

# Adaptive General Scale Interpolation Based on Weighted Autoregressive Models

Mading Li, Jiaying Liu, *Member, IEEE*, Jie Ren, and Zongming Guo, *Member, IEEE*

**Abstract**—The autoregressive (AR) model has been widely used in signal processing for its effective estimation, especially in image processing. Many dedicated  $2\times$  interpolation algorithms adopt the AR model to describe the strong correlation between low-resolution (LR) pixels and high-resolution (HR) pixels. However, these AR model-based methods closely depend on the fixed relative position between LR pixels and HR pixels that are nonexistent in the general scale interpolation. In this paper, we present an adaptive general scale interpolation algorithm that is capable of arbitrary scaling factors considering the nonstationarity of natural images. Different from other dedicated  $2\times$  interpolation methods, the proposed AR terms are modeled by pixels with their adjacent unknown HR neighbors. To compensate for the information loss caused by mismatches of AR models, we consider a weighting scheme suitable for general scale situations based on the pixel similarity to increase accuracy of the estimation. Comprehensive experiments demonstrate the effectiveness of the proposed method on general scaling factors. The maximum gain of peak signal-to-noise ratio is 2.07 dB compared with segment adaptive gradient angle in  $1.5\times$  enlargements. To evaluate the performance in resolution adaptive video coding, we have also tested our method on Joint Scalable Video Model codec and obtained better subjective quality and rate-distortion performance.

**Index Terms**—Autoregressive (AR) model, general scale, interpolation, pixel similarity.

## I. INTRODUCTION

WITH rapid developments in modern technology, image and video processing is playing an increasingly important role. In the field of consumer electronics, the variety of media-playing devices makes it necessary for the single image/video source to be transformed into different resolutions. H.264/Scalable Video Coding (SVC) [2], the expansion of the current video coding standard H.264/Advanced Video Coding (AVC), provides support for spatial scalability. However, it only supports certain scaling ratios and is unable to meet the requirements of the practical resolution adaptivity. In the latest High Efficiency Video Coding (HEVC) standard,

the spatial and temporal scalability of HEVC still calls for proposals [3]. Thus, an adaptive general scale interpolation is urgently needed to deal with the challenge of the arbitrary resolution display.

Image interpolation is a process that generates high-resolution (HR) images utilizing the information from low-resolution (LR) images. The key task of image interpolation is to estimate the HR pixels interpolated into the LR image. Conventional polynomial-based interpolation algorithms, such as bilinear and bicubic [4], apply a convolution on every interpolated pixel. Pixels are estimated by a polynomial comprised of adjacent pixel values, which are usually weighted by their distance from the interpolated pixel. As a result, the complexity of polynomial-based interpolation methods is rather low and these interpolation methods are easy to implement. However, these methods apply the same convolution kernel on every interpolated pixel. Thus, they do not distinguish pixels in the smooth region from those in the high-frequency region. Furthermore, these methods are based on the assumption that local areas in images are continuous. Natural images have lots of discontinuous features. Therefore, polynomial-based interpolation methods produce noticeable reconstruction artifacts and blur the image, especially across edges.

Since the edge structure is one of the most salient features in natural images for the human visual system, many edge-directed interpolation methods are published to produce better reconstructed images with sharp edges. An effective way is to detect the orientation of the edge and interpolate along the direction [5], [6]. For example, Wang and Ward [5] proposed an orientation-adaptive interpolation method. They detected the local orientation of an interpolated pixel by bilinearly interpolating the gradient of its four nearest LR neighbors estimated by an improved Sobel operator. Then, they determined a parallelogram that enclosed the interpolated pixel and used pixels on its vertices to interpolate the pixel's value. Giachetti and Asuni [6] presented an iterative curve based interpolation (ICBI). The two-step algorithm first filled missing pixels with the average of two neighbors in the direction of the lowest second-order derivative and assigned an energy component at the new pixel location. In the second step, an iterative greedy procedure minimized the global energy by modifying the interpolated pixel values. In these methods, the interpolation is rather simple once the orientation of the edge is obtained. Hence, the key step is the detection of edge orientation. The more accurate the edge orientation detection is, the better interpolation quality the method can produce. Nevertheless, the detection of edge orientation is easily affected by noises. Thus, it is not a trivial work to make

Manuscript received November 8, 2013; revised February 11, 2014, April 16, 2014, June 4, 2014, and July 8, 2014; accepted July 31, 2014. Date of publication August 13, 2014; date of current version February 4, 2015. This work was supported in part by the National Natural Science Foundation of China under Contract 61101078, in part by the National High-Tech Technology Research and Department Program (863 Program) of China under Grant 2014AA015205, and in part by the Beijing Natural Science Foundation under Contract 4142021. This paper was recommended by Associate Editor L.-P. Chau. (*Corresponding author: J. Liu.*)

The authors are with Institute of Computer Science and Technology, Peking University, Beijing 100871, China (e-mail: martinli0822@pku.edu.cn; liujiaying@pku.edu.cn; jren2012@gmail.com; guozongming@pku.edu.cn).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TCSVT.2014.2347531

a precise detection of edges and the accuracy of the detection is unstable and unguaranteed.

To avoid the difficulty of detecting the edge orientation, edge orientations are fused with weights to improve the accuracy of edge detection [7]–[9]. Li and Nguyen [7] predefined 16 discrete directions in a  $7 \times 7$  neighborhood structure and each direction contributes to the estimated edge direction. The weight of each direction is calculated by pixel intensity variation (large weight along edges and small weight across edges), imposing a geometric regularity constraint on the interpolated image through the Markov random field model. Zhang and Wu [8] interpolated missing HR pixels by fusing estimations along two orthogonal directions using linear minimum mean square-error estimation. Similarly, Zhou *et al.* [9] proposed a directional cubic convolution (DCC) interpolation. They applied bicubic interpolation along estimated strong edge directions, while fusing two orthogonal directional bicubic interpolation results on weak edges. However, the number of the fusing edges was finite in these methods that reduced the adaptivity of these methods.

In contrast with direct orientation detection, [10] and [11] used autoregressive (AR) models and corresponding parameters to characterize the edge direction information. The new edge-directed interpolation (NEDI) was proposed in [10]. The parameters of the diagonal-direction AR model in the LR image were computed by a least-squares method. Based on the geometric duality between the LR covariance and the HR covariance, they estimated HR pixels by their neighboring LR pixels using the corresponding AR model parameters in the LR image. Zhang and Wu [11] further proposed a soft-decision adaptive interpolation (SAI) based on NEDI. They added a cross-direction AR model and more correlation constraints between LR pixels and HR pixels and produced state-of-the-art results.

However, these AR-based algorithms are based on the assumption that images are piecewise stationary. To account for the fact that the natural image is not always stable in local windows, [12] proposed an implicit piecewise AR (IPAR) model-based image interpolation algorithm based on similarity modulated block estimation. In IPAR, a similarity probability model between HR blocks was proposed to manage the nonstationarity of image signals. Similarly, Hung and Siu [13] proposed an improved edge-directed interpolation algorithm by modeling the weights of residuals for parameter estimation using the bilateral filter. Moreover, to solve the mismatch of geometric duality between HR and LR geometric structures in  $2 \times$  enlargement, Hung and Siu [14] further proposed a robust soft-decision interpolation using the weighted least-squares estimation. Nevertheless, weighting schemes provided by these methods are still limited in the  $2 \times$  enlargement.

As for practical applications such as image compression and video coding, interpolation is a widely used fundamental technology. In H.264/SVC, a bilinear interpolation filter is adopted as the up-sampling operation for residual. However, bilinear interpolation often produces lots of artifacts, such as aliasing and blurring. These artifacts not only affect the visual quality of video sequences but also reduce the compression efficiency.

With the rapid growth of the mobile devices, resolution of video sequences and images differs greatly between devices. Thus, general scale enlargement methods are urgently required. Nevertheless, interpolation methods aforementioned are lack of general scale capability. These state-of-the-art interpolations [6], [8]–[16] have rather impressive performances, while they can only deal with enlargements whose magnification is  $2^i$ , ( $i = 1, 2, \dots$ ).

Some learning-based methods can accommodate arbitrary scaling factors. Super-resolution methods based on sparse coding [17]–[19] are the representative research results of these methods. These methods sought the sparse representation of small patches in the input LR image and tried to generate the HR output by the coefficients of the representation. They trained two dictionaries for LR and HR image patches and the learned dictionary pair is a compact representation adapted to the natural images of interest. The image enlargement performance of this kind of methods is relatively good, yet the learning step is really time consuming. To apply for different scaling factors, these methods need to learn dictionaries for each factor. Thus, these methods are not suitable for general scaling.

The iterative back-projection (IBP) [20] is another classic method capable of general scaling. The IBP process reconstructed the HR image from the LR image (or a set of LR images) by minimizing the reconstruction error iteratively. However, the big solution space of IBP made it difficult to obtain the optimal solution. To improve the performance of the IBP process, the regularized IBP methods by nonlocal means based filters [21] and bilateral filters [22] have been proposed. Nevertheless, the textures of the reconstructed images are still unsatisfactory.

Bicubic interpolation supports arbitrary scaling factor, but it also produces noticeable fuzzy areas and ringing artifacts. In current H.264/SVC reference software Joint Scalable Video Model (JSVM), the scaling factor must have a value of 1.5 or 2 based on the Scalable Baseline profile [2]. Wu *et al.* [23] proposed an adaptive resolution up-conversion method based on H.264/SVC. Similar to SAI, the method used two directional AR models constructed of pixel neighbors. However, the instability of the natural image in local areas was not considered that could lead to an inaccurate estimation of AR models. Meanwhile, the general scale capability of Wu's work is also restricted by H.264/SVC and only supports certain scaling ratios. Tang *et al.* [24], [25] proposed two kinds of general scale interpolation methods based on the AR model. One of them used the geodesic distance weighting to simultaneously measure both the spatial distance and color difference [24], the other combined the AR model with convolution kernel constraint to obtain good results [25]. However, the computational cost of geodesic distance is rather high in [24] and the convolution kernel should be the same with the down-sampled one in [25]. Lately, Zwart and Frakes [1] proposed a segment adaptive gradient angle (SAGA) interpolation based on approximations of image isophotes. Their method had rather impressive performance for  $2 \times$  enlargement but the interpolation on arbitrary scaling factors was not satisfactory.

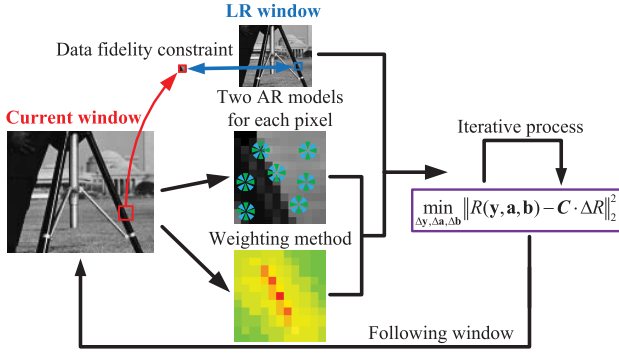


Fig. 1. Flowchart of the proposed method.

To solve issues mentioned above, we propose a novel general scale image interpolation algorithm robust to down-sampling methods. The proposed method can accommodate arbitrary scaling factors, while considering the instability of natural images in local areas. As shown in Fig. 1, the interpolation algorithm can be optimized by adjusting the composition of AR models and introducing a data fidelity constraint. The new AR models are constructed by the pixel's unknown HR neighbors. Inspired by weighted least-squares interpolations [12]–[14], we propose a weighting scheme suitable for general scale situations to determine the similarity between pixels in a local window, which can be used to adjust the formulation of the AR models. Finally, since the HR pixels and parameters of two AR models are both unknown, the structured total least-squares solution (STLS) [26] can be utilized to linearize the objective function and solve the problem by an iterative process. The main contribution of our work is to produce a general scale interpolation framework that suitable for all kinds of down-sampling methods and initial values. Comprehensive experiments show that our method produces the best image quality both from the subjective perspective and objective criterion. Comparing with state-of-the-art interpolation algorithms, the proposed method preserves better image details, especially along edge structures. The maximum gain of peak signal-to-noise ratio (PSNR) is 2.07 dB compared with SAGA in 1.5 $\times$  enlargements. In addition, the performance of the proposed method in 2 $\times$  enlargement is comparable with state-of-the-art dedicated 2 $\times$  interpolation algorithms. When tested on JSVM codec, the largest gain of our method over JSVM is 1 dB.

The rest of this paper is organized as follows. Section II gives a brief introduction of the two AR models and discusses the superiority of the proposed method compared with prior works. Section III generalizes the interpolation algorithm to general scaling factor, and then describes the proposed image interpolation algorithm based on the weight distribution in the local window. Experimental results and analysis of the proposed method are presented in Section IV. Finally, Section V concludes this paper.

## II. AR MODEL

### A. Prior Works

The AR model is often used in statistics and signal processing. In image interpolation, an AR model is considered

as a representation of the local image structure. It can obtain an estimation of pixels based on the given information that means that every unknown interpolated pixel in an image can be estimated by its known adjacent neighbors with certain weights. The AR model is defined as

$$X(m, n) = \sum_{(i, j) \in \Omega} X(m+i, n+j) \cdot \varphi(i, j) + \sigma \quad (1)$$

where  $\sigma$  is the estimation error,  $\Omega$  and  $\varphi$  are the adjacent neighbors and their weights (parameters of the AR model) to pixel  $X(m, n)$ , respectively. The AR model and its parameters attempt to characterize the local structure of the local window  $W$ . Based on the assumption that images maintain stability in the local window  $W$ , the parameters of all AR models in  $W$  are considered the same. Thus, the parameters can be computed by solving the linear least-squares problem

$$\min_{\varphi} \|\mathbf{X} - \mathbf{X}_n \cdot \varphi\|_2^2 \quad (2)$$

where  $\mathbf{X}$  is an  $M \times 1$  vector consisting of all  $M$  pixels in  $W$ .  $\mathbf{X}_n$  is an  $M \times l$  matrix where the  $i$ th row of  $\mathbf{X}_n$  consists of  $l$  adjacent neighbors of the  $i$ th pixel in  $\mathbf{X}$ . Specifically, in 2 $\times$  enlargement,  $\mathbf{X}_n$  is completely constructed by LR pixels. In this case, the AR parameters  $\varphi$  should be an  $l \times 1$  vector representing all the weights of the  $l$  adjacent neighbors. The least-squares problem computes the parameters  $\varphi$  that minimizes the estimation error. After obtaining the parameters of the AR model, unknown pixels in  $W$  can be estimated by LR pixels with the same model parameters  $\varphi$ .

For better estimation, two kinds of AR models in different directions are applied. To estimate a pixel's value, one of them uses the pixel's diagonal neighbors, while the other uses the pixel's cross-direction neighbors. A pixel's diagonal neighbors are pixels on top-left, top-right, bottom-left, and bottom-right of the pixel; its cross-direction neighbors are the four-connected neighbors. Two sets of model parameters can be calculated by performing these two AR models in a local window. Therefore, constraints on pixels are stronger and the unknown pixels can be estimated more precisely.

These types of AR models are used in [10], [11], [13], [14], [24], and [25]. However, there are two drawbacks. First, in general scale cases there may be insufficient LR pixels in the local window  $W$  to estimate unknown HR pixels. Second, the stationary assumption aforementioned does not always hold in most natural images. The solutions of these two problems will be given in Sections III-A and III-B.

### B. Superiority Compared With Prior Works

In this section, the superiority of the proposed method compared with [24] and [25] is discussed. Similar to the proposed method, these prior works also use AR models and weighted least-squares to interpolate images at arbitrary scaling factors. The common problem of utilizing AR models in general scaling situations is that the parameters of AR models cannot be preestimated as in 2 $\times$  enlargement.

TABLE I

PSNR RESULTS AND THEIR VARIANCES OF DIFFERENT METHODS USING LR INPUT GENERATED BY DIFFERENT DOWN-SAMPLING METHODS (1.7 $\times$ )

Images	Down-sample	Bicubic		SAI+BC		DCC+BC		Proposed	
		PSNR	Variance	PSNR	Variance	PSNR	Variance	PSNR	Variance
Cameraman	Bilinear	27.69	0.0121	28.18	0.0303	28.12	0.0220	28.70	0.0050
	Bicubic	27.64		28.51		28.40		28.68	
	Spline	27.90		28.59		28.46		28.84	
Lighthouse	Bilinear	29.08	0.0116	29.18	0.0046	29.16	0.0217	29.62	0.0057
	Bicubic	29.22		29.34		29.44		29.80	
	Spline	29.34		29.28		29.49		29.69	
Bike	Bilinear	27.56	0.0712	28.76	0.1691	28.62	0.1166	28.79	0.0583
	Bicubic	28.07		29.53		29.27		29.33	
	Spline	28.16		29.70		29.40		29.27	
Flinstones	Bilinear	28.89	0.0986	29.45	0.1405	29.42	0.1051	29.85	0.0689
	Bicubic	29.52		30.20		30.06		30.42	
	Spline	29.59		30.29		30.14		30.40	
Flower	Bilinear	33.70	0.0691	34.46	0.1711	34.34	0.1330	34.64	0.0654
	Bicubic	34.21		35.25		35.04		35.21	
	Spline	34.29		35.41		35.17		35.14	
Average		29.66	0.0525	30.41	0.1031	30.30	0.0797	30.56	0.0406

Thus, unlike dedicated  $2\times$  interpolation algorithms, such as NEDI and SAI, there is no closed-form solution for HR pixels.

To tackle this problem, [24] and [25] both use the Gauss–Seidel method to optimize the parameters of AR models and HR pixels, i.e., to alternatively fix one set of variables (e.g., the parameters of AR models) and optimize on the other set (e.g., the HR pixels). Although can be proved convergent, the Gauss–Seidel method suffers from the slow rate of convergence, which results in a large number of iterations. The proposed method optimizes the parameters of AR models and HR pixels at the same time and converges faster (less than three iterations for most of the window).

The other drawback of the AR model is that it strongly relies on the piecewise stationary of natural images. The method proposed in [24] uses the geodesic distance weighting for both parameter and data estimation, considering that the geodesic distance can simultaneously measure the spatial distance and color difference. However, the geodesic distance is not actually suitable for such scenario. The geodesic distance tends to slit up the local window. In that case, similar pixels in different regions (say similar pixels on both sides of an edge) have distinct weights. The proposed method uses a simpler yet effective weighting scheme and produces more reliable results.

The method proposed in [25] focuses on the convolution kernel of the down-sampling process and combines AR models with a convolution-based image degradation model. The experiments show that the results of their method are rather impressive if the convolution kernel in the degradation model is the same as the real degradation process. In reality, however, such prior knowledge is not always known to the processor. Meanwhile, the method still relies on the piecewise stationary of natural images. The proposed method is robust to the degradation process, which can be observed in Table I. The proposed method is also robust to the initial value and the down-sampling process that generates the down-sampling matrix  $\mathbf{D}$ , as shown in Table II. In this paper, we are trying to set up a general scaling framework adequate to

TABLE II

PSNR RESULTS AND THEIR VARIANCES OF THE PROPOSED METHOD USING DIFFERENT INITIAL VALUES AND DOWN-SAMPLING MATRICES (1.7 $\times$ )

Images	Initialization	Method to produce $\mathbf{D}$		Variance
		Bilinear	Bicubic	
Pepper	Bilinear	35.51	35.61	0.0077
	Bicubic	35.44	35.67	
Woman	Bilinear	35.19	35.21	0.0040
	Bicubic	35.09	35.27	
Cameraman	Bilinear	28.85	28.91	0.0072
	Bicubic	28.79	28.68	
Lighthouse	Bilinear	29.59	29.72	0.0067
	Bicubic	29.62	29.80	
Airplane	Bilinear	33.66	33.70	0.0043
	Bicubic	33.66	33.82	
Monarch	Bilinear	34.97	35.07	0.0083
	Bicubic	35.17	35.20	

different degradation processes, different initializations, and different down-sampling matrices.

### III. GENERALIZED IMAGE INTERPOLATION ALGORITHM

In this section, we present a generalized image interpolation method. The difference between  $2\times$  enlargement methods and the generalized method is elaborated at first. Then, we introduce a weighting scheme to define the similarity of two pixels that will be used to assign weights for all the pixels in the local window. Finally, our generalized image interpolation algorithm is described in detail.

#### A. Generalization of Interpolation at Arbitrary Scaling Factors

In the HR image reconstructed by most AR-based interpolation methods, there are always plenty of pixels that are extracted from the LR image directly. In other words, these pixels are exactly the same as the corresponding pixels in the LR image. We name these pixels *fixed*-pixels and *inter*-pixels for other pixels. In a local region in the HR image, the more *fixed*-pixels, the more information we can get to

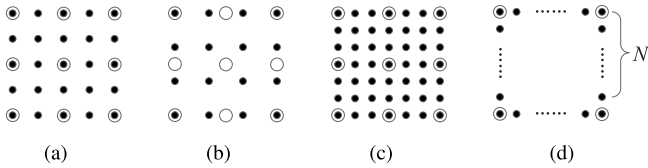


Fig. 2. HR pixels (black dots) and LR pixels (white circles) in the local region at different scaling factors. The overlapped pixels are *fixed*-pixels, while other HR pixels are *inter*-pixels. (a) Scaling factor = 2.0. (b) Scaling factor = 1.5. (c) Scaling factor = 3.0. (d) More general situation. A general scaling factor, such as 1.7, is likely to produce a rather large  $N$ .

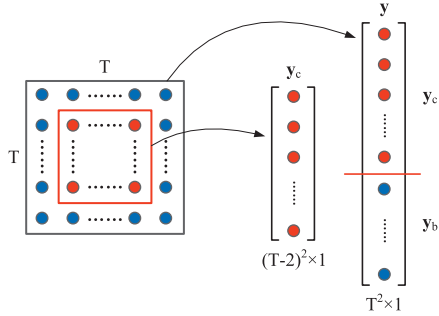


Fig. 3. Illustration of  $\mathbf{y}$  and  $\mathbf{y}_c$ . Let  $T$  be the side-length of the local square window,  $\mathbf{y}$  is a  $T^2 \times 1$  vector constructed by all pixels (red and blue dots) in the window;  $\mathbf{y}_c$  is a  $(T-2)^2 \times 1$  vector consists of pixels in the local window excluding the pixels on the boundaries of the window (red dots only).

interpolate *inter*-pixels. Fig. 2 shows that the relative location between *fixed*-pixels and *inter*-pixels varies as the scaling factor changes. When the scaling factor is 2, there is only one *inter*-pixel between adjacent *fixed*-pixels [Fig. 2(a)]. Hence, it is a good way to estimate *inter*-pixels from their neighboring *fixed*-pixels.

However,  $2\times$  enlargement is just one of the special circumstances of the image enlargement. In Fig. 2(d),  $N$  is the distance between adjacent *fixed*-pixels. It is equivalent to the minimal integral multiples of the scaling factor. In most cases, as  $N$  increases, there are fewer *fixed*-pixels around *inter*-pixels. Temporarily, we assume that the locality stationary still remains and apply two AR models on pixels as dedicated  $2\times$  interpolation algorithms. Different with AR-model based  $2\times$  interpolation algorithms, *inter*-pixels are estimated by its neighbor pixels (instead of its neighbor *fixed*-pixels) no matter what types of pixels they are. The formation is modeled as

$$\min_{\mathbf{y}, \mathbf{a}, \mathbf{b}} \{ \alpha \|\mathbf{y}_c - \mathbf{A}\mathbf{y}\|_2^2 + \beta \|\mathbf{y}_c - \mathbf{B}\mathbf{y}\|_2^2 \} \quad (3)$$

where the  $T^2 \times 1$  vector  $\mathbf{y}$  consists of pixels in a local  $T \times T$  window of the HR image and  $\mathbf{y} = \begin{bmatrix} \mathbf{y}_c \\ \mathbf{y}_b \end{bmatrix}$  (the formation of  $\mathbf{y}$  and  $\mathbf{y}_c$  is shown in Fig. 3). Vector  $\mathbf{y}_c$  consists of pixels in the local window excluding the pixels on the boundaries of the window and vector  $\mathbf{y}_b$  consists of pixels on the boundaries. Since some of the neighbors of pixels in  $\mathbf{y}_b$  are out of the window, only the pixels in  $\mathbf{y}_c$  can be the center of an AR model.  $\alpha$  and  $\beta$  are the coefficients that control the weight of two AR models.  $\mathbf{A}$  and  $\mathbf{B}$  are both  $(T-2)^2 \times T^2$  and they

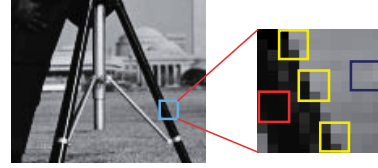


Fig. 4. Similar patches in the local window. Patches in yellow frames are similar to each other but different from those in red or navy blue.

are defined as

$$\mathbf{A}(i, j) = \begin{cases} a_k, & \text{if } \mathbf{y}_j \text{ is the } k\text{th pixel of} \\ & \mathbf{y}_i \text{'s diagonal neighbors} \\ 0, & \text{otherwise} \end{cases}$$

$$\mathbf{B}(i, j) = \begin{cases} b_k, & \text{if } \mathbf{y}_j \text{ is the } k\text{th pixel of} \\ & \mathbf{y}_i \text{'s cross-direction neighbors} \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

where  $a_k$  and  $b_k$  are the  $k$ th element of  $\mathbf{a}$  and  $\mathbf{b}$ .  $\mathbf{a} = (a_1, a_2, a_3, a_4)$  and  $\mathbf{b} = (b_1, b_2, b_3, b_4)$  are the parameters of two AR models, respectively. Different from  $2\times$  enlargement, these AR parameters cannot be computed by (2) because all pixels in  $\mathbf{y}$  are unknown. Therefore, the iterative process introduced in Section III-C can be used to solve this issue.

Since *fixed*-pixels are not utilized in AR models for HR images, these valuable pixels can be used to build a data fidelity constraint as the HR image reconstruction term [27]. For a local window in an HR image, we down-sample it by bicubic interpolation and compare it to the corresponding window in LR image. The constraint is modeled as

$$\|\mathbf{x} - \mathbf{D}\mathbf{y}_c\|_2^2 < \varepsilon \quad (5)$$

where the vector  $\mathbf{x}$  consists of pixels in the corresponding window of the LR image and matrix  $\mathbf{D}$  represents the down-sampling process;  $\varepsilon$  controls the maximum value of the constraint. The pixels on the boundaries of the local window are not estimated by the AR models, thus  $\mathbf{y}_c$  is adopted to perform the data fidelity constraint. Such constraint is useless in dedicated  $2\times$  interpolation algorithms because the pixels down-sampled are exactly the same with the corresponding pixels in the LR image.

### B. Weight Distribution in a Local Window

On the basis of the assumption that the image is piecewise stationary, all AR models of the same type in the whole local window share the same AR parameters. Therefore, when the statistics in the local window are stationary, AR models can impose appropriate constraints on the window and produce rather good image reconstruction. However, natural images do not maintain stability in most local windows. For example, as shown in Fig. 4, there are significant differences between an edge-crossing area and a smooth area in a local window. Thus, estimations of AR models in this window are not robust. To solve this issue, we introduce a weighting scheme to measure the similarity between the pixel to-be-output (usually the center pixel) and other pixels in the local window.

Naturally, we prefer to assign large weight on the pixel (in other words, its corresponding AR model) that is more

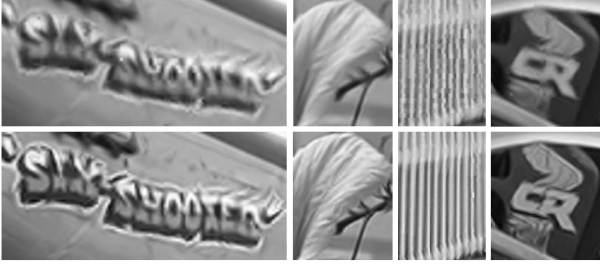


Fig. 5. Comparison between interpolation results without and with the proposed weighting scheme. The first row shows the interpolation results without the proposed weighting scheme. The second row shows the interpolation results with the proposed weighting scheme.

similar to the center pixel, and vice versa. The similarity weight between two pixels depends on the similarity of the local structure centered at two pixels ( $3 \times 3$  patches in Fig. 4). Moreover, the similarity is measured as a decreasing function of the weighted Euclidean distance [28]. The weight is made up of two parts. One is the similarity of two pixels' local structures, the other measures the spatial distance between them. The weight  $w(m, n)$  between two pixels  $m$  and  $n$  is defined as

$$w(m, n) = w_{ls}(m, n) \cdot w_d(m, n) \quad (6)$$

where  $w_{ls}(m, n)$  represents the similarity of two pixels' local structures.  $w_d(m, n)$  represents the distance between two pixels. They are described as

$$w_{ls}(m, n) = e^{-\|L_m - L_n\|_2^2 / \varepsilon_1} \quad (7)$$

$$w_d(m, n) = e^{-\|P_m - P_n\|_2^2 / \varepsilon_2} \quad (8)$$

where  $L_m$  and  $L_n$  represent vectors consisting of the eight-connected neighborhood of  $m$  and  $n$ ,  $P_m$  and  $P_n$  are the spatial coordinates of  $m$  and  $n$ , respectively.  $\varepsilon_1$  and  $\varepsilon_2$  control the shape of the exponential function. In our experiments,  $\varepsilon_1$  and  $\varepsilon_2$  are set to be 17 and 33, empirically.

After obtaining all pixels' weights to the center pixel of the local window  $W$ , a diagonal  $(T-2)^2 \times (T-2)^2$  weight matrix  $\mathbf{W}$  can be formed, which represents the weight distribution in the current local window. Fig. 5 shows the effectiveness of the proposed weighting scheme. The interpolation without the proposed weighting scheme tends to mix up the texture region and the smooth region. The interpolation results with the proposed weighting scheme preserve the sharpness of the textures.

### C. Generalized Interpolation Algorithm

By adding the weight matrix  $\mathbf{W}$  to (3), combining (5) with a Lagrangian multiplier  $\lambda$ , we can get the objective function described below

$$\min_{\mathbf{y}, \mathbf{a}, \mathbf{b}} \left\{ \alpha \|\mathbf{W}(\mathbf{y}_c - \mathbf{A}\mathbf{y})\|_2^2 + \beta \|\mathbf{W}(\mathbf{y}_c - \mathbf{B}\mathbf{y})\|_2^2 + \lambda \|\mathbf{x} - \mathbf{D}\mathbf{y}_c\|_2^2 \right\}. \quad (9)$$

The objective function presented in (9) minimizes the sum of the estimation error of the two weighted AR models and

the data fidelity term. For convenience, the formula can be represented by a least-squares problem as

$$\min_{\mathbf{y}, \mathbf{a}, \mathbf{b}} \|\mathbf{R}(\mathbf{y}, \mathbf{a}, \mathbf{b})\|_2^2 \quad (10)$$

where  $\mathbf{R}(\mathbf{y}, \mathbf{a}, \mathbf{b})$  is the residue vector, representing the estimation residue. It is described as

$$\mathbf{R}(\mathbf{y}, \mathbf{a}, \mathbf{b}) = \begin{bmatrix} \sqrt{\alpha} \mathbf{W}(\mathbf{y}_c - \mathbf{A}\mathbf{y}) \\ \sqrt{\beta} \mathbf{W}(\mathbf{y}_c - \mathbf{B}\mathbf{y}) \\ \sqrt{\lambda} (\mathbf{x} - \mathbf{D}\mathbf{y}_c) \end{bmatrix}. \quad (11)$$

The least-squares problem in (10) is nonlinear. To make it easier to be solved, the STLS is utilized to linearize the problem. Let  $\Delta \mathbf{y} = \begin{bmatrix} \Delta \mathbf{y}_c \\ \Delta \mathbf{y}_b \end{bmatrix}$ ,  $\Delta \mathbf{a}$  and  $\Delta \mathbf{b}$  be the small changes in  $\mathbf{y}$ ,  $\mathbf{a}$  and  $\mathbf{b}$ , respectively. To better constrain the pixels in the window, pixels on the boundaries of the window are kept unchanged. In other words, every element in  $\Delta \mathbf{y}_b$  is zero. As a result, some part of the matrices' product can be omitted.  $\mathbf{A}$  and  $\mathbf{B}$  can be decomposed to  $[\mathbf{A}_c, \mathbf{A}_b]$  and  $[\mathbf{B}_c, \mathbf{B}_b]$ , respectively. Let  $T$  be the side-length of the square local window. Then,  $\mathbf{A}_c$  and  $\mathbf{B}_c$  are  $(T-2)^2 \times (T-2)^2$  matrices and they are composed by the first  $(T-2)^2$  columns of  $\mathbf{A}$  and  $\mathbf{B}$ , respectively. Whereas  $\mathbf{A}_b$  and  $\mathbf{B}_b$  are composed by the remaining columns.

Let  $R_1(\mathbf{y}, \mathbf{a}) = \sqrt{\alpha} \mathbf{W}(\mathbf{y}_c - \mathbf{A}\mathbf{y})$ , thus

$$\begin{aligned} R_1(\mathbf{y} + \Delta \mathbf{y}, \mathbf{a} + \Delta \mathbf{a}) &= \sqrt{\alpha} \mathbf{W}[(\mathbf{y}_c + \Delta \mathbf{y}_c) - (\mathbf{A} + \Delta \mathbf{A})(\mathbf{y} + \Delta \mathbf{y})] \\ &= \sqrt{\alpha} \mathbf{W}[(\mathbf{y}_c - \mathbf{A}\mathbf{y}) + \Delta \mathbf{y}_c - \Delta \mathbf{A}\mathbf{y} - \mathbf{A}\Delta \mathbf{y} - \Delta \mathbf{A}\Delta \mathbf{y}] \\ &= \sqrt{\alpha} \mathbf{W}[(\mathbf{y}_c - \mathbf{A}\mathbf{y}) + \Delta \mathbf{y}_c - (\mathbf{A}_c \Delta \mathbf{y}_c + \mathbf{A}_b \Delta \mathbf{y}_b) - \Delta \mathbf{A}\mathbf{y}] \\ &= \sqrt{\alpha} \mathbf{W}[(\mathbf{y}_c - \mathbf{A}\mathbf{y}) - (-\mathbf{I} + \mathbf{A}_c) \Delta \mathbf{y}_c - \Delta \mathbf{A}\mathbf{y}] \\ &= R_1(\mathbf{y}, \mathbf{a}) - \sqrt{\alpha} \mathbf{W}[(-\mathbf{I} + \mathbf{A}_c) \Delta \mathbf{y}_c + \Delta \mathbf{A}\mathbf{y}] \end{aligned} \quad (12)$$

where  $\Delta \mathbf{y}_b$  and the higher order infinitesimal  $\Delta \mathbf{A}\Delta \mathbf{y}$  have been omitted.

Similarly, the residue vector  $\mathbf{R}(\mathbf{y}, \mathbf{a}, \mathbf{b})$  can be linearized as

$$\begin{aligned} R(\mathbf{y} + \Delta \mathbf{y}, \mathbf{a} + \Delta \mathbf{a}, \mathbf{b} + \Delta \mathbf{b}) &= R(\mathbf{y}, \mathbf{a}, \mathbf{b}) - \begin{bmatrix} \sqrt{\alpha} \mathbf{W}[(-\mathbf{I} + \mathbf{A}_c) \Delta \mathbf{y}_c + \mathbf{E}_1 \Delta \mathbf{a}] \\ \sqrt{\beta} \mathbf{W}[(-\mathbf{I} + \mathbf{B}_c) \Delta \mathbf{y}_c + \mathbf{E}_2 \Delta \mathbf{b}] \\ \sqrt{\lambda} \mathbf{D} \Delta \mathbf{y}_c \end{bmatrix} \end{aligned} \quad (13)$$

where  $\mathbf{E}_1$  and  $\mathbf{E}_2$  are  $(T-2)^2 \times 4$  matrices and constructed as follows: the  $k$ th row of  $\mathbf{E}_1$  is a vector constructed by four diagonal neighbors of pixel  $\mathbf{y}_k$ ; the  $k$ th row of  $\mathbf{E}_2$  is a vector constructed by four cross-direction neighbors of pixel  $\mathbf{y}_k$ .  $\mathbf{E}_1 \Delta \mathbf{a}$  is just an equivalent representation of  $\Delta \mathbf{A}\mathbf{y}$  to simplify the matrix.

Let

$$\mathbf{C} = \begin{bmatrix} \sqrt{\alpha} \mathbf{W}(-\mathbf{I} + \mathbf{A}_c) & \sqrt{\alpha} \mathbf{W} \mathbf{E}_1 & \mathbf{0} \\ \sqrt{\beta} \mathbf{W}(-\mathbf{I} + \mathbf{B}_c) & \mathbf{0} & \sqrt{\beta} \mathbf{W} \mathbf{E}_2 \\ \sqrt{\lambda} \mathbf{D} & \mathbf{0} & \mathbf{0} \end{bmatrix}$$

$$\Delta \mathbf{R} = [\Delta \mathbf{y} \quad \Delta \mathbf{a} \quad \Delta \mathbf{b}]^T.$$

For convenient representation, we rewrite (13) as

$$\min_{\Delta \mathbf{y}, \Delta \mathbf{a}, \Delta \mathbf{b}} \|\mathbf{R}(\mathbf{y}, \mathbf{a}, \mathbf{b}) - \mathbf{C} \cdot \Delta \mathbf{R}\|_2^2. \quad (14)$$

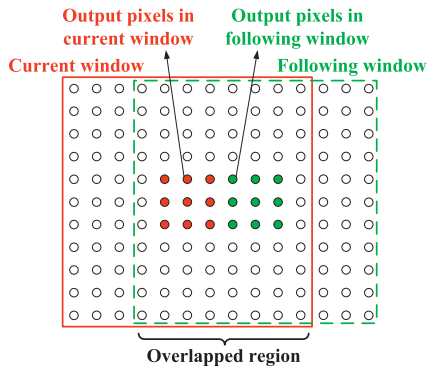


Fig. 6. Possible configuration used in the proposed algorithm. The size of windows is set to be  $11 \times 11$ . The square in the red solid line represents the current window while the square in the green dashed line represents its following window. Red dots and green dots are the output pixels of the current window and the following window, respectively.

Hence, the nonlinear problem (10) is transformed to a linear problem with  $\Delta \mathbf{y}$ ,  $\Delta \mathbf{a}$ , and  $\Delta \mathbf{b}$  as unknown quantities. Therefore, given the initial values of  $\mathbf{y}$ ,  $\mathbf{a}$ , and  $\mathbf{b}$ , we can obtain  $\Delta R$  and use it to update  $\mathbf{y}$ ,  $\mathbf{a}$ , and  $\mathbf{b}$  for the next iteration. In our implementation, bicubic interpolation is used as the initial up-scaling method and the down-sampling process in the data fidelity term.  $\mathbf{a}$  and  $\mathbf{b}$  are initialized as  $(1/4, 1/4, 1/4, 1/4)$ . Weight coefficients  $\alpha$  and  $\beta$  are empirically set to be 0.2 and 0.3, respectively. The stopping criterion of the iteration is either the maximum value of  $\Delta R < 1.0$  or the maximal number of iterations 20 being reached, whichever is reached first.

The iterative process is computationally expensive. To alleviate the complexity of the proposed method, the proposed algorithm is only applied on high-frequency areas. In Fig. 6, the output pixels of two adjacent windows are shown. In our experiment, we output the center  $3 \times 3$  pixels at once. It may slightly reduce the performance, but can lead to a 9 times speedup. Meanwhile, to avoid blocking artifacts, we produce an overlapping region between adjacent windows. The offset is set to be 3 pixels. It can be seen that every pixel except for the pixels in boundary areas of the whole image can be processed.

#### IV. EXPERIMENTAL RESULTS

In this section, the proposed general scale interpolation method is evaluated in two aspects. First, we test the proposed interpolation method on the image enlargement and compare it with other state-of-the-art interpolation methods. PSNR is selected as the objective evaluation criterion. Second, we test the proposed method on JSVM (the reference software of H.264/SVC) [29]. The rate-distortion ( $R$ - $D$ ) curves are compared with evaluate the performance of the proposed method.

##### A. Image Enlargement

The proposed interpolation method is implemented on MATLAB 7.10 platform. To evaluate the general applicability of our method, we choose 1.5 and 1.7 as the scaling factors and

TABLE III  
AVERAGE PSNR RESULTS OF DIFFERENT METHODS UNDER  
THE EXPERIMENT SETUP OF ICBI ( $2\times$ , *Setup2*)

Methods	Bicubic	NEDI	ICBI	SAI	DCC	Proposed
PSNR	30.36	29.71	31.07	30.78	30.66	30.73

compare interpolation results with bicubic interpolation and SAGA [1]. To compare with other state-of-the-art dedicated  $2\times$  interpolation algorithms in general scale cases, we simulate the general scale process by adding a bicubic interpolation after the dedicated  $2\times$  interpolation. In addition, to evaluate the performance of the proposed method when the scaling factor is 2, the state-of-the-art dedicated  $2\times$  interpolation algorithms, such as NEDI [10], SAI [11], DCC [9], ICBI [6], and SAGA [1], are used as comparisons. Our testing images are selected from the Kodak database [30] and the University of Southern California - the Signal and Image Processing Institute image database [31].

For a scaling factor  $s$ , we first generate the LR image by down-sampling the original HR image by a factor of  $1/s$ . After that, different interpolation methods are used to obtain HR images from the LR image. In  $2\times$  enlargement, the size of HR results generated by  $n \times n$  LR images using ICBI is  $(2n-1) \times (2n-1)$  instead of exact  $2n \times 2n$ , which is because ICBI does not reconstruct the last column and the last row. Thus, the left-top  $(2n-1) \times (2n-1)$  part of the original HR image (and the results of other methods) is used to compute the PSNR and structural similarity (SSIM) to prevent the pixel shift.

The proposed method is also robust to down-sampling methods. It is because the two AR models and the weighting scheme only consider the structure in the local window. Meanwhile, the down-sampling matrix is generated by the same method with the initial up-scaling. Thus, the differences produced by down-sampling methods do not affect the proposed method. Table I shows the robustness of the proposed method under different down-sampling methods. The average variance of the proposed method's PSNR results for three kinds of down-sampling methods is the smallest, which shows the robustness of the proposed method. Meanwhile, the proposed method produces the highest PSNR results.

In our experiment, the direct down-sampling is applied for  $2\times$  enlargement and bicubic down-sampling for other scaling factors (direct down-sampling cannot be used for arbitrary scaling factors). For simplicity, such experiment setup is regarded as *Setup1* in the rest of this paper. It should be noticed that PSNR/SSIM results for NEDI and ICBI are underestimated due to the half pixel shift under *Setup1*. To fairly compare the performance of the interpolation methods, we have added the experiments ( $2\times$  enlargement) using bilinear as down-sampling method (*Setup2*). *Setup2* does not produce the half pixel shift for those methods. The PSNR results of different interpolation methods in the added experiments are presented in Table III. As can be observed, ICBI performs slightly better than SAI, DCC, and the proposed method. The performance of the proposed method is competitive with SAI and DCC, as it is under *Setup1* (Table V), which further

TABLE IV  
PSNR (dB) RESULTS OF DIFFERENT METHODS AT ARBITRARY SCALING FACTORS (*Setup1*)

Images	Scale	Bicubic	SAI+BC	DCC+BC	ICBI+BC	SAGA	Proposed
<i>Cameraman</i>	1.7	27.64	28.51	28.40	28.02	27.73	28.68
<i>House</i>		24.68	24.88	24.80	24.37	24.51	24.99
<i>Ruler</i>		12.83	12.80	12.85	12.39	12.67	12.95
<i>Lighthouse</i>		29.22	29.34	29.44	28.94	28.18	29.80
<i>Monarch</i>		34.27	35.23	35.12	35.25	34.03	35.20
<i>Lena</i>	1.5	37.67	38.13	37.86	37.95	37.24	38.29
<i>Baboon</i>		26.19	26.49	26.27	25.99	25.95	26.50
<i>Bike</i>		29.94	31.03	30.78	30.75	29.66	31.26
<i>Tulip</i>		38.23	39.76	39.28	39.49	37.88	39.95
<i>House</i>		26.23	26.32	26.26	26.10	25.92	27.00
Average		28.69	29.25	29.11	28.93	28.38	29.46

TABLE V  
PSNR (dB) RESULTS AND SSIM INDEX (IN THE BRACKETS) OF DIFFERENT METHODS, SCALING FACTOR  $s = 2$  (*Setup1*)

Images	Bicubic	NEDI	SAI	DCC	ICBI	SAGA	Proposed
<i>Airplane</i>	30.90 (0.965)	31.31 (0.966)	31.79 (0.967)	31.56 (0.960)	30.73 (0.962)	30.50 (0.934)	31.59 (0.967)
<i>Bike</i>	25.75 (0.950)	25.56 (0.943)	26.83 (0.955)	26.47 (0.953)	25.37 (0.945)	26.35 (0.951)	26.29 (0.952)
<i>Flinstones</i>	27.07 (0.962)	26.31 (0.961)	27.94 (0.967)	27.84 (0.963)	27.12 (0.961)	27.17 (0.963)	27.87 (0.966)
<i>Flower</i>	31.95 (0.978)	31.71 (0.977)	32.98 (0.981)	32.74 (0.978)	31.95 (0.978)	32.38 (0.979)	32.48 (0.980)
<i>Lena</i>	34.09 (0.972)	33.93 (0.973)	34.85 (0.974)	34.52 (0.967)	34.14 (0.970)	34.52 (0.974)	34.57 (0.973)
<i>Monarch</i>	30.51 (0.986)	30.50 (0.987)	31.76 (0.988)	31.61 (0.984)	31.19 (0.986)	31.17 (0.987)	31.57 (0.988)
<i>Parrot</i>	33.66 (0.981)	33.80 (0.982)	34.82 (0.983)	34.49 (0.978)	33.43 (0.980)	34.14 (0.982)	34.34 (0.982)
<i>Pepper</i>	32.96 (0.957)	33.19 (0.963)	33.71 (0.960)	33.36 (0.954)	33.07 (0.954)	33.60 (0.962)	33.58 (0.958)
<i>Ruler</i>	12.10 (0.849)	11.83 (0.844)	11.51 (0.840)	11.97 (0.845)	10.83 (0.812)	12.87 (0.859)	12.64 (0.861)
<i>Barbara</i>	24.52 (0.879)	22.44 (0.851)	23.59 (0.864)	23.62 (0.860)	22.87 (0.851)	24.29 (0.874)	23.30 (0.860)
<i>Lighthouse</i>	27.07 (0.927)	26.51 (0.924)	26.79 (0.926)	27.26 (0.923)	26.36 (0.920)	27.33 (0.929)	27.26 (0.929)
Average	28.23 (0.946)	27.92 (0.943)	28.78 (0.946)	28.68 (0.942)	27.91 (0.938)	28.57 (0.945)	28.68 (0.947)

verifies the robustness of the proposed method. However, when using *Setup1*, the performance of ICBI is not even better than bicubic. In this case, the objective evaluation criterion may be inaccurate. We recommend readers refer to the subjective comparisons, such as Figs. 8 and 11–13 to evaluate the performances of the compared methods.

If the scaling factor  $s$  is 1.5 or 1.7, for SAI, DCC, and ICBI, the LR image is first magnified by a factor of 2, then followed by the bicubic interpolation by a factor of  $s/2$  (denoted as SAI+BC, DCC+BC, and ICBI+BC in Tables I and IV). Most of the interpolation algorithms do not process pixels on the boundaries of images, including the proposed method. Thus, PSNR between the processed images and the original images is calculated regardless of the 5-pixel-width boundaries of the images.

The other weight coefficient  $\lambda$  is set to be 0.5 empirically. The window size is also important to our algorithm (a larger size window may impact the locality stationary, a small one cannot provide enough structural information), we set it to be  $11 \times 11$ . Results of  $1.5\times$ ,  $1.7\times$ , and  $2\times$  magnifications are shown in Tables IV and V.

As can be observed from Table IV, the proposed method generates the best results in most cases. In our experiments, *Barbara* is a very interesting image because the bicubic interpolation always produces the best result in all cases. The proposed method is particularly sensitive to edges, because of the weight distribution in a rather large window. In addition, the initial value is given by the bicubic interpolation, which

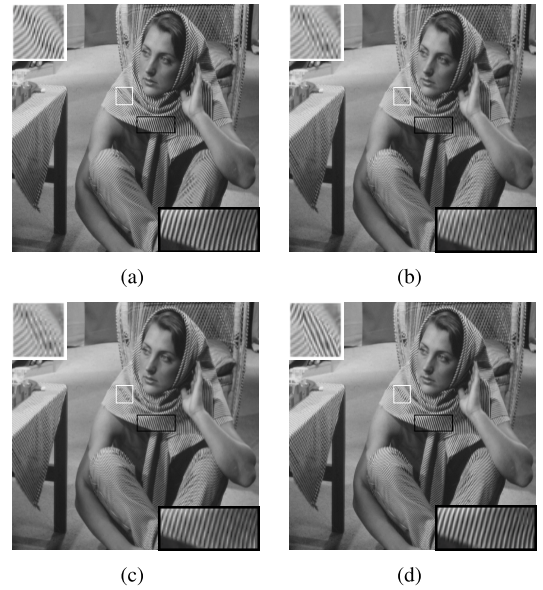


Fig. 7. Local magnification of *Barbara* in aliasing part ( $1.5\times$  enlargements). (a) Original image. (b) Result of bicubic. (c) Result of SAGA. (d) Result of the proposed work. The white block with its corresponding magnification on the left-top corner of each image shows the severe aliasing area, which is not well processed by the proposed method. The black block with its corresponding magnification on the right-bottom corner of each image shows the slight aliasing area, which is clearly reconstructed by the proposed method.

will easily cause optical aliasing [Fig. 7(b)]. The proposed method can alleviate slight aliasing but cannot properly process severe aliasing [Fig. 7(d)]. The proposed method



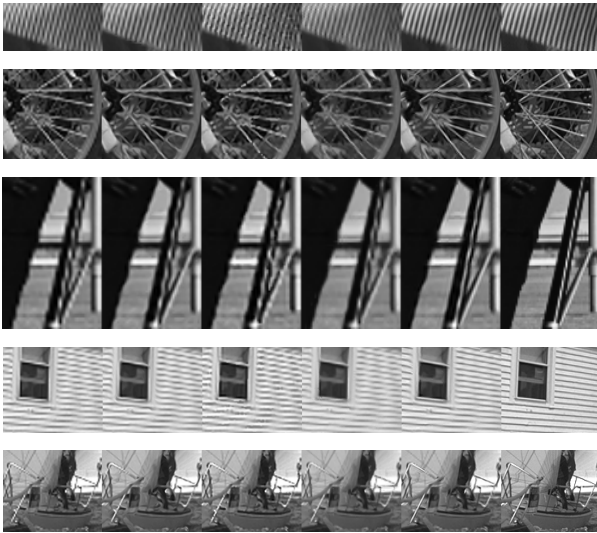


Fig. 8. Subjective comparison with other general scale interpolation algorithms applied on different images with scaling factor  $s = 1.7$ . From left to right: bicubic, DCC + BC, ICBI + BC, SAGA, the proposed method, and the original image. From top to bottom: *Barbara*, *Bike*, *Cameraman*, *Lighthouse*, and *Sailboat*.

considers the severe aliasing as edges on a false direction and reconstructs the false edges more sharply than SAGA [Fig. 7(c)]. In Fig. 7, a severe aliasing and a slight aliasing are marked in white and black blocks, while their magnifications are shown in the left-top corner and right-bottom corner of the image, respectively. In *Barbara*, there are a lot more severe aliasing artifacts in bicubic results. Thus, the proposed method produces false edges more than the correct ones and the quality of the reconstructed image is not good enough.

To evaluate the general applicability of our method,  $2\times$  enlargement experiments are also conducted and Table V shows the results. The performance of the proposed method is comparable with other state-of-the-art dedicated  $2\times$  interpolation algorithms. Moreover, the proposed method is designed for general scaling factors to make images/videos more adaptive to all kinds of media-playing devices. Thus, the proposed method has greater advantages in the general scale situation and its performance is basically satisfactory in  $2\times$  situation. In addition, our method obtains better SSIM index [32]. Compared with PSNR, the SSIM index is more consistent relative to visual perception [33]. It is well known that the human visual system is more sensitive to the edge structures of images. As a result, our method has a better performance on the SSIM index over DCC and ICBI, and is equivalent with SAI.

As pointed out before, in our experiments, the input LR images for  $2\times$  enlargement are directly down-sampled and that of  $1.5\times$  and  $1.7\times$  enlargements are down-sampled by bicubic. The applied experiment setup causes half pixel shift for NEDI/ICBI. To make a fair comparison, we have also tested the proposed method using the experiment setup that does not produce the half pixel shift for NEDI/ICBI. The PSNR results are presented in Table III.

The superiority of the subjective quality is also demonstrated in Figs. 8–11. In general scale cases, the proposed

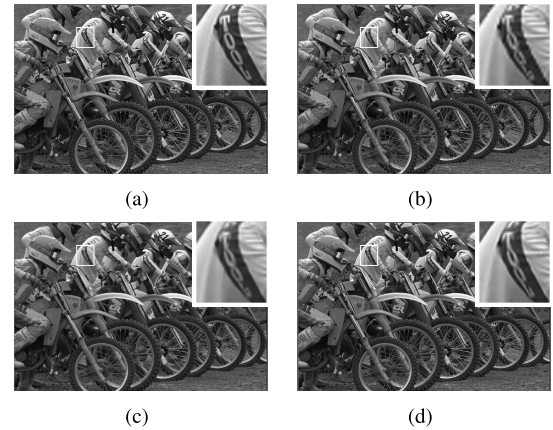


Fig. 9. Subjective image quality comparison of reconstructed HR images by different methods in  $1.7\times$  enlargements. Local magnification is shown in the upper-right corner of each image. (a) Original image *Bike*. (b) Result of bicubic. (c) Result of SAGA. (d) Result of the proposed method.

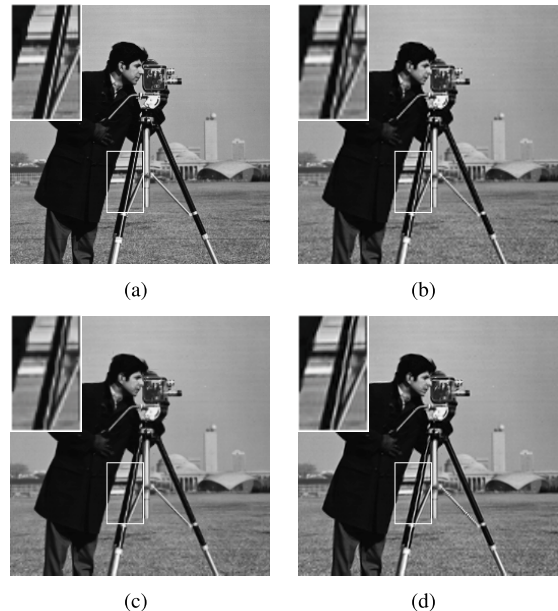


Fig. 10. Subjective image quality comparison of reconstructed HR images by different methods in  $1.5\times$  enlargements. Local magnification is shown in the upper-left corner of each image. (a) Original image *Cameraman*. (b) Result of bicubic. (c) Result of SAGA. (d) Result of the proposed method.

method presents sharper edges and better visual quality than bicubic and SAGA. As shown in Fig. 9, the proposed method presents more distinct letters on the rider's sleeve. In Fig. 10, the proposed method preserves the sharpness of both horizontal and beveled edges. As shown, bicubic interpolation produces evident zigzag artifacts, while SAGA somewhat blurs the edges. Our method preserves the sharpness of edges and obtains satisfactory restoration. Fig. 8 shows more comparison of  $1.7\times$  interpolation results.

As for  $2\times$  enlargement situations, we compared some images to show the superiority of the proposed method on long and sharp edges (Fig. 11). Bicubic interpolation presents fuzzy areas around the edges. For most of the edges, NEDI, SAI, DCC, ICBI, and SAGA present rather good results, but there are still some jags and ringing effects on a few sharp

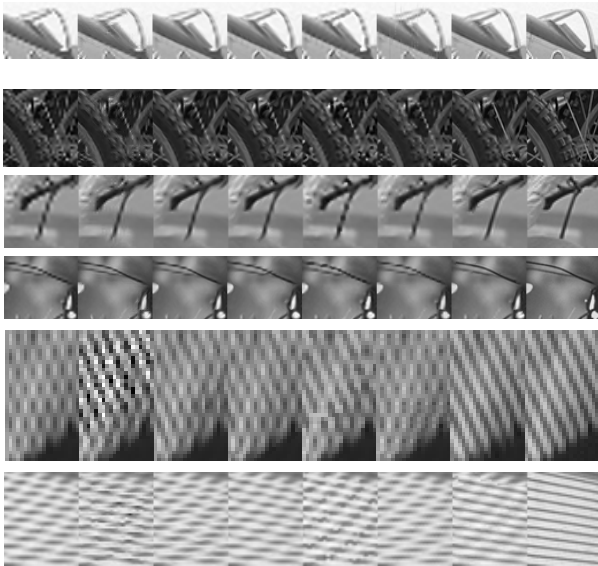


Fig. 11. Subjective comparison with other state-of-the-art  $2\times$  dedicated interpolation algorithms applied on different images with scaling factor  $s = 2$ . From left to right: bicubic, NEDI, SAI, DCC, ICBI, SAGA, the proposed method, and the original image. From top to bottom: *Airplane*, *Bike*, *Flower*, *Monarch*, *Barbara*, and *Lighthouse*.



Fig. 12. Subjective image quality comparison of reconstructed HR images by different methods in  $3\times$  enlargements. From left to right: bicubic, SAI+BC, DCC+BC, ICBI+BC, the proposed method, and the original image.

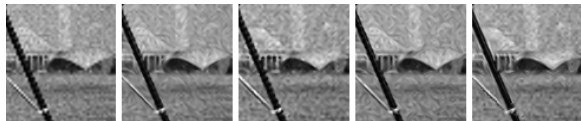


Fig. 13. Subjective image quality comparison of reconstructed HR images under noisy environment by different methods in  $2\times$  enlargement. The LR input is corrupted by Gaussian noise. From left to right: bicubic, SAI, DCC, ICBI, and the proposed method.

edges, which affect the subjective image quality. The proposed method presents the sharpest edges and best subjective quality. Fig. 12 shows the  $3\times$  enlargement results of a region of the *Airplane* image. It further shows the effectiveness of the proposed method when dealing with slight aliasing artifacts.

Since there exists a noise term in the AR model, the proposed method is also applied to noisy LR images. Fig. 13 shows the performance under the noisy environment. The proposed method produces the sharpest edge structures (the tripod leg) while maintaining textures (in the lawn) despite the noise.

Currently, the code is implemented in MATLAB. Due to the iterative process, the processing time is relatively long. To give readers an intuitive impression of the time performance of the proposed method, we list the processing time on different images and the average processing time on frames from different videos in Table VI. Theoretically, the time complexity of the proposed method is  $O(kPQ)$ , where  $k$  is a constant,  $P$  and  $Q$  are the height and width of the HR

TABLE VI  
PROCESSING TIME (SECONDS) OF DCC, ICBI, THE PROPOSED METHOD, AND ScSR ON DIFFERENT IMAGES ( $4\times$ ) AND VIDEOS ( $2\times$ )

	Name	DCC	ICBI	Proposed	ScSR
Images	<i>Lighthouse</i>	9.39	254.46	592.97	9031.00
	<i>Barbara</i>	6.20	150.94	552.15	9022.18
	<i>Monarch</i>	6.37	196.96	821.94	9022.47
	<i>Airplane</i>	6.24	221.75	1096.10	9021.35
Videos	<i>Akiyo</i>	1.13	22.02	95.39	-
	<i>Foreman</i>	1.13	20.35	119.73	-
	<i>Highway</i>	1.13	21.88	84.03	-
	<i>Harbour</i>	1.12	31.24	195.97	-

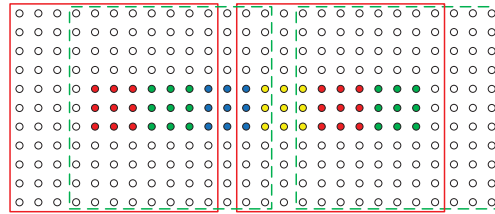


Fig. 14. Possible parallel implementation. The squares in the same color represent windows being processed in parallel in the same pass. The dots in four different colors represent output pixels in different passes.

image, respectively. In the experiment, the processing time of the proposed method is about 2–6 times slower than ICBI and much faster than the learning-based method Sparse coding Super Resolution (ScSR) [17] (under the default configuration except for the scaling factor). The processing speed can be improved by implementing the code in C and using the parallel interpolation scheme. In principle, any two nonoverlapping windows can be processed simultaneously.

For example, to promote the processing speed, the following 16-pass parallel implementation can be used. As shown in Fig. 14, the offset between adjacent windows is set to be 12 in the same pass. Fig. 14 only shows the output of four different passes in the horizontal direction. By outputting center  $3 \times 3$  pixels of every window in each horizontal and vertical pass, all the pixels (except the pixels on the boundaries of the whole image) can be processed after the 16-pass procedure. Such implementation can avoid the data dependence in the parallel process and may lead to a significant promotion of processing speed. The only difference between the original method and the parallel one is the output order, which affects the initial value of local windows. Since the method is robust to the initial value, we believe the performance of the parallel implementation should be as good as the original one.

### B. Resolution Adaptive Video Coding

To compare the performance of the proposed method and the up-sampling method in H.264/SVC [2], we have also tested the proposed method on the JSVM 9.15 codec [29]. Abundant sequences from the coding community [34] have been tested in the experiment to evaluate our method in the codec. The overall coding performance of the modified encoder is compared with the original encoder of JSVM.

In the experiment, two spatial layers are encoded. The enhancement layer was encoded with Common Intermediate

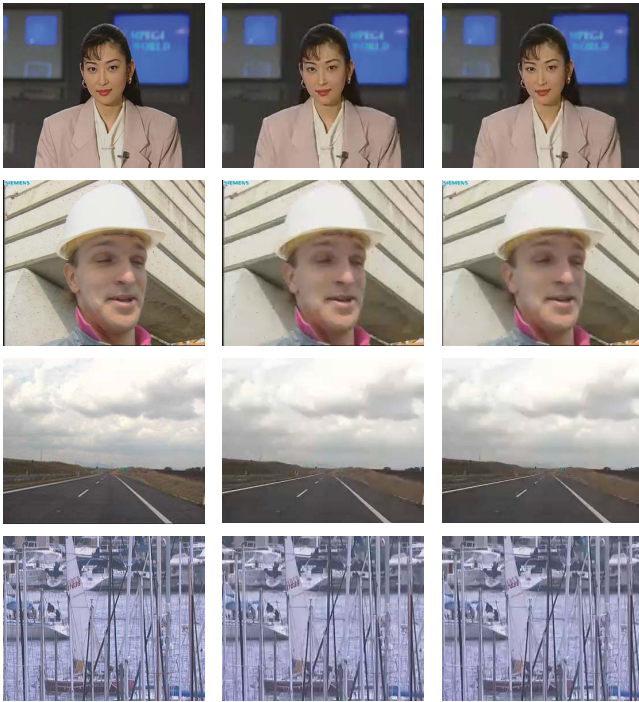


Fig. 15. Subjective quality comparison of frames in the reconstructed video sequences. From top to bottom: the second frame of *Akiyo*; the second frame of *Foreman*; the fourth frame of *Highway*; and the fourth frame *Harbor*. From left to right: original frame; the result of JSVM; and the result of the proposed method.

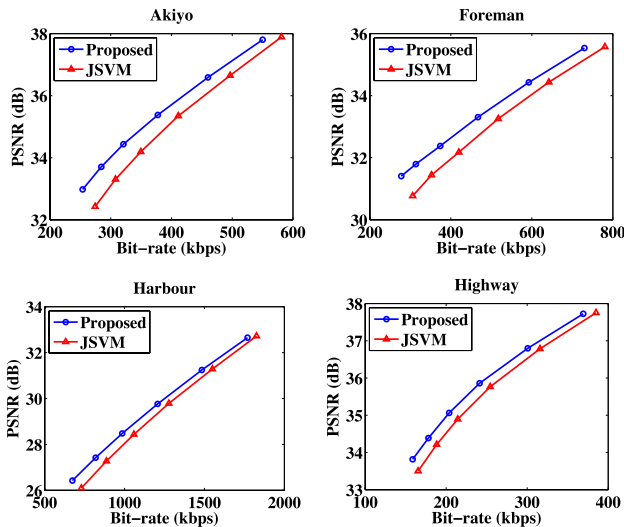


Fig. 16.  $R$ - $D$  performance of different sequences. The largest gain of the proposed method over JSVM is more than 1 dB, observed in *Akiyo* sequences.

Format sequences and the base layer was encoded with Quarter Common Intermediate Format sequences. The scaling ratio between the base layer and the enhancement layer is 2. MPEG4 13-tap dyadic down-sampling filter is utilized as the down-sampling method.

The subjective quality of frames in video sequences is compared in Fig. 15. As shown, the results of the proposed method have better subjective quality than that of JSVM. Textures are preserved by the proposed method, especially edge structures.

We set the Quantization Parameter (QP) of the base layer to be 32, and change the QP of the enhancement layer from 32 to 42 to plot  $R$ - $D$  curves for different sequences (Fig. 16). As shown, the proposed method outperforms the method in JSVM. In the experiments, it is noticed that interlayer prediction is applied in most of the frames because of the precise reconstruction of the proposed method.

## V. CONCLUSION

In this paper, we present an interpolation algorithm that is suitable for arbitrary scaling factors considering the non-stationarity statistics of natural images. The AR models are constructed by the pixel's neighbors instead of its available LR neighbors to solve the general scale interpolation problem. The similarity of pixels in the local window is exploited to alleviate the inaccurate estimation of AR models caused by the local instability of natural images. The experimental results show that our proposed algorithm outperforms the state-of-the-art methods and produces much better image quality with sharp edges. Furthermore, when tested in JSVM codec, our method also produces better subjective quality and  $R$ - $D$  performance.

## REFERENCES

- [1] C. M. Zwart and D. H. Frakes, "Segment adaptive gradient angle interpolation," *IEEE Trans. Image Process.*, vol. 22, no. 8, pp. 2960–2969, Aug. 2013.
- [2] C. A. Segall and G. J. Sullivan, "Spatial scalability within the H.264/AVC scalable video coding extension," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, pp. 1121–1135, Sep. 2007.
- [3] Z. Shi, X. Sun, and F. Wu, "Spatially scalable video coding for HEVC," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jul. 2012, pp. 1091–1096.
- [4] R. Keys, "Cubic convolution interpolation for digital image processing," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 29, no. 6, pp. 1153–1160, Dec. 1981.
- [5] Q. Wang and R. K. Ward, "A new orientation-adaptive interpolation method," *IEEE Trans. Image Process.*, vol. 16, no. 4, pp. 889–900, Apr. 2007.
- [6] A. Giachetti and N. Asuni, "Real-time artifact-free image upscaling," *IEEE Trans. Image Process.*, vol. 20, no. 10, pp. 2760–2768, Oct. 2011.
- [7] M. Li and T. Q. Nguyen, "Markov random field model-based edge-directed image interpolation," *IEEE Trans. Image Process.*, vol. 17, no. 7, pp. 1121–1128, Jul. 2008.
- [8] D. Zhang and X. Wu, "An edge-guided image interpolation algorithm via directional filtering and data fusion," *IEEE Trans. Image Process.*, vol. 15, no. 8, pp. 2226–2238, Aug. 2006.
- [9] D. Zhou, X. Shen, and W. Dong, "Image zooming using directional cubic convolution interpolation," *IET Image Process.*, vol. 6, no. 6, pp. 627–634, Aug. 2012.
- [10] X. Li and M. T. Orchard, "New edge-directed interpolation," *IEEE Trans. Image Process.*, vol. 10, no. 10, pp. 1521–1527, Oct. 2001.
- [11] X. Zhang and X. Wu, "Image interpolation by adaptive 2-D autoregressive modeling and soft-decision estimation," *IEEE Trans. Image Process.*, vol. 17, no. 6, pp. 887–896, Jun. 2008.
- [12] J. Ren, J. Liu, W. Bai, and Z. Guo, "Similarity modulated block estimation for image interpolation," in *Proc. 18th IEEE Int. Conf. Image Process.*, Sep. 2011, pp. 1177–1180.
- [13] K.-W. Hung and W.-C. Siu, "Improved image interpolation using bilateral filter for weighted least square estimation," in *Proc. 17th IEEE Int. Conf. Image Process.*, Sep. 2010, pp. 3297–3300.
- [14] K.-W. Hung and W.-C. Siu, "Robust soft-decision interpolation using weighted least squares," *IEEE Trans. Image Process.*, vol. 21, no. 3, pp. 1061–1069, Mar. 2012.
- [15] X. Zhang, S. Ma, Y. Zhang, L. Zhang, and W. Gao, "Nonlocal edge-directed interpolation," in *Proc. 10th Pacific Rim Conf. Multimedia*, Dec. 2009, pp. 1197–1207.
- [16] X. Gao, J. Zhang, F. Jiang, X. Fan, S. Ma, and D. Zhao, "High-quality image interpolation via local autoregressive and nonlocal 3-D sparse regularization," in *Proc. Vis. Commun. Image Process.*, Nov. 2012, pp. 1–6.

- [17] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Trans. Image Process.*, vol. 19, no. 11, pp. 2861–2873, Nov. 2010.
- [18] W. Dong, D. Zhang, G. Shi, and X. Wu, "Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization," *IEEE Trans. Image Process.*, vol. 20, no. 7, pp. 1838–1857, Jul. 2011.
- [19] J. Ren, J. Liu, and Z. Guo, "Context-aware sparse decomposition for image denoising and super-resolution," *IEEE Trans. Image Process.*, vol. 22, no. 4, pp. 1456–1469, Apr. 2013.
- [20] M. Irani and S. Peleg, "Motion analysis for image enhancement: Resolution, occlusion, and transparency," *J. Vis. Commun. Image Represent.*, vol. 4, no. 4, pp. 324–335, Dec. 1993.
- [21] W. Dong, D. Zhang, G. Shi, and X. Wu, "Nonlocal back-projection for adaptive image enlargement," in *Proc. 16th IEEE Int. Conf. Image Process.*, Nov. 2009, pp. 349–352.
- [22] S. Dai, M. Han, Y. Wu, and Y. Gong, "Bilateral back-projection for single image super resolution," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jul. 2007, pp. 1039–1042.
- [23] X. Wu, M. Shao, and X. Zhang, "Improvement of H.264 SVC by model-based adaptive resolution upconversion," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2010, pp. 4205–4208.
- [24] K. Tang, O. C. Au, Y. Guo, J. Pang, and J. Li, "Arbitrary factor image interpolation using geodesic distance weighted 2D autoregressive modeling," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, May 2013, pp. 2217–2221.
- [25] K. Tang, O. C. Au, Y. Guo, J. Pang, J. Li, and L. Fang, "Arbitrary factor image interpolation by convolution kernel constrained 2-D autoregressive modeling," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2013, pp. 996–1000.
- [26] J. B. Rosen, H. Park, and J. Glick, "Structured total least norm for nonlinear problems," *SIAM J. Matrix Anal. Appl.*, vol. 20, no. 1, pp. 14–30, 1998.
- [27] J. Sun, J. Zhu, and M. F. Tappen, "Context-constrained hallucination for image super-resolution," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 231–238.
- [28] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2005, pp. 60–65.
- [29] *Joint Video Team JSVM Reference Software*. [Online]. Available: [http://ip.hhi.de/imagecom\\_G1/savce/downloads/SVC-Reference-Software.htm](http://ip.hhi.de/imagecom_G1/savce/downloads/SVC-Reference-Software.htm), accessed 6, Jan. 2011.
- [30] *True Color Kodak Images*. [Online]. Available: <http://r0k.us/graphics/kodak/>, accessed 27, Jan. 2013.
- [31] *The USC-SIP1 Image Database*. [Online]. Available: <http://sipi.usc.edu/database/>, accessed 22, Aug. 2014.
- [32] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
- [33] Z. Wang and A. C. Bovik, "Mean squared error: Love it or leave it? A new look at signal fidelity measures," *IEEE Signal Process. Mag.*, vol. 26, no. 1, pp. 98–117, Jan. 2009.
- [34] *Derf's Test Media Collection*. [Online]. Available: <http://media.xiph.org/video/derf/>, accessed 1, May 2013.



**Mading Li** was born in Toronto, ON, Canada, in 1991. He received the B.S. degree in computer science from Peking University, Beijing, China, in 2009, where he is currently working toward the Ph.D. degree with the Institute of Computer Science and Technology.

His research interests include image and video processing, image interpolation, and error concealment.



**Jiaying Liu** (S'09–M'10) received the B.E. degree in computer science from Northwestern Polytechnic University, Xi'an, China, in 2005 and the Ph.D. (Hons.) degree in computer science from Peking University, Beijing, China, in 2010.

She was a Visiting Scholar with University of Southern California, Los Angeles, CA, USA, from 2007 to 2008. She is currently an Associate Professor with Institute of Computer Science and Technology, Peking University. Her research interests include image processing, sparse signal representation, and video compression.



University in 2013.

**Jie Ren** was born in Shenyang, China, in 1984. He received the B.S. (Hons.) degree from Beijing Normal University, Beijing, China, in 2007 and the Ph.D. (Hons.) degree in computer science from Peking University, Beijing, in 2013.

He is a Deputy Researcher with the Software Research Center Beijing, Ricoh Inc., Beijing. His research interests include image and video processing and stereo vision-based recognition.

Dr. Ren received the Distinguished Doctoral Dissertation Award in Computer Science from Peking



**Zongming Guo** (M'09) received the B.S. degree in mathematics, and the M.S. and Ph.D. degrees in computer science from Peking University, Beijing, China, in 1987, 1990, and 1994, respectively.

He is a Professor with Institute of Computer Science and Technology, Peking University. His research interests include video coding, processing, and communication.

Dr. Guo is an Executive Member of the China-Society of Motion Picture and Television Engineers. He received the Prize of the State Administration of Radio Film and Television Award in 2004, the Prize of the Ministry of Education Science and Technology Progress Award in 2006, the Prize of the National Science and Technology Award in 2007, the Wang Xuan News Technology Award and the Chia Tai Teaching Award in 2008, the Government Allowance granted by the State Council in 2009, and the Distinguished Doctoral Dissertation Advisor Award of Peking University in 2012 and 2013.