Image Interpolation Via Regularized Local Linear Regression

Xianming Liu, Debin Zhao, Ruiqin Xiong, *Member, IEEE*, Siwei Ma, *Member, IEEE*, Wen Gao, *Fellow, IEEE*, and Huifang Sun, *Fellow, IEEE*

***Abstract*—The linear regression model is a very attractive tool to design effective image interpolation schemes. Some regression-based image interpolation algorithms have been proposed in the literature, in which the objective functions are optimized by ordinary least squares (OLS). However, it is shown that interpolation with OLS may have some undesirable properties from a robustness point of view: even small amounts of outliers can dramatically affect the estimates. To address these issues, in this paper we propose a novel image interpolation algorithm based on regularized local linear regression (RLLR). Starting with the linear regression model where we replace the OLS error norm with the moving least squares (MLS) error norm leads to a robust estimator of local image structure. To keep the solution stable and avoid overfitting, we incorporate the ℓ_2-norm as the estimator complexity penalty. Moreover, motivated by recent progress on manifold-based semi-supervised learning, we explicitly consider the intrinsic manifold structure by making use of both measured and unmeasured data points. Specifically, our framework incorporates the geometric structure of the marginal probability distribution induced by unmeasured samples as an additional local smoothness preserving constraint. The optimal model parameters can be obtained with a closed-form solution by solving a convex optimization problem. Experimental results on benchmark test images demonstrate that the proposed method achieves very competitive performance with the state-of-the-art interpolation algorithms, especially in image edge structure preservation.**

***Index Terms*—Edge preservation, image interpolation, moving least squares, ordinary least squares, robust estimation.**

I. INTRODUCTION

IMAGE interpolation, which addresses the problem of rescaling a low-resolution (LR) image to a high-resolution (HR) version, is one of the most elementary imaging research topics. It has a wide range of applications from digital photography to video communication, satellite remote sensing, object recognition, consumer electronics, and many other important fields. The main purpose of image interpolation is to recover sharp edges and textures, while suppressing pixel blocking, blurring, and other visual artifacts.

Image interpolation is still a widely studied and unsolved problem in image processing field. A number of methods have been suggested in the literature [1]–[16]. Among these existing methods, the simplest techniques for image interpolation are based on classical data-invariant linear filters, such as the bilinear, bicubic [2], and cubic spline algorithms [3]. These linear methods have a relatively low complexity, but suffer from the inability to adapt to varying pixel structures which results in blurred edges and annoying artifacts.

As well known, the human visual system (HVS), which is the ultimate receiver of the rescaled images, is highly sensitive to distortions of spatial coherence of edges. It is agreed that for many applications, the main emphasis of image interpolation should be on the perceptual quality of images. That is, the interpolated images should be artifact-free and visually pleasing. Many algorithms have been proposed to improve the subjective quality of the interpolated images by imposing more accurate models [4]–[13].

Spatial adaptive interpolation algorithms, which adjust the interpolation coefficients to better match the local structure, have received more and more attention. Among various spatial adaptive algorithms, interpolation along local edge directions is a good idea. This is because, based on geometric constraint of edges, estimation along the edge orientation is optimal in the sense of best inferring unknown pixels. In one of the earliest papers proposed to reduce edge artifacts, Jensen *et al.* [4] propose to estimate the orientation of each edge in the image by using projections onto an orthonormal basis, and modify the interpolation process to avoid interpolating across the edges. The algorithm in [5] is applied to a linearly expanded image as a post-process to enhance edges using a weighted average of neighboring pixels chosen explicitly by the Canny edge detector. In LAZA [6], Battiato *et al.* use simple rules and configurable thresholds to explicitly detect edges and update the interpolation process accordingly. They further extend this work by using local gradient information based on a neural network to achieve better edge sharpness and computation efficiency [7], [8].

The methods mentioned above are all based on explicit detection of edges. The problem with such methods is that the penalty to image quality is high if the estimated edge direction is not ac-

Manuscript received June 23, 2010; revised January 22, 2011; accepted April 10, 2011. Date of publication May 12, 2011; date of current version November 18, 2011. This work was supported by National Science Foundation of China (60736043, 61073083, 60803068), and the Major State Basic Research Development Program of China (973 Program 2009CB320905). This paper was recommended by Associate Editor Prof. Rafael Molina.

X. Liu and D. Zhao are with the School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China (e-mail: xmliu@jdl.ac.cn; dbzhao@jdl.ac.cn).

R. Xiong, S. Ma, and W. Gao are with the National Engineering Laboratory for Video Technology, and Key Laboratory of Machine Perception (MoE), School of Electrical Engineering and Computer Science, Peking University, Beijing 100871, China (e-mail: rqxiong@pku.edu.cn; swma@pku.edu.cn; wgao@pku.edu.cn).

H. Sun is with Mitsubishi Electric Research Laboratories, Cambridge, MA 02139 USA (e-mail: hsun@merl.com).

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/TIP.2011.2150234

curate, which maybe happen due to the difficulty in determining the edge direction. In [9], Zhang and Wu propose to interpolate a missing pixel in preset multiple directions, and then fuse the directional interpolation results by minimum mean square-error estimation. Directional filtering and estimation have proved to be effective to preserve the edges in image interpolation. In [10], a method named ICBI is proposed to use local second order information to adapt the interpolation and an iterative refinement is further exploited to remove artifacts while preserving image features and texture.

Edge detection and image interpolation are two problems with a chicken-and-egg flavor because solving one makes the other almost trivial. However, the accuracy of existing edge detection algorithms remains insufficient, especially in the scenario of image interpolation where the HR image information is incomplete. In view of the difficulty with explicit edge detection, alternatively, auto-regression (AR)-based methods have been extensively studied, which integrate edge direction information into AR model parameters. Li and Orchard propose a new edge-directed interpolation (NEDI) method [11], which exploits the geometric duality between the LR covariance and the HR one to first estimate local covariance coefficients from the LR image and then use these coefficients to adapt the interpolation at the HR image. Furthermore, the improved new edge-directed interpolation (INEDI) method [12] modifies NEDI by varying the size of the training window according to the edge size and achieves better performance. Recently, Zhang and Wu propose the named SAI algorithm [13], which learns and adapts varying scene structures using a 2-D piecewise AR model, and interpolates the missing pixels in a group by a soft-decision manner. SAI achieves promising results in both objective and subjective performance, and is one of the best performed image interpolation algorithms. Besides, the AR model has been successfully applied to other applications, such as frame-rate up-conversion [28], [29].

Central to most AR-based image interpolation algorithms is the definition of an effective and efficient approximation function. NEDI and SAI are both based on *ordinary least squares* (OLS) approximations, they naturally handle uniform noise and generate smooth HR images. However, it is well known from statistics that OLS-based approaches are highly sensitive to outliers. The small amount of outliers will severely affect the accuracy of interpolation model, and further make the estimated intensity values depart far from the true ones. As such, the OLS-based loss function is not a good choice from a robustness point of view, contrary to other loss functions such as ℓ_1 loss or Vapnik's ϵ-insensitive loss [25]. From a computational point of view on the other hand, ℓ_1 loss involves solving a quadratic programming problem, there is no simple analytical formula for the solution.

In statistics, moving least squares (MLS) [15], [16] is a robust technique for data fitting, which provides a high level of control over the function reconstruction, and allows for a tunable amount of data smoothing during interpolation. This flexibility provides the capacity for handling outliers. MLS achieves a good tradeoff between effectiveness and efficiency as it can give more reliable results while still minimizing a least-squares criterion.

To achieve robustness and computation efficiency simultaneously, in this paper, we propose a novel image interpolation algorithm based on a graph-Laplacian regularized local linear regression (RLLR) model. Starting with the linear regression model where we replace the OLS error norm with the MLS error norm leads to a robust estimator of local image structure. Moving weights are incorporated into the objective function in order to express the relative importance of the image samples in estimating the parameters of model. This idea is similar to *kernel regression* [14], to which the difference is that our method imposes the ℓ_2-norm as a complexity penalty term to keep the solutions stable and avoid overfitting, and we design an efficient *patch-based bilateral moving weight* to better keep local texture structure and reduce the influence of outliers in regression. Moreover, motivated by recent progress on manifold based semi-supervised learning [17], we explicitly consider the intrinsic manifold structure by making use of both measured and unmeasured data points in HR image reconstruction. Specifically, our framework incorporates the geometric structure of the marginal probability distribution induced by unmeasured samples as an additional local smoothness preserving constraint. Considering the underlying geometry of the image, the proposed algorithm allows us to remove the artifacts that may arise when performing interpolation, such as blocking and blurring. Computed examples demonstrate its effectiveness by visual comparisons and quantitative measures.

The rest of this paper is organized as follows. In Section II, we give a brief description of ordinary least squares and moving least squares. Section III presents the proposed interpolation framework. Section IV details the implementation of the proposed framework, including the design of moving weights, optimization solution, complexity analysis, and regularization parameters setting. Experimental results are presented in Section V. Section VI concludes the paper.

II. Ordinary Least Squares and Moving Least Squares

In this section, we will first give a brief discussion about ordinary least squares (OLS) and moving least squares (MLS) on fitting one dimensional data, and then revisit them on two-dimensional image signals. In this way, we have a intuitive understanding of the robustness property of MLS, and it provides us a strong motivation for using MLS in the local adaptive image interpolation method which will be introduced in the next section.

A. One-Dimensional Signals

Given a training set $S = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_k, y_k)\}$ of points $\mathbf{x}_i$ with corresponding output y_i, we want to compute an approximation to the data points. Let us consider the following linear regression model:

$$f(\mathbf{x}, \mathbf{w}) = \mathbf{w}^T \phi(\mathbf{x}) = \sum_{i=1}^{d} w_i \phi_i(\mathbf{x}) \tag{1}$$

where $\mathbf{w}$ is the d-dimensional *weight vector* and $\phi(\mathbf{x}) = (\phi_1, \ldots, \phi_d) \in \Re^{d \times 1}$ is the *basic function vector*. We want to best interpolate S with minimum approximation error.

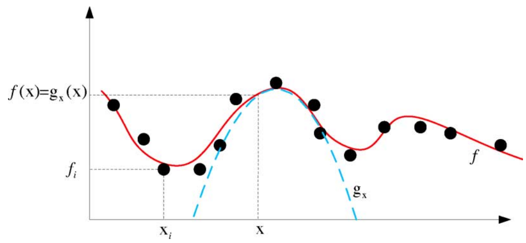


Fig. 1. MLS reconstruction of a given set of scattered data points $(\mathbf{x}_i, f_i)$ in 1-D. The local MLS approximation $g_{\mathbf{x}}$ for the point $\mathbf{x}$ is shown in blue. Its computation and evaluation at every point of the domain yields the complete MLS reconstruction shown in red.

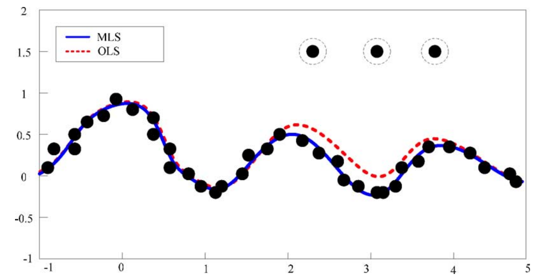


Fig. 2. Toy data example. Three outliers are marked with black dashed circle. Red dashed curve: OLS. Blue solid curve: MLS.

A common criterion for approximation error is OLS, which can be expressed as

$$J_{\text{ols}}(\mathbf{w}) = \sum_{i=1}^{k} \|\mathbf{w}^T \phi(\mathbf{x}_i) - y_i\|^2. \tag{2}$$

Setting the derivative to 0, the optimal $\mathbf{w}$ can be represented by

$$\mathbf{w}^*_{\text{ols}} = (\mathbf{\Phi}^T\mathbf{\Phi})^{-1}\mathbf{\Phi}^T\mathbf{y} \tag{3}$$

where $\mathbf{y} = (y_1, \ldots, y_k)^T$, and $\mathbf{\Phi} = (\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_k))^T \in \Re^{k\times d}$.

OLS obtains a globally defined function $f(\mathbf{x})$ that approximates the given scalar value y_i at point $\mathbf{x}_i$ in the least squares sense. It considers all samples in the training set with equal importance in the process of minimization. As a consequence, OLS is sensitive to outliers, a small amount of outliers within the data can severely bias the least squares estimation.

Opposite to the global-linear property of OLS, MLS is with a local-linear but global-nonlinear manner. The basic idea of MLS approximation is to start with an arbitrary fixed point $\mathbf{x}$ and then move it over the entire domain where the variable is defined. As illustrated in Fig. 1, a continuous approximation $f(\mathbf{x})$ (shown in red) is reconstructed from a set of k data points by computing and evaluating a local approximation $g_{\mathbf{x}}$ at $\mathbf{x}$ (shown in blue)

$$g_{\mathbf{x}}(\mathbf{x}) = \mathbf{w}_{\mathbf{x}}^T \phi(\mathbf{x}). \tag{4}$$

At each point, the approximation function is locally minimized and evaluated using weighted least squares by fitting the neighboring data points [15], which is formulated as

$$j_{mls}(\mathbf{w}_{\mathrm{x}}) = \sum_{i=1}^{k} \theta(\mathbf{x}, \mathbf{x}_i) \left\|\mathbf{w}_{\mathbf{x}}^T \phi(\mathbf{x}_i) - y_i\right\|^2 \tag{5}$$

where $\theta(\mathbf{x}, \mathbf{x}_i)$ is the similarity weight of the sample $\mathbf{x}_i$ for the current interpolation point $\mathbf{x}$. Setting the derivative to 0, the optimal $\mathbf{w}_{\mathbf{x}}$ can be represented by

$$\mathbf{w}^*_{\mathbf{x}} = (\mathbf{\Phi}^T\mathbf{\Theta}\mathbf{\Phi})^{-1}\mathbf{\Phi}^T\mathbf{\Theta}\mathbf{y} \tag{6}$$

where $\mathbf{\Theta} = \text{diag}(\theta(\mathbf{x}, \mathbf{x}_1), \ldots, \theta(\mathbf{x}, \mathbf{x}_k)) \in \Re^{k\times k}$ is a matrix with similarity weights on the diagonal line.

MLS is a appealing technique for data fitting, which provides a high level of control over the function reconstruction, and allows a tunable amount of data smoothing during interpolation. This flexibility provides the capacity for handling outliers and avoiding the artifacts incurred by OLS based reconstruction schemes. Furthermore, the MLS approximation is a continuous reconstruction with well-defined, smooth derivatives which allows for high-quality shape preserving [16]. In MLS, overlapping local neighborhoods and collectively analysis can provide information about the global geometry. Its computation and evaluation at every point of the domain yields the complete MLS reconstruction shown in red.

To illustrate the advantage of MLS over OLS, in Fig. 2, some toy data points are simulated including three outliers (points with dashed circle). OLS (red dashed curve) is clearly affected by the outlying observations. MLS (blue solid curve), which uses bilateral weights [18] as moving weights, improves the fits remarkably well. This is because that the bilateral weights not only consider the geometric closeness but also the label similarity. The three samples with dashed circle receive smaller weights exactly equal to zero since their values are far from the neighboring points. Therefore, they are locally recognized as outliers in regression and the MLS performs much better in this example.

B. *Two-Dimensional Signals*

In the above, we show the approximation power of MLS on simple one dimensional signals. Now let us turn to two dimensional image signals, and discuss why MLS can bring benefits on the task of image interpolation compared with OLS.

In order to generate high-quality HR images, in image interpolation, homogeneous regions should be represented as smoothly as possible, while heterogeneous regions should be as separably as possible. The previous OLS-based methods, such as NEDI and SAI, consider all samples in the local neighborhood with equal importance. They are not well suited for situations where there are some outliers in the measurements. For example, in the situation where the local neighborhood is on the boundary of two regions, the regions on either side of the boundary may be well approximated with a second-order polynomial model *but not near edges* where the model switches from one to another [19]. For such case, the model estimated by OLS will diffuse the information from both sides of discontinuity. Therefore, edges will be smoothed and do not appear as sharp as they should. This multi-model situation often occurs in the image interpolation application due to the fact that in natural images there are a lot of abrupt changes going from one to another object.

One way to deal with the multi-model situation is to first estimate edge that separates the different regions, and then estimate model parameters on each side of the edge. However, this way will introduce another difficult problem: edge detection. In this paper, we turn to a less principled approach. Instead of a multi-model approach we stick to a simpler one-model approach where we use a statistical robust estimator. A robust estimator will only consider the data points from the homogeneous region and disregard the samples from heterogeneous regions as being *statistical outliers*, which is achieved by incorporating moving weights into the objective function to express the relative importance of samples in estimation.

Another important effect making OLS-based estimates questionable is that when collecting measurements from a local neighborhood some samples may have quite different geometric structure from that of the current sample to interpolate. Taking into account that in NEDI and SAI the geometric duality is the basic principle to guide model estimation, the mismatch of geometric duality is also a kind of outlier, which is called *structure outliers*. Since in OLS all samples contribute equally to the final decision, the structure outliers will disturb the estimation and bias from the ground true model parameters will be induced.

To alleviate the influence of duality mismatch, we embed structure information into the moving weights to choose samples with the same or similar geometric structure as that of the current sample. It allows us to consider part of the measurements from the local neighborhood to belong to the model we are interested in, and disregard all other measurements as being outliers which therefore are not relevant in estimating the model parameters. Evidently measurements that are outliers to the ground true model contribute slightly to the final decision. Reducing the influence of the large errors leads to robust error norms.

III. Framework of Regularized Local Linear Regression

In this section, we first describe the interpolation model used in the proposed method. Then we detail the proposed image interpolation framework including kernel ridge regression and graph-based Laplacian regularization.

A. The Interpolation Model

Without loss of generality, the LR image is a down sampled version of the associated HR image by a factor of two. As illustrated in Fig. 3, the black dots represent the measured samples in the LR image and the rest blue and blank dots represent the unmeasured samples in the HR image. The key issue of image interpolation is how to infer the intensity of a missing sample in the HR image according the information hidden in the neighboring pixels.

The proposed approach is based on subdivision of the global image domain Ω into smaller overlapping domain $\Omega = \{\Omega_1, \ldots, \Omega_k\}$. The model parameters are estimated on the fly for each pixel using sample statistics of a local covering.

Supposing $\mathbf{x}_i \in \Re^2$ centered on Ω_i is the current pixel to interpolate in the HR image. To estimate the intensity value of

$$\mathbf{x_i}$$

we utilize the strategy of linear weighting of a set of candidates $\phi(\mathbf{x}_i)$, which is the intensity vector of the k-nearest neighbors of $\mathbf{x}_i$ from all surrounding directions. Specially, we consider a linear affine transformation function $f_i(\cdot; \mathbf{a}_i, b_i)$ defined as follows:

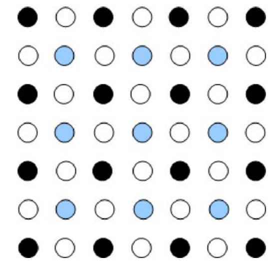


Fig. 3. Formation of an LR image from an HR image by downsampling with a factor of two. The black dots represent the LR image pixels and the rest blue and blank dots represent the missing HR samples.

$$f_i(\mathbf{x}_i) = \langle \mathbf{a}_i, \phi(\mathbf{x}_i) \rangle + b_i \tag{7}$$

where $\mathbf{a}_i$ and b_i are the *weight vector* and *bias* of the linear estimator; $f_i(\mathbf{x}_i)$ is the estimated intensity value of $\mathbf{x}_i$ and $\langle \cdot, \cdot \rangle$ is the inner product. One often deals with the bias term b_i by appending each instance with an additional dimension

$$\Phi(\mathbf{x}_i)^T \leftarrow \left[\phi(\mathbf{x}_i)^T, 1\right], \mathbf{w}_i^T \leftarrow \left[\mathbf{a}_i^T, b_i\right] \tag{8}$$

then the linear transformation function becomes

$$f_i(\mathbf{x}_i) = \mathbf{w}_i^T \Phi(\mathbf{x}_i). \tag{9}$$

Since the function f_i is defined for each point but not shared by all data points in the local neighborhood Ω_i, the proposed method performs nonlinear transformation globally but linear transformation locally. Taking this property into account, we refer to the interpolation model as *local linear regression model*. Note that the local neighborhood can be defined in a number of different ways. Although some methods may be better than others, we keep it simple in this paper by using the same neighborhood size for all missing points in the HR image.

B. Kernel Ridge Regression

We now proceed to devise the MLS-based image interpolation algorithm more formally. Given a local covering $\Omega_i = \{\mathbf{x}_1, \ldots, \mathbf{x}_l, \mathbf{x}_{l+1}, \ldots, \mathbf{x}_{l+u}\}$. The first l points $X_l = \{\mathbf{x}_j\}_{j=1}^{j=l}$ are measured examples, since they are from the LR image and their intensity values are $Y_l = \{y_j\}_{j=1}^{j=l}$. The rest points $X_u = \{\mathbf{x}_j\}_{j=l+1}^{j=l+u}$ are unmeasured samples to be estimated.

Image interpolation is an ill-posed problem. In order to infer the missing samples, one needs to have some prior knowledge on the HR image to be estimated. A reasonable assumption made with the natural image source is that it can be modeled as a locally stationary Gaussian process, i.e., the neighboring samples maybe have the same or similar transformation function. Following this assumption, the optimal model parameter vector $\mathbf{w}_i$ is found by projecting the function f_i onto the neighboring measured examples. Given a set of measured examples $(\mathbf{x}_j, y_j)$, $j = 1, \ldots, l$, we can estimate $\mathbf{w}_i$ by minimizing

$$J(\mathbf{w}_i) = \sum_{j=1}^{l} \theta(\mathbf{x}_i, \mathbf{x}_j) \left\| y_i - \mathbf{w}_i^T \Phi(\mathbf{x}_j) \right\|^2 + \lambda \| L(\mathbf{w}_i) \|^q \tag{10}$$

where the second term is the ℓ_q-norm, which is imposed as an additional estimator complexity penalty on MLS to design more stable empirical minimization.

In our work, we focus on the convex class of regularization functions in (10) which eliminates $q < 1$. We consider the practical range of interest of q-value is $1 \leq q \leq 2$. There are some popular regularization techniques which can be added into the objective function to define $L(\mathbf{w}_i)$ and q, such as Tikhonov regularizer [20] and Total-Variation regularizer [21]. For Tikhonov regularizer, $L(\mathbf{w}_i) = \mathbf{w}_i$ and $q = 2$; For Total-Variation regularizer, $L(\mathbf{w}_i) = \nabla(\mathbf{w}_i)$ and $q = 1$. In practical experiments, we utilize the Tikhonov regularizer as the complexity penalty of estimator, which can be formulated as

$$J(\mathbf{w}_i) = \sum_{j=1}^{l} \theta(\mathbf{x}_i, \mathbf{x}_j) \left\| y_i - \mathbf{w}_i^T \Phi(\mathbf{x}_j) \right\|^2 + \lambda \|\mathbf{w}_i\|^2. \quad (11)$$

This is also known as kernel ridge regression (KRR). The penalty term is stable because it does not depend on data. When λ is large, this term dominates and we have very stable solutions close to $\mathbf{w}_i = \mathbf{0}$. By choosing λ properly, an appropriate amount of ridge's stability translates into good statistical properties of the KRR estimator.

It is easy to see that the solution of KRR estimator takes the form

$$\mathbf{w}_i^* = (\mathbf{\Phi}^T \mathbf{\Theta} \mathbf{\Phi} + \lambda \mathbf{I})^{-1} \mathbf{\Phi}^T \mathbf{\Theta} \mathbf{y} \quad (12)$$

where $\mathbf{I}$ is the identity matrix.

C. Graph-Based Laplacian Regularization

In the previous subsection, we incorporate a complexity penalty term into the MLS framework to keep the solution stable and avoid overfitting. From a algebraic point of view, regularization of MLS is done by tuning the objective function into a strictly convex one, therefore guaranteeing a unique solution. However, regularization is not only a way of gaining an algebraic stability in the reconstruction process. From a Bayesian point of view, regularization should also be a way of exploiting some *prior* information, such as the probability density function (pdf) of images. In this way, a properly chosen regularization can direct the solution toward a better quality outcome by taking into account the proper characteristics of the objective image.

Motivated by recent progress on manifold based semi-supervised learning [17], we refine the KRR interpolation framework by exploring additional discrimination information hidden in unmeasured samples. From a geometric perspective, there is a probability distribution p on $\Re^2 \times \Re$ in HR image. The available LR samples are $(\mathbf{x}, y)$ pairs generated according to $p(\mathbf{x}, y)$, the rest missing samples are simply $\mathbf{x} \in \Re^2$ drawn according to the marginal distribution $p(\mathbf{x})$ of p. In the previous subsection, the induced loss function only expresses relationships between the current unmeasured sample and its neighboring measured samples. By assuming that the support of $p(\mathbf{x})$ is a compact manifold, it is more reasonable to incorporate the geometric structure of the marginal distribution $p(\mathbf{x})$ induced by unmeasured samples into the image interpolation framework. It is usually assumed that there is a specific relationship between $p(\mathbf{x})$ and $p(y|\mathbf{x})$. In another word, we assume that if two points $\mathbf{x}_1$ and $\mathbf{x}_2$ are close in the intrinsic geometry of $p(\mathbf{x})$, then the conditional distribution $p(y|\mathbf{x}_1)$ and $p(y|\mathbf{x}_2)$ should be similar, i.e., $p(y|\mathbf{x})$ should vary smoothly along the geodesics in the intrinsic geometry of $p(\mathbf{x})$.

In this following, we proceed to incorporate such knowledge into the loss function through an additional local smoothness preserving penalty. Given measured samples and unmeasured ones in the local neighborhood, we consider the following loss function:

$$\begin{aligned} J(\mathbf{w}_i) = &\sum_{j=1}^{l} \theta(\mathbf{x}_i, \mathbf{x}_j) \left\| y_j - \mathbf{w}_i^T \Phi(\mathbf{x}_j) \right\|^2 \\ &+ \lambda \|\mathbf{w}_i\|^2 \\ &+ \eta \sum_{m=l+1}^{l+u} \sum_{n=l+1}^{l+u} \theta(\mathbf{x}_m, \mathbf{x}_n) \left\| \mathbf{w}_m^T \Phi(\mathbf{x}_m) \right. \\ &\left. - \mathbf{w}_n^T \Phi(\mathbf{x}_n) \right\|^2 \end{aligned} \quad (13)$$

where $\theta(\mathbf{x}_m, \mathbf{x}_n)$ are the edge weights in the data adjacency graph. The additional regularization term is the graph-Laplacian penalty, which restricts the intrinsic geometric information of the marginal distribution $p(\mathbf{x})$ therefore preserves the smoothness of transformation functions. It imposes a smaller number of equivalence classes on the transformation function space, therefore can guarantee a better generalization error. The final objective function can be viewed as a generalization of KRR to the transductive setting. The parameters $\lambda > 0$ and $\eta > 0$ control the relative contribution of two regularization terms in the objective function. As a result, the task of transformation function learning is to minimize the above cost function:

$$\mathbf{w}_i^* = \arg\min_{\mathbf{w}} J(\mathbf{w}_i). \quad (14)$$

IV. Implementation Details

The proposed algorithm interpolates the unmeasured pixels in the HR image in two pass in a coarse to fine progression. The procedure of the two passes is illustrated in Fig. 3, in which the black dots are the LR image pixels (measured data) and the rest are the HR image pixels to interpolate (unmeasured data). The first pass is to interpolate the pixels located on the cross, which are regarded as measured points in the second pass. The remaining unmeasured points are to be interpolated in the second pass. In the following, let us consider some key issues in our approach: moving weights design, optimization solution, complexity analysis, and regularization parameters setting.

A. Patch-Based Bilateral Moving Weights

In the proposed RLLR framework, moving weights provide the prior with the flexibility to model explicitly the local salient features of an image. When local characteristics of the image differ significantly across spatial domain, setting these control weights in regression can efficiently handle the statistical and structure outliers. Some efforts in other image processing tasks have been initiated in this direction to determine the similarity of local pattern for better spatial adaptation. In particular, the Radial Basis Function (RBF) kernel computes weights decreasing with distance from the neighborhood center, as formulated as follows:

$$\theta(\mathbf{x},\mathbf{x}') = \exp\left\{-\frac{\|\mathbf{x}-\mathbf{x}'\|^2}{\sigma_s^2}\right\}, \sigma_s > 0. \quad (15)$$

The intuition of RBF is that an image typically varies slowly over space, so neighboring pixels are likely to have similar values. However, the assumption of slow spatial variations fails at edges, which are consequently blurred by RBF. To reduce this unpleasant effect, *bilateral filter* [18] is proposed to combine gray levels based on both their geometric closeness and their photometric similarity. The bilateral weights can be represented by the following equation:

$$\theta(\mathbf{x},\mathbf{x}') = \frac{1}{C_{\mathbf{x}}}\exp\left\{-\frac{\|\mathbf{x}-\mathbf{x}'\|^2}{\sigma_s^2}\right\} \times \exp\left\{-\frac{\|y(\mathbf{x})-y(\mathbf{x}')\|^2}{\sigma_f^2}\right\}, \quad \sigma_s > 0, \sigma_f > 0 \quad (16)$$

where $C_{\mathbf{x}}$ is the normalization factor.

RBF and bilateral filter can be regarded as the neighborhood filter since they are both local. Neighborhood filters perform well in presence of moderate noise, but the comparison of the grey level or color values at a single pixel is no more robust when these values get noisier. This drawback is overcome by the *non-local-means* weights [22], in which each weight is proportional to the similarity between the local neighborhood of the pixel being processed and the neighborhood corresponding to other image pixels. The non-local-means weight is defined as follows:

$$\theta(\mathbf{x},\mathbf{x}') = \frac{1}{C_{\mathbf{x}}}\exp\left\{-\frac{G\cdot\|SW(\mathbf{x})-SW(\mathbf{x}')\|^2}{\sigma_p^2}\right\}, \quad \sigma_p > 0 \quad (17)$$

where G is a Gaussian kernel used to take into account the distance between the central pixel and other pixels in the patch, and $SW(\mathbf{x})$ represents the pixel patch whose components are intensity values of pixels in the similarity window centered on $\mathbf{x}$. This patch comparison permits a reliable similarity measure involving pixels which can fall far away from each other.

Image priors in a product form are very attractive since they have the ability to enforce simultaneously many properties on an image. In this paper, we combine the edge-preserving property of bilateral filter and the robust property of non-local-means weights to design efficient moving weights, which are called *patch-based bilateral moving weights* as define as

$$\theta(\mathbf{x},\mathbf{x}') = \frac{1}{C_{\mathbf{x}}}\exp\left\{-\frac{\|\mathbf{x}-\mathbf{x}'\|^2}{\sigma_s^2}\right\} \times \exp\left\{-\frac{G\cdot\|SW(\mathbf{x})-SW(\mathbf{x}')\|^2}{\sigma_p^2}\right\}, \quad \sigma_s > 0, \sigma_q > 0. \quad (18)$$

B. Optimization Solution and Complexity Analysis

In practical experiments, for simplicity we do not exploit all unmeasured samples for the additional local smoothness preserving penalty, but only a subset $\{x_p\}_{p=l+1}^{p=i-1}$ which have been already interpolated before $\mathbf{x}_i$ with the estimated intensity values $\{\widehat{y}_p\}_{p=l+1}^{p=i-1}$. It means that our algorithm consists of selecting a transformation function that fits best the values of measured samples and the estimated intensity values of unmeasured samples provided previous.

Let d be the dimension of the weighting candidates set $\Phi(\mathbf{x})$ in the interpolation model and let $\mathrm{Y}_l \in \Re^{l\times 1}$ denote the column matrix whose components are the intensity values of the measured samples, $\mathrm{Y}_u \in \Re^{u\times 1}$ the column matrix whose components are the intensity values of previous estimated unmeasured samples and $u = i-l-1$. Let $\mathbf{\Phi}_l = [\Phi(\mathbf{x}_1),\ldots,\Phi(\mathbf{x}_l)] \in \Re^{d\times l}$ denote the matrix whose columns are the components of the values by Φ of the measured samples, and similarly $\mathbf{\Phi}_u = [\Phi(\mathbf{x}_{l+1}),\ldots,\Phi(\mathbf{x}_{i-1})] \in \Re^{d\times u}$ the matrix corresponding to the previous estimated unmeasured samples. The final loss function can then be formulated in a matrix form:

$$\begin{aligned} J(\mathbf{w}_i) = &\left(\mathrm{Y}_l - \mathbf{\Phi}_l^T\mathbf{w}_i\right)^T \mathbf{\Theta}(\mathbf{x}_i)\left(\mathrm{Y}_l - \mathbf{\Phi}_l^T\mathbf{w}_i\right) \\ &+ \lambda\mathbf{w}_i^T\mathbf{w}_i \\ &+ \eta\left(\mathrm{Y}_u - \mathbf{\Phi}_u^T\mathbf{w}_i\right)^T \mathbf{\Psi}(\mathbf{x}_i)\left(\mathrm{Y}_u - \mathbf{\Phi}_u^T\mathbf{w}_i\right) \end{aligned} \quad (19)$$

where $\mathbf{\Theta}(\mathbf{x}_i) = \mathrm{diag}(\theta(\mathbf{x}_i,\mathbf{x}_1),\ldots,\theta(\mathbf{x}_i,\mathbf{x}_l))$ is a matrix whose entries in diagonal are moving weights of $\mathbf{x}_i$ with respect to the measured samples and similarly $\mathbf{\Psi}(\mathbf{x}_i) = \mathrm{diag}(\theta(\mathbf{x}_i,\mathbf{x}_{l+1}),\ldots,\theta(\mathbf{x}_i,\mathbf{x}_{i-1}))$ is a matrix where entries in diagonal are the weights of $\mathbf{x}_i$ with respect to the previous estimated unmeasured samples.

To derive the optimal transformation vector $\mathbf{w}_i$, we take the derivative of the loss function J in (19) with respect to $\mathbf{w}_i$ and set the derivative to 0, then the optimal $\mathbf{w}_i$ can be represented by

$$\begin{aligned} \mathbf{w}_i^T = &\left(\mathrm{Y}_l^T\mathbf{\Theta}(\mathbf{x}_i)\mathbf{\Phi}_l^T + \eta\mathrm{Y}_u^T\mathbf{\Psi}(\mathbf{x}_i)\mathbf{P}(\mathbf{x}_i)^T\right) \\ &\times\left(\mathbf{\Phi}_l\mathbf{\Theta}(\mathbf{x}_i)\mathbf{\Phi}_l^T + \lambda\mathbf{I} + \eta\left(\Theta_u^T\mathbf{1}\right)\mathbf{\Phi}_u\mathbf{\Phi}_u^T\right)^{-1} \end{aligned} \quad (20)$$

where $\mathbf{P}(\mathbf{x}_i) \in \Re^{d\times u}$ is the matrix whose columns are all the repetition of $\Phi(\mathbf{x}_i)$, $\Theta_u = [\theta(\mathbf{x}_i,\mathbf{x}_{l+1}),\ldots,\theta(\mathbf{x}_i,\mathbf{x}_{i-1})]^T$ is the column vector whose components are moving weights of unmeasured samples, $\mathbf{1}$ is the column vector whose components are all 1.

This result gives a closed-form solution based on the inversion of a matrix in $\Re^{d\times d}$. Let $T(d)$ be the time complexity of computing the inverse of a matrix in $\Re^{d\times d}$, and $T(d) = \mathrm{O}(d^3)$ using standard method or $T(d) = \mathrm{O}(d^{2.376})$ with the method of Coppersmith and Winogard. The time complexity of the computation of $\mathbf{w}_i$ from $\mathbf{\Phi}_l$, $\mathbf{\Phi}_u$, Y_l, and Y_u is thus in $\mathrm{O}(T(d)+(l+u)d^2)$. Note that in our method we set $\Phi(\mathbf{x})$ as the four 8-connected neighboring samples of $\mathbf{x}$; thus, $d = 5$ (with an additional dimension for appending).

As a consequence, the overall computation complexity of our method is $\mathrm{O}(T(d) + (l+u)d^2) \times N$, where N is the number of unmeasured samples whose model parameters are needed to estimate. For the offline image interpolation scenario, we can fully exploit the advantage of our method to achieve the best performance. For the online scenario, we can manage the computational complexity by reducing the number of samples to estimate. One way is to apply the RLLR method only on edge pixels (pixels near an edge); for nonedge pixels (pixels in smooth regions), we still use simple bilinear or bicubic interpolation. Another way is to generalize our method from point-wise to piece-wise (block-based), i.e., a block of unmeasured samples share the same model parameters.

C. Regularization Parameters Setting

One important issue of the proposed method is to determine appropriate values for two regularization parameters λ and η.

For the KRR parameter λ, we generalize the previous work on spectral regression discriminant analysis [26] to the kernel version, and can derive a data-adaptively optimal solution.

For (6) and (12), $\mathbf{\Theta}$ is a diagonal matrix, we obtain

$$\mathbf{\Phi}^T\mathbf{\Theta}\mathbf{\Phi} = \mathbf{\Phi}^T\left(\mathbf{\Theta}^{\frac{1}{2}}\right)^T\mathbf{\Theta}^{\frac{1}{2}}\mathbf{\Phi} = \hat{\mathbf{\Phi}}^T\hat{\mathbf{\Phi}} \tag{21}$$

where $\hat{\mathbf{\Phi}} = \mathbf{\Theta}^{(1/2)}\mathbf{\Phi}$. We perform singular value decomposition (SVD) on $\hat{\mathbf{\Phi}}$, and have $\hat{\mathbf{\Phi}} = \mathbf{USV}^T$, where $\mathbf{U}$ and $\mathbf{V}$ are unitary matrices, and $\mathbf{S}$ is the singular value matrix.

In a local neighborhood, the linear regression model can be represented as

$$\mathbf{\Phi}\mathbf{w} + \varepsilon = \mathbf{y} \tag{22}$$

where ε is an $l \times 1$ vector of random error with $E[\varepsilon] = 0$ and $Var[\varepsilon] = \sigma^2\mathbf{I}_l$. For the MLS estimator $\hat{\mathbf{w}}$, we can derive

$$\hat{\mathbf{w}} = \mathbf{w} + (\mathbf{\Phi}^T\mathbf{\Theta}\mathbf{\Phi})^{-1}\mathbf{\Phi}^T\mathbf{\Theta}\varepsilon. \tag{23}$$

According to (6) and (12), we can derive the relationship between MLS estimator $\hat{\mathbf{w}}$ and KRR estimator $\tilde{\mathbf{w}}$ as follows:

$$\tilde{\mathbf{w}} = (\mathbf{\Phi}^T\mathbf{\Theta}\mathbf{\Phi} + \lambda\mathbf{I})^{-1}(\mathbf{\Phi}^T\mathbf{\Theta}\mathbf{\Phi})\hat{\mathbf{w}} = \mathbf{R}\hat{\mathbf{w}}. \tag{24}$$

A good parameter λ should reduce the mean square error (MSE) of KRR estimator $\tilde{\mathbf{w}}$. Therefore, it is necessary to derive $E[D^2(\lambda)]$, where $D(\lambda)$ denotes the distance from KRR estimator $\tilde{\mathbf{w}}$ to the ground true model weights $\mathbf{w}$. We have

$$\begin{aligned} E[D^2(\lambda)] &= E[(\tilde{\mathbf{w}} - \mathbf{w})^T(\tilde{\mathbf{w}} - \mathbf{w})] \\ &= E[(\hat{\mathbf{w}} - \mathbf{w})^T\mathbf{R}^T\mathbf{R}(\hat{\mathbf{w}} - \mathbf{w})] \\ &\quad + (\mathbf{R}\mathbf{w} - \mathbf{w})^T(\mathbf{R}\mathbf{w} - \mathbf{w}). \end{aligned} \tag{25}$$

Substituting (23) and (24) into (25), we have

$$\begin{aligned} E[D^2(\lambda)] &= \sigma^2(\mathrm{Tr}(\mathbf{\Phi}^T\mathbf{\Theta}\mathbf{\Phi} + \lambda\mathbf{I})^{-1} \\ &\quad - \lambda\mathrm{Tr}(\mathbf{\Phi}^T\mathbf{\Theta}\mathbf{\Phi} + \lambda\mathbf{I})^{-2}) \\ &\quad + \lambda^2\mathbf{w}^T(\mathbf{\Phi}^T\mathbf{\Theta}\mathbf{\Phi} + \lambda\mathbf{I})^{-2}\mathbf{w}. \end{aligned} \tag{26}$$

Let $\mathbf{c} = [c_1, c_2, \ldots, c_d]^T$, and $\mathbf{c} = \mathbf{V}^T\mathbf{w}$, we can obtain

$$\begin{aligned} E[D^2(\lambda)] &= \sigma^2\left(\sum_{i=1}^{d}\frac{1}{s_i^2 + \lambda} - \sum_{i=1}^{d}\frac{\lambda}{(s_i^2 + \lambda)^2}\right) \\ &\quad + \lambda^2\mathbf{c}^T(\mathbf{S}^T\mathbf{S} + \lambda\mathbf{I})^{-2}\mathbf{c} \\ &= \sigma^2\sum_{i=1}^{d}\frac{s_i^2}{(s_i^2 + \lambda)^2} + \lambda^2\sum_{i=1}^{d}\frac{\mathbf{c}_i^2}{(s_i^2 + \lambda)^2} \end{aligned} \tag{27}$$

where s_i represents the ith largest singular value of $\hat{\mathbf{\Phi}}$.

Note that in the above equation the first term is monotonically decreasing while the second term is monotonically increasing. Taking the derivative with respect to λ, we find the minimum of MSE falls in the interval of λ:

$$\left[\min\left\{\frac{\sigma^2}{c_i^2}\right\}, \max\left\{\frac{\sigma^2}{c_i^2}\right\}\right]_{i=1}^{d}. \tag{28}$$

The optimal λ should translate an appropriate amount of ridges's stability into the robust property on perturbation in the parameter space. Therefore, we induce a noise error υ on $\{s_i\}$, and estimate the optimal λ by solving the following minimization problem:

$$\begin{aligned} &\lambda^* = \arg\min_{\lambda} E[\|\tilde{\mathbf{w}}(\lambda, s + \upsilon) - \tilde{\mathbf{w}}(\lambda, s)\|^2] \\ &\text{s.t.} \quad \lambda^* \in \left[\min\left\{\frac{\sigma^2}{c_i^2}\right\}, \max\left\{\frac{\sigma^2}{c_i^2}\right\}\right]_{i=1}^{d}. \end{aligned} \tag{29}$$

The problem can be further relaxed to the following simple form:

$$\lambda^* = \arg\min_{\lambda}\sum_{i=1}^{l}\left(\lambda - s_i^2\right)^2. \tag{30}$$

Finally, we can obtain the optimal solution for each local covering

$$\lambda^* = \frac{1}{L}\sum_{i=1}^{l}s_i^2 \tag{31}$$

where L is a normalized parameter to keep λ in (0, 1), which is set as 10^6 in the practice implementation. Since $\{s_1^2, \ldots, s_l^2\}$ are the eigenvalues of symmetric matrix $\mathbf{\Phi}^T\mathbf{\Theta}\mathbf{\Phi}$, the sum is the trace of $\mathbf{\Phi}^T\mathbf{\Theta}\mathbf{\Phi}$.

For Graph-Laplacian regularization parameter η, there is no reliable optimal method to estimate it, we just empirically set it as a value from $\{10^{-3}, 10^{-2}, 10^{-1}\}$.

V. Experimental Results and Analysis

In this section, extensive experimental results are presented to demonstrate the superiority of the proposed RLLR algorithm. For thoroughness and fairness of our comparison study, we select a large test images set including ten widely used images in the literature and a computer generated image: the letter A, as illustrated in Fig. 4. We downsample these HR images by a factor of two in both row and column dimensions to get the corresponding LR images, from which the original HR images are reconstructed by the proposed and competing methods. Both direct and average downsampling images are considered. Note that our method can be easily generalized to enlarge images with a factor greater than two, in the same way as stated in [27]. For color images in RGB representation, each channel can be treated independently as a grayscale image. The interpolated images of the three channels are then recombined to give the final image.

For comprehensive comparison, the RLLR algorithm is compared with some representative work in the literature. More specifically, eight approaches are included in our comparative study: 1) LAZA [6]; 2) NEDI [11]; 3) DFDF [9]; 4) KR [14]; 5)

Fig. 4. Eleven sample images in the test set.

TABLE I
OBJECTIVE QUALITY COMPARISON OF EIGHT INTERPOLATION ALGORITHMS (IN dB) FOR DIRECT DOWNSAMPLING

Method	LAZA		NEDI		DFDF		KR		ICBI		INEDI		SAI		RLLR	
	PSNR	EPSNR	PSNR	EPSNR	PSNR	EPSNR	PSNR	EPSNR	PSNR	EPSNR	PSNR	EPSNR	PSNR	EPSNR	PSNR	EPSNR
Airplane	30.17	19.48	28.69	15.42	30.53	19.44	29.11	16.01	30.08	18.68	30.66	19.65	30.72	19.25	**30.97**	**19.86**
Lena	33.36	28.31	33.57	27.75	33.96	28.10	33.96	27.97	34.05	26.99	34.11	27.86	**34.72**	**28.97**	34.41	28.65
Flowers	25.65	20.37	25.62	19.94	25.74	20.40	25.79	20.23	25.20	19.24	25.89	20.66	25.96	**20.78**	**26.20**	20.75
Girl	31.84	29.50	31.84	28.30	31.81	29.41	31.92	29.10	31.27	28.82	32.24	29.30	31.77	29.60	**32.29**	**29.74**
Door	32.28	26.23	32.14	25.73	32.27	25.93	32.20	25.86	31.71	24.33	32.41	25.99	32.46	26.07	**32.60**	**26.39**
Peppers	31.51	22.42	29.30	17.83	31.87	22.53	31.02	20.53	31.70	22.18	32.05	22.43	31.84	22.08	**32.15**	**22.77**
Splash	33.54	19.74	31.38	15.49	33.79	19.79	33.38	19.07	33.32	**21.29**	33.69	21.28	33.54	19.14	**33.99**	19.87
Baboon	20.09	18.06	19.91	17.75	19.87	18.01	19.99	17.79	19.06	16.56	19.55	18.06	19.92	18.04	**20.67**	**18.77**
Tower	38.45	28.70	39.91	28.36	39.69	27.64	40.28	28.57	38.94	25.93	40.74	28.19	41.49	28.82	**41.97**	**29.43**
Butterfly	28.70	21.10	28.96	21.02	29.67	20.76	29.54	21.25	29.91	20.26	30.07	21.36	29.93	20.46	**30.32**	**21.56**
Letter	26.07	13.53	26.47	13.59	27.21	14.22	27.02	13.13	26.71	15.13	**27.86**	**14.81**	27.23	13.98	27.43	14.75
Average	30.15	22.49	29.79	21.02	30.58	22.38	30.38	21.78	30.17	21.77	30.84	22.69	30.88	22.47	**31.18**	**22.96**

ICBI [10]; 6) INEDI [12]; 7) SAI [13]; 8) the proposed RLLR approach.

There are a few parameters involved in the RLLR algorithm. As for the sizes of similarity window and local training window, we empirically set as 7×7 and 21×21, respectively. The variance parameters σ_s and σ_p in computing moving weights are fixed to 0.05.

A. Noise-Free Images Interpolation

First, let us consider the objective and subjective quality of eight algorithms on noise-free images. We quantify the objective performance of all methods by PSNR and edge PSNR (EPSNR). Tables I and II tabulate the objective performance of the eight different methods for direct downsampling and average downsampling, respectively. It can be observed that the proposed RLLR algorithm achieves the highest average PSNR value for both cases. Since PSNR is an average quality measurement over the whole image, we exploit EPSNR to focus on fidelity of image edges. In our study, the Sobel edge filter is used to locate the edge in the original image, and the PSNR of the pixels on the edge are used to generate the EPSNR. Compared with SAI, the average EPSNR gains of our method are 0.49 and 0.55 dB, respectively. It demonstrates the proposed RLLR algorithm produces significantly smaller interpolation errors along edges than the competing methods for both cases.

Although PSNR and EPSNR can measure the intensity difference between two images, it is well-known that they may fail to describe the visual perception quality of the image. How to evaluate the visual quality of an image is a very difficult problem and an active research topic. In the literature, the SSIM index proposed in [23] is one of the most commonly used metric for image visual quality assessment. Recently, another powerful image quality assessment metric, named FSIM, has been proposed [24]. In our study, we use SSIM and FSIM to measure the visual quality of these interpolation algorithms. From Table III, it can be seen for direct downsampling RLLR achieves the highest average SSIM and FSIM scores among the competing methods. For average downsampling, RLLR achieves the competitive results with SAI, as depicted in Table IV.

Given the fact that human visual system (HVS) is the ultimate receiver of the enlarged images, we also show the subjective comparison results. In the test image *Airplane*, a sharp diagonal line is visible on the side of airscrew. The test image *Butterfly* exhibits strong and sharp edges in varying directions. Such characteristics make them prime images to test edge blurring

TABLE II
OBJECTIVE QUALITY COMPARISON OF EIGHT INTERPOLATION ALGORITHMS (IN dB) FOR AVERAGE DOWNSAMPLING

Method	LAZA		NEDI		DFDF		KR		ICBI		INEDI		SAI		RLLR	
	PSNR	EPSNR	PSNR	EPSNR	PSNR	EPSNR	PSNR	EPSNR	PSNR	EPSNR	PSNR	EPSNR	PSNR	EPSNR	PSNR	EPSNR
Airplane	30.35	20.12	29.26	16.31	29.76	17.68	30.83	20.13	30.98	19.94	31.08	20.39	31.12	20.00	**31.26**	**20.58**
Lena	33.29	28.47	33.82	28.23	33.82	28.32	34.15	28.56	34.71	28.29	34.35	28.42	**34.90**	**29.35**	34.57	29.16
Flowers	26.27	20.96	26.37	20.72	26.28	20.89	26.48	21.05	26.46	20.88	26.75	21.42	**26.82**	**21.56**	26.8	21.44
Girl	32.62	29.99	32.69	28.99	32.61	29.75	32.84	30.22	32.74	30.29	33.01	29.83	32.92	30.44	**33.01**	**30.45**
Door	32.55	26.68	32.58	26.41	32.41	26.44	32.7	26.49	32.64	25.82	32.77	26.61	**32.98**	26.82	32.89	**26.89**
Peppers	31.96	23.04	30.07	18.71	31.55	21.45	32.54	23.24	32.71	23.16	32.66	23.10	32.58	22.81	**32.79**	**23.51**
Splash	33.89	20.62	32.06	16.44	33.86	20.09	34.35	20.68	34.05	20.31	34.01	20.08	34.17	20.03	**34.49**	**20.71**
Baboon	21.05	19.01	20.93	18.77	20.78	18.71	20.94	18.94	20.73	18.25	20.91	**19.22**	21.08	19.14	**21.15**	19.11
Tower	37.76	29.54	39.79	29.46	39.31	29.64	39.69	29.00	40.15	28.03	40.50	29.39	41.15	29.99	**41.31**	**30.57**
Butterfly	28.44	21.28	29.07	21.63	29.21	21.50	29.55	21.15	30.37	21.50	30.38	22.05	29.9	21.05	**30.43**	**22.37**
Letter	26.08	13.52	26.52	14.41	27.08	13.17	27.23	14.23	26.89	**15.75**	26.85	14.62	26.1	12.22	**27.46**	14.73
Average	30.39	23.02	30.29	21.83	30.61	22.51	31.03	23.06	31.13	22.93	31.21	23.19	31.25	23.04	**31.47**	**23.59**

TABLE III
SUBJECTIVE QUALITY COMPARISON OF EIGHT INTERPOLATION ALGORITHMS FOR DIRECT DOWNSAMPLING

Method	LAZA		NEDI		DFDF		KR		ICBI		INEDI		SAI		RLLR	
	SSIM	FSIM	SSIM	FSIM	SSIM	FSIM	SSIM	FSIM	SSIM	FSIM	SSIM	FSIM	SSIM	FSIM	SSIM	FSIM
Airplane	0.9122	0.9794	0.9110	0.9782	0.9144	0.9804	0.9049	0.9798	0.9085	0.9781	0.9166	0.9811	0.9171	0.9817	**0.9188**	**0.9823**
Lena	0.9109	0.9855	0.9112	0.9862	0.9129	0.9871	0.9097	0.9875	0.9112	0.9868	0.9175	0.9876	**0.9184**	**0.9884**	0.9180	0.9879
Flowers	0.6850	0.9421	0.6884	0.9412	0.6844	0.9390	0.6648	0.9440	0.6651	0.9316	0.6974	0.9428	0.6973	0.9415	**0.7056**	**0.9465**
Girl	0.7809	0.9688	0.7842	0.9684	0.7741	0.9656	0.7675	0.9701	0.7523	0.9590	**0.7933**	**0.9717**	0.7702	0.9645	0.7924	0.9715
Door	0.8618	0.9632	0.8576	0.9621	0.8607	0.9637	0.8501	0.9633	0.8501	0.9575	0.8625	0.9622	0.8643	**0.9648**	**0.8669**	0.9640
Peppers	0.8735	0.9788	0.8737	0.9777	0.8732	0.9790	0.8701	0.9798	0.8638	0.977	0.8822	0.9816	0.8752	0.9796	**0.8804**	**0.9811**
Splash	0.9281	0.9811	0.9285	0.9829	0.9296	0.9829	0.9245	0.9824	0.9241	0.9826	0.9328	**0.9852**	0.9298	0.9825	**0.9332**	0.9848
Baboon	0.5410	0.8375	0.5416	0.8369	0.5326	0.8364	0.4923	0.8236	0.5073	0.8359	0.5364	0.8357	0.5414	0.8433	**0.5602**	**0.8461**
Tower	0.9875	0.9869	0.9904	0.9909	0.9892	0.9895	0.9897	0.9903	0.9842	0.9846	0.9929	0.9923	0.9919	0.9920	**0.9950**	**0.9927**
Butterfly	0.9416	0.9454	0.9427	0.9462	0.9501	0.9546	0.9468	0.9464	0.9513	0.9549	0.9507	0.9518	0.9554	**0.9575**	**0.9555**	0.9558
Letter	0.9706	0.9738	0.9696	0.9598	0.9786	0.9760	0.9774	0.9721	0.9737	0.9751	**0.9822**	**0.9829**	0.9668	0.9767	0.9801	0.9793
Average	0.8539	0.9584	0.8544	0.9573	0.8545	0.9595	0.8453	0.9581	0.8447	0.9566	0.8603	0.9614	0.8571	0.9611	**0.8641**	**0.9629**

and edge blocking effects. Figs. 5 and 6 illustrate the subjective quality comparison on these two test images. It can be clearly observed that the images reconstructed by the LAZA interpolator suffer from blurred edges, jaggies, and annoying ringing artifacts. The NEDI method is competitive in terms of visual quality, since it can reconstruct sharp large-scale edges well. But it has difficulty with small edges and textures, producing ringing artifacts and spurious small edges. The DFDF method is slightly inferior to the NEDI method in strong edge regions, while it performs better than NEDI in regions containing small-scale features. The SAI method shows improvements over the NEDI and DFDF methods in the regions of small-scale edges and textures, eliminating the visual defects of the NEDI method. However, since samples contribute uniformly in the process of image reconstruction, statistical outliers still confuse the edges and fine textures. The proposed RLLR technique produces the most visually pleasant results among all competing methods. The produced edges in our method are clean and sharp. Thanks to the more powerful kernel weights and the additional local smoothness preserving penalty, our method achieves more wonderful visual quality compared with KR and INEDI. Most visual artifacts appeared in the results of KR and INEDI, such as jaggies and ringings, are eliminated in the proposed method. The outstanding performance of the proposed method is more vivid by observing the error images. As illustrated in Fig. 7, the proposed RLLR algorithm produces smaller interpolation error than other

TABLE IV
SUBJECTIVE QUALITY COMPARISON OF EIGHT INTERPOLATION ALGORITHMS FOR AVERAGE DOWNSAMPLING

Method	LAZA		NEDI		DFDF		KR		ICBI		INEDI		SAI		RLLR	
	SSIM	FSIM	SSIM	FSIM	SSIM	FSIM	SSIM	FSIM	SSIM	FSIM	SSIM	FSIM	SSIM	FSIM	SSIM	FSIM
Airplane	0.9109	0.9819	0.9124	0.9826	0.9155	0.9849	0.9029	0.9827	0.9172	0.9865	0.9156	0.9842	**0.9196**	**0.9865**	0.9184	0.9853
Lena	0.9125	0.9866	0.9156	0.989	0.9181	0.9904	0.9098	0.9893	0.9226	0.9926	0.9183	0.9896	**0.9240**	**0.9919**	0.9216	0.9905
Flowers	0.6956	0.9623	0.7030	0.9632	0.6995	0.9641	0.6733	0.964	0.7006	0.9653	0.7088	0.966	**0.7142**	**0.9676**	0.7121	0.9660
Girl	0.8027	0.9826	0.8048	0.9836	0.8022	0.9836	0.7883	0.9839	0.7975	0.9829	0.8080	0.9851	0.8035	0.9839	**0.8085**	**0.9848**
Door	0.8621	0.9700	0.8611	0.9701	0.8644	0.9729	0.8491	0.9701	0.8641	0.9736	0.8612	0.9699	**0.8698**	**0.9749**	0.8676	0.9710
Peppers	0.8846	0.986	0.8863	0.9866	0.889	0.9889	0.8801	0.9878	0.8886	0.9904	0.8905	0.9894	0.8920	**0.9900**	**0.8924**	0.9898
Splash	0.9307	0.9833	0.9317	0.9858	0.9346	0.9864	0.9262	0.9349	0.9351	0.9388	0.9336	0.9872	0.9356	0.9865	**0.9368**	**0.9879**
Baboon	0.5482	0.8305	0.5477	0.8303	0.5443	0.8339	0.4975	0.7993	0.5509	0.8045	0.5552	0.8046	**0.5598**	**0.8450**	0.5384	0.8011
Tower	0.9844	0.9831	0.9891	0.9887	0.9891	0.9885	0.9859	0.9855	0.9883	0.9883	0.9894	0.9884	0.9913	**0.9913**	**0.9931**	0.9906
Butterfly	0.9375	0.9372	0.9431	0.9416	0.9487	0.9486	0.9425	0.9384	0.9547	0.9528	0.9506	0.9473	**0.9546**	**0.9520**	0.9544	0.9502
Letter	0.9707	0.9736	0.9695	0.9604	0.9788	0.9761	0.9777	0.9720	0.9744	0.9752	0.9728	0.9737	0.9639	0.9664	**0.9802**	**0.9791**
Average	0.8582	0.9616	0.8604	0.9620	0.8622	0.9653	0.8485	0.9553	0.8631	0.9592	0.8640	0.9623	**0.8662**	**0.9669**	0.8658	0.9633

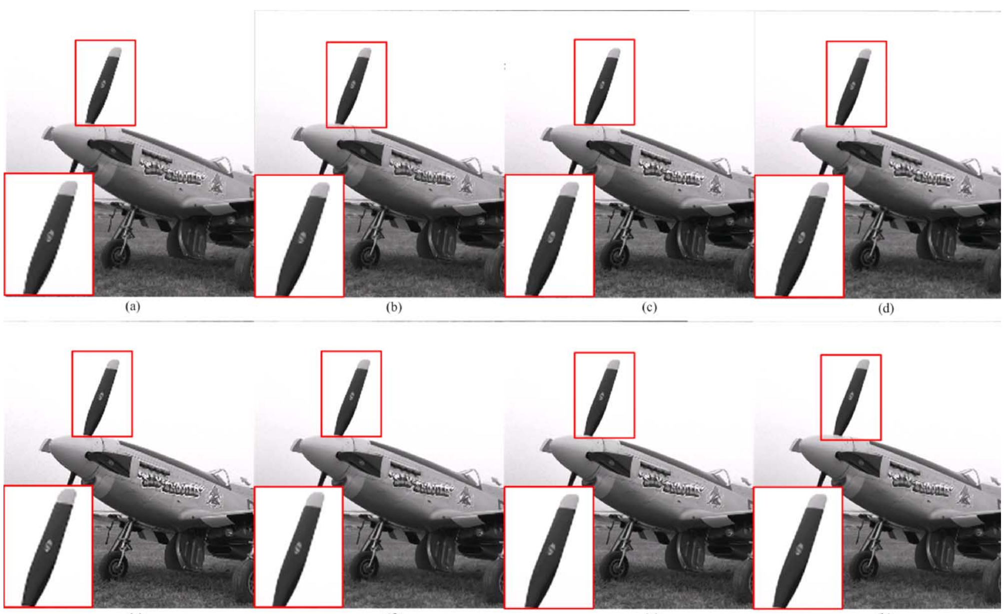


Fig. 5. Subjective quality comparison for *Airplane*. (a) LAZA. (b) NEDI. (c) DFDF. (d) KR. (e) ICBI. (f) INEDI. (g) SAI. (h) RLLR.

methods on the test image *Butterfly*. Such results clearly demonstrate the superiority of the proposed RLLR method in reconstructing the high frequency, such as edges and textures.

B. Noisy Images Interpolation

The experimental results shown above are all based on noise-free image signals. Since we argue the proposed method is more robust compared with other AR-based algorithms, it is interesting to consider noisy image interpolation problems [27]. Specially, we consider compressed image interpolation. That is, the LR images are no longer noise free but compressed by JPEG. We think this case is more general in practical applications. We set the quality factor to 75 and compare the proposed method with the other AR-based algorithms. The experimental results are depicted in Table V. For objective measurements, our method achieves highest average PSNR and EPSNR values. Note that the average gains compared with SAI are up to 0.28 and 0.38 dB with respect to PSNR and edge PSNR, respectively. For subjective measurements, our method achieves highest average SSIM value and the second highest average FSIM score among all methods.

C. Running Time Versus Performance Comparison

In Section IV-B, we give a detailed analysis on the complexity of RLLR. Now let us consider the practical processing time and performance comparison. Table VI gives the PSNR versus average processing times results

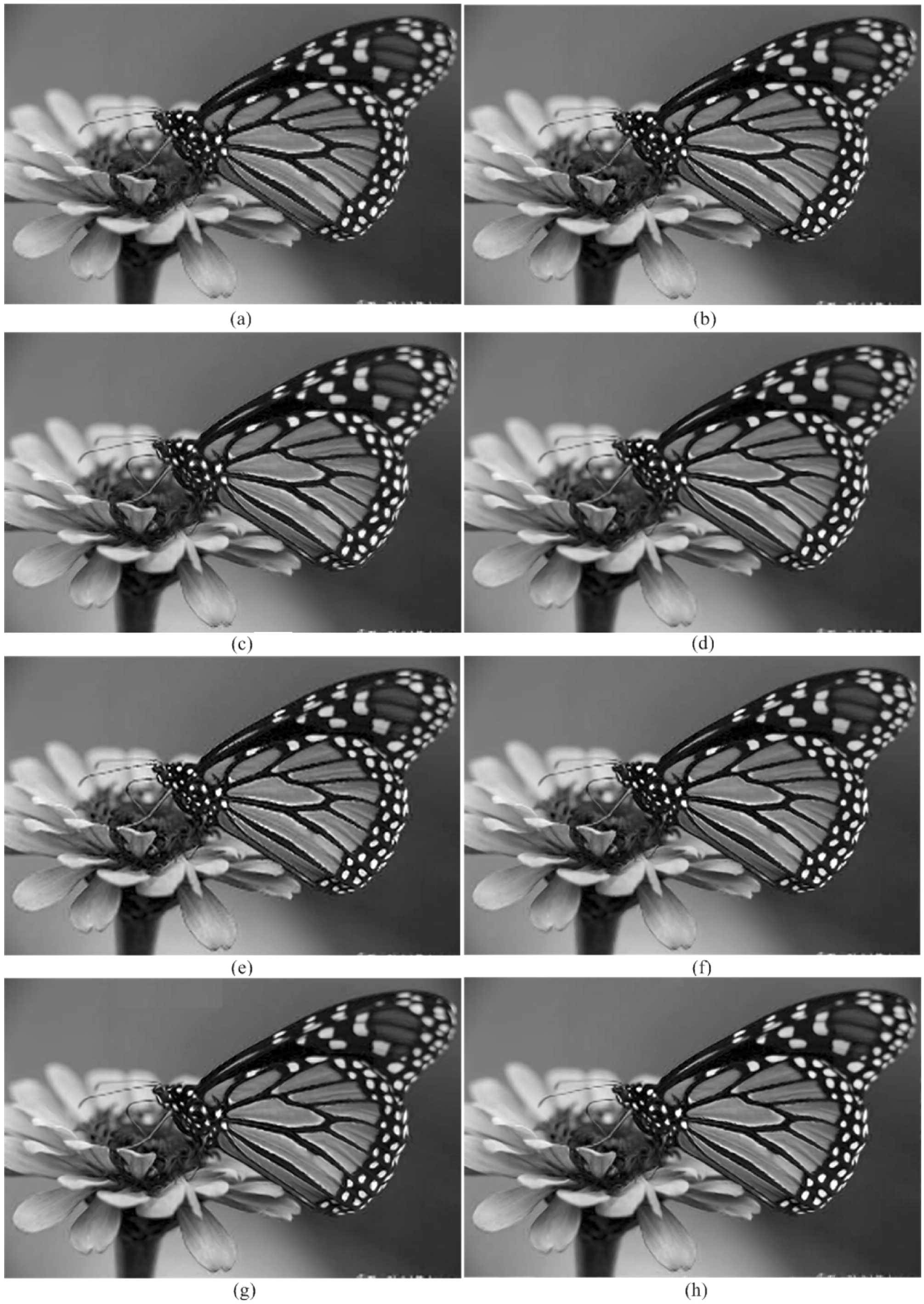


Fig. 6. Subjective quality comparison for *Butterfly*. (a) LAZA. (b) NEDI. (c) DFDF. (d) KR. (e) ICBI. (f) INEDI. (g) SAI. (h) RLLR.

on a typical computer (2.5-GHz Intel Dual Core, 3-GB Memory) of compared algorithms, and for the proposed RLLR algorithm, both offline and online scenario are considered. All methods are run on Matlab. Since there is no Matlab source code available, for fairness of comparison, we omitted the result of SAI. As depicted in Table VI, although the computational complexity is higher, the proposed RLLR model with offline case, which means almost all pixels are interpolated by RLLR, achieves much better quality than other methods. To speed up the proposed algorithm, we can turn to the online case where we control the number of RLLR interpolated pixels by setting a larger threshold for covariance of the four 8-connected samples of the current pixel. In this way, we can fast our algorithm at the expense of some performance loss. For offline case, the covariance threshold is set to 8, and for online case, the threshold is set to 400. From the results, we can find the online case achieve 0.34 dB gain compared with the DFDF method at the expense of about three times running time, and for INEDI, our method achieves 0.12-dB gain using a little less running time. Further study on reducing the computational complexity of the proposed RLLR model is needed.

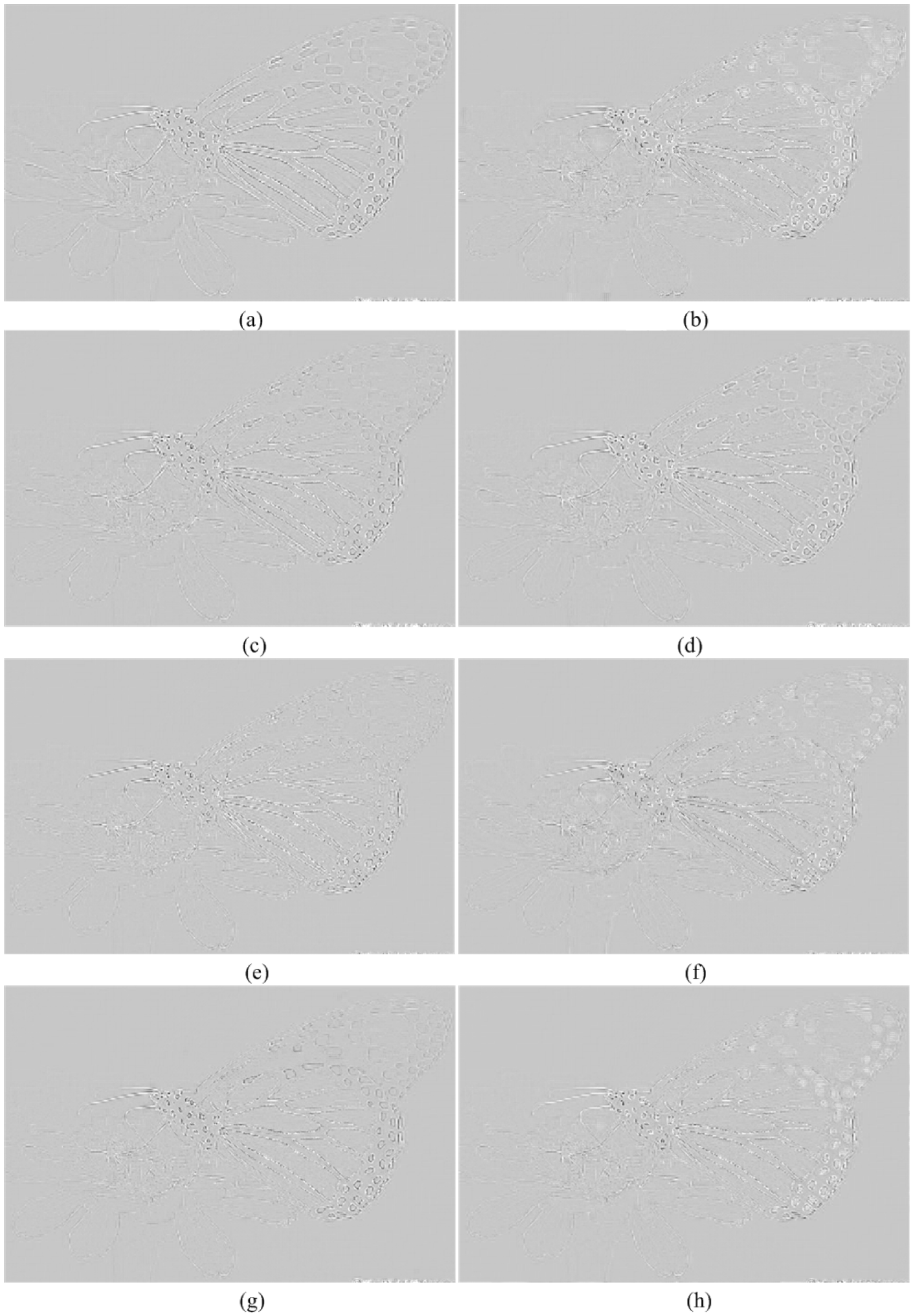


Fig. 7. Reconstruction error comparison for *Butterfly*. (a) LAZA. (b) NEDI. (c) DFDF. (d) KR. (e) ICBI. (f) INEDI. (g) SAI. (h) RLLR.

D. Contribution Analysis

Compared with the NEDI method which is based on OLS, the proposed RLLR method includes three further improvements: moving least squares, kernel ridge regression and graph-Laplacian regularization. We analysis the contribution of each component to the final interpolation performance. As illustrated in Fig. 8, MLS achieves significant PSNR gain over OLS, which demonstrates the proposed patch-based bilateral moving weights design method is efficient. For test images *Flowers*, *Girl*, *Door*, and *Butterfly*, kernel ridge regression further achieves some gains over MLS, while for the test image *Baboon*, kernel ridge regression loses to some extent. This is because in *Baboon* hairs are sharp and discontinuous, ℓ_2-norm destroys the non-smooth property. For such case, ℓ_1-norm may work better. Over kernel ridge regression, graph-Laplacian regularization further improves the quality of generated images.

VI. Conclusion

In this paper, we present a novel image interpolation scheme based on regularized local linear regression. On one hand, we

TABLE V
OBJECTIVE AND SUBJECTIVE QUALITY COMPARISON OF AR BASED INTERPOLATION ALGORITHMS FOR COMPRESSED IMAGES

Method	NEDI				INEDI				SAI				RLLR			
	PSNR	EPSNR	SSIM	FSIM	PSNR	EPSNR	SSIM	FSIM	PSNR	EPSNR	SSIM	FSIM	PSNR	EPSNR	SSIM	FSIM
Airplane	28.15	15.33	0.8730	0.9684	29.84	19.34	0.8803	0.9716	29.87	18.99	0.8791	0.9705	**30.11**	**19.53**	**0.8816**	**0.9725**
Lena	32.17	26.77	0.8715	0.9773	32.54	26.73	0.8773	0.9789	**32.83**	**27.47**	0.8765	0.9779	32.72	27.28	**0.8771**	**0.9785**
Flowers	24.99	19.67	0.6287	0.9311	25.20	20.33	0.6362	0.9320	25.27	20.39	0.6320	0.9302	**25.55**	**20.44**	**0.6447**	**0.9371**
Girl	30.91	27.76	0.7375	0.9605	31.24	28.14	**0.7468**	**0.9646**	30.85	28.55	0.7260	0.9566	**31.25**	**28.62**	0.7438	0.9634
Door	31.02	25.11	0.7946	0.9489	31.26	25.24	0.8009	0.9493	31.25	25.36	0.7986	0.9504	**31.37**	**25.59**	**0.8018**	**0.9509**
Peppers	28.75	17.75	0.8322	0.9686	31.11	22.07	**0.8409**	**0.9732**	30.89	21.76	0.8343	0.9697	**31.12**	**22.28**	0.8385	0.9720
Splash	30.85	15.50	0.8867	0.9721	**33.38**	**20.04**	**0.8908**	**0.9758**	32.62	19.02	0.8857	0.9713	33.01	19.69	0.8896	0.9745
Baboon	19.61	17.44	0.5068	0.8259	19.24	17.77	0.5004	0.8261	19.64	17.82	0.5070	**0.8310**	**20.45**	**18.58**	**0.5281**	0.8175
Tower	36.32	27.41	0.9465	0.9574	36.96	27.20	0.9537	**0.9630**	37.04	27.73	0.9497	0.9588	**37.20**	**28.02**	**0.9525**	0.9599
Butterfly	28.19	20.76	0.9077	0.9191	29.09	21.01	0.9185	0.9277	29.03	20.32	0.9201	0.9275	**29.30**	**21.18**	**0.9216**	**0.9287**
Average	29.10	21.35	0.7985	0.9429	29.99	22.79	0.8046	**0.9462**	29.93	22.74	0.8009	0.9444	**30.21**	**23.12**	**0.8079**	0.9455

TABLE VI
OBJECTIVE QUALITY VERSUS AVERAGE PROCESSING TIMES (dB/s) RESULTS

Method	LAZA		NEDI		DFDF		KR		ICBI		INEDI		RLLR-online		RLLR-offline	
	PSNR	TIME	PSNR	TIME	PSNR	TIME	PSNR	TIME	PSNR	TIME	PSNR	TIME	PSNR	TIME	PSNR	TIME
Airplane(512x768)	30.17	3.13	28.69	30.30	30.53	27.31	29.11	59.95	30.08	9.62	30.66	74.39	30.90	92.69	30.97	168.31
Lena(512x512)	33.36	2.20	33.57	21.08	33.96	17.98	33.96	40.13	34.05	5.39	34.11	54.15	33.99	70.30	34.41	160.20
Flower(480x640)	25.65	2.53	25.62	25.80	25.74	21.16	25.79	46.75	25.20	6.64	25.89	118.48	26.03	115.44	26.20	269.45
Girl(512x768)	31.84	1.97	31.84	21.64	31.81	17.88	31.92	39.58	31.27	3.87	32.24	44.18	31.96	32.67	32.29	200.39
Door(512x768)	32.28	3.16	32.14	31.36	32.27	27.13	32.20	62.14	31.71	8.25	32.41	71.61	32.59	47.72	32.60	213.09
Peppers(512x512)	31.51	1.94	29.30	21.06	31.87	18.06	31.02	40.06	31.70	4.95	32.05	46.7	32.04	43.94	32.15	173.05
Splash(512x768)	33.54	1.88	31.38	20.25	33.79	18.17	33.38	39.70	33.32	3.95	33.69	28.01	33.93	32.44	33.99	117.16
Baboon(256x256)	20.09	1.33	19.91	6.00	19.87	5.14	19.99	10.91	19.06	2.29	19.55	43.31	20.46	38.03	20.67	68.95
Tower(300x300)	38.45	2.02	39.91	7.77	39.69	7.25	40.28	15.18	38.94	2.81	40.74	12.98	40.41	14.59	41.97	33.66
Butterfly(324x492)	29.34	2.20	28.96	14.55	29.67	13.27	29.54	26.80	29.91	6.21	30.07	42.32	30.25	43.69	30.32	87.64
Average	30.62	2.24	30.13	19.98	30.92	17.34	30.72	38.12	30.52	5.40	31.14	53.61	31.26	53.15	31.56	149.19

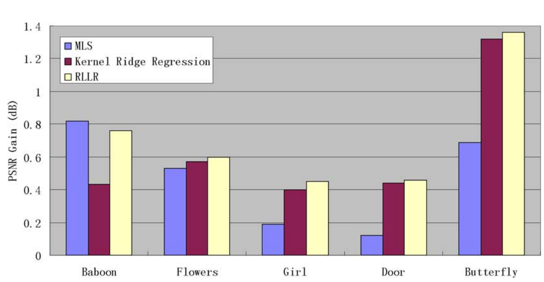


Fig. 8. PSNR gain of three components over OLS.

introduce a robust estimator of local image structure based on moving least squares, which can efficiently handle outliers compared with ordinary least squares-based methods. On the other hand, motivated by recent progress on graph based manifold learning, the intrinsic manifold structure is explicitly considered by making use of both measured and unmeasured data points. In particular, the geometric structure of the marginal probability distribution induced by unmeasured samples is incorporated as an additional local smoothness preserving constraint. The optimal transformation functions can be obtained with a closed-form solution by solving a convex optimization problem, which are smooth and locally linear, and can keep the local image structure wonderfully. Experimental results over a wide range of test images demonstrate that our method achieves very competitive interpolation performance compared the state-of-the-art methods in both objective and subjective visual quality.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their constructive suggestions which helped us improving our manuscript.

REFERENCES

[1] X. Liu, D. Zhao, R. Xiong, S. Ma, and W. Gao, "Image interpolation via regularized local linear regression," in *Proc. IEEE Picture Coding Symp. (PCS2010)*, Nogoya, Japan, Dec. 08–10, 2011, pp. 118–121.

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

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

[4] K. Jensen and D. Anastassiou, "Subpixel edge localization and the interpolation of still images," *IEEE Trans. Image Process.*, vol. 4, no. 3, pp. 285–295, Mar. 1995.

[5] H. Shi and R. Ward, "Canny edge based image expansion," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)1*, 2002, pp. 785–788.

[6] S. Battiato, G. Gallo, and F. Stanco, "A locally-adaptive zooming algorithmfor digital images," *Image Vis. Comput.*, vol. 20, no. 11, pp. 805–812, 2002.

[7] S. Battiato and F. Stanco, "ALZ: Adaptive learning for zooming digital images," in *Proc. IEEE Int. Conf. Consumer Electron.*, Las Vegas, NV, Jan. 10–14, 2007, pp. 1–2.

[8] S. Battiato, E. U. Giuffrida, and F. Rundo, "A cellular neural network for zooming digital colour images," in *Proc. IEEE Int. Conf. Consumer Electron.*, Las Vegas, NV, 2008, pp. 1–2.

[9] L. Zhang and X. Wu, "Image interpolation via directional filtering and data fusion," *IEEE Trans. Image Process.*, vol. 15, no. 8, pp. 2226–2238, Aug. 2006.

[10] A. Giachetti and N. Asuni, "Fast artifacts-free image interpolation," in *British Mach. Vis. Conf.*, 2008.

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

[12] N. Asuni and A. Giachetti, "Accuracy improvements and artifacts removal in edge based image interpolation," in *Proc. 3rd Int. Conf. Comput. Vis. Theory Applicat. (VISAPP08)*, 2008.

[13] 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.

[14] H. Takeda, S. Farsiu, and P. Milanfar, "Kernel regression for image processing and reconstruction," *IEEE Trans. Image Process.*, vol. 16, no. 2, pp. 349–366, Feb. 2007.

[15] H. Wendland, *Scattered Data Approximation*, 1st ed. Cambridge, U.K.: Cambridge Univ. Press, Dec. 2004, vol. 17.

[16] D. Levin, "The approximation power of moving least squares," *Math. Comput.*, vol. 67, no. 224, pp. 1517–1531, 1998.

[17] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from examples," *J. Mach. Learn. Res.*, vol. 7, pp. 2399–2434, 2006.

[18] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proc. IEEE Int. Conf. Comput. Vis.*, 1998, pp. 839–846.

[19] J. van de Weijer and R. van den Boomgaard, "Least squares and robust estimation of local image structure," *Int. J. Comput. Vis. (IJCV)*, vol. 64, no. 2–3, pp. 143–15, Sep. 2005.

[20] A. N. Tikhonov, "Regularization of Incorrectly Posed Problems. Soviet Math.," 1963 Transl.:English, (4).

[21] L. I. Rudin, S. J. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Phys. D*, vol. 60, pp. 259–268, 1992.

[22] A. Buades, B. Coll, and J. M. Morel, "A review of image denoising algorithms, with a new one," *Multiscale Modeling Simulat.*, vol. 4, no. 2, pp. 490–530, 2005.

[23] 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.

[24] L. Zhang, L. Zhang, X. Mou, and D. Zhang, "FSIM: A feature similarity index for image quality assessment," *IEEE Trans. Image Process.*, vol. 20, no. 8, pp. 2378–2386, Aug. 2011.

[25] V. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer, 1995.

[26] W. Chen, C. Shan, and G. Haan, "Optimal regularization parameter estimation for spectral regression discriminant analysis," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 12, pp. 1921–1926, Dec. 2009.

[27] L. Zhang, X. Li, and D. Zhang, "Image denoising and zooming under the LMMSE framework," *IET Image Process.*, http://www4.comp.polyu.edu.hk/~cslzhang/papers.htm, to be published.

[28] Y. Zhang, D. Zhao, X. Ji, R. Wang, and W. Gao, "A spatio-temporal auto regressive model for frame rate upconversion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 9, pp. 1289–1301, Sep. 2009.

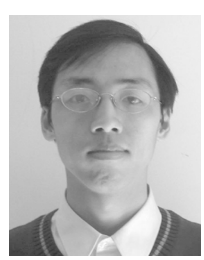
[29] Y. Zhang, D. Zhao, S. Ma, R. Wang, and W. Gao, "Amotion-aligned auto-regressive model for frame rate up conversion," *IEEE Trans. Image Process.*, vol. 19, no. 5, pp. 1248–1258, May 2010.

Xianming Liu received the B.S. and M.S. degrees in computer science from the Harbin Institute of Technology (HIT), Harbin, China, in 2006 and 2008, respectively. He is now working towards the Ph.D. degree at the School of Computer Science and Technology, HIT.

In 2007, he joined the Joint Research and Development Lab (JDL), Chinese Academy of Sciences, Beijing, as a Research Assistant. Since 2009, he has been with the National Engineering Lab for Video Technology, Peking University, Beijing, as a Research Assistant. His research interests include image/video coding, image/video processing, and machine learning.

Debin Zhao received the B.S., M.S., and Ph.D. degrees in computer science from Harbin Institute of Technology (HIT), Harbin, China, in 1985, 1988, and 1998, respectively.

He is now a Professor in the Department of Computer Science, HIT. He has published over 200 technical articles in refereed journals and conference proceedings in the areas of image and video coding, video processing, video streaming and transmission, and pattern recognition.



Ruiqin Xiong (M'08) received the B.S. degree from the University of Science and Technology of China (USTC), Hefei, in 2001 and the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences (ICT, CAS), Beijing, in 2007.

He was a Research Intern at Microsoft Research Asia from August 2002 to July 2007, and a Senior Research Associate at the University of New South Wales, Sydney, Australia, from September 2007 to August 2009. He is now a Researcher at the Institute of Digital Media, Peking University, Beijing, China. His research interests include image and video processing, compression, multimedia communication, channel coding, and distributed coding.

Siwei Ma (M'05) received the B.S. degree from Shandong Normal University, Jinan, China in 1999, and the Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, in 2005.

From 2005 to 2007, he was a post-doctorate with the University of Southern California, Los Angeles. Then he joined the Institute of Digital Media, School of Electrical Engineering and Computer Science, Peking University, where he is currently an Associate Professor. He has published over 70 technical articles in refereed journals and proceedings in the areas of image and video coding, video processing, video streaming, and transmission.

Wen Gao (M'92–SM'05–F'09) received the Ph.D. degree in electronics engineering from the University of Tokyo, Tokyo, Japan, in 1991.

He is a Professor of computer science at Peking University, Beijing, China. Before joining Peking University, he was a Professor of computer science at the Harbin Institute of Technology from 1991 to 1995, and a Professor at the Institute of Computing Technology of Chinese Academy of Sciences. He has published extensively including five books and over 600 technical articles in refereed journals and conference proceedings in the areas of image processing, video coding and communication, pattern recognition, multimedia information retrieval, multimodal interface, and bioinformatics.

Dr. Gao served or serves on the editorial board for several journals, such as the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, IEEE TRANSACTIONS ON MULTIMEDIA, IEEE TRANSACTIONS ON AUTONOMOUS MENTAL DEVELOPMENT, *EURASIP Journal of Image Communications*, and *Journal of Visual Communication and Image Representation*. He chaired a number of prestigious international conferences on multimedia and video signal processing, such as the IEEE ICME and ACM Multimedia, and also served on the advisory and technical committees of numerous professional organizations.

Huifang Sun (M'85–SM'93–F'00) received the B.S. degree in electrical engineering from the Harbin Engineering Institute, Harbin, China, and the Ph.D. degree in electrical engineering from University of Ottawa, Ottawa, ON, Canada.

In 1986, he joined Fairleigh Dickinson University, Teaneck, NJ, as an Assistant Professor and consequently promoted to an Associate Professor. From 1990 to 1995, he was with the David Sarnoff Research Center (Sarnoff Corp.), Princeton, NJ, as a Member of Technical Staff and later promoted to Technology Leader of Digital Video Technology where his activities were MPEG video coding, HDTV and Grand Alliance HDTV development. He joined Mitsubishi Electric Research Laboratories (MERL), in 1995 as a Senior Principal Technical Staff and was promoted as Deputy Director in 1997, Vice President and MERL Fellow in 2003 and now as MERL Fellow. He has coauthored two books and published more than 150 journal and conference papers. He has 60 granted U.S. patents.

Prof. Sun was an Associate Editor of the IEEE TRANSACTION ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY and TC chair of Visual Communications and Image Processing of IEEE Circuits and Systems.