Isophote-Constrained Autoregressive Model With Adaptive Window Extension for Image Interpolation

Wenhan Yang, Jiaying Liu, *Member, IEEE,* Mading Li, and Zongming Guo, *Member, IEEE*

***Abstract*— The autoregressive (AR) model is widely used in image interpolations. Traditional AR models consider utilizing the dependence between pixels to model the image signal. However, they ignore the valuable patch-level information for image modeling. In this paper, we propose to integrate both the pixel-level and patch-level information to depict the relationship between high-resolution and low-resolution pixels and obtain better image interpolation results. In particular, we propose an isophote-constrained AR (ICAR) model to perform AR-flavored interpolation within an identified joint stable region and further develop an AR interpolation with an adaptive window extension. Considering the smoothness along the isophote curve, the ICAR model searches only several successive similar patches along the isophote curve over a large region to construct an adaptive window. These overlapped patches, representing the patch-level structure similarity, are used to construct a joint AR model. To better characterize the piecewise stationarity and determine whether a pixel is suitable for AR estimation, we further propose pixel-level and patch-level similarity metrics and embed them into the ICAR model, introducing a weighted ICAR model. Comprehensive experiments demonstrate that our method can effectively reconstruct the edge structures and suppress jaggy or ringing artifacts. In the objective quality evaluation, our method achieves the best results in terms of both peak signal-to-noise ratio and structural similarity for both simple size doubling (two times) and for arbitrary scale enlargements.**

***Index Terms*— Autoregressive (AR) model, interpolation, isophote, patch-level dependence, pixel-level dependence, similarity metric.**

I. Introduction

NOWADAYS, users employ a variety of devices with diverse display resolutions to browse images and watch videos; consequently, it is imperative to be able to adjust the display size of images or videos to conform to the user's screen resolution. Moreover, people typically use mobile devices to capture pictures and videos that have limited resolution. To produce high-resolution (HR) images from low-resolution (LR) images requires a common but effective technique called image interpolation.

In the past few decades, numerous image interpolation techniques have been proposed that can be roughly classified into three categories: polynomial-based, explicitly adaptive, and implicitly adaptive. The polynomial-based methods, such as Bilinear, Bicubic [1], and Cubic Spline [2], interpolate images by convolving all the image pixels using a predefined kernel, and they are both easy to implement and highly efficient. However, because these methods use fixed-form prediction functions, they cannot be adaptive to various local structures in natural images. In other words, the interpolation process of polynomial-based methods does not align well with the geometric information in images. Therefore, artifacts (e.g., blurring, ringing, jaggies, zippering, and so forth) inevitably appear at the boundaries of different regions, leading to degraded results.

To solve the aforementioned problem, explicitly adaptive methods [3], [4] have been developed that take local structural information into consideration. Local structural information, such as that of edges and local covariance, is important for describing various statistics of an image, such as sharpness or stationarity. Explicitly adaptive methods identify the directions of edges or isophotes explicitly and interpolate along these directions, where the isophote is a constant intensity path. Guided by the local structural information, the results of these methods preserve more structural details such as sharp edges. For example, Wang and Ward [3] proposed constructing a parallelogram for a local region containing an edge where one side of the parallelogram parallels the estimated isophote and the parallelogram encloses the interpolated pixel. Then, the interpolation is performed within the parallelogram. In [4], the segment adaptive gradient angle interpolation (SAGA) builds an isophote lattice and interpolates along it to preserve the structural details of edges or isophotes. Moreover, this method does not require the relative positions between the estimated HR pixels and LR pixels to be fixed. Consequently, general scale enlargements are also supported by this type of method.

However, all these methods suffer from estimation errors on edge directions. Therefore, explicitly adaptive methods with soft edge estimations, which depict the edge direction as a probability distribution rather than a fixed direction, were proposed in [5]–[7]. Li and Nguyen [5] used a length-16 weighting vector to formulate the geometric reg-

Manuscript received February 23, 2016; revised October 3, 2016 and November 17, 2016; accepted December 3, 2016. Date of publication December 13, 2016; date of current version May 4, 2018. This work was supported in part by the National Natural Science Foundation of China under Contract 61472011 and in part by the National Key Technology Research and Development Program of China under Grant 2015AA011605. This paper was recommended by Associate Editor X. Li. *(Corresponding author: Jiaying Liu.)*

W. Yang, J. Liu, and M. Li are with Institute of Computer Science and Technology, Peking University, Beijing 100871, China (e-mail: yangwenhan@pku.edu.cn; liujiaying@pku.edu.cn; martinli0822@pku.edu.cn).

Z. Guo is with the Institute of Computer Science and Technology, Peking University, Beijing 100871, China, and also with the Cooperative Medianet Innovation Center, Shanghai 200240, China (e-mail: 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.2016.2638864

ularity constraint, smoothness and sharpness of edges, and the constraint is imposed on the interpolated image through a Markov random field (MRF). Zhang and Wu [6] proposed interpolating missing HR pixels by fusing the estimations along two orthogonal directions via linear minimum mean square error (MSE) estimation. Zhou *et al.* [7] suggested an edge-directed cubic convolution interpolation scheme that can adapt to the varying edge structures of images. More concretely, the method estimates salient edges (i.e., edges with high contrast on each side) that pass through interpolated pixels to guide the interpolation process for those pixels.

Although explicitly adaptive methods improve interpolation quality to some extent, they do not exploit the statistical information of image features, e.g., the covariance of pixels in a local window. Thus, implicitly adaptive methods were developed. Those methods incorporate statistical priors into the objective function and interpolate images by minimizing the error. For instance, according to the geometric duality, i.e., the consistency between the LR and HR covariances, the new edge-directed interpolation (NEDI) method proposed in [8] first employs LR images to estimate HR covariances by solving a least squares problem. Then, it estimates the HR pixels using their neighboring LR pixels as well as the corresponding covariances. That study was the seminal work for developing autoregressive (AR) model-based interpolation methods. Inspired by NEDI, many improved algorithms have been proposed. However, in the high-frequency regions, where support is not available to provide sufficient information for the least squares covariance estimation, the NEDI solution is overconstrained and results in an unstable solution. To overcome this problem, the iterative curve-based interpolation (ICBI) method [9] forces the AR parameters in opposite directions to be equal and uses the second-order information to remove the artifacts while preserving edges and textures. Other advanced methods [10]–[13] tend to avoid accumulating errors and instead enhance the stability by introducing filters or by estimating the covariance using an offline dictionary.

Soft-decision adaptive interpolation (SAI) [14] extended the framework of NEDI by adding an additional cross-directional AR constraint and then obtaining the final interpolation by solving a constrained least squares problem. Because it benefits from fully exploiting the joint relationships between the HR and LR pixels, SAI's performance was a significant improvement. SAI assumes that the AR parameters within a local window are the same. However, this assumption does not hold in areas with edges or rich structures. To improve the accuracy and the robustness of SAI, robust soft-decision interpolation (RSAI) [15], [16] was proposed to incorporate weights for all the residuals in the SAI cost function. Hung and Siu [17] used a bilateral filter to replace the least squares model to estimate the parameters. Tang *et al.* [18] presented an AR-based interpolation weighted by the pixelwise geodesic distance (PIGD) that considered both spatial distances and color differences. To accommodate interpolation at an arbitrary scale, Li *et al.* [19] presented an adaptive general scale interpolation (AGSI) method. The AR terms in AGSI are modeled between pixels and their adjacent unknown HR neighbors instead of their fixed LR neighbors. A weighting scheme suitable for arbitrary scaling situations based on pixel similarity is embedded into the model to increase the accuracy of the estimation. In sum, the implicitly adaptive methods mentioned earlier include neighborhood and edge information in an energy function. By incorporating the local structural information, these methods yield more adaptive and natural results. However, they exploit only ***pixel-level dependence***. Pixel-level dependence provides local structural information but is limited to the area of an interpolation window; it reflects the stationarity characteristics in the image. In nonstationary regions, the pixel-level dependencies are changing and are difficult to estimate. Therefore, another type of constraint called ***patch-level dependence***—which uses the statistics of salient features that recur within an image—has gained attention from researchers for image modeling. Patch-level dependence has been shown to be effective in image reconstruction related tasks, such as image denoising [20], superresolution [21], and completion [22].

Patch-based interpolations utilize patch-level dependence information to model the statistics of natural images by jointly estimating the HR pixels of a cube that consists of similar patches. The dictionary prior is exploited to form a constraint of the optimization function for the estimation. This type of the interpolation method is more closely related to the sparse representation-based image superresolution methods [21], [23], [24] than to the traditional interpolation methods. In [25], with the help of Taylor dictionaries, the fusion of patterns in nonlocal similar patches is constrained by local image structures, while the redundancy from those patterns enables more accurate estimation for the structural regression. In [26] and [27], sparse representation dictionary learning-based methods were proposed by considering nonlocal AR modeling (NARM) for image interpolation. Due to the introduction of the patch-level prior into the estimation, these methods achieve a desirable interpolation performance. However, compared with traditional interpolations, the computational and space complexities of dictionary-based interpolations are both much higher. This type of method also consumes extra storage. Therefore, they cannot be easily applied in most typical video codecs, postprocessing chip designs, and so on. In addition, these methods strongly heavily on the availability of similar patches in an external database or on nonlocal similar patches within the image itself. However, when such similar patches are not available, the interpolation results degrade significantly.

Recently, example-based methods have been proposed to learn the LR-to-HR mapping, namely, by learning the general dependencies of natural images from very large training sets instead of modeling the image signal from a local perspective. These methods include MRF fusion [28], local self-examples [29], edge statistics [30], support vector regression [31], kernel ridge regression [32], multiscale similarity [33], sparse representation [21], [34], simple function mapping [35], anchor neighbor regression [36], [37], and random forest [38], among others. Using the learned priors from natural images, these methods achieve very promising results in both visual quality and time efficiency. However, these models require a large number of parameters that occupy significant

amounts of storage. Moreover, training these methods requires large amounts of resources: large collected training sets and a long training period. These drawbacks may undermine their practicability to some extent, especially in extreme cases, such as aerial or satellite photography.

In this paper, we focus on designing a more locally adaptive and general computational and storage efficient interpolation model from the perspective of local signals, following the route of AR-flavored methods. We aim at developing an interpolation approach that makes full use of both the pixel-level and the patch-level dependence with low cost in a traditional manner, but that achieves results comparable to the dictionary-based methods. In this paper, using the guidance of the isophote, we propose to incorporate successive similar patches into the AR model for interpolation. The *isophote*, a constant intensity path, is effective in modeling both the pixel-level and the patch-level dependence. The pixel intensity remains the same along the isophote but deviates in the vertical direction. It also identifies the locations where similar patches exist, providing a more efficient way to search for similar patches. With the guidance of an isophote, we successively collect several neighboring similar patches along the isophote. Pixels in these patches are spatially autocorrelated; we integrate that autocorrelation into a joint estimation problem and develop a novel AR model—the isophote-constrained AR (**ICAR**) model. The ICAR model combines both the pixel-level dependencies and the patch-level dependencies among several successive neighboring patches. Through an appropriate simplification, ICAR can be converted to a traditional AR model constructed in an adaptive window extended from a fixed window. Without resorting to complicated dictionary priors, our method achieves state-of-the-art performance while keeping the computational complexity low.

The main contributions of this paper are summarized as follows.

1) Guided by the estimated isophote, we construct an adaptive window by selecting successive neighboring similar patches to model the patch-level dependence. Combining the patch-level dependence of the joint region and the pixel-level dependence of the traditional AR model, we propose a novel ICAR.
2) To reliably measure the merit of each pixel in the local window to the estimation improvement of the ICAR model, a joint pixel-level and patch-level similarity metric is presented, based on the patchwise geodesic distance (PAGD) and the patch difference, respectively. Incorporated with the joint metric, the weighted ICAR model not only depicts the local stationarity of pixels but also maintains the structural consistency of successive neighboring similar patches.
3) Because the optimization problem involved in solving the ICAR model is overconstrained, we relax the constraints and convert the model to an equivalent but simpler interpolation method: adaptive window-extension-based AR interpolation. Closed-form solutions can be derived for the objective functions for the interpolation.

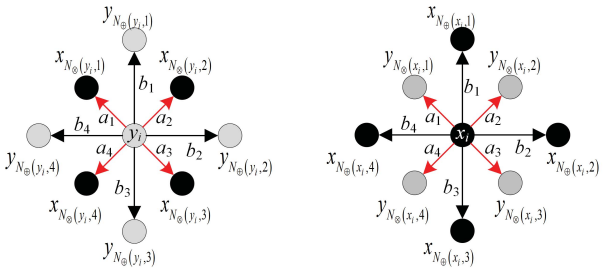


Fig. 1. Notations for the neighboring pixels in the diagonal and cross directions. The gray and black circles represent the HR and LR pixels, respectively. Red arrows: AR relationship in the diagonal directions. Black arrows: AR relationship in the cross directions.

The rest of this paper is organized as follows. Section II introduces the ICAR model. Section III presents the pixel-level and patch-level joint similarity metric and their embedded ICAR model. Section IV explores the underdetermined problem in the ICAR model and addresses it by substituting the ICAR constraint with a more relaxed constraint. Experimental results are presented in Section V and concluding remarks are given in Section VI.

II. Isophote-Constrained Autoregressive Model

In this section, we first review the traditional AR model, which is considered as a representation of pixel-level local image structures. Then, we construct an ICAR, incorporating the patch-level prior into AR modeling. The joint ICAR model not only depicts the local pixel stationarity but also maintains the structural consistency of successive neighboring similar patches.

A. Autoregressive Model for Interpolation

In traditional AR, the value of a pixel is assumed to be correlated with its neighboring pixels; thus, a missing pixel can be predicted by using the correlation information from its neighbors. Formally, let x be a pixel of the input LR image $\mathbf{X}$ and y be the corresponding pixel of the original HR image $\mathbf{Y}$. Given $\mathbf{X}$, image interpolation aims at estimating $\mathbf{Y}$. For a pixel x or y, we use $N_\otimes$ and $N_\oplus$ to index the neighboring pixels in the diagonal and cross directions, respectively. As shown in Fig. 1, the diagonal directions of a pixel are: top-left, top-right, bottom-right, and bottom-left and the cross direction neighbors are the four connected neighbors. $t \in \{1, 2, 3, 4\}$ represents the index of neighbors in the AR model. In Fig. 1, $y_{N_\otimes(x_i,t)}$ and $x_{N_\otimes(y_i,t)}$ represent the tth neighbor of x_i and y_i in the diagonal direction, respectively, while $x_{N_\oplus(x_i,t)}$ and $y_{N_\oplus(y_i,t)}$ represent the tth neighbor of x_i and y_i in the cross direction, respectively. In particular, note that the diagonal neighboring pixels of x_i are HR pixels, while the diagonal neighboring pixels of y_i are LR pixels. The notations $\boldsymbol{a} = [a_1, a_2, a_3, a_4]^T$ and $\boldsymbol{b} = [b_1, b_2, b_3, b_4]^T$ are AR model parameters that depict the covariances between a pixel and its neighbors in the diagonal and cross directions, respectively. The notations $\sigma_i^\oplus$ and $\sigma_i^\otimes$ refer to random perturbations that are independent of the spatial location and the image signal. These perturbations account for both the fractal-like fine details of the image signal and the measurement noise.

Assuming that the AR models for generating HR or LR pixels in a certain direction share the same model parameters in the local space, the AR model for a pixel in the diagonal direction and the cross direction can be represented as follows:

$$x_i = \sum_{t=1}^{4} a_t y_{N_{\otimes}(x_i,t)} + \sigma_i^{\otimes}, \quad y_i = \sum_{t=1}^{4} a_t x_{N_{\otimes}(y_i,t)} + \sigma_i^{\otimes} \tag{1}$$

$$x_i = \sum_{t=1}^{4} b_t x_{N_{\oplus}(x_i,t)} + \sigma_i^{\oplus}, \quad y_i = \sum_{t=1}^{4} b_t y_{N_{\oplus}(y_i,t)} + \sigma_i^{\oplus}. \tag{2}$$

According to (1), the matching error ℓ_m of the modeled pixels can be represented by

$$\ell_{\mathrm{m}} = \sum_{i\in W}\left(y_i - \sum_{t=1}^{4} a_t x_{N_{\otimes}(y_i,t)}\right)^2 + \sum_{i\in W}\left(x_i - \sum_{t=1}^{4} a_t y_{N_{\otimes}(x_i,t)}\right)^2 \tag{3}$$

where W is a square window containing the pixels to be interpolated and a_t is assumed to be fixed in the local interpolation window. The matching error measures the difference between the estimated values of interpolated pixels and the predicted results from the regressions of their neighboring pixels. Based on this concept, the matching error in the square interpolation window can be minimized by a joint estimation framework to generate the best prediction for the HR pixel

$$\min_{\{y_i\}} \left\{ \sum_{i\in W}\left(y_i - \sum_{t=1}^{4} a_t x_{N_{\otimes}(y_i,t)}\right)^2 + \sum_{i\in W}\left(x_i - \sum_{t=1}^{4} a_t y_{N_{\otimes}(x_i,t)}\right)^2 \right\}. \tag{4}$$

In traditional AR interpolation, the local window W used to calculate the matching error is a fixed square, and the image signal is assumed to be stationary within W. Different from (1) in which both the HR and LR pixels are modeled simultaneously, (2) builds the relationships among the HR pixels and LR pixels separately. Thus, (2) is often used as constraints to avoid overfitting.

B. Isophote-Constrained AR

Based on the assumption that the image is stationary locally, all the AR models of the same type in the entire local window share the same AR parameters. Then, the AR models can impose appropriate constraints on the window and produce fairly good image reconstructions. However, natural images do not maintain stability in most local windows. For example, as shown in Fig. 2, there are significant differences between an edge-crossing area and a smooth area in a local window. Thus, estimations by AR models in this window are not robust. Some pixels in the fixed squared window do not contribute to the correct AR model estimation, while some pixels outside the window actually have quite similar AR relationships. Therefore, the size and the shape of the window must be more flexible, and a screening method should be introduced to import some useful pixels such that an adaptive window area can be composed of patches with similar and consistent structures.

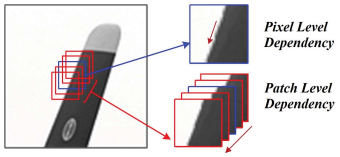


Fig. 2. Illustrations for the two types of dependencies exploited in the proposed ICAR. The pixel-level dependence (blue border) describes the statistical properties among pixels within the center patch, while the patch-level dependence (red borders) among similar neighboring patches models the variational trend across patches.

We propose a novel ICAR model. The joint estimation is extended from a single patch to multiple patches. Guided by the estimated isophote, successive adjacent similar patches along the isophote should have similar AR parameters. The isophote is estimated as the local path along which the sum of the differences between the successive adjacent patches is the smallest. Then, the region consisting of similar patches along the isophote is constructed for the joint estimation. The isophote reflects the local structural similarity and maintains the local consistency of the AR parameters within the local adaptive window. As described in Fig. 2, along the isophote, the basic patch (in the blue frame) and its similar patches (in red frames) are considered as a whole for the estimation. Pixels in the basic patch and similar patches are estimated jointly. The goal is to minimize the overall matching error using a least squares problem as follows:

$$\min_{\{y_i\}} \left\{ \sum_{k=1}^{n}\sum_{i\in W_k}\left[\left(y_i - \sum_{t=1}^{4} a_t x_{N_{\otimes}(y_i,t)}\right)\right]^2 + \sum_{k=1}^{n}\sum_{i\in W_k}\left[\left(x_i - \sum_{t=1}^{4} a_t y_{N_{\otimes}(x_i,t)}\right)\right]^2 \right\} \tag{5}$$

while satisfying the following constraint:

$$\sum_{k=1}^{n}\sum_{i\in W_k}\left[\left(x_i - \sum_{t=1}^{4} b_t x_{N_{\oplus}(x_i,t)}\right)\right]^2 \approx \sum_{k=1}^{n}\sum_{i\in W_k}\left[\left(y_i - \sum_{t=1}^{4} b_t y_{N_{\oplus}(y_i,t)}\right)\right]^2 \tag{6}$$

where W_k is the kth neighboring patch of the current interpolated patch on the isophote, and n is the number of patches in the estimated joint region. With the AR relationship in cross directions, (6) forces the variance of the distribution of x_i to be equal to that of y_i to avoid overfitting.

III. Pixel/Patch-Level Dependence-Based Similarity Metric

Piecewise stationarity is one of the intrinsic characteristics of natural images, and it reflects the local consistency of image patterns, such as long edges or continuous textures. An interpolation window for constructing the AR model estimations may contain several stationary regions. Thus, pixels in different stable regions benefit from AR estimation differently. To further depict the piecewise stationarity and measure whether a pixel is suitable for estimating

the AR model, we propose two similarity metrics based on the pixel-level dependence and the patch-level dependence, respectively, and use those metrics as weighting terms to constrain the AR model. Then, we present a weighted ICAR model that uses the two weighting terms.

A. Pixel-Level Metric Based on Patchwise Geodesic Distance

The stationarity assumption is that the AR parameters in an image will be similar within a local area. Based on this assumption, AR models pixel interpolations in a local window by using the same parameters. However, due to the diversity of natural image contents, the stationarity assumption may not hold true even within a very small image region. Therefore, with the uniform stationarity, piecewise stationarity is more reliable. Piecewise stationarity assumes that an image is composed of several connected and stable regions. The statistics do not change within a stable region but they do change across the boundaries of different regions. Applied to interpolation, a pixel has strong correlations with pixels in the same *stable region*, but weaker relations with pixels outside of that stable region, regardless of the distance between them. To determine whether two examined pixels are in the same stable region, a pixel-level metric based on bilateral distance was proposed in [39]. The basic idea for designing the appropriate distance metric is to calculate the spatial correlation within a local window from the following three aspects.

1) *Spatial Proximity:* Nearby pixels tend to be in the same stable region.
2) *Neighboring Similarity:* Pixels with similar neighboring structural patterns tend to be in the same stable region.
3) *Intensity Differences:* Pixels with similar intensities tend to be in the same stable region.

A suitable distance used to calculate the pixel-level similarity metric should satisfy the above-mentioned three criteria. We propose a PAGD. Before that, we first present two widely used distance metrics: bilateral distance and PIGD.

The bilateral distance used in [39] combines three terms to measure the relationships between pixels. Let c and p denote the center pixel and one arbitrary pixel within the same patch, respectively. Let D denote the distance defined for pixel-level similarity. The bilateral distance D^b is then calculated as follows:

$$\begin{aligned} D^b(p,c) &= D^i(p,c) + \alpha_1 D^n(p,c) + \alpha_2 D^l(p,c) \\ D^n(p,c) &= ||N(p) - N(c)||_2^2 \\ D^i(p,c) &= ||I(p) - I(c)||_2^2 \\ D^l(p,c) &= ||L(p) - L(c)||_2^2. \end{aligned} \tag{7}$$

Here, D^n, D^i, and D^l represent the neighboring similarity, the intensity difference, and the spatial proximity, respectively. For a given pixel, $N(\cdot)$ is a 1D neighboring structure pattern vector consisting of the intensity of eight-connected neighborhood pixels, $I(\cdot)$ contains the intensity, and $L(\cdot)$ holds the spatial coordinates indicating the spatial locations. The weights α_1 and α_2 balance the importance of these three terms.

The method proposed in [18] uses the PIGD to measure the probability of two pixels being correlated. The basic idea behind PIGD is to figure out whether two pixels are in the same connected component, that is, a region consisting of pixels with similar intensities. This measure incorporates both the spatial distance and the intensity distance between two pixels; thus, it is relatively robust to outliers. The PIGD $D^g(p,c)$ is defined as the length of the shortest path connecting p and c

$$D^g(p,c) = \min_{P=(p_1,\ldots,p_n)\in\mathcal{P}(p,c)} \sum_{k=2}^{n} |I(p_k) - I(p_{k-1})| \tag{8}$$

where $p_1 = p$, $p_n = c$, and $\mathcal{P}(p,c)$ is the set containing all paths connecting p and c. For a given path P, p_k is a pixel located in $P = (p_1, \ldots, p_n)$ and $I(p_k)$ is the intensity of p_k.

One can observe that the PIGD satisfies two of the three criteria listed earlier: the spatial closeness and the intensity difference. It is worth noting that the stable region and the connected component are different, although in many scenarios, the statistics remain stable in the connected component. For example, in Fig. 3, the two pixels A and B are on the boundary of the same component but in different local structures. Although the PIGD between them is small, they cannot be interpolated using the same AR parameters. To handle the failures of the PIGD, the neighboring structure pattern is introduced when accumulating the differences along paths. To enable the distance calculation to be capable of determining whether two pixels have similar local structures, we, therefore, present a more powerful measurement—PAGD. PAGD measures the difference between patches rather than between pixels. Thus, it has the capacity of better representing the structural pattern and modeling the stable region. By taking the neighboring similarity into account, PAGD sums up the differences of patches centered at the pixels along a path

$$D^p(p,c) = \min_{P\in\mathcal{P}_{p,c}} \sum_{i=2}^{n} \|M(p_i) - M(p_{i-1})\|_1 \tag{9}$$

where $M(\cdot)$ is the operator that extracts a local patch centered at a pixel as a vector and L_1 norm is used to sum the difference between two neighboring patches, such as the difference between $M(p_i)$ and $M(p_{i-1})$. The PAGD accumulates the minimum values of the patch differences along all the connected paths. A large geodesic distance means that there are no short connected paths between the corresponding patches of the given pixels p and c.

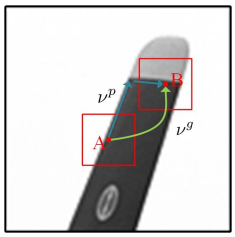


Fig. 3. AR parameters of pixels A and B may be different even if the PIGD between them is small. The pattern information around the pixels helps to handle the situation.

The distance is then converted to pixel-level similarity[1]

$$\nu(p,c) = \exp\left\{\frac{-D^p(p,c)}{\beta}\right\} \tag{10}$$

where β is a user-defined parameter that controls the shape of the exponential function. Details about the parameter settings are discussed in Section V-B. Similarly, it is also possible to calculate ν with D^b and D^g. Depending on whether p is an HR or LR pixel, ν is notated as ν^H or ν^L, respectively. These are incorporated into the objective function (12) in Section III-C for the image interpolation. In practice, the PAGD usually makes the metric more reasonable by incorporating the patch-level neighboring similarity.

B. Patch-Level Metric Based on Patch Distance

The pixel-level metric measures whether two pixels are in the same stable region and further share the same AR parameters. Comparatively, guided by the intuition that two pixels located in two similar patches should have similar AR parameters, a patch-level metric is proposed to measure the similarity between the AR parameters of two pixels based on whether one patch containing a given pixel is similar to one of the patches that contains the other given pixel. We calculate the patch distances between compared patches, which refer to the rectangle regions used as the basic unit for comparison, to represent the patch-level similarity. The patch distances are calculated between the *center compared patch* centered at the HR pixel currently being estimated and other compared patches. For any given pixel, there are many compared patches covering it. The minimum distance between those patches and the center compared patch is used to calculate the patch-level metric. The MSE is applied to calculate the patch difference. The patch-level similarity metric ω_i is defined as

$$\omega_i = \begin{cases} 1, & \text{if } p \in P_B \\ \exp\left\{-\dfrac{\min ||P_C(k) - P_B||_2^2}{\gamma}\right\}, & \forall k, \quad \text{s.t. } p \in P_C(k) \\ & \wedge\ p \notin P_B \end{cases} \tag{11}$$

where p is a pixel in the interpolation window, γ is a controlling parameter, set to 2×10^4 in our method, and $P_C(k)$ is the kth compared patch. If a pixel is in the center of the compared patch P_B, the patch distance of p is zero and its similarity metric is computed as 1. Otherwise, the similarity metric of p is set to the value calculated with the minimum MSE between the center compared patch and other compared patches containing p.

C. Weighted ICAR Model

To model the piecewise stationarity in natural images with the pixel-level dependence and the patch-level dependence, we incorporate the proposed patch-level and pixel-level similarity metrics into the ICAR model as weighting terms to improve the estimation accuracy of the AR model. More concretely, the matching error of the pixels within a joint region in (5) and (6) is incorporated with the weighting terms. Thus, the interpolation is represented as a weighted least squares estimation

$$\begin{aligned} \min_{\{y_i\}} & \left\{ \sum_{k=1}^{n} \omega_k \sum_{i \in W_k} \nu_i^H \left(y_i - \sum_{t=1}^{4} a_t x_{N_\otimes(y_i,t)} \right)^2 \right. \\ & \left. + \sum_{k=1}^{n} \omega_k \sum_{i \in W_k} \nu_i^L \left(x_i - \sum_{t=1}^{4} a_t y_{N_\otimes(x_i,t)} \right)^2 \right\} \\ \text{s.t. } & \sum_{k=1}^{n} \omega_k \sum_{i \in W_k} \nu_i^L \left(x_i - \sum_{t=1}^{4} b_t x_{N_\oplus(x_i,t)} \right)^2 \\ & \approx \sum_{k=1}^{n} \omega_k \sum_{i \in W_k} \nu_i^H \left(y_i - \sum_{t=1}^{4} b_t y_{N_\oplus(y_i,t)} \right)^2 \end{aligned} \tag{12}$$

where W_k is the kth neighboring patch of the current interpolated patch in the isophote direction, n is the number of patches in the estimated joint region, ω_k is the similarity metric based on the patch-level dependence that measures the similarity between the current interpolated patch and W_k, and ν_i^H and ν_i^L are the similarity metrics obtained based on the pixel-level dependence that measure the similarity between the center interpolated pixel and the HR and LR pixels, respectively. The problem in (12) is a constrained least squares problem. By introducing the Lagrange multiplier λ, the optimization problem is reformulated as follows:

$$\begin{aligned} \min_{\{y_i\}} & \left\{ \sum_{k=1}^{n} \omega_k \sum_{i \in W_k} \nu_i^H \left(y_i - \sum_{t=1}^{4} a_t x_{N_\otimes(y_i,t)} \right)^2 \right. \\ & + \sum_{k=1}^{n} \omega_k \sum_{i \in W_k} \nu_i^L \left(x_i - \sum_{t=1}^{4} a_t y_{N_\otimes(x_i,t)} \right)^2 \\ & + \lambda \left[\sum_{k=1}^{n} \omega_k \sum_{i \in W_k} \nu_i^H \left(y_i - \sum_{t=1}^{4} b_t y_{N_\oplus(y_i,t)} \right)^2 \right. \\ & \left. \left. - \sum_{k=1}^{n} \omega_k \sum_{i \in W_k} \nu_i^L \left(x_i - \sum_{t=1}^{4} b_t x_{N_\oplus(x_i,t)} \right)^2 \right] \right\}. \end{aligned} \tag{13}$$

By minimizing (13), the missing HR pixels in the joint region can be estimated. Its local optimal solution is given by Karush–Kunhn–Tucker conditions [40]. In practice, however, it is not necessary to solve for exact solutions. Setting λ to 0.5 provides a good tradeoff between loss and penalty terms and usually gives a reasonably good solution with much lower computational complexity. To observe the weighting terms in the objective function, we reformulate (13) as follows:

$$\begin{aligned} \min_{\{y_i\}} & \left\{ \sum_{k=1}^{n} \sum_{i \in W_k} \omega_k \nu_i^H \left(y_i - \sum_{t=1}^{4} a_t x_{N_\otimes(y_i,t)} \right)^2 \right. \\ & + \sum_{k=1}^{n} \sum_{i \in W_k} \omega_k \nu_i^L \left(x_i - \sum_{t=1}^{4} a_t y_{N_\otimes(x_i,t)} \right)^2 \\ & \left. + \lambda \sum_{k=1}^{n} \sum_{i \in W_k} \omega_k \nu_i^H \left(y_i - \sum_{t=1}^{4} b_t y_{N_\oplus(y_i,t)} \right)^2 \right\}. \end{aligned} \tag{14}$$

[1]The proposed similarity metric in this section is used to measure the pixel-level property, although the corresponding distance may be calculated with patch-level information.

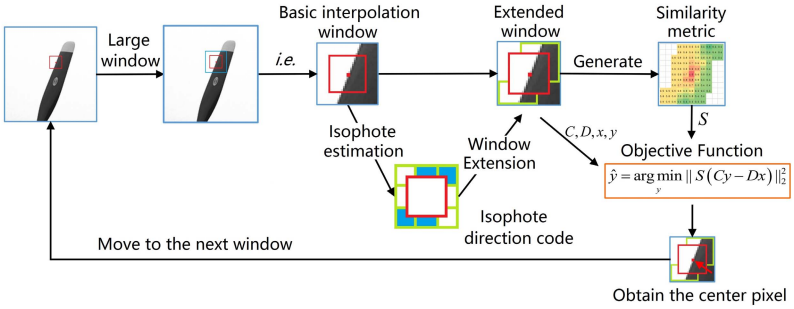


Fig. 4. Flowchart of the proposed method.

Here, the constant term in (13) has been removed, under the circumstance that $\lambda = 0.5$. The weighting terms based on the pixel-level dependence and the patch-level dependence, such as $\omega_k v_i^H$ and $\omega_k v_i^L$, jointly measure whether a pixel in (14) shares the same AR parameters as the current interpolated HR pixel.

IV. Adaptive Window-Based AR Interpolation: A Relaxed ICAR

We introduced the ICAR model in Section III. Next, we apply the model to solve interpolation problems in practice. A relaxed method—adaptive window-based interpolation—is described from the AR parameter estimation to its implementation.

A. Adaptive Window-Based Interpolation: A Relaxed ICAR

Based on (14), a closed-form solution for HR pixels $\mathbf{y}$ is given that can theoretically be used for the interpolation. However, when patches involved in the joint estimation are very similar, the optimization function in (14) is overconstrained. In this ill-conditioned case, the solution of (14) is unstable and sensitive to small disturbances in y. Thus, it is important to relax the constraints to generally obtain a properly conditioned problem. Because replicate pixels in overlapped patches cannot provide additional useful information, a joint estimation is built on the nonoverlapped parts of the current interpolated patch and neighboring patches $\{W_k | k = 1, \ldots, n\}$ in (14). Thus, constructing such a joint region for the estimation is equivalent to extending the center interpolation window in isophote directions, forming an irregular interpolation window for constructing the AR models. In this way, we relax the ICAR model from a joint estimation of the multipatch constraint to a joint estimation of the multipixel constraint in an irregular interpolation window.

With the preceding definitions *patch*, *window*, and *compared patch*, as shown in Fig. 5, the *window extension* process is defined. In the window extension process, our method starts with the basic interpolation window, a patch enlarged from the center patch. We estimate the isophote using the information of center compared patch (the compared patch at the center) and other compared patches. Next, the isophote direction is obtained and represented by the isophote direction code (Fig. 5). According to the patch differences in eight directions, the final isophote direction can be represented as an 8-b code. If the MSE between the center compared patch and one of other compared patches in a direction is less than a threshold, the corresponding bit is set to 1; otherwise, it is set to 0. The example in Fig. 5(a) shows that patch distances between the center compared patch and other compared patches jointly determine the estimate of the isophote direction and, consequently, determine the isophote direction code. The code guides the interpolation window to extend along the isophote direction. The window extension process builds an irregular interpolation window containing the basic interpolation window and extension windows in various directions. After the window extension process, the ICAR model is built according to the information of all pixels in the interpolation window. The similarity metrics mentioned in Section III are incorporated as weighting terms into the ICAR model and a novel objective function is obtained. The similarity metrics are calculated based on both the pixel-level dependence in the interpolation window and the patch-level dependence between the center compared patch and other compared patches along the isophote. Finally, the function can be solved with a closed-form solution. The entire process is shown in Fig. 4.

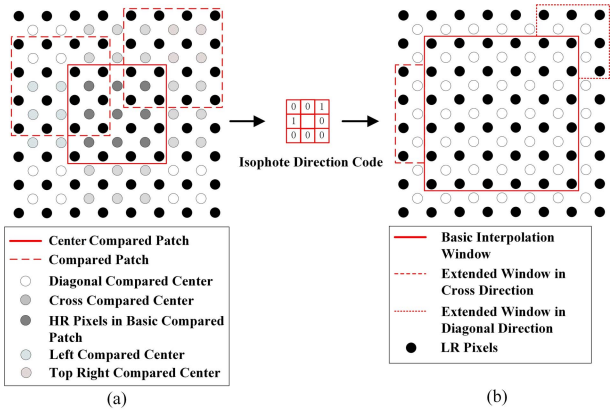


Fig. 5. Illustrations for the window extension. (a) Center locations of the compared patches along the cross and diagonal directions. (b) Extended result along the diagonal top-right and the cross left direction. Images (a) and (b) are the experimental setting *extension1* in Section V-E.

We now further illustrate some details about the isophote estimation and the adaptive window extension. The size of the center window is set to 6×6 in the LR image (11×11 in the HR image). The sizes of the compared windows, including the center compared window and other compared windows, are set to 4×4 in the LR image (7×7 in the HR image) as shown in Fig. 5. The extension windows are irregular, as shown in Fig. 5(b). The initial interpolation window is

set as the center window. We search for similar compared patches in eight directions. For example, the six blue HR pixels on the left of the center compared patch are considered to be the centers of compared patches; the four red HR pixels in the top-right direction are also considered to be the centers of compared patches as described in Fig. 5(a). An example of the isophote direction code is shown in Fig. 4: in this example, it is 11000110. Another detailed example is shown in Fig. 5, in which the MSE between the center compared patch and one of the compared patches on the left and one of those in the top-right direction are below the threshold. Thus, the isophote direction code is 00100100 and the interpolation window extends from the center window to an adaptive window as shown in Fig. 5(b). In Fig. 5(b), the extension windows are bounded by dashed line boxes in the left and the top-right directions. To generate the isophote direction code, six patches are compared in each cross direction, and four patches are compared in each diagonal direction.

Then, the optimization function in (14) can be converted to

$$\min_{\{y_i\}} \left\{ \sum_{i\in\rho(W_b)} v_i^H \omega_i^H \left(y_i - \sum_{t=1}^{4} a_t x_{N_\otimes(y_i,t)} \right)^2 + \sum_{i\in\rho(W_b)} v_i^L \omega_i^L \left(x_i - \sum_{t=1}^{4} a_t y_{N_\otimes(x_i,t)} \right)^2 + \lambda \sum_{i\in\rho(W_b)} v_i^H \omega_i^H \left(y_i - \sum_{t=1}^{4} b_t y_{N_\oplus(y_i,t)} \right)^2 \right\} \quad (15)$$

where W_b is the basic interpolation window, ρ is the window-extension operator, $\rho(W_b)$ indicates the extended interpolation window of W_b, and ω_i^H and ω_i^L, corresponding to ω_k in Section III-B, measure the similarity between the center compared patch and the ith compared patch centered at the ith HR and LR pixels, respectively. Details of the similarity metrics and the weighting scheme are elaborated in Section III.

B. AR Parameter Estimation

The AR model parameters ($\boldsymbol{a}$ and $\boldsymbol{b}$) are estimated by the LR pixels in the interpolation window after extension. Specifically, $\boldsymbol{b}$ can be obtained by solving the following weighted least squares estimation problems:

$$\hat{\boldsymbol{b}} = \arg\min_{\{b_t\}} \sum_{i\in\rho(W_b)} v_i^L \omega_i^L \left(x_i - \sum_{t=1}^{4} b_t x_{N_\oplus(x_i,t)} \right)^2. \quad (16)$$

Because the HR neighbors of x_i are unknown, $\boldsymbol{a}$ cannot be estimated directly. However, we impose the geometric duality assumption [8], which assumes that the relationships between LR pixels and those between HR pixels are the same (as shown in Fig. 6). Thus, $\boldsymbol{a}$ can be estimated from the LR image. Specifically, the relationship between x_i and its diagonal neighbor $y_{N_\otimes(x_i,t)}$ can be estimated from x_i and $x_{N_\otimes^L(x_i,t)}$ as shown in Fig. 6, where $N_\otimes^L(\cdot)$ are the LR neighboring operators to denote the subscripts of LR neighboring pixels in the diagonal direction.

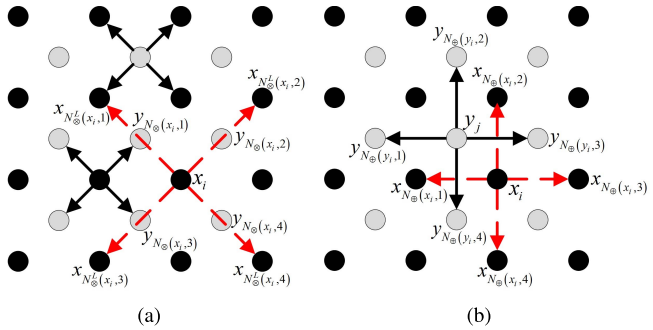


Fig. 6. Parameter estimations based on the geometric duality between HR and LR pixels. (a) Estimation of parameter $\boldsymbol{a}$. The relationship between x_i and its diagonal neighbor $y_{N_\otimes(x_i,t)}$ is approximated from x_i and $x_{N_\otimes^L(x_i,t)}$. (b) Estimation of parameter $\boldsymbol{b}$. The relationship between y_i and its cross neighbor $y_{N_\oplus(y_i,t)}$ is approximated directly from x_i and $x_{N_\oplus(y_i,t)}$ ($x_{N_\oplus^L(y_i,t)}$).

Based on this assumption, we estimate the AR parameters at the HR scale with those from the LR scale. That is, the relationship between x_i and its diagonal neighbor $y_{N_\otimes(x_i,t)}$ can be deduced approximately from x_i and $x_{N_\otimes^L(x_i,t)}$ as shown in Fig. 6, where $N_\otimes^L(\cdot)$ are the LR neighboring operators to denote the subscripts of LR neighbors in the diagonal direction. Thus, as discussed in (17), the best estimator for $\boldsymbol{a}$ is the solution of

$$\hat{\boldsymbol{a}} = \arg\min_{\{a_t\}} \sum_{i\in\rho(W_b)} v_i^L \omega_i^L \left(x_i - \sum_{t=1}^{4} a_t x_{N_\otimes^L(x_i,t)} \right)^2. \quad (17)$$

C. Implementation

Let $\boldsymbol{y} = [y_1, y_2, \ldots, y_m]^T$ and $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]^T$. n is the number of the LR pixels and m is the number of the HR pixels in the interpolation window after extension. The best estimator of $\boldsymbol{a}$ and $\boldsymbol{b}$ can be represented in matrix form

$$\hat{\boldsymbol{a}} = \arg\min_{\boldsymbol{a}} ||\boldsymbol{S_l}(\boldsymbol{A}\boldsymbol{a} - \boldsymbol{x})||_2^2, \quad \hat{\boldsymbol{b}} = \arg\min_{\boldsymbol{b}} ||\boldsymbol{S_l}(\boldsymbol{B}\boldsymbol{b} - \boldsymbol{x})||_2^2 \quad (18)$$

where $\boldsymbol{S_l}$ is a diagonal matrix consisting of the similarity probabilities of LR pixels, and $\boldsymbol{A}$ and $\boldsymbol{B}$ are matrices with sizes of $n\times 4$. The ith row in matrix $\boldsymbol{A}$ consists of x_i's LR neighbors in the diagonal direction, and the ith row in matrix $\boldsymbol{B}$ consists of x_i's LR neighbors in the cross direction

$$\boldsymbol{S_l} = \begin{bmatrix} v_1^L w_1 & 0 & \ldots & 0 \\ 0 & v_2^L w_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & v_n^L w_n \end{bmatrix}$$

$$\boldsymbol{A} = \begin{bmatrix} x^L_{N_\otimes(x_1,1)} & x^L_{N_\otimes(x_1,2)} & x^L_{N_\otimes(x_1,3)} & x^L_{N_\otimes(x_1,4)} \\ x^L_{N_\otimes(x_2,1)} & x^L_{N_\otimes(x_2,2)} & x^L_{N_\otimes(x_2,3)} & x^L_{N_\otimes(x_2,4)} \\ \vdots & \vdots & \vdots & \vdots \\ x^L_{N_\otimes(x_n,1)} & x^L_{N_\otimes(x_n,2)} & x^L_{N_\otimes(x_n,3)} & x^L_{N_\otimes(x_n,4)} \end{bmatrix}$$

$$\boldsymbol{B} = \begin{bmatrix} x_{N_\oplus(x_1,1)} & x_{N_\oplus(x_1,2)} & x_{N_\oplus(x_1,3)} & x_{N_\oplus(x_1,4)} \\ x_{N_\oplus(x_2,1)} & x_{N_\oplus(x_2,2)} & x_{N_\oplus(x_2,3)} & x_{N_\oplus(x_2,4)} \\ \vdots & \vdots & \vdots & \vdots \\ x_{N_\oplus(x_n,1)} & x_{N_\oplus(x_n,2)} & x_{N_\oplus(x_n,3)} & x_{N_\oplus(x_n,4)} \end{bmatrix}.$$

The analytical solution of the weighted least squares problems in (18) can be represented as follows:

$$\hat{\boldsymbol{a}} = \left(\boldsymbol{A}^T \boldsymbol{S}_l^2 \boldsymbol{A}\right)^{-1} \boldsymbol{A}^T \boldsymbol{S}_l^2 \boldsymbol{x}, \quad \hat{\boldsymbol{b}} = \left(\boldsymbol{B}^T \boldsymbol{S}_l^2 \boldsymbol{B}\right)^{-1} \boldsymbol{B}^T \boldsymbol{S}_l^2 \boldsymbol{x}. \quad (19)$$

For piecewise stationary images, such as logos and images of text, the neighboring patterns are simple and highly similar in a local patch. Thus, multicollinearity may exist between the model parameters, which results in the expansion of variances and imprecise estimations. To solve this problem, we deploy the weighted ridge regression to enhance the stability of the solution

$$\hat{\boldsymbol{a}} = \left(\mathbf{A}^{\mathbf{T}} \mathbf{S}_{\mathbf{l}}^{\mathbf{2}} \mathbf{A} + r\mathbf{I}\right)^{-1} \mathbf{A}^{\mathbf{T}} \mathbf{S}_{\mathbf{l}}^{\mathbf{2}} \mathbf{x}, \quad \hat{\boldsymbol{b}} = \left(\mathbf{B}^{\mathbf{T}} \mathbf{S}_{\mathbf{l}}^{\mathbf{2}} \mathbf{B} + r\mathbf{I}\right)^{-1} \mathbf{B}^{\mathbf{T}} \mathbf{S}_{\mathbf{l}}^{\mathbf{2}} \mathbf{x} \quad (20)$$

where $\mathbf{I}$ is the identify matrix, and the ridge parameter r is used to alleviate the singularity in the estimation. Larger r values lead to more stable—but biased—results. In our method, we set $r = 250$.

Thus, according to the optimization function in (15), we can obtain the estimator of $\boldsymbol{y}$ by solving

$$\hat{\boldsymbol{y}} = \arg\min_{\boldsymbol{y}} ||\boldsymbol{S}(\boldsymbol{C}\boldsymbol{y} - \boldsymbol{D}\boldsymbol{x})||_2^2 \quad (21)$$

where $\boldsymbol{S}$, $\boldsymbol{C}$, and $\boldsymbol{D}$ are defined as

$$\begin{aligned}
\boldsymbol{S} &= \operatorname{diag}[\nu_1^H \omega_1^H, \ldots, \nu_{k_1+k_2}^H \omega_{k_1+k_2}^H, \\
&\quad \nu_1^L \omega_1^L, \ldots, \nu_{k_3}^L \omega_{k_3}^L, \nu_1^H \omega_1^H, \ldots, \nu_{k_1}^H \omega_{k_1}^H] \\
\boldsymbol{C} &= \begin{bmatrix} \boldsymbol{I}_{(k_1+k_2)\times(k_1+k_2)} \\ \boldsymbol{C}^1_{k_3\times(k_1+k_2)} \\ \lambda \boldsymbol{C}^2_{k_1\times(k_1+k_2)} \end{bmatrix}, \quad \boldsymbol{D} = \begin{bmatrix} \boldsymbol{D}^1_{(k_1+k_2)\times(k_3+k_4)} \\ \boldsymbol{D}^2_{k_3\times(k_3+k_4)} \\ \mathbf{0}_{k_1\times(k_3+k_4)} \end{bmatrix}
\end{aligned}$$

where $\boldsymbol{S}$ is a diagonal matrix composed of the similarity probability of the HR and LR pixels. The k_1, k_2, k_3, and k_4 are the numbers of four types of pixels in the interpolation window, as shown in Fig. 7. In Fig. 7, the gray HR pixels can only be the centers in the diagonal direction constraints, while the blue LR pixels and the red HR pixels can be the centers in both the diagonal direction and cross direction constraints. The black LR pixels only engage in other pixels' constraints; they cannot be the center of any constraints. For example, in Fig. 7, $k_1 = 4$, $k_2 = 19$, $k_3 = 11$, and $k_4 = 27$. The submatrices of $\boldsymbol{C}$ and $\boldsymbol{D}$ are defined as

$$\begin{aligned}
\boldsymbol{C}^1 &= \{c_1(i,j)\}(i = 1, 2, \ldots, k_3, j = 1, 2, \ldots, (k_1 + k_2)) \\
c_1(i,j) &= \begin{cases} \hat{a}_t, & j \in \{N_{\otimes}(x_i, t) | t = 1, 2, 3, 4\} \\ 0, & \text{otherwise} \end{cases} \\
\boldsymbol{C}^2 &= \{c_2(i,j)\}(i = 1, 2, \ldots, k_1, j = 1, 2, \ldots, (k_1 + k_2)) \\
c_2(i,j) &= \begin{cases} 1, & i = j, \\ -\hat{b}_t, & j \in \{N_{\oplus}(y_i, t), t = 1, 2, 3, 4\} \\ 0, & \text{otherwise} \end{cases} \\
\boldsymbol{D}^1 &= \{d_1(i,j)\}(i = 1, 2, \ldots, k_1 + k_2, \\
&\qquad j = 1, 2, \ldots, (k_3 + k_4)) \\
d_1(i,j) &= \begin{cases} \hat{a}_t, & j \in \{N_{\otimes}(y_i, t) | t = 1, 2, 3, 4\} \\ 0, & \text{otherwise} \end{cases} \\
\boldsymbol{D}^2 &= [\boldsymbol{I}_{k_3\times k_3} \quad \mathbf{0}_{k_3\times k_4}]
\end{aligned}$$

where $\boldsymbol{I}$ is a unit matrix and $\mathbf{0}$ is a zero matrix.

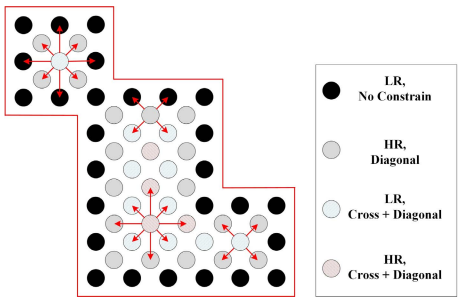


Fig. 7. Example of the window extension result.

The closed-form solution of (21) can be represented as follows:

$$\hat{\boldsymbol{y}} = (\boldsymbol{C}^T \boldsymbol{S}^2 \boldsymbol{C})^{-1} \boldsymbol{C}^T \boldsymbol{S}^2 \boldsymbol{D}\boldsymbol{x}.$$

Finally, the best estimators of the HR pixels $\hat{\boldsymbol{y}} = [\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_{k_1+k_2}]$ in the extended interpolation window are obtained. The proposed algorithm outputs the center HR pixel in the center window. The estimators of the HR pixels are the linear combinations of LR pixels within the extended interpolation window in a 2D adaptive interpolation filter form. They contain similar pattern information between pixels and between patches along the isophote.

In the experiments, the adaptive window-based extension interpolation is performed only in areas with high frequencies, because simpler methods such as Bicubic interpolation are sufficient for smooth regions. These high-frequency regions are identified using the local variances estimated from the LR pixels. Using this strategy, the pattern information in (21) is diversified and singular solutions are seldom found.

D. Relation to RSAI

In SAI, the constructed optimization problem is solved by orthogonal least squares, which strongly assumes that the AR parameters of all pixels are the same. RSAI [15] addresses the problem of local AR parameter inconsistency using weighted orthogonal least squares. The residuals (prediction errors) are weighted according to the geometric similarity between the pixel of interest and the residuals. This is modeled as a simplified version of (14) constructed in a rather large rectangle interpolation window W as follows:

$$\begin{aligned}
\min_{\{y_i\}} \Bigg\{ & \sum_{i\in W} \nu_i^H \left(y_i - \sum_{t=1}^{4} a_t x_{N_{\otimes}(y_i,t)} \right)^2 \\
& + \sum_{i\in W} \nu_i^L \left(x_i - \sum_{t=1}^{4} a_t y_{N_{\otimes}(x_i,t)} \right)^2 \\
& + \lambda \sum_{i\in W} \nu_i^H \left(y_i - \sum_{t=1}^{4} b_t y_{N_{\oplus}(y_i,t)} \right)^2 \Bigg\}. \quad (22)
\end{aligned}$$

Comparing (15) and (22), the formulation for our final relaxed problem is similar to RSAI to some extent. However, these methods are essentially different in two aspects. First, the weighting scheme of RSAI functions mainly at the pixel level and does not consider the similarity between large regions. Second, our method is equivalent to interpolating in a large

rectangular window W, without window extension but with a hard-thresholding operation to remove the residues of the pixels that have a low weight value in W. Thus, compared with RSAI, our weighing scheme is a more general framework with more diversified information between large regions and is more adaptive in its function form.

V. Experimental Results

In this section, experimental results are presented to demonstrate the performance of the proposed method. First, we elaborate on the setting for the experiments. Second, we perform comparisons between pixel-level metrics based on different distances. Third, we compare the adaptive window-extension-based interpolation method with other state-of-the-art methods for a two times enlargement, using both objective and subjective quality evaluations. Finally, we present a comparison of arbitrary scale enlargements.

A. Experimental Setting

All the evaluations were performed using the MATLAB 8.20 platform. Bicubic interpolation was performed with the MATLAB built-in functions. The source code for the other compared methods was kindly provided by their authors. To ensure the thoroughness and fairness of our comparison study, we performed tests on 15 widely used benchmark images selected from the Kodak database[2] and the USC-SIPI image database.[3] To further evaluate the performance of the proposed method on long repetitive edge patterns, eight images from the Urban data set[4] were also tested in our experiments.

To evaluate the effectiveness of the proposed adaptive window-extension-based interpolation method, we compared it with state-of-the-art interpolation algorithms, including the NEDI [8], the SAI [14], the ICBI [9], the AGSI [19], the segment adaptive gradient interpolation (SAGA) [4], the sparse mixing estimators (SMEs) [41], and the NARM [26]. In our experiments, 2 and 4 were chosen as scaling factors. We also compared the results of 1.5 times and 1.7 times enlargements (see the supplementary material).[5] We used two objective quality assessment criteria, peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) [42] to evaluate the quality of the interpolation results.

For a given scaling factor s, we first generate the LR image by downsampling the original HR image by a factor of $1/s$. Then, we applied the different interpolation methods to obtain HR images from the LR image. For the downsampling process, in two times enlargement, we adopted two popular configurations. The first one is the "direct downsampling" generally used as the experimental setting in the previous studies [4], [8], [14], [19], which we denote as *Setup1*. For a scaling factor s, we extract the top-left pixel from every $s \times s$ nonoverlapped patch to generate the downsampled LR image. Note that, in this configuration, the PSNR/SSIM results for NEDI and ICBI are underestimated due to the half-pixel shift [43] under *Setup1*; thus, they are not presented to avoid potential confusions. To fairly compare the performance of all methods, we also tested using LR and HR images obtained both via Bilinear interpolation, which we denote as *Setup2*, using the settings given in [9]. Because the HR and LR images are both generated via Bilinear interpolation, *Setup2* does not produce the half-pixel shift for those methods.

TABLE I

Quantitative Evaluation of PSNR (dB) and SSIM Index Using Pixel-Level Similarity Metrics Based on Different Distances (Two Times, *Setup1*). PIGD and PAGD Denote the PIGD and PAGD, Respectively

Image	Criteria	Uniform	Bilateral	PIGD	PAGD
Woman	PSNR	31.27	31.34	31.33	**31.35**
	SSIM	0.8955	0.8963	0.8958	**0.8962**
Tulip	PSNR	35.71	35.85	35.81	**35.87**
	SSIM	0.9601	0.9589	**0.9590**	0.9589
Cameraman	PSNR	25.99	26.06	26.06	**26.12**
	SSIM	0.8663	0.8663	0.8657	**0.8667**
Airplane	PSNR	29.62	30.05	30.03	**30.08**
	SSIM	0.9161	0.9171	0.9165	**0.9175**
Bike	PSNR	26.28	26.31	26.28	**26.33**
	SSIM	0.8764	0.8772	0.8766	**0.8775**
Ruler	PSNR	11.37	11.81	12.31	**12.74**
	SSIM	0.6679	0.6921	0.7177	**0.7263**
Average	PSNR	26.71	26.90	26.97	**27.08**
	SSIM	0.8637	0.8680	0.8717	**0.8739**

The interpolation algorithms dedicated to two times enlargements are followed by a Bicubic downsampling process to perform arbitrary scaling interpolations. If the size of the interpolation result does not match that of the original HR image, we extend the last row and the last column of the result to match the size.

B. Effectiveness of Patchwise Geodesic Distance-Based Pixel-Level Similarity Metric

To show the effectiveness of the pixel-level similarity metric based on the proposed PAGD, we provide a comparison of the interpolation results incorporated with pixel-level similarity metrics based on different distances. We compared the results of the image enlargement using a fixed window size. Four metrics (mentioned in Section III-A) are involved in the comparisons: uniform (without similarity metrics), bilateral distance based, PIGD-based, and PAGD based. The selected scaling factor is 2. The objective quality evaluation is shown in Table I. As illustrated, the proposed AR model embedded with the pixel-level similarity metric based on the PAGD acquires higher PSNR results than the ones based on the other three distances. Especially for the *Ruler* image, the proposed AR model incorporated with the proposed metric gains a PSNR 0.4 dB higher than the one with the metric based on the PIGD. This result illustrates the proposed model's outstanding ability to model the piecewise stationary regions.

Following [39], the parameters of the bilateral distance are set to $\beta = 50\,000$, $\alpha_1 = 3000$, and $\alpha_2 = 6000$. However, the value of β for the PIGD is not mentioned in [18]. In our experimental setting, the β values for v^g and v^p are set to the maximum values of D^g and D^p in the extended window.

C. Comparison in two times Enlargement

As can be observed from Table II, the proposed method generates better or competitive PSNR and SSIM results in

[2] http://r0k.us/graphics/kodak/

[3] http://sipi.usc.edu/database/

[4] https://github.com/jbhuang0604/SelfExSR

[5] Available at http://www.icst.pku.edu.cn/struct/AWI/supple.pdf

TABLE II

QUANTITATIVE EVALUATION OF PSNR (dB) AND SSIM INDEX USING DIFFERENT INTERPOLATORS ON 15 IMAGES (TWO TIMES, *Setup1*)

Image	Criteria	Bicubic	SAI	SAGA	AGSI	SME	NARM	Proposed
Child	PSNR	35.49	35.63	35.41	35.49	35.53	35.52	**35.71**
	SSIM	0.9840	**0.9840**	0.9830	0.984	**0.9840**	0.982	**0.9840**
Lena	PSNR	34.01	34.76	34.49	34.49	34.61	**35.09**	34.77
	SSIM	0.9725	0.9741	0.9732	0.973	0.9740	**0.976**	0.9739
Tulip	PSNR	33.82	35.71	35.26	35.26	35.08	**36.04**	35.84
	SSIM	0.9860	**0.9891**	0.9876	0.988	0.9884	0.988	0.9890
Cameraman	PSNR	25.51	25.99	26.02	25.55	**26.24**	26.05	26.12
	SSIM	0.8590	0.8660	0.8680	0.863	0.8660	**0.871**	0.8670
Monarch	PSNR	31.93	33.08	32.87	32.87	32.69	**34.10**	33.34
	SSIM	0.9876	0.9893	0.9888	0.989	0.9889	**0.991**	0.9895
Airplane	PSNR	29.40	29.62	29.72	29.87	30.00	30.05	**30.09**
	SSIM	0.9650	0.9670	0.9670	0.967	0.9670	0.969	**0.9680**
Caps	PSNR	31.25	31.64	31.53	31.53	31.60	**31.77**	31.71
	SSIM	0.9672	0.9691	0.9683	0.968	0.9697	0.969	**0.9698**
Statue	PSNR	31.36	31.78	31.80	31.35	31.55	31.72	**31.96**
	SSIM	0.9710	**0.9740**	0.9730	0.972	**0.9740**	0.973	**0.9740**
Sailboat	PSNR	30.12	30.69	30.56	30.39	**30.90**	30.64	30.87
	SSIM	0.9620	0.9660	0.9650	0.963	**0.9690**	0.969	0.9660
Woman	PSNR	31.17	31.27	31.31	31.26	31.15	**31.57**	31.35
	SSIM	0.9580	0.9570	0.9580	0.958	0.9580	**0.961**	0.9580
Bike	PSNR	25.41	26.28	25.85	25.85	26.08	26.23	**26.33**
	SSIM	0.9477	0.9524	0.9498	0.950	0.9527	0.952	**0.9531**
Lighthouse	PSNR	26.97	26.70	27.13	27.16	27.23	**27.36**	27.04
	SSIM	0.9272	0.9257	0.9288	0.929	0.9301	**0.931**	0.9286
Barbara	PSNR	**24.46**	23.55	23.26	23.26	23.98	23.86	23.36
	SSIM	**0.8786**	0.8638	0.8597	0.860	0.8730	0.867	0.8625
Ruler	PSNR	11.98	11.37	12.35	12.51	11.51	11.88	**12.71**
	SSIM	0.8490	0.8400	0.8580	0.861	0.8390	0.836	**0.8590**
Slope	PSNR	26.74	26.63	27.84	27.68	**27.06**	26.89	26.98
	SSIM	0.9640	0.9480	0.9711	0.971	0.9646	0.954	**0.9654**
Average	PSNR	28.64	28.98	29.03	28.97	29.01	**29.25**	29.21
	SSIM	0.9452	0.9444	0.9466	0.9463	0.9466	0.9460	**0.9472**

TABLE III

QUANTITATIVE EVALUATION OF PSNR (dB) AND SSIM INDEX USING DIFFERENT INTERPOLATORS ON *URBAN* DATA SET (TWO TIMES, *Setup1*)

Image	Criteria	Bicubic	SAI	SAGA	SME	AGSI	NARM	Proposed
65	PSNR	23.00	23.80	23.61	23.49	23.73	23.83	**23.93**
	SSIM	0.8291	0.8555	0.8488	0.8423	0.8492	**0.8621**	0.8537
66	PSNR	22.15	22.17	22.49	**22.46**	21.76	22.10	22.15
	SSIM	0.7681	0.7728	0.7784	**0.7760**	0.7616	0.7756	0.7719
67	PSNR	19.59	19.72	19.93	19.48	19.52	19.75	**19.85**
	SSIM	0.7022	0.7117	0.7189	0.6951	0.7025	**0.7209**	0.7146
68	PSNR	24.16	24.71	24.90	**24.98**	24.31	24.81	24.79
	SSIM	0.7977	0.8200	0.8206	**0.8254**	0.8016	0.8222	0.8191
69	PSNR	23.22	23.76	23.85	23.75	23.52	**23.95**	**23.95**
	SSIM	0.7644	0.7920	0.7918	0.7874	0.7770	**0.8058**	0.7981
70	PSNR	20.00	20.60	20.75	20.58	20.03	20.53	**20.79**
	SSIM	0.6729	0.7049	0.7085	0.6993	0.6731	0.7108	**0.7141**
71	PSNR	23.07	**23.21**	23.40	22.94	22.91	23.00	**23.21**
	SSIM	0.8329	0.8361	0.8394	0.8296	0.8288	**0.8385**	0.8348
72	PSNR	26.26	26.80	26.95	**27.33**	26.77	26.94	26.85
	SSIM	0.7762	0.7921	0.7971	0.7924	0.7839	**0.8066**	0.7877
Average	PSNR	22.68	23.10	23.24	23.13	22.82	23.11	**23.19**
	SSIM	0.7679	0.7856	0.7879	0.7809	0.7722	**0.7928**	0.7868

most cases for *Setup1*. It is worth noting that, using this configuration, the PSNR and SSIM results for NEDI and ICBI are underestimated due to the half-pixel shift. Thus, to avoid potential confusions, in Tables II and III, we only compare the objective quality for other methods fairly. Compared with the traditional Bicubic method, the proposed method significantly improves the objective quality of the generated HR images, with an average gain of 0.60 dB in PSNR. Our method also outperforms other AR-based interpolation methods, such as SAI and AGSI. The average PSNR gains of our method compared with SAI and AGSI are 0.26 and 0.24 dB, respectively. SAGA is an explicit edge-directed method; our method gains 0.36 dB over SAGA. Our method also outperforms SME in most cases, achieving better performance on average. Even when compared with the learning-based interpolation method NARM, our method shows comparable performance.

In the experiments, *Ruler* is a noteworthy image, because—except for SAGA and our proposed method—all the other methods produce poor results. In this particular image, the stationary assumption is invalid in most areas. Most interpolation methods fail to model the piecewise stationarity and, consequently, degrade significantly.

As shown in Table III, when tested on images from the Urban data set, the proposed method achieves better PSNR and SSIM results in most cases and, on average, obtains the best PSNR and SSIM results. These results demonstrate the

Fig. 8. Visual comparisons between different algorithms for the image *Barbara* (two times). (a) Original. (b) Bicubic. (c) SAI. (d) SAGA. (e) AGSI. (f) SME. (g) NARM. (h) Proposed. The bottom-left subimage shows the local magnification of interpolation results, and the bottom-right subimage shows the difference map between the local original image and the interpolation result.

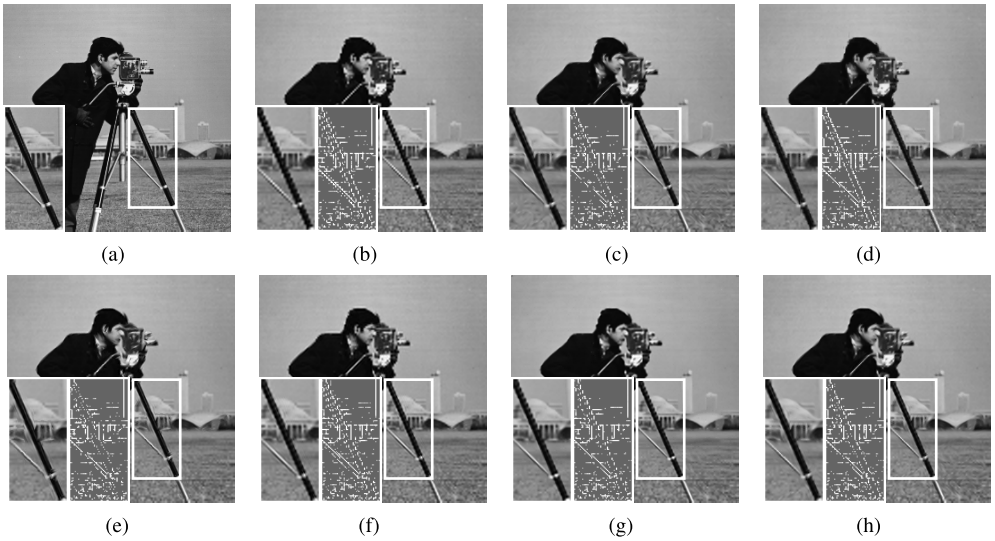


Fig. 9. Visual comparisons between different algorithms for the image *Cameraman* (two times). (a) Original. (b) Bicubic. (c) SAI. (d) SAGA. (e) AGSI. (f) SME. (g) NARM. (h) Proposed. The bottom-left subimage is the local magnification of interpolation results, and the bottom-right subimage is the difference map between the local original image and the interpolation result.

effectiveness of the proposed isophote-guided model and of the window-extension AR model-based interpolation, especially their ability to reconstruct long and repetitive edges.

To fairly compare the performance of the interpolation methods, we also evaluated all the methods in *Setup2* using the 25 testing images from [9]. The PSNR is calculated based on the results, whose 50-pixel borders are cropped out. As shown in Table IV, ICBI performs slightly better than SAI, NARM, and the proposed method. However, despite the pixel shift, the proposed method achieves a performance similar to SAI and NARM in the two times situation just as it does under *Setup1* (Tables II and III). This experiment also shows that our method is robust to downsampled configurations. Note that, we particularly recommend that readers refer to the subjective comparisons.

Subjective comparisons are also given in Figs. 8 and 9. In the two times enlargements, *Barbara* and *Cameraman* are used in the experiment. Figs. 8 and 9 show the subjective comparisons and the reconstruction errors. For *Barbara*, downsampling causes the stripes' directions to change; therefore, the interpolation results based on the misleading LR image have severe prediction errors. In the subregions of *Barbara* in Fig. 8, most methods make incorrect predictions and produce interpolation errors. Due to the robust modeling capacity of the proposed PAGD-based pixel-level similarity metric, the proposed method can suppress artifacts and reduce errors. For *Cameraman*, all methods exhibit strong and sharp edges. The proposed method presents desirable local results and obtains the darkest difference maps. Dark difference maps signify that the interpolation results contain fewer errors and are more similar to the corresponding original HR images.

It can be clearly observed that images interpolated by Bicubic interpolation suffer from blurred edges, jaggies, and annoying ringing artifacts. NEDI and ICBI improve the

TABLE IV

AVERAGE PSNR RESULTS (dB) OF DIFFERENT METHODS UNDER THE EXPERIMENT SETUP OF ICBI (TWO TIMES, *Setup2*)

Methods	**Bicubic**	**NEDI**	**ICBI**	**SAI**	**NARM**	**Proposed**
PSNR	30.36	29.71	31.07	30.78	30.95	30.81

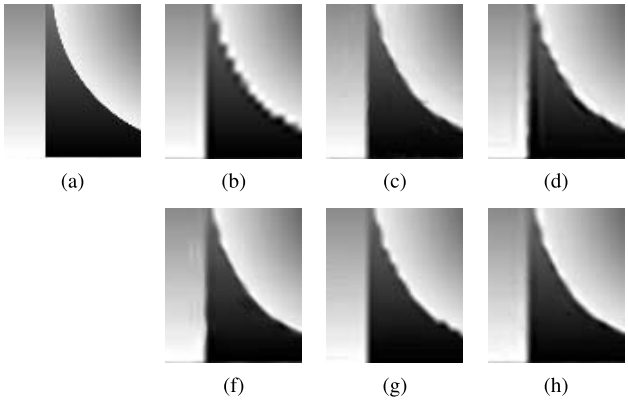


Fig. 10. Visual comparisons between different algorithms for the image *Slope* (four times). (b) Bicubic. (c) NEDI. (d) ICBI. (f) SAI. (g) NARM. (h) Proposed.

reconstruction quality to some extent but at a lower quality than SAI. SAGA shows an improvement over these methods in the edge and texture regions, reducing the visual defects of these methods. AGSI improves a lot from the piecewise-stationarity modeling. NARM achieves a desirable performance when nonlocal redundancy exists. Benefiting from the combination of the pixel-level dependence and the patch-level dependence, the proposed method achieves better visual quality compared with all the other methods. Our algorithm produces fewer interpolation errors than other methods, which clearly demonstrates the superiority of the proposed method in reconstructing the high-frequency portions of images, such as edges and textures.

D. Comparison in Arbitrate Scale Enlargement

To evaluate the capacities of all methods in a more challenging large scale enlargement, we tested Bicubic, NEDI, ICBI, SAI, NARM, and the proposed method at a scaling factor of 4. Fig. 10 shows the four times results of a region from the *Slope* image. The proposed method presents the most continuous and the sharpest edge—the same as SAI—but has slightly fewer aliasing artifacts in the vertical direction, which demonstrates the superiority of the proposed method in large scale enlargement.

We also chose to compare the results at lower noninteger factor enlargements to evaluate the generality of all the methods, namely, whether the compared methods (including our method), which are specially designed for two times enlargement with a specific lattice scheme, are capable of working well with Bicubic interpolation to achieve enlargements at arbitrary scales. The reason to evaluate general scale enlargement capabilities is due to the practical need to adapt to various device resolutions. The growing variety of media-playing devices in consumer electronics recently makes it necessary for a single image/video source to be scaled for display at different resolutions. Thus, the capacity to work at general scale factor enlargements has become a basic requirement for image interpolation. For lack of space, these results are presented in the supplementary material.

TABLE V

QUANTITATIVE EVALUATION OF PSNR RESULTS (dB) AND RUNNING TIME (SECONDS) USING DIFFERENT COMPARED PATCH SIZES

Metric	PSNR			Running Time		
Patch size	5	7	11	5	7	11
Cameraman	30.07	30.08	**30.09**	11.20	21.22	25.44
Tulip	35.79	**35.84**	35.81	31.20	128.69	138.72
Airplane	26.08	**26.12**	26.09	25.80	137.77	149.47
Lighthouse	27.11	**27.23**	27.20	20.12	120.12	125.67

TABLE VI

QUANTITATIVE EVALUATION OF PSNR RESULTS (dB) AND RUNNING TIME (SECONDS) USING DIFFERENT WINDOW EXTENSION SCHEMES

Metric	PSNR		Running Time	
Scheme	*Extension1*	*Extension2*	*Extension1*	*Extension2*
Airplane	**30.08**	30.01	346.97	442.23
Barbara	**23.36**	23.24	281.69	381.23
Cameraman	**26.12**	26.06	57.21	63.12
Lighthouse	27.04	**27.05**	390.50	430.12

TABLE VII

QUANTITATIVE EVALUATION ON PSNR RESULTS (dB) AND RUNNING TIME (SECONDS) USING DIFFERENT SIZES TO CALCULATE THE PATCH-GEODESIC DISTANCE

Metric	PSNR			Running Time		
Patch size	1	3	5	1	3	5
Cameraman	26.06	**26.12**	26.09	11.20	21.22	32.65
Tulip	35.81	**35.87**	35.76	41.30	128.69	160.34
Airplane	30.03	**30.08**	30.07	35.21	137.77	174.23
Lighthouse	26.97	27.08	**27.11**	39.12	120.12	180.76

E. Window/Patch Size Evaluation

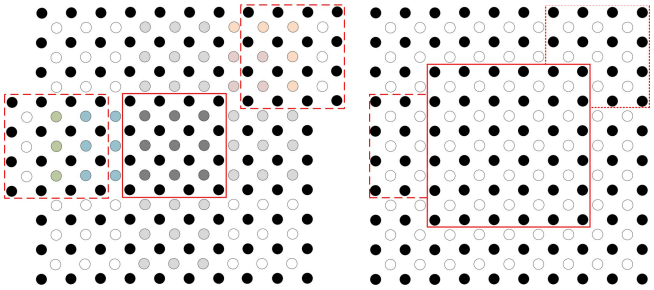
Here, we mainly consider three settings: the size of compared patches, the scheme used for the window extension, and the patch size to calculate the patch-geodesic distance to measure pixel-level similarity. We evaluate these three factors empirically.

1) Size of Compared Patches: The size of compared patches is determined by an evaluation guided by some generally accepted observations. Intuitively, we want to choose a patch that is capable of representing the AR parameters of the center HR pixel. A small 5×5 HR pixel^2 patch does not provide useful context information, while in a larger of 11×11 HR pixel^2, the influence of the frontier pixels may play a more dominant role in AR parameters than the center pixel. The empirical results on four images shown in Table V support that intuition; we chose a 7×7 HR pixel^2 for the patch size of the compared patches, which leads to the best PSNR results in the evaluation.

2) Window Extension Mechanism: The method of window extension in Fig. 5 (denoted as *extension1*) is primarily determined by manual settings. Here, we further compare the scheme that extends the window to a broader range as shown in Fig. 11(a) and (b). We superpose a further extension scheme on *extension1*: when the MSEs between the center

TABLE VIII
THEORETICAL TIME COMPLEXITY OF EACH STEP OF OUR PROPOSED METHOD

Steps	**P1-PGD**	**P2-AWE**	**P3-WOLS**	**All**
Time Complexity	$O\left(l_g HW H_g^2\right)$	$O\left(n_c HW H_c^2\right)$	$O\left(H_i^2 HW\right)$	$O\left(HW\left(l_g H_g^2 + n_c H_c^2 + H_i^2\right)\right)$



(a) (b)

Fig. 11. Images (a) and (b) demonstrate the experimental setting *extension2*. Compared with *extension1*, *extension2* considers and may construct a larger interpolation window.

compared patch and the compared patches centered at the green and orange pixels are below a threshold, we extend the interpolation window in the manner shown in Fig. 11. The new addition with the original *extension1* is denoted as *extension2*. We compare the performances of these two schemes on four images in terms of both PSNR and running time, as shown in Table VI. Clearly, *extension2* achieves no better performance even with the additional computational burden; thus, we chose to use the *extension1* scheme throughout this paper.

3) Patch Size Used to Calculate the Patchwise Geodesic Distance: This patch size is determined by making a tradeoff between computation time and reconstruction quality as shown in Table VII. It can be observed that increasing the patch size from 3 to 5 does not boost the performance despite imposing a greater computational burden; thus, we chose to calculate the patch-geodesic distance using a patch size of 3×3 HR pixel^2 in our method.

F. Complexity Analysis

To obtain a thorough understanding of the capacity of our method for practical applications, we provide the theoretical time complexity analysis, evaluate the practical running time, and briefly comment on the space complexity.

1) Theoretical Time Complexity: Theoretically, the complexity of our proposed method involves three steps: patch-geodesic distance (P1-PGD), adaptive window extension (P2-AWE), and weighted ordinary least squares (P3-WOLS). The theoretical time complexity of each step is shown in Table VIII. Here, H and W are the height and width of the image, respectively, and H_g is the patch size and the iteration number (equal to 3) to calculate the patch-geodesic distance. H_c, n_c (equal to 49), and H_i (less than 15 pixels) are the patch size of the compared patches, the number of the compared patches, and the window size of the interpolation window, respectively. Our method maintains approximately the same theoretical time complexity as other, simpler AR-flavored methods, such as SAI and NEDI, whose time complexities are approximately equal to that of the WOLS step.

TABLE IX
RUNNING TIME (SECONDS) OF DIFFERENT METHODS AND EACH STEP OF OUR METHOD

	Methods	NEDI	ICBI	ScSR	SRCNN
Images	*Airplane*	186.28	95.50	1891.43	30.42
	Barbara	135.65	60.41	1325.67	24.52
	Cameraman	30.74	13.69	325.97	25.27
	Lighthouse	201.96	93.40	2051.23	25.76
	Methods	Proposed	P1-PGD	P2-AWE	P3-WOLS
Images	*Airplane*	346.97	128.75	28.40	189.82
	Barbara	281.69	84.69	18.99	178.01
	Cameraman	57.21	18.93	4.29	33.99
	Lighthouse	390.50	129.75	28.34	232.41

2) Practical Running Time: We empirically compared the running time of NEDI, ICBI, sparse coding super-resolution (ScSR), super-resolution convolutional neutral network (SRCNN), and the proposed method on the *Airplane*, *Barbara*, *Cameraman*, and *Lighthouse* images, as shown in Table IX. These methods are implemented solely in MATLAB to create a fair comparison platform.

As Table IX shows, our proposed method is slower than ICBI and SRCNN and maintains the same running time as NEDI, but is much faster than the learning-based method ScSR. Our current version requires a relatively long processing time to interpolate one image. This result occurs largely because the MATLAB implementation is not fully optimized. The running time of our method could be improved considerably by being implemented in C. Furthermore, the interpolation process for each pixel is data-independent for all three steps; thus, it could easily be optimized using a parallel implementation scheme, even one based on highly parallel-capable GPU devices.

3) Space Complexity: In addition to time complexity, space complexity is another important property when considering practicability. In extreme cases, such as satellite or aerial photography, it is expensive to store large dictionaries or thousands of model parameters. In contrast, our method has a low space complexity; it does not need extra space to store large numbers of model parameters. This is a desirable characteristic that can be of benefit to some real applications in extreme conditions.

VI. CONCLUSION AND DISCUSSION

In this paper, we proposed an ICAR model to perform AR-flavored interpolation and further develop an AR interpolation with an adaptive window extension. The model integrates both pixel-level and patch-level information to depict the relationships between the HR and LR pixels and, consequently, yields better image interpolation results. Guided by the isophote smoothness prior, our method searches within

several successive similar patches along the isophote in a large region to construct an adaptive window. These overlapped patches, representing the patch-level structure similarity, are used to construct a joint AR model. To better characterize the piecewise stationarity and determine whether a pixel is suitable for AR estimation, we further propose pixel-level and patch-level similarity metrics and embed them into the ICAR model, introducing a weighted ICAR model. Comprehensive experiments demonstrate the effectiveness of our method using both objective and subjective evaluations. Although our interpolation method achieves very promising results in visual quality, its computational cost as shown in Table IX makes it away from the practical use. In the future, we plan to accelerate our method and shorten its running time, e.g., via parallelism or a precomputed mapping function to estimate the AR parameters based on local structures in a similar way to anchored regression [36], which will make our method more practical.

REFERENCES

[1] R. G. Keys, "Cubic convolution interpolation for digital image processing," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 29, no. 6, pp. 1153–1160, Dec. 1981.

[2] H. S. Hou and H. Andrews, "Cubic splines for image interpolation and digital filtering," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 26, no. 6, pp. 508–517, Dec. 1978.

[3] 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.

[4] 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.

[5] 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.

[6] L. 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.

[7] 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.

[8] X. Li and M. T. Orchard, "New edge-directed interpolation," *IEEE Trans. Image Process.*, vol. 10, no. 10, pp. 1521–1527, Oct. 2001.

[9] A. Giachetti and N. Asuni, "Real-time artifact-free image upscaling," *IEEE Trans. Image Process.*, vol. 20, no. 10, pp. 2760–2768, Oct. 2011.

[10] X. Liu, D. Zhao, R. Xiong, S. Ma, W. Gao, and H. Sun, "Image interpolation via regularized local linear regression," *IEEE Trans. Image Process.*, vol. 20, no. 12, pp. 3455–3469, Dec. 2011.

[11] K. S. Ni and T. Q. Nguyen, "An adaptable k-nearest neighbors algorithm for MMSE image interpolation," *IEEE Trans. Image Process.*, vol. 18, no. 9, pp. 1976–1987, Sep. 2009.

[12] W. S. Tam, C. W. Kok, and W. C. Siu, "Modified edge-directed interpolation for images," *J. Electron. Imag.*, vol. 19, no. 1, p. 013011, 2010.

[13] D. T. Vo, J. Sole, Y. Peng, C. Gomila, and T. Q. Nguyen, "Selective data pruning-based compression using high-order edge-directed interpolation," *IEEE Trans. Image Process.*, vol. 19, no. 2, pp. 399–409, Feb. 2010.

[14] 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.

[15] H. 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.

[16] H. W. Hung and W. C. Siu, "Improved image interpolation using bilateral filter for weighted least square estimation," in *Proc. Int. Conf. Image Process.*, Sep. 2010, pp. 3297–3300.

[17] K. W. Hung and W. C. Siu, "Fast image interpolation using the bilateral filter," *IET Image Process.*, vol. 6, no. 7, pp. 877–890, Oct. 2012.

[18] 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.

[19] M. Li, J. Liu, J. Ren, and Z. Guo, "Adaptive general scale interpolation based on weighted autoregressive models," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 2, pp. 200–211, Feb. 2014.

[20] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising with block-matching and 3D filtering," *Proc. SPIE*, vol. 6064, pp. 354–365, Feb. 2006.

[21] 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.

[22] J. Sun, L. Yuan, J. Jia, and H.-Y. Shum, "Image completion with structure propagation," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 861–868, 2005.

[23] W. Dong, L. 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.

[24] 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.

[25] H. Zhang, J. Yang, Y. Zhang, and T. S. Huang, "Image and video restorations via nonlocal kernel regression," *IEEE Trans. Cybern.*, vol. 43, no. 3, pp. 1035–1046, Jun. 2013.

[26] W. Dong, L. Zhang, R. Lukac, and G. Shi, "Sparse representation based image interpolation with nonlocal autoregressive modeling," *IEEE Trans. Image Process.*, vol. 22, no. 4, pp. 1382–1394, Apr. 2013.

[27] Y. Romano, M. Protter, and M. Elad, "Single image interpolation via adaptive nonlocal sparsity-based modeling," *IEEE Trans. Image Process.*, vol. 23, no. 7, pp. 3085–3098, Jul. 2014.

[28] W. T. Freeman, T. R. Jones, and E. C. Pasztor, "Example-based super-resolution," *IEEE Comput. Graph. Appl.*, vol. 22, no. 2, pp. 56–65, Mar./Apr. 2002.

[29] G. Freedman and R. Fattal, "Image and video upscaling from local self-examples," *ACM Trans. Graph.*, vol. 30, no. 2, p. 12, 2011.

[30] R. Fattal, "Image upsampling via imposed edge statistics," *ACM Trans. Graph.*, vol. 26, no. 3, 2007, Art. no. 95.

[31] K. S. Ni and T. Q. Nguyen, "Image superresolution using support vector regression," *IEEE Trans. Image Process.*, vol. 16, no. 6, pp. 1596–1610, Jun. 2007.

[32] K. I. Kim and Y. Kwon, "Example-based learning for single-image super-resolution," in *Proc. Joint Pattern Recognit. Symp.*, 2008, pp. 456–465.

[33] D. Glasner, S. Bagon, and M. Irani, "Super-resolution from a single image," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2009, pp. 349–356.

[34] K. I. Kim and Y. Kwon, "Single-image super-resolution using sparse regression and natural image prior," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 6, pp. 1127–1133, Jun. 2010.

[35] C.-Y. Yang and M.-H. Yang, "Fast direct super-resolution by simple functions," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 561–568.

[36] R. Timofte, V. De Smet, and L. Van Gool, "Anchored neighborhood regression for fast example-based super-resolution," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1920–1927.

[37] R. Timofte, V. De Smet, and L. Van Gool, "A+: Adjusted anchored neighborhood regression for fast super-resolution," in *Proc. Asian Conf. Comput. Vis.*, 2014, pp. 111–126.

[38] J.-J. Huang, W.-C. Siu, and T.-R. Liu, "Fast image interpolation via random forests," *IEEE Trans. Image Process.*, vol. 24, no. 10, pp. 3232–3245, Oct. 2015.

[39] J. Ren, J. Liu, W. Bai, and Z. Guo, "Similarity modulated block estimation for image interpolation," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2011, pp. 1177–1180.

[40] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge Univ. Press, 2004.

[41] S. Mallat and G. Yu, "Super-resolution with sparse mixing estimators," *IEEE Trans. Image Process.*, vol. 19, no. 11, pp. 2889–2900, Nov. 2010.

[42] 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.

[43] N. Asuni and A. Giachetti, "TESTIMAGES: A large-scale archive for testing visual devices and basic image processing algorithms," in *Proc. Eurograph. Italian Chapter Conf.*, 2014, pp. 63–70.

Wenhan Yang received the B.S. degree in computer science from Peking University, Beijing, China, in 2012, where he is currently pursuing the Ph.D. degree with the Institute of Computer Science and Technology.

He was a Visiting Scholar with the National University of Singapore, Singapore, from 2015 to 2016. His current research interests include image processing, sparse representation, image restoration, and deep learning-based image processing.

Jiaying Liu (S'08–M'10) received the B.E. degree in computer science from Northwestern Polytechnic University, Xi'an, China, in 2005, and the Ph.D. degree (Hons.) in computer science from Peking University, Beijing, China, in 2010.

She is currently an Associate Professor with the Institute of Computer Science and Technology, Peking University. She has authored over 80 technical articles in refereed journals and proceedings, and holds 15 granted patents. Her current research interests include image/video processing, compression, and computer vision.

Dr. Liu was a Visiting Scholar with the University of Southern California, Los Angeles, from 2007 to 2008. She was a Visiting Researcher with Microsoft Research Asia in 2015 supported by Star Track for Young Faculties. She has also been a TC Member in APSIPA IVM since 2015, and an APSIPA Distinguished Lecturer from 2016 to 2017, and a CCF Senior Member.

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 2013, where he is currently pursuing the Ph.D. degree with the Institute of Computer Science and Technology.

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

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

Dr. Guo is the Executive Member of the China-Society of Motion Picture and Television Engineers. He was a recipient of the First Prize of the State Administration of Radio Film and Television Award in 2004, the First Prize of the Ministry of Education Science and Technology Progress Award in 2006, the Second 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.