition, image interpolation has a wide range of practical applications and commercial prospects in remote sensing, cloud-of-thing system [1], surveillance [2,3], medical imaging [4], multimedia processing [5,6], and consumer electronic [7] .

In the registration and fusion of non-homologous images, image size adjustment is needed. Especially in the visual measurement process of large machined parts, due to the angle of view, the images captured by different cameras have different dimensions [8]. This will lead to dimensional distortion. In order to reduce visual artifacts and adapt to the requirements of arbitrary magnification, a more flexible image interpolation method is needed. Although the visual quality of simple linear interpolation methods is not sufficient, these methods still are widely used due to their low computational complexity and interpolation flexibility. Linear interpolation methods include nearest-neighbor interpolation [9,10], bilinear interpolation [11,12], bicubic interpolation [11–14], cubic spine interpolation [15,16] and iterative linear interpolation [17,18], which predict the gray intensity of an unknown pixel according to the distance between the unknown pixel and the reference pixels. The pixels at the boundary, edge or texture of a low-resolution (LR) image have a greater impact on image interpolation than those perpendicular to the edge. Therefore, an isotropic low-pass filter is employed to weaken the high-frequency components of the edges, and the edges are consequently smoothed, which will lead to artifacts such as blurring, ringing, checkerboard effects, edge discontinuities and jagging.

2. Related works

In related works, contributions have been made in reducing the visual artifacts. However, those methods are mainly suitable for integer magnification of images. Some edge-guided interpolation algorithms have been designed to correct the artifacts stemming from isotropic filters. The explicit methods [19,20] were developed to consider local structural information such as edges or isophotes. However, for images containing many complex structures, an edge map estimated with these methods tends to be unpredictable, which increases the intensity variation of the interpolation. Implicit adaptive methods have been proposed, which embed the local structures into an objective function that can be solved using linear or nonlinear optimization methods. These methods can model the image’s local patterns and estimate the unknown pixels around the edges in texture-rich areas. Li and Orchard [21] proposed an edge adaptive interpolation method according to the geometric duality of edges, named new edge directed interpolation (NEDI). Geometric duality between the low-resolution covariance and the high-resolution covariance, couple the pair of pixels along the same orientation, enables NEDI to estimate the high-resolution covariance from its low-resolution counterpart with a qualitative model characterizing the relationship between the covariance and the resolution. Because the inaccurate estimation of covariance in texture-rich regions, jagging and color infidelity are inescapable on some

RGB samples. Analogous to NEDI, other studies have been conducted with geometrical similarity measurement [22–26] to improve interpolation accuracy. Chang and Kevin [27] used collaborative representation and exploiting non-local self-similarity of natural images and introduced the external HR information into the interpolation process. Zhang and Wu [22] proposed soft-decision adaptive interpolation (SAI), which learnt and adapted varying pixel structures by using a piecewise autoregressive (PAR) model in the local rectangular window of the image. To implicitly use AR models for better solving nonlinear problems at the edges in image interpolation, irregular windows were employed by Guo et al. [23–25], which were extended adaptively from the root windows according to the geometric features, such as isophotes and curvatures. The similarities between pixels were calculated using patch-geodesic distance. Cheng et al. [28] used Fast Fourier Transformation (FFT) multichannel interpolation to reconstruct or approximate the continuous signals from a series of discrete points. A spatial interpolation can predict the values of the unknown points by processing the surrounding variables with meaningful values within the same region. In our previous work [26], a spatial general autoregressive model (SGAR), which is a uniform expression for both linear and nonlinear AR models, was employed to implement a noise-insensitive and edge-preserving interpolator. Although it still uses regular windows, its nonlinear description ability was improved by introducing nonlinear terms into the model. The similarities between pixels in the image windows were implicitly exploited by the robust parameter estimation method named generalized M-estimator.

In contrast to the above interpolation strategies, the curvature-based interpolation algorithms introduced high-order derivative information of image intensity to achieve high-accuracy interpolation from the rough results. Giachetti and Asuni [29] proposed the iterative curvature based interpolation (ICBI) method, which generates high-resolution (HR) pixels within the grid along the directions with the lowest second-order derivative and subsequently updates the values of the HR pixels by minimizing the local variations of curvature of the image intensity. However, the hole-filling strategy limits the application with non-integer upscaling factors. Kim and Cha [30] proposed the curvature interpolation method (CIM) based on a partial differential equation (PDE). Because it is known to tend to converge to a piecewise constant image [31,32]. These methods share the characteristic that the final interpolation results are modified by an iterative procedure upon the initial values provided by other methods. It was emphasized that better interpolation accuracy was achieved, which was largely due to the first-step interpolation method.

Therefore, an arbitrary scale factor image interpolation based on the SGAR model was used to implement image interpolation from three aspects: image window adaptive extension by gradient angle, model regularization by the elastic network, and refinement of the result accuracy by the curvature constraints. The rest of the paper is structured as follows. Section 3 introduces the SGAR model-based image interpolation method and its implementation. Experimental results and a comparison study with some existing popular image interpolation techniques are presented in Section 4. Section 5 contains our conclusions.

3. SGAR model-based image interpolation method and its implementation

3.1. Image description based on SGAR model

AR models are effective tools for image modeling [22–26,33–35]. The linear AR model in vectorization form is shown as follows.

$$\hat{\boldsymbol{y}} = h_{\theta}\left(\boldsymbol{x}\right) = \boldsymbol{\theta}^{T} \cdot \boldsymbol{x} \tag{1}$$

where $\hat{\boldsymbol{y}}$ is the prediction value, $\theta = \left[\theta_0, \theta_1, \ldots, \theta_n\right]$ is the model's parameter vector, containing the intercept term θ_0 and the feature weights θ_1 to θ_n, $\boldsymbol{x}$ is the sample's feature vector, and h_θ is the hypothesis function, using the model parameters θ.

However, the data distributions of a digital image are more complex. To better fit the image model, the product of the pixels can be involved in the polynomial regression, which is adopted as a new feature, such as the SGAR model that was initially explored based on Weierstrass theory for digital image adaptive filtering in our previous work [36]. The pixels in the image follow a certain regular pattern that indicates the direction of gray scale change. In other words, every unknown interpolated pixel in a piecewise image can be estimated by its known adjacent neighbor with certain weights. The original images are broken up into small fragments. Based on the stationarity assumption in piecewise images [22], we model the fragments as a locally stationary Gaussian process. Inevitably, some fragments with abrupt and unnatural gray scale changes are brought into the modeling process and have negative effect on describing the pattern of the image [36]. An adaptive filter was implemented based on the SGAR model. The new filter removed these artifacts while effectively conserving detailed image information. This was because the SGAR model fuses both linear and nonlinear AR models into a uniform expression[26]. However, the high-order polynomial regression model will probably overfit the training data. Moreover, simple linear regression has the problem of underfitting. For an image window, it could be modeled as

$$x(\boldsymbol{p}) = \sum_{i=1}^{r}\left\{\sum_{\boldsymbol{s}_1 \in \boldsymbol{A}} \cdots \sum_{\boldsymbol{s}_i \in \boldsymbol{A}} \phi\left(\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_i\right) \prod_{k=1}^{i} x\left(\boldsymbol{s}_k\right)\right\} + a\left(\boldsymbol{p}\right) \tag{2}$$

where $x(\boldsymbol{p})$ is the forecasted value of a pixel, $x(\boldsymbol{s}k)$ is the reference value of a pixel in the LR image, $a(\boldsymbol{p})$ is the modeling residuals. $\boldsymbol{s}$ is a two-dimensional vector representing the location of the specific pixel. $\boldsymbol{A}$ is a set of vectors representing the location of adjacent reference pixels. The composition of set $\boldsymbol{A}$ is discussed in section 2.3. $\phi(\boldsymbol{s}1, \boldsymbol{s}2, \ldots, \boldsymbol{s}i)$ is the model parameters. r is the order of the model, which indicates the dimension of vector ϕ. When r is 1, the SGAR model is degenerated into AR model as is shown in Eq (3). The dimension of the vector ϕ is only 8 and consists of the eight adjacent pixels around the anchor pixel.

$$x(\boldsymbol{p}) = \sum_{\boldsymbol{s}_i \in \boldsymbol{A}} \phi(\boldsymbol{s}_i) x(\boldsymbol{s}_i) + a(\boldsymbol{p}) \tag{3}$$

When r is 2, the SGAR model can also be rewritten as Eq (4). The dimension of vector ϕ is 44 and consists of 8 adjacent pixels around the anchor pixel, 36 self-multiplication and product of two different pixels are taken into consideration. We will explain how to obtain the parameters of the SGAR model in section 2.4.

$$x(\boldsymbol{p}) = \sum_{\boldsymbol{s}_i \in \boldsymbol{A}} \phi(\boldsymbol{s}_i) x(\boldsymbol{s}_i) + \sum_{\boldsymbol{s}_a \in \boldsymbol{A}} \sum_{\boldsymbol{s}_b \in \boldsymbol{A}} \phi(\boldsymbol{s}_a, \boldsymbol{s}_b) x\left(\boldsymbol{s}_a\right) \cdot x\left(\boldsymbol{s}_b\right) + a(\boldsymbol{p}) \tag{4}$$

3.2. Interpolation accommodate arbitrary scaling factor

The relationship between the coordinates of the pixels of the HR image and those of the corresponding LR image must first be clarified.

$$\begin{cases} u = i \times \dfrac{1}{h} \\ v = j \times \dfrac{1}{h} \end{cases} \tag{5}$$

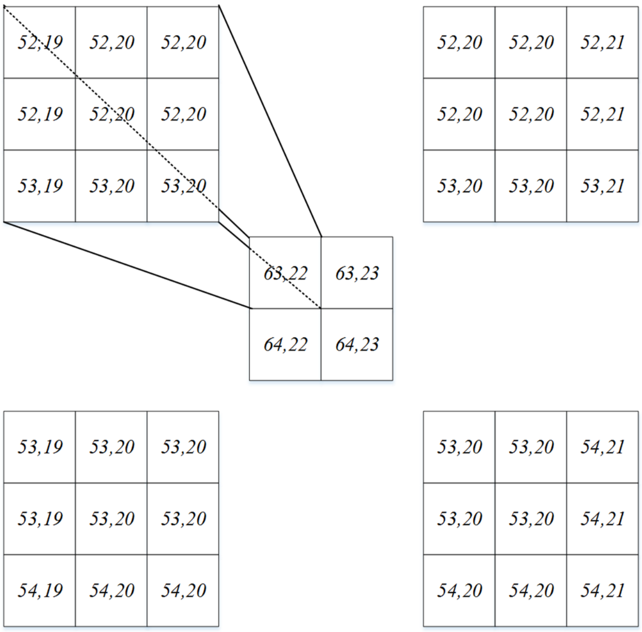
where (u, v) represents a pixel in the LR image, (i, j) represents a pixel in the HR image, and h is an arbitrary scaling factor.

The reference pixels in the LR image of a reconstructed pixel in the HR image is determined with the following mathematical expression.

$$\begin{cases} u_1 = \varphi(u) \\ v_1 = \varphi(v) \end{cases}, \begin{cases} u_2 = \rho(u) \\ v_2 = \rho(v) \end{cases}, \begin{cases} u_3 = u_1 + 1 \\ v_3 = v_1 + 1 \end{cases} \tag{6}$$

where $\varphi(\cdot)$ is a function that tends toward negative infinity, and $\rho(\cdot)$ is a function that rounds to the nearest decimal or integer.

When r is 1, the product of the LR pixels and SGAR model parameter vector ϕ is regard as the value of interpolated pixel. For the convenience of understanding, taking 1.25× interpolation as an example, the coordinates of HR pixels are *(63,22), (63,23), (64,22), (64,23),* the LR reference pixels coordinate at (*52, 19*), (*52,20*), (*53,19*) and (*53,20*) are used to interpolate the HR pixel at (63, 22), other HR pixels interpolation schemes are shown in Figure 1:



52,19	52,20	52,20
52,19	52,20	52,20
53,19	53,20	53,20

52,20	52,20	52,21
52,20	52,20	52,21
53,20	53,20	53,21

63,22	63,23
64,22	64,23

53,19	53,20	53,20
53,19	53,20	53,20
54,19	54,20	54,20

53,20	53,20	54,21
53,20	53,20	54,21
54,20	54,20	54,21

Figure 1. Pixel arrangement in interpolation process.

When r is 2, the parameters of SGAR model are estimated by modified samples. For each sample, it consist of an anchor pixel at the center of 3 × 3 image block, 8 LR reference pixels and 36 parameters generated by these pixels. Then the LR pixels used in the interpolation process share the same arrangement with the samples used in parameter estimation process.

3.3. Gradient guided model extension

The gradient guided method [20,37] was employed to adaptively extend the parameters estimation window to increase the training dataset and ultimately improve the interpolation accuracy. The gradient angles were calculated by the Scharr operator, which is formulated as follows.

$$Scharr_x = \begin{bmatrix} -3 & 0 & 3 \\ -10 & 0 & 10 \\ -3 & 0 & 3 \end{bmatrix}, Scharr_y = \left(Scharr_x\right)' = \begin{bmatrix} -3 & -10 & -3 \\ 0 & 0 & 0 \\ 3 & 10 & 3 \end{bmatrix} \tag{7}$$

where Gx and Gy are the gradients of the x and y directions, respectively, according to the Scharr operator.

$$\begin{cases} G_x = \left[3f(x-1,y+1)+10f(x,y+1)+3f(x+1,y+1)\right]- \\ \qquad \left[3f(x-1,y-1)+10f(x,y-1)+3f(x+1,y-1)\right] \\ G_y = \left[3f(x-1,y-1)+10f(x-1,y)+3f(x-1,y+1)\right]- \\ \qquad \left[3f(x+1,y-1)+10f(x+1,y)+3f(x+1,y+1)\right] \end{cases} \tag{8}$$

Therefore, the gradient angle is

$$\theta = \arctan\left(\frac{G_y}{G_x}\right)\cdot\frac{180}{\pi} \tag{9}$$

The gradient angle space was divided into eight regions according to an interval of 45°, and the eight regions were placed into four groups.

$$\begin{cases} [-22.5, 22.5) \cup [157.5, 202.5) \\ [22.5, 67.5) \cup [202.5, 247.5) \\ [67.5, 112.5) \cup [247.5, 292.5) \\ [112.5, 157.5) \cup [292.5, 337.5) \end{cases} \tag{10}$$

According to the gradient direction of each pixel in the LR image, the image windows involved in modeling the image with the SGAR model would be extended in four directions as shown in Figure 2.

The gradient directions of the n pixels were grouped according to Eq (10), and the frequency of each group was counted. The extension direction was subsequently determined according to the maximum of the four frequencies. The number of fitting samples is determined by the gradient based extension method. In fact, 15 samples were used in the horizontal and vertical directions, and 19 samples are used in the other two directions.

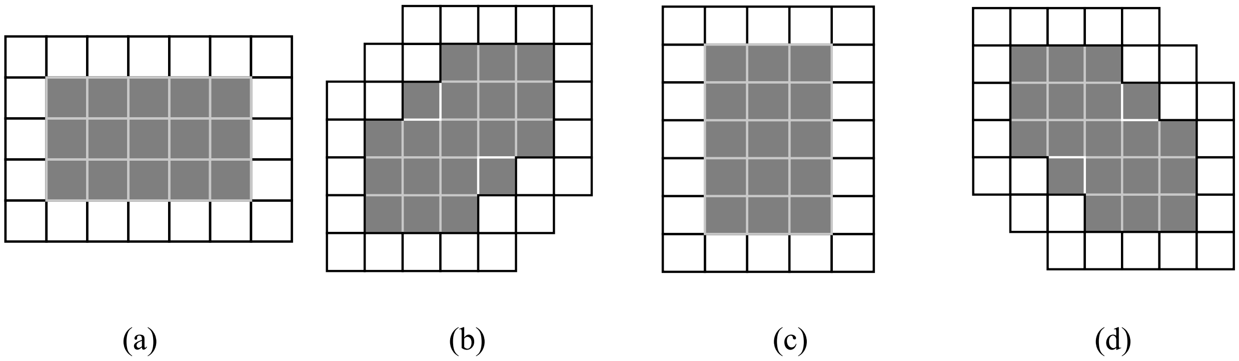


Figure 2. Adaptive window extension method based on gradient guidance. (a) Image windows extended horizontally. (b) Image windows extended in the direction of 45°. (c) Image windows extended vertically. (d) Image windows extended in the direction of 135°.

3.4. Regularized SGAR model-based on elastic network

The earlier works suggested that the patterns of the AR models containing only a linear relationship between pixels are sufficient. The least-square (LS) method [18], which is highly sensitive to outliers, was used to solve the parameters of the interpolation model with cross-direction constraints. To address the noise-sensitive problem of the ordinary LS, some methods were proposed [38]. Among them, regularization is a common method. The l1-norm and l2-norm regularization terms were added to the objective function by Liu et al. to enhance the stability of the LS solution. The l2-norm penalty term [27] (i.e., ridge regression) was used to regularize the objective function. Weighted ridge regression (WRR) [19] was adopted to restrain the expansion of variance and thus achieve more reliable estimations by modulating weights into the regression to evaluate the reliability of each sample. In contrast to the above methods, the products of the pixels were employed to increase the description capacity of the SGAR model to suit various digital images, and the overfitting problem was noted.

The SGAR model adopted the products of pixels as the new features, so the descriptive power of the model was improved compared with some interpolation methods that consider only simple linear relations [15]. Therefore, the model was especially effective for rebuilding the local nonlinear relationship of image windows [22,23,26,35]. However, it may cause overfitting problems because of the higher degree of freedom [26]. To address this problem, the regularization method could be used to reduce the freedom of models and keep the complexity of the object function in a reasonable state. Ridge regression [23,39], Lasso regression [40] and Elastic Network [41] are the three most commonly used methods.

Therefore, the mean square error (MSE) was calculated to represent the intensity differences between the reference pixels and the predicted pixels. The parameters of the SGAR model will obtained by minimizing the loss function. The loss function of Elastic Network is shown as:

$$L\left(\boldsymbol{X},\boldsymbol{\theta},\alpha,t\right)=\frac{1}{m}\sum_{i=1}^{m}\left(\boldsymbol{\theta}^{T}\cdot\boldsymbol{X}^{(i)}-\boldsymbol{y}^{(i)}\right)^{2}+t\alpha\sum_{j=1}^{n}\left|\theta_{j}\right|+\frac{1-t}{2}\alpha\sum_{j=1}^{n}\theta_{j}^{2} \tag{11}$$

where m is the number of the samples in the dataset, n is the number of fit parameters, $\boldsymbol{X}(i)$ is the vector combined with all features of sample i, $\boldsymbol{\theta}$ is the vector of the model parameter, $\boldsymbol{y}(i)$ is the anchor pixel of the region, and is the original value of LR pixel at the center of 3 × 3 image block, α is the hyper-parameter that dominates the regularization level of the model, and t is the hyper-parameter that dominates the mixing ratio. In the experiment, the value of t was manually set to 0.5.

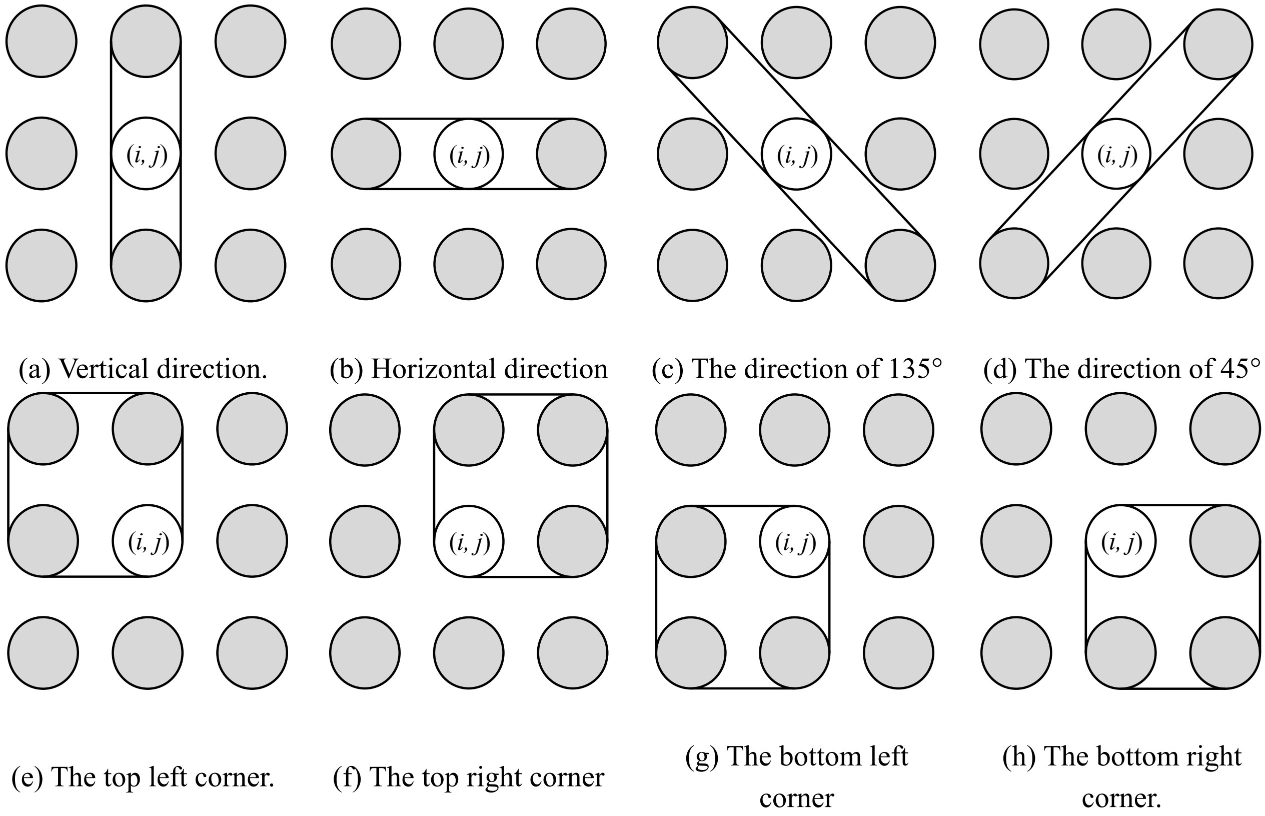
3.5. Regularized SGAR model-based on elastic network

The SGAR model learns the structural relationship between pixels and their neighbors in the LR image to recover as much missing information as possible during down-sampling [42]. However, we know that it is very difficult to recover HR images in high-frequency areas where the gray level of the image changes rapidly, such as textures and gradients. Only a relatively small part of the missing information can be restored. Compared with low-frequency areas, modeling of image windows with the SGAR model in these areas is more likely to encounter the problem of insufficient generalization ability, which makes interpolation stability difficult to control. Therefore, in these regions, a strategy that updates the interpolated pixels by minimizing the local variation of the second derivative of the image was adopted. By transforming the two-dimensional image plane into three-dimensional space, the surface optimization methods [43] were employed to solve the problem of artifacts.

List 1. The process of iterative curvature method.

1:	Input $U(i,j)$
2:	Calculate The second-order derivatives of $U(i,j)$ in 8-directions $d_1 = (U(i-1,j)+U(i+1,j))/2 - U(i,j)$ $d_2 = (U(i,j-1)+U(i,j+1))/2 - U(i,j)$ $d_3 = (U(i-1,j-1)+U(i+1,j+1))/2 - U(i,j)$ $d_4 = (U(i-1,j+1)+U(i+1,j-1))/2 - U(i,j)$ $d_5 = U(i-1,j)+U(i,j-1)-U(i-1,j-1)-U(i,j)$ $d_6 = U(i-1,j)+U(i,j+1)-U(i+1,j-1)-U(i,j)$ $d_7 = U(i,j-1)+U(i+1,j)-U(i+1,j-1)-U(i,j)$ $d_8 = U(i,j+1)+U(i+1,j)-U(i+1,j+1)-U(i,j)$
3:	If $d_i \ge Threshold$, then $d_i \in \boldsymbol{A}$ which is a set.
4:	$d_m = \min\left(\left\|d_k\right\|\right)$, where $d_k \in \boldsymbol{A}$, $k = 1,2,\ldots m$, m is the element numbers of $\boldsymbol{A}$.
5:	Update $U(i,j)$: $U(i,j) = U(i,j) + dm$
6:	Repeat steps 1–5 until the stopping rules: 1) Set A is an empty set or 2) the maximum iterations were reached.

An analogous filtering-based method proposed by Gong et al. [43] was explored to reduce the regularization part of the variational energy while guaranteeing non-increasing total energy. That is, the iterative curvature method was modulated into SGAR mode. The second-order derivatives of $p(i, j)$ on the HR image were calculated in eight different directions, as shown in Figure 3, and were used as an approximation of the curvatures. The eight second-order derivatives were thresholded, and a set consisting of the thresholded second-order derivatives was obtained. $P(i, j)$ was updated according to the absolute minimum of the elements of the set. The second-order derivatives of the digital image window were calculated repeatedly until the curvature of $p(i, j)$ in all directions was below the threshold or the maximum iterations were reached. The curvatures around $p(i, j)$ gradually adapted to the directions that shared a similar curvature change tendency. Thus, the accuracy of the interpolation results obtained in the first step was improved with the following algorithm process in List 1.



(a) Vertical direction. (b) Horizontal direction (c) The direction of 135° (d) The direction of 45°

(e) The top left corner. (f) The top right corner (g) The bottom left corner (h) The bottom right corner.

Figure 3. Curvature descriptions of 3 × 3 region in 8 different directions.

4. Results and discussions

Experiments were carried out on a dataset provided by ICBI [29] with different magnification factors to systematically verify the performance of the proposed method in terms of objective measurements, subjective visual effect and computational cost. For thoroughness and fairness, the chosen dataset contains images with different resolutions and various objects—animals, flowers and buildings—with a wide range of color and natural textures, which include typical and unique digital images for interpolation research.

Table 1. Comparison of 1.7× magnification interpolation results with four non-integer methods.

Image	Scale	Criterion	AREA	Nearest	Bicubic	Bilinear	Proposed
Zebra	1.7×	MSE	81.207	228.335	75.005	77.776	**59.349**
		PSNR	29.035	24.545	29.380	29.222	**30.397**
		SSIM	0.924	0.864	**0.929**	0.920	0.926
		FSIM	0.9817	0.9512	0.9791	0.9772	**0.9877**
		CC	0.9935	0.9810	0.9939	0.9939	**0.9952**
Bench	1.7×	MSE	104.015	231.486	91.697	107.383	**79.586**
		PSNR	27.960	24.486	28.507	27.821	**29.122**
		SSIM	0.920	0.864	**0.928**	0.913	0.925
		FSIM	0.9760	0.9499	0.9761	0.9705	**0.9847**
		CC	0.9891	0.9749	0.9903	0.9889	**0.9916**
Bird	1.7×	MSE	71.808	156.785	**61.920**	74.543	63.665
		PSNR	29.569	26.178	**30.212**	29.407	30.092
		SSIM	0.914	0.847	**0.920**	0.907	0.916
		FSIM	0.9802	0.9511	0.9814	0.9753	**0.9868**
		CC	0.9921	0.9824	**0.9932**	0.9919	0.9929
Clock	1.7×	MSE	44.927	104.802	42.651	45.213	**36.236**
		PSNR	31.606	27.927	31.831	31.578	**32.539**
		SSIM	0.961	0.932	**0.964**	0.959	0.962
		FSIM	0.9875	0.9693	0.9862	0.9841	**0.9921**
		CC	0.9950	0.9880	0.9952	0.9950	**0.9959**
Butterfly 1	1.7×	MSE	97.058	249.318	87.326	95.717	**74.768**
		PSNR	28.260	24.163	28.719	28.321	**29.394**
		SSIM	0.939	0.875	0.946	0.936	**0.947**
		FSIM	0.9849	0.9600	0.9836	0.9806	**0.9904**
		CC	0.9929	0.9813	0.9936	0.9932	**0.9945**
Bee	1.7×	MSE	37.375	106.105	29.757	36.878	**26.581**
		PSNR	32.405	27.873	33.395	32.463	**33.885**
		SSIM	0.964	0.916	**0.970**	0.964	**0.970**
		FSIM	0.9916	0.9696	0.9907	0.9895	**0.9952**
		CC	0.9961	0.9886	0.9968	0.9962	**0.9972**
Carrousel	1.7×	MSE	77.964	183.253	72.883	78.882	**62.649**
		PSNR	29.212	25.500	29.505	29.161	**30.162**
		SSIM	0.952	0.913	**0.955**	0.949	0.953
		FSIM	0.9823	0.9592	0.9809	0.9786	**0.9882**
		CC	0.9948	0.9877	0.9951	0.9948	**0.9958**
Sunflower	1.7×	MSE	27.130	75.759	23.167	26.798	**18.957**
		PSNR	33.796	29.336	34.482	33.850	**35.353**
		SSIM	0.963	0.932	**0.967**	0.960	0.964
		FSIM	0.9907	0.9763	0.9906	0.9883	**0.9945**
		CC	0.9980	0.9945	0.9983	0.9981	**0.9986**

Continued on next page

Image	Scale	Criterion	AREA	Nearest	Bicubic	Bilinear	Proposed
Puppet	1.7×	MSE	13.923	34.978	13.314	13.920	**10.774**
		PSNR	36.694	32.693	36.888	36.694	**37.807**
		SSIM	0.967	0.938	**0.970**	0.966	**0.970**
		FSIM	0.9932	0.9813	0.9913	0.9915	**0.9958**
		CC	0.9975	0.9937	0.9976	0.9975	**0.9981**
Eagle	1.7×	MSE	38.275	83.231	39.204	40.307	**33.764**
		PSNR	32.302	28.928	32.197	32.077	**32.846**
		SSIM	0.945	0.905	**0.946**	0.942	0.945
		FSIM	0.9907	0.9746	0.9880	0.9885	**0.9932**
		CC	0.9958	0.9908	0.9957	0.9956	**0.9963**
Sheep	1.7×	MSE	34.092	73.470	**31.319**	35.503	32.320
		PSNR	32.804	29.470	**33.173**	32.628	33.036
		SSIM	0.939	0.885	**0.947**	0.934	0.941
		FSIM	0.9909	0.9721	0.9903	0.9885	**0.9941**
		CC	0.9963	0.9919	**0.9966**	0.9961	0.9965
Giraffe	1.7×	MSE	20.370	62.295	18.833	18.987	**14.790**
		PSNR	35.041	30.186	35.382	35.346	**36.431**
		SSIM	0.969	0.929	**0.973**	0.969	0.972
		FSIM	0.9927	0.9719	0.9905	0.9908	**0.9957**
		CC	0.9977	0.9930	0.9979	0.9979	**0.9983**
Tiger	1.7×	MSE	102.932	201.494	**101.077**	107.526	102.251
		PSNR	28.005	25.088	**28.084**	27.816	28.034
		SSIM	0.878	0.806	**0.883**	0.867	0.874
		FSIM	0.9852	0.9578	0.9816	0.9820	**0.9892**
		CC	0.9851	0.9702	**0.9853**	0.9846	0.9851
Cat	1.7×	MSE	33.662	62.693	**31.593**	35.278	34.880
		PSNR	32.859	30.159	**33.135**	32.656	32.705
		SSIM	0.917	0.867	**0.923**	0.910	0.912
		FSIM	0.9886	0.9687	0.9883	0.9861	**0.9914**
		CC	0.9851	0.9702	**0.9853**	0.9846	0.9851
Guitar	1.7×	MSE	84.518	158.135	68.741	88.767	**66.554**
		PSNR	28.861	26.141	29.759	28.648	**29.899**
		SSIM	0.932	0.896	**0.941**	0.925	0.938
		FSIM	0.9833	0.9659	0.9854	0.9784	**0.9914**
		CC	0.9927	0.9863	**0.9931**	0.9924	0.9924
Dragonfly	1.7×	MSE	22.511	52.164	20.837	22.784	**18.960**
		PSNR	34.607	30.957	34.942	34.554	**35.353**
		SSIM	0.968	0.942	**0.971**	0.967	0.969
		FSIM	0.9931	0.9804	0.9917	0.9913	**0.9957**
		CC	0.9924	0.9877	0.9947	0.9932	**0.9949**
Church	1.7×	MSE	62.413	139.430	57.258	64.413	**55.518**
		PSNR	30.178	26.687	30.552	30.041	**30.687**
		SSIM	0.935	0.879	**0.940**	0.931	0.938
		FSIM	0.9784	0.9458	0.9759	0.9729	**0.9848**
		CC	0.9970	0.9930	0.9972	0.9970	**0.9975**

Continued on next page

Image	Scale	Criterion	AREA	Nearest	Bicubic	Bilinear	Proposed
Tower	1.7×	MSE	37.601	79.741	35.826	39.162	**35.180**
		PSNR	32.379	29.114	32.589	32.202	**32.668**
		SSIM	0.943	0.906	**0.946**	0.938	0.940
		FSIM	0.9793	0.9497	0.9765	0.9750	**0.9844**
		CC	0.9927	0.9834	0.9933	0.9925	**0.9935**
Butterfly 2	1.7×	MSE	66.696	178.208	66.707	66.773	**48.798**
		PSNR	29.890	25.622	29.889	29.885	**31.247**
		SSIM	0.961	0.919	0.965	0.959	**0.966**
		FSIM	0.9855	0.9561	0.9802	0.9818	**0.9910**
		CC	0.9953	0.9900	0.9955	0.9951	**0.9956**
House	1.7×	MSE	74.397	146.167	**68.578**	78.266	74.777
		PSNR	29.415	26.482	**29.769**	29.195	29.393
		SSIM	0.913	0.866	**0.919**	0.905	0.904
		FSIM	0.9757	0.9447	0.9746	0.9695	**0.9811**
		CC	0.9908	0.9745	0.9906	0.9910	**0.9932**
Lion	1.7×	MSE	61.003	116.098	**58.801**	64.213	61.114
		PSNR	30.277	27.483	**30.437**	30.055	30.269
		SSIM	0.884	0.812	**0.889**	0.872	0.880
		FSIM	0.9849	0.9613	0.9835	0.9809	**0.9894**
		CC	0.9892	0.9785	**0.9900**	0.9888	0.9891
Stained Glass	1.7×	MSE	245.082	492.326	**221.710**	254.356	226.619
		PSNR	24.238	21.208	**24.673**	24.076	24.578
		SSIM	0.871	0.797	**0.879**	0.858	0.867
		FSIM	0.9570	0.9198	0.9591	0.9467	**0.9726**
		CC	0.9907	0.9821	**0.9910**	0.9903	0.9907
Colorful	1.7×	MSE	66.784	150.067	63.122	67.522	**56.865**
		PSNR	29.884	26.368	30.129	29.836	**30.582**
		SSIM	0.923	0.873	**0.927**	0.919	0.922
		FSIM	0.9880	0.9702	0.9869	0.9853	**0.9917**
		CC	0.9809	0.9602	**0.9826**	0.9805	0.9821
Newspaper	1.7×	MSE	89.945	181.939	82.357	94.110	**79.896**
		PSNR	28.591	25.532	28.974	28.394	**29.106**
		SSIM	0.917	0.868	**0.924**	0.911	0.920
		FSIM	0.9827	0.9579	0.9804	0.9782	**0.9885**
		CC	0.9881	0.9754	0.9891	0.9877	**0.9894**
Wheel	1.7×	MSE	176.331	335.015	**180.833**	187.125	191.301
		PSNR	25.668	22.880	**25.558**	25.409	25.314
		SSIM	0.827	0.738	**0.826**	0.807	0.802
		FSIM	0.9701	0.9253	0.9611	0.9648	**0.9743**
		CC	0.9908	0.9745	0.9906	0.9910	**0.9932**

Comparisons of non-integer factors are studied with bicubic interpolation, bilinear interpolation, nearest neighbor interpolation and AREA interpolation. These methods are well-established in Open Source Computer Vision Library (OpenCV). The mean square error (MSE), peak signal to noise ratio (PSNR), structural similarity index (SSIM), feather similarity index (FSIM) and Correlation coefficient (CC) were used to evaluate the result of the interpolation algorithm. The experiments are conducted in non-integer interpolation factors to substantiate the performance of proposed method in arbitrary scaling. In 1.7× and 2.2× enlargement, images with 512 × 512 pixels are resized into 435 × 435 pixels and 563 × 563 pixels with bicubic method, and used as the reference for computing the MSE, PSNR and SSIM index. The images with 256 × 256 pixels are enlarged to the same size by proposed method and other four conventional non-integer interpolation methods. The experimental results are tabulated in Tables 1 and 2.

Table 2. Comparison of 2.2× magnification interpolation results with four non-integer methods.

Image	Scale	Criterion	AREA	Nearest	Bicubic	Bilinear	Proposed
Zebra	2.2×	MSE	88.816	206.365	50.628	76.558	**46.788**
		PSNR	28.646	24.984	31.087	29.291	**31.429**
		SSIM	0.903	0.848	0.922	0.902	**0.925**
		FSIM	0.8965	0.8323	**0.8975**	0.8956	**0.8975**
		CC	0.94262	0.97264	0.97502	0.97506	**0.97513**
Bench	2.2×	MSE	109.459	208.750	74.247	106.305	**64.341**
		PSNR	27.738	24.935	29.424	27.865	**30.046**
		SSIM	0.906	0.858	0.923	0.901	**0.928**
		FSIM	0.8993	0.8463	0.9026	0.8963	**0.9030**
		CC	0.93808	0.96811	0.97060	0.97168	**0.97178**
Bird	2.2×	MSE	75.389	142.926	53.552	74.440	**49.275**
		PSNR	29.358	26.580	30.843	29.413	**31.205**
		SSIM	0.898	0.839	0.921	0.893	**0.926**
		FSIM	0.8986	0.8398	**0.9022**	0.8959	**0.9022**
		CC	0.95257	0.97605	0.97802	**0.97807**	**0.97807**
Clock	2.2×	MSE	47.199	95.372	31.825	44.876	**28.044**
		PSNR	31.391	28.337	33.103	31.611	**33.652**
		SSIM	0.956	0.931	0.965	0.955	**0.966**
		FSIM	0.9432	0.9102	0.9449	0.9420	**0.9452**
		CC	0.98293	**0.98430**	0.98424	0.98428	0.94669
Butterfly 1	2.2×	MSE	103.625	228.543	61.539	94.279	**56.611**
		PSNR	27.976	24.541	30.239	28.387	**30.602**
		SSIM	0.924	0.866	0.949	0.924	**0.951**
		FSIM	0.9154	0.8609	**0.9174**	0.9142	**0.9174**
		CC	0.97389	0.97616	0.97630	**0.97641**	0.96715
Bee	2.2×	MSE	40.711	93.276	22.502	36.456	**20.329**
		PSNR	32.034	28.433	34.609	32.513	**35.050**
		SSIM	0.954	0.912	0.972	0.958	**0.973**
		FSIM	0.9389	0.8971	**0.9398**	0.9385	0.9397
		CC	0.98425	0.98562	0.98565	**0.98569**	**0.98569**

Continued on next page

Image	Scale	Criterion	AREA	Nearest	Bicubic	Bilinear	Proposed
Carrousel	2.2×	MSE	81.653	166.234	55.864	77.604	**49.037**
		PSNR	29.011	25.924	30.660	29.232	**31.226**
		SSIM	0.943	0.909	0.954	0.942	**0.956**
		FSIM	0.9189	0.8715	**0.9202**	0.9176	0.9200
		CC	0.96606	0.98261	**0.98406**	0.98396	0.98402
Sunflower	2.2×	MSE	29.199	66.942	16.333	26.295	**14.375**
		PSNR	33.477	29.874	36.000	33.932	**36.555**
		SSIM	0.958	0.932	0.967	0.957	**0.968**
		FSIM	0.9525	0.9226	**0.9530**	0.9519	**0.9530**
		CC	0.98441	0.99236	0.99301	0.99304	**0.99305**
Puppet	2.2×	MSE	14.994	31.026	9.121	13.794	**8.177**
		PSNR	36.372	33.214	38.530	36.734	**39.005**
		SSIM	0.962	0.936	0.972	0.962	**0.973**
		FSIM	0.9582	0.9342	**0.9593**	0.9573	0.9591
		CC	0.98381	0.99165	0.99240	**0.99247**	**0.99247**
Eagle	2.2×	MSE	38.812	73.190	30.466	39.360	**26.828**
		PSNR	32.241	29.486	33.293	32.180	**33.845**
		SSIM	0.939	0.905	0.947	0.936	**0.949**
		FSIM	0.9490	0.9227	**0.9496**	0.9487	0.9494
		CC	0.97598	0.98676	**0.98786**	0.98733	0.98732
Sheep	2.2×	MSE	35.217	65.627	26.054	35.116	**24.500**
		PSNR	32.663	29.960	33.972	32.676	**34.239**
		SSIM	0.929	0.883	0.948	0.926	**0.950**
		FSIM	0.9389	0.9011	**0.9415**	0.9372	0.9410
		CC	0.97840	0.98865	**0.98950**	0.98944	0.98945
Giraffe	2.2×	MSE	22.205	53.935	12.282	18.577	**11.382**
		PSNR	34.666	30.812	37.238	35.441	**37.569**
		SSIM	0.959	0.923	0.972	0.963	**0.973**
		FSIM	0.9408	0.8983	**0.9416**	0.9405	0.9415
		CC	0.97902	0.98989	0.99078	0.99074	**0.99079**
Tiger	2.2×	MSE	104.056	182.616	89.693	105.770	**82.569**
		PSNR	27.958	25.515	28.603	27.887	**28.963**
		SSIM	0.858	0.793	0.874	0.846	**0.880**
		FSIM	0.9147	0.8711	**0.9171**	0.9134	0.9167
		CC	0.92847	0.95987	**0.96311**	0.96260	0.96287
Cat	2.2×	MSE	34.597	57.043	30.381	35.208	**28.759**
		PSNR	32.740	30.569	33.305	32.664	**33.543**
		SSIM	0.904	0.861	0.916	0.898	**0.918**
		FSIM	0.9368	0.9007	**0.9391**	0.9355	0.9385
		CC	0.96745	0.98197	0.98320	0.98325	**0.98326**

Continued on next page

Image	Scale	Criterion	AREA	Nearest	Bicubic	Bilinear	Proposed
Guitar	2.2×	MSE	87.945	147.489	62.059	88.695	**52.371**
		PSNR	28.689	26.443	30.203	28.652	**30.940**
		SSIM	0.923	0.891	0.939	0.917	**0.944**
		FSIM	0.9292	0.8952	0.9369	0.9246	**0.9376**
		CC	0.97516	0.98627	0.98703	0.98839	**0.98857**
Dragonfly	2.2×	MSE	23.496	46.874	15.687	22.488	**14.230**
		PSNR	34.421	31.421	36.175	34.611	**36.599**
		SSIM	0.963	0.940	0.971	0.963	**0.972**
		FSIM	0.9635	0.9400	**0.9648**	0.9627	0.9646
		CC	0.98166	0.99039	0.99118	0.99121	**0.99124**
Church	2.2×	MSE	64.442	126.812	46.051	63.768	**42.204**
		PSNR	30.039	27.099	31.498	30.085	**31.877**
		SSIM	0.925	0.878	0.942	0.922	**0.945**
		FSIM	0.8850	0.8210	**0.8870**	0.8844	0.8864
		CC	0.95441	0.97650	**0.97845**	0.97832	0.97838
Tower	2.2×	MSE	38.860	71.499	29.564	38.900	**26.737**
		PSNR	32.236	29.588	33.423	32.231	**33.860**
		SSIM	0.938	0.907	0.946	0.932	**0.948**
		FSIM	0.8976	0.8486	**0.8989**	0.8968	0.8985
		CC	0.97411	0.98649	0.98745	0.98758	**0.98760**
Butterfly 2	2.2×	MSE	70.169	157.232	41.024	65.577	**36.018**
		PSNR	29.669	26.165	32.000	29.963	**32.566**
		SSIM	0.951	0.914	0.967	0.952	**0.969**
		FSIM	0.9107	0.8601	**0.9119**	0.9095	**0.9119**
		CC	0.92690	0.96274	**0.96583**	0.96551	0.96565
House	2.2×	MSE	101.537	130.791	61.677	77.698	**57.183**
		PSNR	28.065	26.965	30.230	29.227	**30.558**
		SSIM	0.880	0.861	0.912	0.893	**0.913**
		FSIM	0.8872	0.8298	**0.8904**	0.8857	0.8897
		CC	0.94697	0.97134	0.97348	**0.97367**	0.97358
Lion	2.2×	MSE	62.116	104.635	51.312	63.442	**47.896**
		PSNR	30.199	27.934	31.029	30.107	**31.328**
		SSIM	0.866	0.803	0.887	0.854	**0.891**
		FSIM	0.9146	0.8718	**0.9184**	0.9121	0.9176
		CC	0.95686	0.97589	**0.97759**	0.97748	**0.97759**
Stained Glass	2.2×	MSE	253.151	445.729	198.079	252.215	**181.234**
		PSNR	24.097	21.640	25.162	24.113	**25.548**
		SSIM	0.851	0.789	0.871	0.837	**0.875**
		FSIM	0.8532	0.7783	**0.8610**	0.8473	0.8601
		CC	0.90247	0.94826	0.95220	0.95272	**0.95295**

Continued on next page

Image	Scale	Criterion	AREA	Nearest	Bicubic	Bilinear	Proposed
Colorful	2.2×	MSE	70.714	137.821	50.291	67.007	**46.159**
		PSNR	29.636	26.738	31.116	29.870	**31.488**
		SSIM	0.906	0.861	0.919	0.904	**0.922**
		FSIM	0.9347	0.8934	**0.9360**	0.9339	0.9356
		CC	0.96315	0.98112	0.98265	0.98278	**0.98282**
Newspaper	2.2×	MSE	92.886	166.308	71.003	93.358	**63.312**
		PSNR	28.451	25.922	29.618	28.429	**30.116**
		SSIM	0.904	0.859	0.919	0.899	**0.924**
		FSIM	0.9132	0.8684	0.9161	0.9105	**0.9165**
		CC	0.93967	0.96737	0.96981	0.97002	**0.97009**
Wheel	2.2×	MSE	180.859	297.585	158.770	188.328	**148.868**
		PSNR	25.557	23.395	26.123	25.382	**26.403**
		SSIM	0.799	0.725	0.815	0.776	**0.817**
		FSIM	0.8580	0.8014	**0.8615**	0.8559	0.8607
		CC	0.81563	0.89332	**0.90193**	0.90154	0.90170

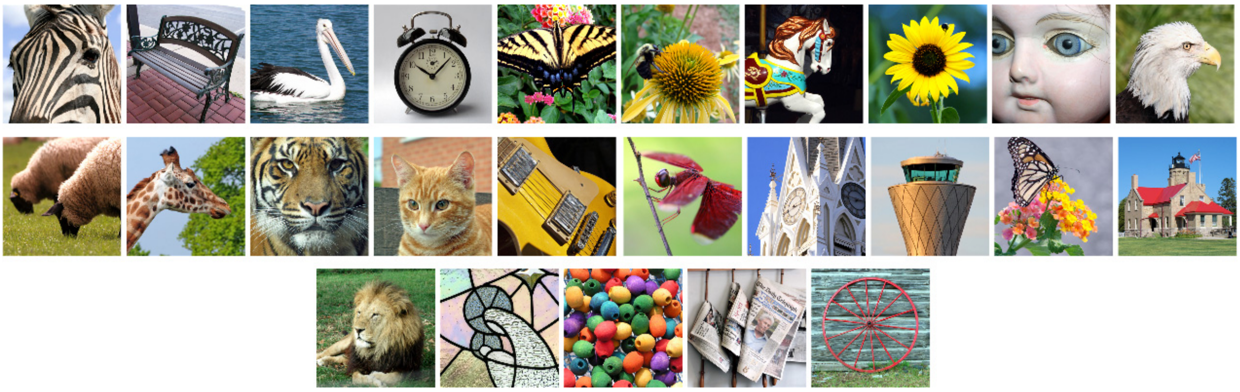


Figure 4. Images used in the experiments.

In 1.7× interpolation experiment of 25 images, our method obtains best MES index in 17 of 25 images, best PSNR index among 18 of 25 interpolated HR images, best SSIM index in 4 of 25 images, best FSIM index in 19 of 25 images and best CC index in 17 of 25 images. In 2.2× interpolation experiment, proposed method achieved best performance over MSE, PSNR and SSIM index in all of 25 images, best FSIM index in 9 of 25 images and best CC index in 15 of 25 images.

In comparison with other edge-directed methods NEDI [21], improved NEDI (i-NEDI) [34] and ICBI [29] are invited to evaluate the result of the interpolation algorithm. All of 25 images with 512 × 512 pixels are the references for computing the MSE, PSNR and SSIM, the 25 images with 256 × 256 pixels are enlarged by propose method and other interpolation methods. In order to prevent the pixel-shift in ICBI and our proposed method, the left-top 511 × 511 part of original images (and the results of other methods) is used to compute the MSE, PSNR and SSIM. The experimental results are tabulated in Table 3.

Table 3. Comparison of 2× magnification interpolation results with four excellent methods.

Image	Scale	Criterion	NEDI	Bicubic	i-NEDI	ICBI	SGAR	Proposed
Zebra	2×	MSE	151.503	148.421	152.44	**144.954**	146.616	145.153
		PSNR	26.327	26.416	26.3	**26.518**	26.469	26.513
		SSIM	0.863	0.869	0.866	**0.872**	0.871	**0.872**
		FSIM	**0.9547**	0.9510	0.9512	0.9523	0.9522	0.9519
		CC	0.9876	0.9880	0.9876	0.9874	**0.9881**	0.9878
Bench	2×	MSE	161.811	157.186	147.207	**145.943**	147.807	146.195
		PSNR	26.041	26.167	26.452	**26.489**	26.434	26.481
		SSIM	0.869	0.876	0.883	**0.883**	0.882	**0.883**
		FSIM	**0.9533**	0.9510	0.9526	0.9531	0.9532	0.9525
		CC	**0.9880**	0.9843	0.9827	0.9842	0.9847	0.9832
Bird	2×	MSE	117.786	114.073	116.195	112.192	112.748	**110.911**
		PSNR	27.42	27.559	27.479	27.631	27.610	**27.681**
		SSIM	0.857	0.867	0.867	**0.873**	**0.873**	**0.873**
		FSIM	0.9521	0.9533	0.9529	0.9532	**0.9534**	0.9530
		CC	0.9843	**0.9874**	0.9868	0.9869	**0.9874**	0.9872
Clock	2×	MSE	80.781	77.446	75.958	74.529	75.464	**74.407**
		PSNR	29.058	29.241	29.325	29.408	29.353	**29.415**
		SSIM	0.936	0.938	**0.942**	0.941	0.940	0.940
		FSIM	**0.9721**	0.9703	0.9719	0.9713	0.9712	0.9709
		CC	0.9875	0.9915	0.9908	0.9914	**0.9919**	0.9912
Butterfly 1	2×	MSE	168.61	162.884	172.659	159.534	161.906	**157.403**
		PSNR	25.862	26.012	25.759	26.102	26.038	**26.161**
		SSIM	0.884	0.891	0.886	0.893	0.891	**0.895**
		FSIM	**0.9622**	0.9604	0.9611	0.9615	0.9608	0.9609
		CC	**0.9915**	0.9880	0.9873	0.9870	0.9881	0.9877
Bee	2×	MSE	61.689	60.26	63.025	57.927	58.620	**56.663**
		PSNR	30.229	30.331	30.136	30.502	30.450	**30.598**
		SSIM	0.907	0.925	0.908	0.913	0.911	**0.929**
		FSIM	**0.9733**	0.9719	0.9711	0.9723	0.9719	0.9719
		CC	0.9881	0.9937	0.9932	0.9932	0.9932	**0.9938**
Carrousel	2×	MSE	130.863	126.369	127.243	123.344	124.221	**120.354**
		PSNR	26.963	27.114	27.084	27.22	27.189	**27.326**
		SSIM	0.913	0.920	0.920	0.920	0.919	**0.924**
		FSIM	0.9618	0.9628	**0.9635**	0.9628	0.9634	0.9634
		CC	0.9911	0.9915	0.9914	0.9916	0.9916	**0.9918**
Sunflower	2×	MSE	50.277	42.696	44.174	41.959	42.132	**38.675**
		PSNR	31.117	31.827	31.679	31.903	31.885	**32.256**
		SSIM	0.916	0.934	0.918	0.919	0.918	**0.940**
		FSIM	**0.9788**	0.9775	0.9782	0.9785	0.9782	0.9781
		CC	0.9963	0.9968	0.9968	0.9969	0.9968	**0.9971**

Continued on next page

Image	Scale	Criterion	NEDI	Bicubic	i-NEDI	ICBI	SGAR	Proposed
Puppet	2×	MSE	23.822	21.949	21.833	**20.907**	21.189	21.05
		PSNR	34.361	34.717	34.74	**34.928**	34.870	34.898
		SSIM	0.944	0.949	0.948	0.950	0.949	**0.951**
		FSIM	**0.9813**	0.9806	0.9805	0.9808	0.9810	0.9805
		CC	0.9958	0.9961	0.9961	**0.9963**	0.9963	**0.9963**
Eagle	2×	MSE	67.783	65.684	64.984	64.385	65.075	**64.216**
		PSNR	29.82	29.956	30.003	30.043	29.997	**30.054**
		SSIM	0.905	0.909	0.910	**0.911**	0.910	**0.911**
		FSIM	**0.9758**	0.9756	0.9753	0.9755	0.9757	0.9753
		CC	0.9926	0.9928	**0.9929**	**0.9929**	**0.9929**	**0.9929**
Sheep	2×	MSE	77.659	53.97	57.411	52.907	53.019	**52.839**
		PSNR	29.229	30.809	30.541	30.896	30.886	**30.901**
		SSIM	0.887	0.898	0.892	0.900	0.900	**0.902**
		FSIM	0.9711	0.9742	0.9725	0.9745	**0.9746**	0.9743
		CC	0.9914	0.9940	0.9936	**0.9941**	0.9939	**0.9941**
Giraffe	2×	MSE	39.514	37.885	41.076	37.192	37.490	**37.067**
		PSNR	32.163	32.346	31.995	32.426	32.392	**32.441**
		SSIM	0.938	0.943	0.938	0.942	0.942	**0.944**
		FSIM	0.9749	0.9752	0.9740	**0.9754**	**0.9754**	0.9752
		CC	0.9955	0.9957	0.9954	**0.9958**	**0.9958**	**0.9958**
Tiger	2×	MSE	183.369	171.484	173.456	**166.573**	167.652	167.954
		PSNR	25.498	25.789	25.739	**25.915**	25.887	25.879
		SSIM	0.782	0.801	0.799	**0.808**	0.806	**0.808**
		FSIM	0.9605	0.9616	0.9596	0.9616	**0.9617**	0.9614
		CC	0.9732	0.9749	0.9746	0.9756	**0.9760**	0.9754
Cat	2×	MSE	59.565	56.445	57.088	**55.344**	55.440	55.431
		PSNR	30.381	30.615	30.565	**30.7**	30.693	30.693
		SSIM	0.852	0.864	0.861	0.866	0.866	**0.868**
		FSIM	0.9704	0.9717	0.9702	0.9717	**0.9720**	0.9715
		CC	0.9870	0.9877	0.9875	0.9879	**0.9883**	0.9879
Guitar	2×	MSE	123.978	108.122	107.388	100.344	101.682	**98.526**
		PSNR	27.197	27.792	27.821	28.116	28.058	**28.195**
		SSIM	0.883	0.897	0.895	0.898	0.897	**0.906**
		FSIM	0.9627	0.9662	0.9672	**0.9691**	0.9686	0.9686
		CC	0.9904	0.9916	0.9917	0.9922	**0.9925**	0.9924
Dragonfly	2×	MSE	33.055	31.859	32.668	30.377	30.714	**30.02**
		PSNR	32.938	33.098	32.99	33.305	33.257	**33.357**
		SSIM	0.950	0.953	0.951	0.953	0.952	**0.955**
		FSIM	0.9828	0.9827	0.9825	0.9832	**0.9833**	0.9828
		CC	0.9955	0.9957	0.9956	**0.9959**	0.9954	**0.9959**
Church	2×	MSE	107.069	103.032	102.779	**99.755**	100.272	100.064
		PSNR	27.834	28.001	28.012	**28.141**	28.119	28.128
		SSIM	0.886	0.890	0.895	**0.897**	0.896	0.895
		FSIM	**0.9471**	0.9446	0.9445	0.9449	0.9446	0.9446
		CC	0.9873	0.9878	0.9878	**0.9882**	0.9881	0.9881

Continued on next page

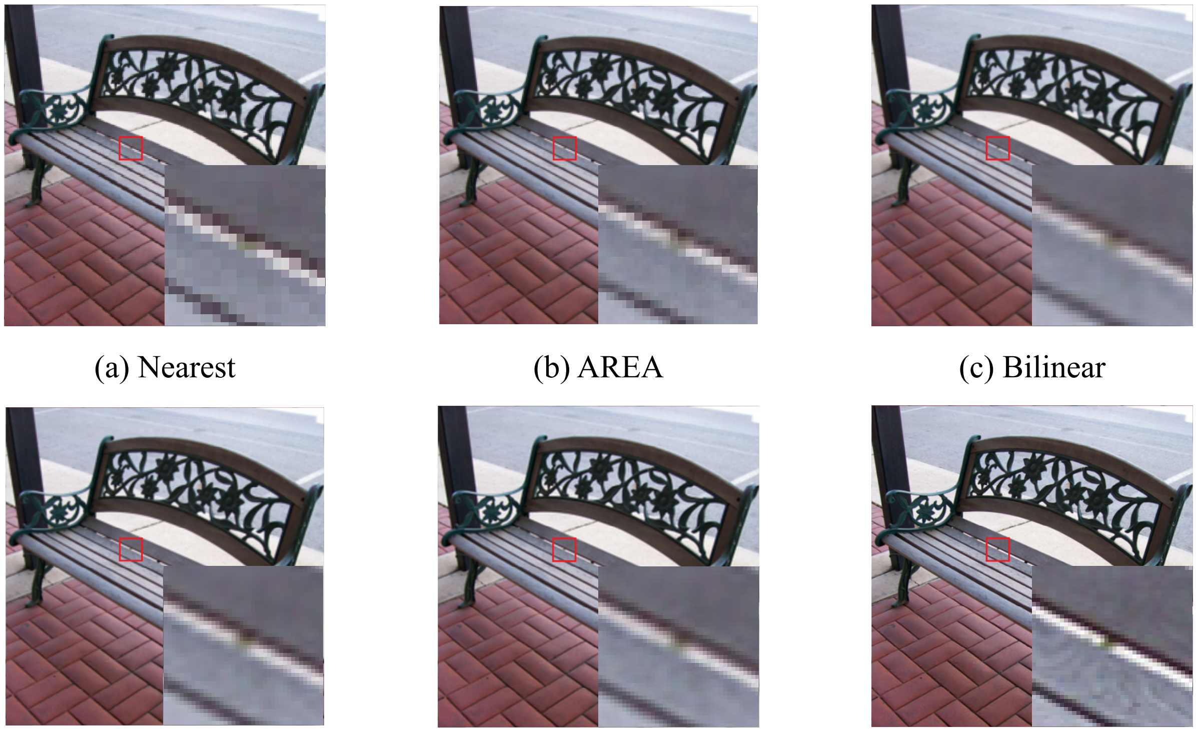
Image	Scale	Criterion	NEDI	Bicubic	i-NEDI	ICBI	SGAR	Proposed
Tower	2×	MSE	65.253	62.152	60.207	59.533	60.026	**59.289**
		PSNR	29.985	30.196	30.334	30.383	30.347	**30.401**
		SSIM	0.899	0.903	0.907	0.907	0.906	**0.908**
		FSIM	**0.9490**	0.9487	0.9481	0.9479	0.9479	0.9475
		CC	0.9918	0.9922	0.9924	0.9925	**0.9928**	0.9925
Butterfly 2	2×	MSE	149.758	110.03	112.736	106.043	108.439	**105.334**
		PSNR	26.377	27.716	27.61	27.876	27.779	**27.905**
		SSIM	0.924	0.930	0.930	**0.934**	0.933	**0.934**
		FSIM	0.9565	0.9586	0.9581	**0.9592**	0.9588	0.9588
		CC	0.9783	0.9841	0.9837	0.9846	0.9841	**0.9848**
House	2×	MSE	119.413	113.792	118.699	112.433	113.106	**110.616**
		PSNR	27.36	27.57	27.386	27.622	27.596	**27.693**
		SSIM	0.852	0.861	0.855	0.863	0.862	**0.865**
		FSIM	0.9459	**0.9462**	0.9445	0.9461	**0.9462**	0.9457
		CC	0.9824	0.9833	0.9825	0.9834	0.9832	**0.9837**
Lion	2×	MSE	106.518	99.932	105.505	99.009	99.477	**98.498**
		PSNR	27.857	28.134	27.898	28.174	28.154	**28.197**
		SSIM	0.790	0.809	0.799	0.812	0.811	**0.814**
		FSIM	0.9601	0.9625	0.9606	0.9627	**0.9629**	0.9623
		CC	0.9836	0.9846	0.9838	0.9848	**0.9855**	0.9848
Stained Glass	2×	MSE	400.909	387.65	387.415	379.12	378.228	**377.372**
		PSNR	22.1	22.246	22.249	22.343	22.353	**22.363**
		SSIM	0.787	0.798	0.802	**0.806**	0.805	0.804
		FSIM	0.9245	0.9228	**0.9304**	0.9278	0.9270	0.9272
		CC	0.9683	0.9692	0.9691	0.9698	**0.9706**	0.9700
Colorful	2×	MSE	114.975	112.91	113.786	110.039	110.317	**108.504**
		PSNR	27.525	27.603	27.57	27.715	27.704	**27.776**
		SSIM	0.851	0.857	0.853	0.857	0.857	**0.861**
		FSIM	**0.9719**	0.9713	0.9710	0.9718	**0.9719**	0.9715
		CC	0.9893	0.9895	0.9894	0.9898	**0.9903**	0.9899
Newspaper	2×	MSE	154.087	143.778	144.271	**138.333**	139.746	138.528
		PSNR	26.253	26.554	26.539	**26.722**	26.677	26.715
		SSIM	0.860	0.871	0.872	**0.876**	0.875	**0.876**
		FSIM	0.9575	0.9587	0.9589	**0.9596**	0.9595	0.9592
		CC	0.9794	0.9815	0.9806	0.9815	**0.9821**	0.9814
Wheel	2×	MSE	286.172	271.404	277.259	266.933	267.872	**265.138**
		PSNR	23.565	23.795	23.702	23.867	23.852	**23.896**
		SSIM	0.711	0.724	0.726	**0.737**	0.736	0.734
		FSIM	**0.9357**	0.9334	0.9295	0.9316	0.9314	0.9314
		CC	0.9332	0.9371	0.9353	0.9377	0.9374	**0.9382**

In 2× interpolation experiment, proposed method together with our previous SGAR method achieved best MSE in 18 of 25 images, best PSNR in 18 of 25 images, best SSIM index in 21 of 25 images, best FSIM index in 8 of 25 images and best CC index in 21 of 25 images. According to the PNSR index, 7 of the 25 interpolated HR images obtained the best effect with the ICBI method, and the other 18 interpolated HR images obtained the best effect with the method proposed in this paper.

The ratio was as high as 72%. For these 18 images, the PSNRs of the proposed method were improved to different degrees compared to those of the other four methods. The maximum, minimum and average increments were 1.672, 0.005 and 0.267 dB, respectively, and the maximum, minimum and average relative increments were 5.79, 0.02 and 0.94%, respectively. Compared with NEDI, the maximum, minimum and average increases in PSNR were 1.672, 0.234 and 0.547 dB, respectively, and the relative maximum, minimum and average increases were 5.79, 0.78 and 1.94%, respectively. Compared with bicubic, the maximum, minimum and average increases in PSNR were 0.429, 0.063 and 0.182 dB, respectively, and the relative maximum, minimum and average increases were 1.45, 0.22 and 0.63%, respectively. Compared with i-NEDI, the maximum, minimum and average increases in PSNR were 0.577, 0.051 and 0.281 dB, respectively, and the relative maximum, minimum and average increases were 1.82, 0.17 and 0.98%, respectively. Compared with ICBI, the maximum, minimum and average increases in PSNR were 0.353, 0.005 and 0.06 dB, respectively, and the relative maximum, minimum and average increases were 1.11, 0.02 and 0.21%, respectively. According to the SSIM index, 17 of 25 images obtain the best effect with the ICBI method, compare with the second best ICBI method, the maximum, minimum and average increments were 0.02099, 0.00019 and 0.00251, respectively, and the maximum, minimum and average relative increments were 2.284, 0.021 and 0.283%, respectively. Compared with bicubic, which accommodate non-integer scaling factors, the maximum, minimum and average increases in SSIM were 0.00976, 0.00124 and 0.00461, respectively, and the relative maximum, minimum and average increases were 1.35, 0.13 and 0.52%, respectively. Two points should be specified: First, both bicubic and the proposed method can achieve non-integer image magnification. Compared with the bicubic method, the proposed method improved the interpolation effect to a certain extent in all 25 experimental images. Second, both ICBI and the proposed method use the iterative curvature method. Thus, the two methods achieved the best interpolation effect among these edge-directed methods. The difference is that the method presented in this paper uses the SGAR model to predict the unknown pixels in the first step the increasing trend is obvious.

Compared with the most popular non-integer interpolate methods and other conventional edge-directed methods, the HR images obtained by the proposed method have significantly fewer blurring effects. As is shown in Figures 5 and 6, these defects are especially obvious among non-integer methods. In the visual comparison of 2× magnification, these defects also exist in Figures 7(a) and 8(a). It can be seen that the interpolation method based on the SGAR model has a better image description ability than the interpolation method based on the B-spline theory. The isotropic low-pass filter can enhance the smoothing effect, while the SGAR model is more suitable for revealing the distribution pattern of the image. Furthermore, the interpolation method based on the SGAR model has better color fidelity than NEDI, as shown in Figure 9. As shown in Figures 7 (d) and 8 (d), the HR image interpolated by ICBI has more visual defects. To analyze the reason, the proposed method used a curvature iterative method based on the discrete features of the image to describe the curvature change of anchor pixels in more directions, which helps eliminate interpolation defects such as artifacts in edge areas and texture-rich areas.

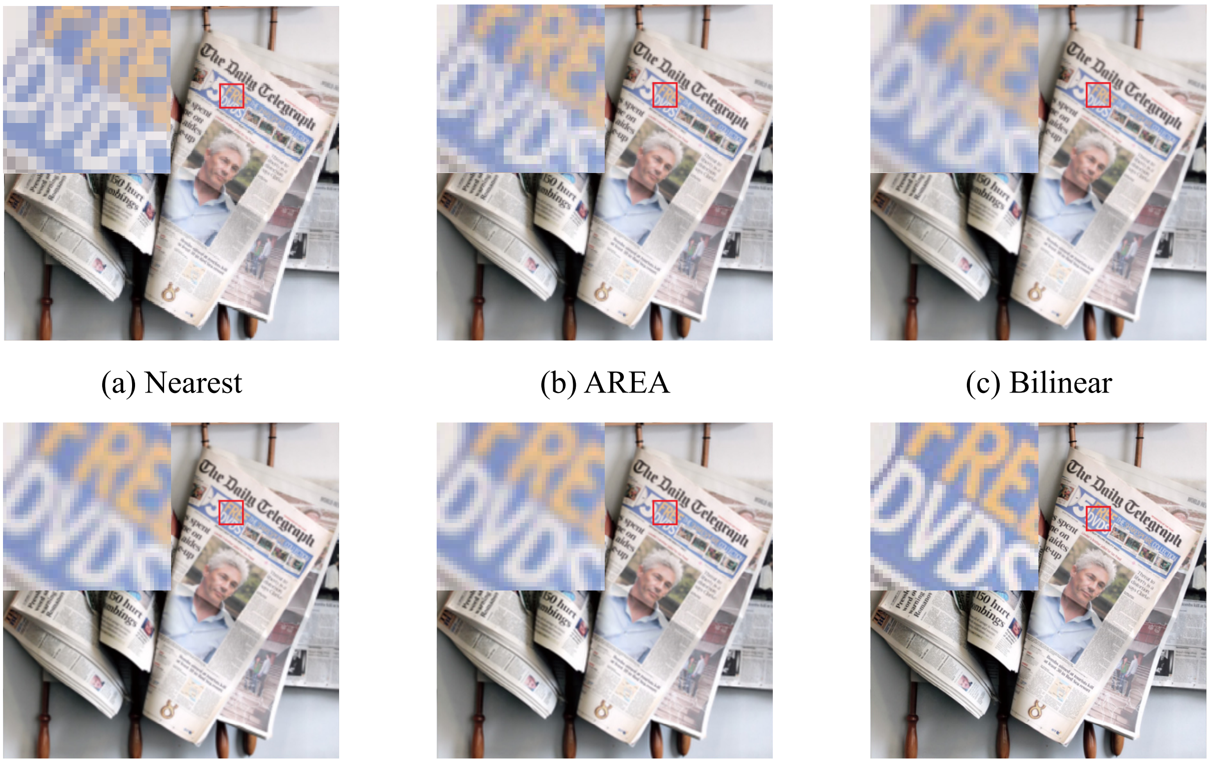
In machine vision system, especially in online visual inspection system. Due to the vibration and overheating problems, image noise will inevitably in the poor working conditions. In order to better combine with the potential application scenarios, gaussian noise with mean value of 0 and variance of 0.005 is added to gray image. The images used in this part of the experiment are shown in Figure 10.



(a) Nearest (b) AREA (c) Bilinear

(d) Bicubic (e) Proposed method (f) Original image

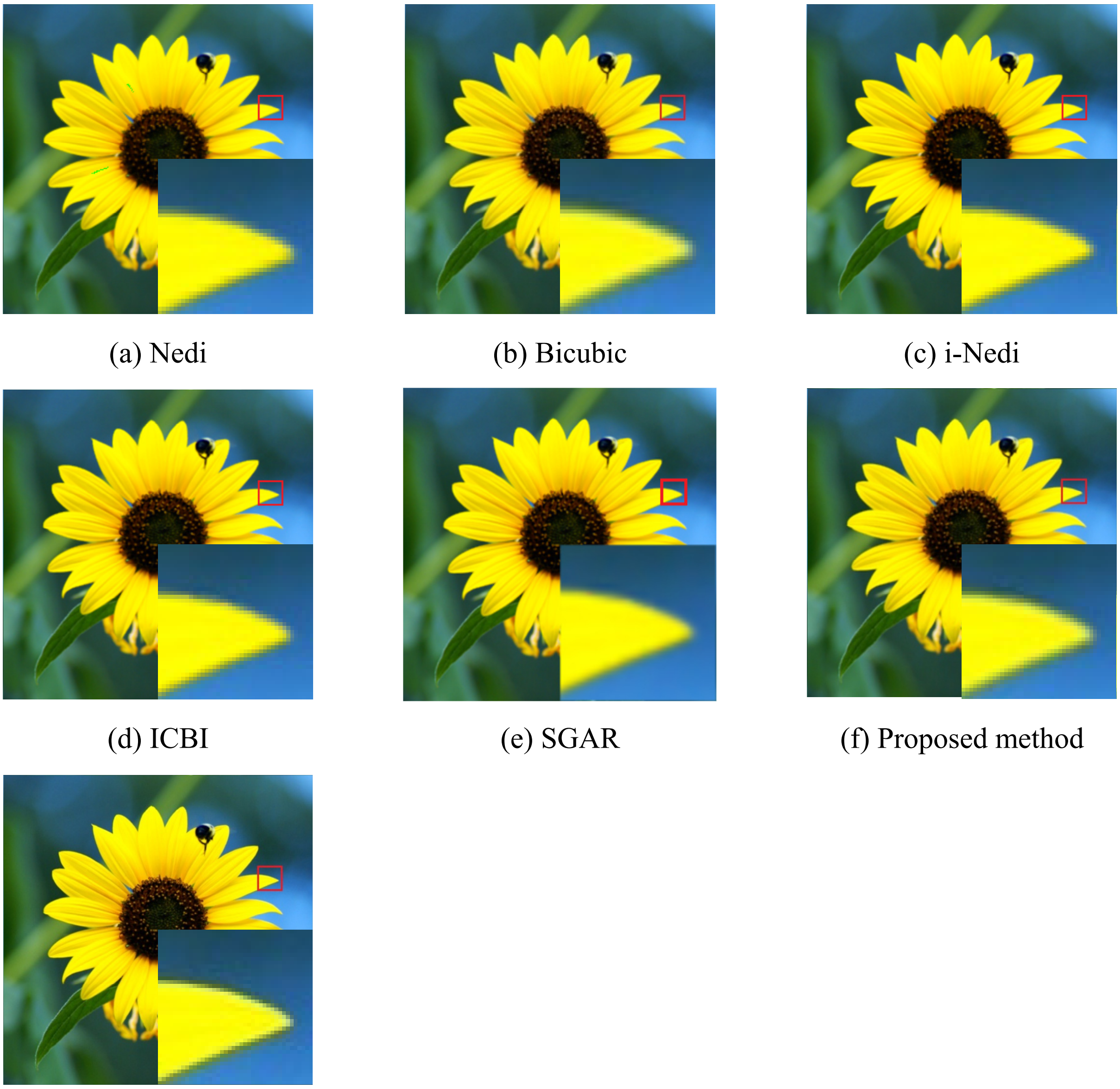
Figure 5. Visual comparisons of 2.2× magnification on the bench image.



(a) Nearest (b) AREA (c) Bilinear

(d) Bicubic (e) Proposed method (f) Original image

Figure 6. Visual comparisons of 2.2× magnification on the newspaper image.



(a) Nedi

(b) Bicubic

(c) i-Nedi

(d) ICBI

(e) SGAR

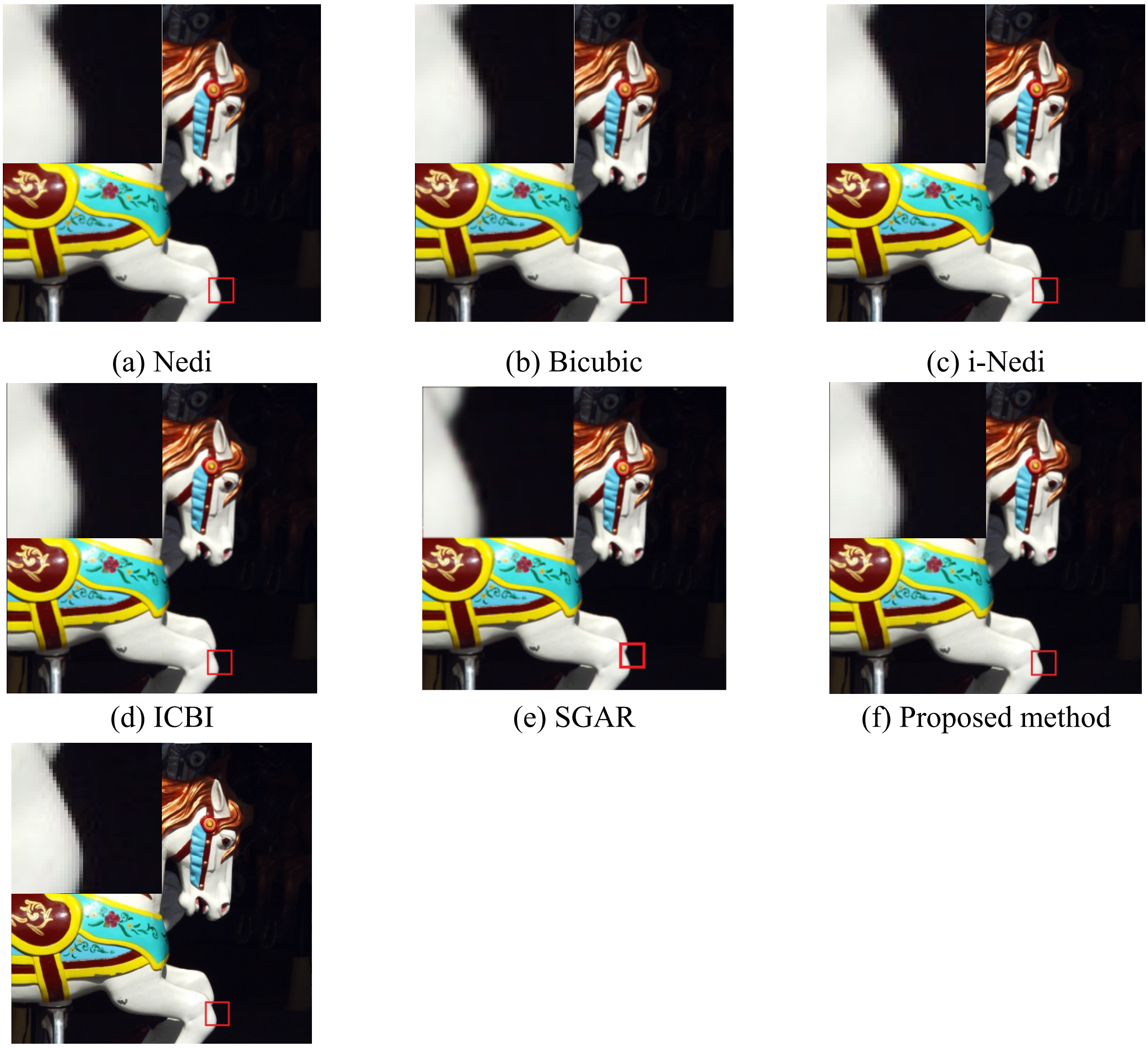
(f) Proposed method

(g) Original image

Figure 7. Visual comparisons of 2× magnification on the sunflower image.

We add the latest learning-based Meta-SR [44] to the comparison. In comparing with the traditional Bicubic interpolation method and Meta-SR [44], our previous SGAR method and the proposed method have made better progress in objective indicators on all of 5 images. Although, on some objective indicators, the edge-based method like NEDI and i-NEDI are better results than ours, these methods also magnify the impact of noise on visual quality. These artifacts are especially obvious in Figure 11.

Frankly, the naive for-loop and iterative process make the processing time relatively long, the proposed method is much slower than ICBI, let along other popular non-integer interpolation methods well-established in OpenCV. However, the processing speed can be improved with parallel threads simultaneously process on NVIDIA CUDA devices and the research will continue in future works.



(a) Nedi

(b) Bicubic

(c) i-Nedi

(d) ICBI

(e) SGAR

(f) Proposed method

(g) Original image

Figure 8. Visual comparisons of 2× magnification on the Carrousel image.

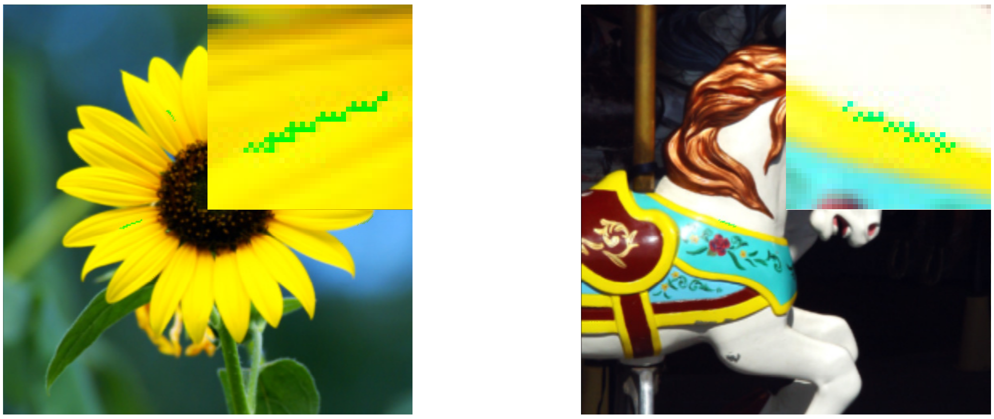


Figure 9. Color infidelity in comparing methods.

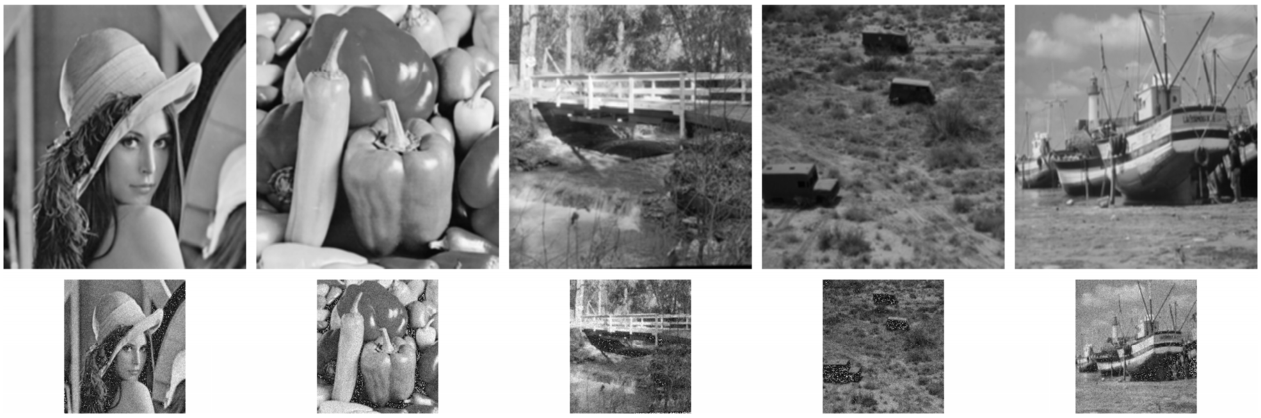
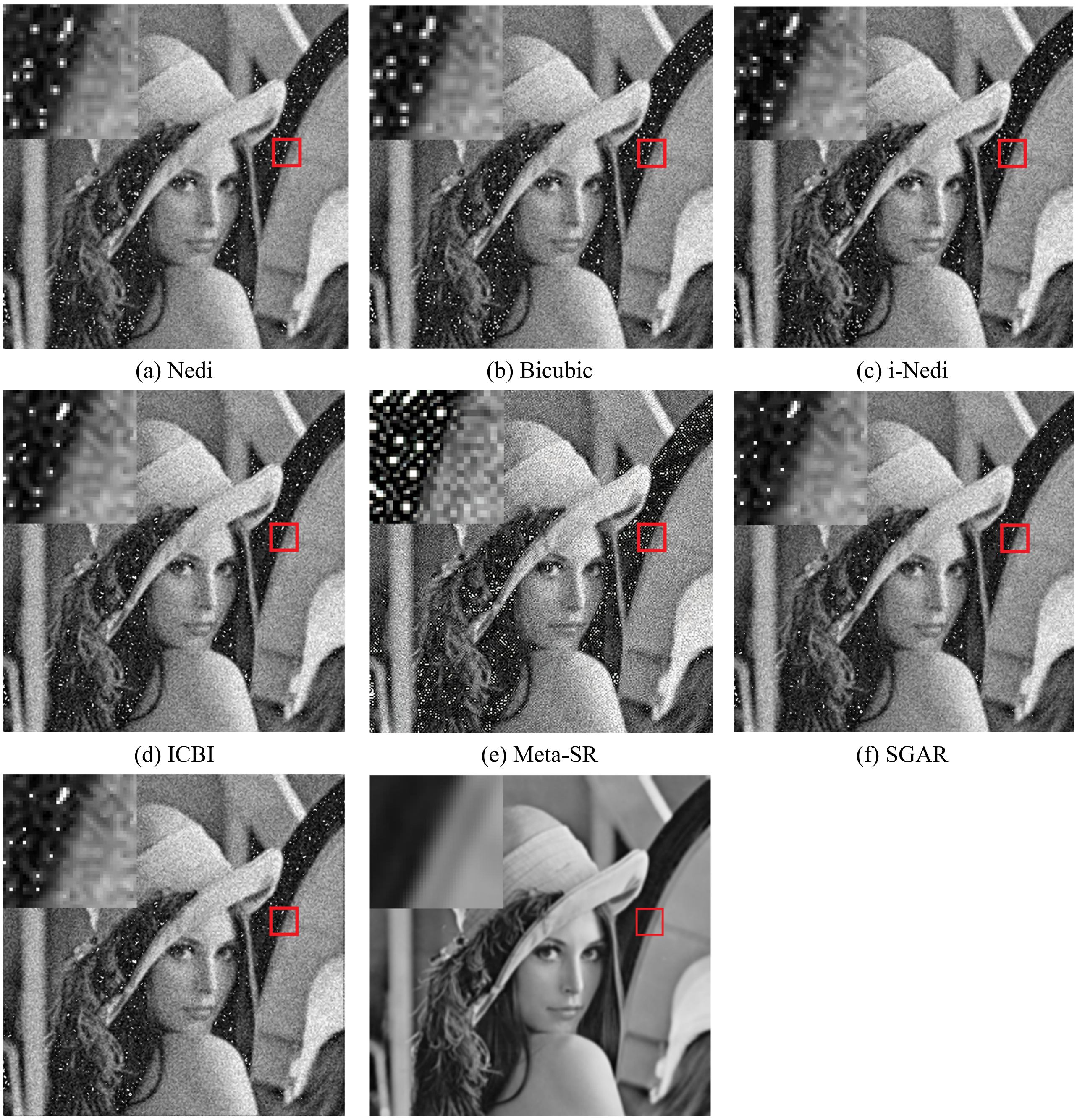


Figure 10. Noise images.

Table 4. Comparison of 2× magnification interpolation results on noise images.

Image	Scale	Criterion	NEDI	Bicubic	i-NEDI	ICBI	Meta-SR	SGAR	Proposed
Image 1	2×	MSE	444.632	461.051	407.923	416.077	1240.412	**361.525**	402.718
		PSNR	21.651	21.493	22.025	21.939	17.195	**22.549**	22.081
		SSIM	0.421	0.417	**0.428**	0.401	0.225	0.422	0.418
		FSIM	0.763	0.758	0.757	0.752	0.752	**0.768**	0.762
		CC	0.923	0.921	0.932	0.930	0.820	**0.939**	0.932
Image 2	2×	MSE	521.417	548.597	491.627	528.783	1295.417	**467.874**	531.563
		PSNR	20.959	20.738	21.214	20.898	17.007	**21.430**	20.875
		SSIM	**0.433**	0.420	0.430	0.399	0.228	0.419	0.415
		FSIM	**0.770**	0.761	0.758	0.751	0.690	0.766	0.762
		CC	0.923	0.920	0.929	0.924	0.836	**0.932**	0.923
Image 3	2×	MSE	439.341	489.277	441.469	466.365	1158.269	**327.368**	440.931
		PSNR	21.703	21.235	21.682	21.444	17.493	**22.980**	21.687
		SSIM	0.464	0.461	0.475	0.436	0.238	**0.507**	0.454
		FSIM	0.807	0.799	0.796	0.791	0.734	**0.839**	0.802
		CC	0.784	0.771	0.803	0.789	0.605	**0.939**	0.797
Image 4	2×	MSE	391.395	414.673	**383.998**	402.288	1054.442	411.358	391.838
		PSNR	22.205	21.954	**22.288**	22.085	17.901	21.989	22.200
		SSIM	0.437	0.440	0.450	0.417	0.241	**0.457**	0.434
		FSIM	0.784	0.786	0.782	0.776	0.713	**0.805**	0.786
		CC	0.898	0.894	**0.905**	0.900	0.780	0.810	0.902
Image 5	2×	MSE	358.182	368.718	354.381	376.873	900.455	349.492	**347.985**
		PSNR	22.590	22.464	22.636	22.369	18.586	22.696	**22.715**
		SSIM	0.505	0.527	**0.539**	0.504	0.304	0.437	0.521
		FSIM	0.836	0.842	0.838	0.833	0.787	0.789	**0.843**
		CC	0.931	0.930	**0.935**	0.930	0.858	0.912	**0.935**



(a) Nedi

(b) Bicubic

(c) i-Nedi

(d) ICBI

(e) Meta-SR

(f) SGAR

(g) Proposed method

(h) Original image

Figure 11. Visual comparisons of 2× magnification on the noise image.

5. Conclusions

Based on our and others' previous work, this paper introduced a new method for image interpolation via integration. The new method is based on the SGAR model and can accommodate arbitrary scaling factors. First, the paper discussed how to use the SGAR model to describe the image window, including the establishment of a linear autoregressive model, the SGAR model and the relationship between anchor pixels and their neighboring pixels. By grouping the gradient directions, the adaptive extension direction of the image window was determined, and an image window adaptive extension method based on the gradient angles for the SGAR model was formed. Because the product terms were introduced into the SGAR model, the degree of freedom of the model was increased, and the ability to describe the model was enhanced, but this may cause overfitting problems. Therefore, an elastic network was introduced into the solution of the objective function to address the overfitting problem. Finally, the curvatures were calculated in eight directions, and the interpolation results were updated accordingly to improve the interpolation accuracy. Experiments on 25 images show that the objective measures of the proposed method were improved to a certain extent. Subjective visual effect evaluations were carried out, and much better results were achieved. Therefore, the method presented in this paper improved the objective index of image interpolation and enhanced the subjective visual effect.

Acknowledgments

Mr. Hu reports grants from National Natural Science Foundation of China, grants from Postgraduate Research & Practice Innovation Program of Jiangsu Province, during the conduct of the study.

This work was supported in part by the National Natural Science Foundation of China under Grant No. 51705238 and the Postgraduate Research & Practice Innovation Program of Jiangsu Province Grant No. SJCX19_0491.

Conflict of interests

We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

References

1. Y. Tian, M. M. Kaleemullah, M. A. Rodhaan, B. Song, A. Al-Dhelaan, T. Ma, A privacy preserving location service for cloud-of-things system, *J. Parallel Distrib. Comput.*, **123** (2019), 215–222.
2. W. Wang, W. Zhang, Z. Jin, K. Sun, R. Zou, C. Huang, et al., *A Novel Location Privacy Protection Scheme with Generative Adversarial Network*, International Conference on Big Data and Security Springer, 2019.
3. L. Wang, X. Shu, W. Zhang, Y. Chen, *Design and Optimization of Evaluation Metrics in Object Detection and Tracking for Low-Altitude Aerial Video*, International Conference on Big Data and Security, 2019.
4. Y. Liu, M. Pang, *Research on Medical Image Encryption Method Based on Chaotic Scrambling and Compressed Sensing*, International Conference on Big Data and Security, 2019.
5. B. Song, M. M. Hassan, A. Alamri, A. Alelaiwi, Y. Tian, M. Pathan, et al., A two-stage approach for task and resource management in multimedia cloud environment, *Computing*, **98** (2016), 119–145.
6. Z. Pan, C. N. Yang, V. S. Sheng, N. Xiong, W. Meng, Machine Learning for Wireless Multimedia Data Security, *Sec. Commun. Networks*, **2019** (2019), 7682306.
7. S. Ousguine, F. Essannouni, L. Essannouni, M. Abbad, D. Aboutajdine, *A New Image Interpolation Using Laplacian Operator*, International Symposium on Ubiquitous Networking, 2016, Singapore.

8. S. Pan, L. Han, Y. Tao, Q. Liu, *Study on Indicator Recognition Method of Water Meter Based on Convolution Neural Network*, International Conference on Big Data and Security, 2019, Springer.
9. J. A. Parker, R. V. Kenyon, D. E. Troxel, Comparison of Interpolating Methods for Image Resampling, *IEEE Trans. Med. Imaging*, **2** (1983), 31–39.
10. R. Hanssen, R. Bamler, Evaluation of interpolation kernels for SAR interferometry, *IEEE Trans. Geosci. Remote Sens.*, **37** (1999), 318–321.
11. T. M. Lehmann, C. Gonner, K. Spitzer, Survey: interpolation methods in medical image processing, *IEEE Trans. Med. Imaging*, **18** (1999), 1049–1075.
12. D. Y. Han, *Comparison of Commonly Used Image Interpolation Methods*, Proceedings of the 2nd international conference on computer science and electronics engineering, Atlantis Press, 2013.
13. A. Amanatiadis, I. Andreadis, *Performance evaluation techniques for image scaling algorithms*, IEEE International Workshop on Imaging Systems and Techniques, 2008.
14. Y. Li, F. Qi, Y. Wan, *Improvements On Bicubic Image Interpolation*, 2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), 2019.
15. H. Hsieh, H. Andrews, Cubic splines for image interpolation and digital filtering, *IEEE Trans. Acoust. Speech Signal Process.*, **26** (1978), 508–517.
16. S. Abbas, M. Irshad, M. Z. Hussain, Adaptive image interpolation technique based on cubic trigonometric B-spline representation, *IET Image Process.*, **12** (2018), 769–777.
17. D. D. Muresan, T. W. Parks, Adaptively quadratic (AQua) image interpolation, *IEEE Trans. Image Process.*, **13** (2004), 690–698.
18. C. Chen, C. Lai, Iterative Linear Interpolation Based on Fuzzy Gradient Model for Low-Cost VLSI Implementation, *IEEE Trans. VLSI Syst.*, **22** (2014), 1526–1538.
19. Q. Wang and R. K. Ward, A New Orientation-Adaptive Interpolation Method, *IEEE Trans. Image Process.*, **16** (2007), 889–900.
20. C. M. Zwart, D. H. Frakes, Segment Adaptive Gradient Angle Interpolation, *IEEE Trans. Image Process.*, **22** (2013), 2960–2969.
21. X. Li, M. T. Orchard, New edge-directed interpolation, *IEEE Trans. Image Process.*, **10** (2001), 1521–1527.
22. X. Zhang, X. Wu, Image Interpolation by Adaptive 2-D Autoregressive Modeling and Soft-Decision Estimation, *IEEE Trans. Image Process.*, **17** (2008), 887–896.
23. Q. Wang, J. Liu, W. Yang, Z. Guo, *Adaptive autoregressive model with window extension via explicit geometry for image interpolation*, IEEE International Conference on Image Processing (ICIP), 2015.
24. W. Yang, J. Liu, S. Yang, Z. Guo, *Novel autoregressive model based on adaptive window-extension and patch-geodesic distance for image interpolation*, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2015.
25. W. H. Yang, J. Y. Liu, M. D. Li, Z. Gao, Isophote-Constrained Autoregressive Model With Adaptive Window Extension for Image Interpolation, *IEEE Trans. Circuits Syst. Video Technol.*, **28** (2018), 1071–1086.
26. F. Hao, J. Shi, R. Chen, S, Zhu, Z. Zhang, Noise-insensitive and edge-preserving resolution upconversion scheme for digital image based on the spatial general autoregressive model, *IET Image Process.*, **10** (2016), 280–288.
27. K. Chang, P. L. K. Ding, B. Li, Single Image Super-resolution Using Collaborative Representation and Non-local Self-Similarity, *Signal Process.,* **149** (2018), 49–61.

28. D. Cheng, K. I. Kou, FFT Multichannel Interpolation and Application to Image Super-resolution, *Signal Process.*, **162** (2019), 21–34.
29. A. Giachetti, N. Asuni, Real-Time Artifact-Free Image Upscaling, *IEEE Trans. Image Process.*, **20** (2011), 2760–2768.
30. H. Kim, Y. Cha, S. Kim, Curvature Interpolation Method for Image Zooming, *IEEE Trans. Image Process.*, **20** (2011), 1895–1903.
31. A. Marquina, S. Osher, *A New Time Dependent Model Based on Level Set Motion for Nonlinear Deblurring and Noise Removal*, International Conference on Scale-Space Theories in Computer Vision, 1999.
32. T. F. Chan, J. J. Shen, *Image processing and analysis: variational, PDE, wavelet, and stochastic methods*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2005.
33. J. Hu, Y. Luo, Single-image superresolution based on local regression and nonlocal self-similarity, *J. Electron. Imaging*, **23** (2014), 033014.
34. N. Asuni, A. Giachetti, Accuracy Improvements and Artifacts Removal in Edge Based Image Interpolation, *VISAPP (1)*, **8** (2008), 58–65.
35. W. Dong, L. Zhang, R. Lukac, G. Shi, Sparse Representation Based Image Interpolation With Nonlocal Autoregressive Modeling, *IEEE Trans. Image Process.*, **22** (2013), 1382–1394.
36. F. Hao, J. Shi, Z. Zhang, R, Chen, S. Zhu, Canny edge detection enhancement by general auto-regression model and bi-dimensional maximum conditional entropy, *Optik*, **125** (2014), 3946–3953.
37. M. Li, J. Liu, J. Ren, Z. Guo, Adaptive General Scale Interpolation Based on Weighted Autoregressive Models, *IEEE Trans. Circuits Syst. Video Technol.*, **25** (2015), 200–211.
38. A. Ullah, M. Z. Asghar, A. Habib, S. Aleem, F. M. Kundi, A. M. Khattak, *Optimizing the Efficiency of Machine Learning Techniques*, International Conference on Big Data and Security, 2019.
39. R. M. Rifkin, R. A. Lippert, *Notes on regularized least squares*, Massachusetts Institute of Technology Computer Science and Artificial Intelligence Laboratory, 2007.
40. K. Koh, S. J. Kim, S. Boyd, An interior-point method for large-scale l1-regularized logistic regression, *J. Mach. Learn. Res.*, **8** (2007), 1519–1555.
41. J. Friedman, T. Hastie, R. Tibshirani, Regularization paths for generalized linear models via coordinate descent, *J. Stat. Software*, **33** (2010), 1.
42. H. Wang, W. Guan, K. Zhang, *Over-Sampling Multi-classification Method Based on Centroid Space*, International Conference on Big Data and Security, 2019.
43. Y. Gong, I. F. Sbalzarini, Curvature filters efficiently reduce certain variational energies, *IEEE Trans. Image Process.*, **26** (2017), 1786–1798.
44. X. Hu, H. Mu, X. Zhang, Z. Wang, T. Tan, J. Sun, *Meta-SR: A Magnification-Arbitrary Network for Super-Resolution*, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.