

Single image super-resolution by approximated Heaviside functions

Liang-Jian Deng*

Weihong Guo[†]

Ting-Zhu Huang[‡]

Abstract

Image super-resolution is a process to enhance image resolution from one single or multiple low resolution images. It is widely used in medical imaging, satellite imaging, target recognition, etc. In this paper, we solve the problem of single image super-resolution from an image intensity function estimation perspective. We conduct continuous modeling and assume that the unknown image intensity function is defined on a continuous domain and belongs to a space with a redundant basis. The selection of the redundant basis is based on an observation: an image is consisted of smooth components and non-smooth components, and we use two classes of approximated Heaviside functions (AHFs) to represent them respectively. The coefficients of the redundant basis are computed iteratively from the given low-resolution image. In addition, we apply the proposed iterative model to image patches to reduce computation and storage. Comparisons with some existing competitive methods show the effectiveness of the proposed method.

Key words: Single image super-resolution, Approximated Heaviside functions, Iterative model

1 Introduction

Image super-resolution (SR) is to estimate a high-resolution (HR) image from one or multiple low-resolution (LR) images. When there is only *one* low-resolution input, it is called *single* image super-resolution. The other case is called *multiple* image super-resolution. Compared to multiple image super-resolution, single image super-resolution is more practical when only one low-resolution image is available. Obviously, it is more challenging. In this paper, we address single image super-resolution. Low-resolution images but not high-resolution images are sometimes available due to the limitation of hardware devices and high cost. For instance, it needs more cost and time to get high-resolution images in medical imaging MRI. Due to long distance and air turbulence, we can not get high-resolution images for synthetic aperture radar (SAR) and satellite imaging. In target recognition, we may not always get high-resolution videos to make accurate recognition. Thus, it is important and useful to develop a more effective image super-resolution algorithm.

There are mainly four categories of image super-resolution methods: interpolation-based methods, learning-based methods, statistics-based methods and other methods. In Tab. 1 we briefly list some advantages and disadvantages of methods in each category. Methods in different category are however not completely independent. For instance, some of statistics-based methods may involve learning strategies.

Nearest-neighbor interpolation and bicubic interpolation are two classical interpolation methods. Nearest-neighbor interpolation estimates intensity of an unknown location using that of its nearest neighbor point. It however often creates jaggy effect. Bicubic interpolation is to interpolate unknown intensity by utilizing a cubic kernel, and may cause blur effect. More interpolation methods have been proposed recently [39, 43, 69, 7, 25, 24, 54]. In [24], it introduces a contour stencils method to

*School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu, Sichuan, 611731, P. R. China. (liangjian1987112@126.com).

[†]The Corresponding Author. Department of Mathematics, Case Western Reserve University, Cleveland, OH, 44106, USA. (wxg49@case.edu).

[‡]School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu, Sichuan, 611731, P. R. China. (tingzhuhuang@126.com).

estimate the image contours based on total variation (TV) along curves. This method can distinguish lines of different orientations, curves, corners, etc., with a computationally efficient formula so that the resulted high-resolution images preserve image details well. In [54], Wang et al. propose a fast image super-resolution method on the displacement field. This method is based on two stages, one of which uses an interpolation technique to pick up the low-frequency information and the other employs a reconstruction method to recover local high-frequency structures.

Learning-based methods are an important tool for many image applications, e.g., image retrieval [63], image reranking [62], image classification [61]. Recently, many image super-resolution methods are learning-based [23, 22, 47, 50, 28, 21, 46, 35, 34, 41, 76, 57, 48, 9, 19, 60, 59, 58, 75, 30, 16, 67, 36, 3, 51, 52, 15, 72, 27, 38]. Learning-based methods need two large training data sets, one formed by low-resolution images and the other by the corresponding high-resolution images. A relation between the two training data sets is learned and then applied to a given low-resolution image to obtain a high-resolution image. They heavily rely on the selection of training data and involve expensive computation. They are not single image super-resolution approaches in a strict sense as they require extra training data. In particular, most of the learning-based methods are based on the optimization framework that can be used in many image applications, e.g., image deblurring [73, 74, 12, 14, 40], image decomposition [42], etc. In [59], Yang et al. propose a sparse signal representation method based on the optimization frame for single image super-resolution. [15] by Dong et al. was the first work to apply a deep learning method for single image super-resolution.

Table 1: Four categories for image super-resolution. Note that these categories sometimes are not completely independent. For instance, some of statistics-based methods may involve learning strategies. In particular, since there are no common advantages and disadvantages for “Other methods”, we do not present them in the table.

Interpolation-based methods	e.g., Nearest-neighbor interpolation, bicubic interpolation, contour stencils [25, 24]. More references: [39, 43, 69, 7, 54]	<i>Advantage:</i> fast. <i>Disadvantage:</i> sometimes encounter jaggy or blur effect.
Learning-based methods	e.g., Sparse coding [60, 59], deep learning method [15]. More references: [36, 3, 51, 52], etc.	<i>Advantage:</i> accurate. <i>Disadvantage:</i> extra data for training, slow speed.
Statistics-based methods	e.g., Maximum a Posterior (MAP), Maximum Likelihood estimator (MLE). References: [17, 6, 18].	<i>Advantage:</i> accurate. <i>Disadvantage:</i> encounter some artifacts, e.g., blur effect.
Other methods	e.g., hybrid method [11], frequency technique [4], etc. More references: [31, 32, 37, 53, 5].	<i>Advantage:</i> - <i>Disadvantage:</i> -

Statistics-based methods such as Maximum a Posterior (MAP) and Maximum Likelihood estimator (MLE) are a popular tool for image super-resolution [17, 6, 18, 45]. Fattal [18] utilizes statistical edge dependency relating edge features in low and high resolution images to preserve sharp edges. In [17], an alternative approach using ℓ_1 norm minimization and a bilateral prior based robust regularization, is proposed by Farsin et al.

Many other methods have also been proposed for image super-resolution. Examples are hybrid method [11], pixel classification method [1, 2], frequency technique [4], reconstruction method [31, 32, 37, 10, 49] and others [53, 5, 20, 44, 65, 8, 64, 66, 70, 71, 68].

Most existing image super-resolution methods are based on discrete models. In this paper, we propose a continuous model based on approximated Heaviside functions (AHFs) for single image super-resolution. Since an image is normally consisted of smooth components and non-smooth components such as step edges, we model an image as a sum of two classes of AHFs, one representing smooth components and the other depicting non-smooth components. Due to the sparsity of non-smooth

components, we design a ℓ_1 regularization model and solve it by alternating direction method of multipliers (ADMM). An iterative model based on the ℓ_1 model is proposed to preserve more details. We apply the coefficients, computed by the iterative model, to generate high-resolution images at any upscaling factors. The iterative model is applied to image patches, aiming to get cheap computation and storage. For images with smooth background, the proposed method may cause ring artifacts and we develop a strategy utilizing image gradient to reduce them. Numerical experiments show that the proposed method is a competitive method comparing with existing excellent methods.

To the best of our knowledge, this is the first work to use AHFs on image super-resolution. The proposed method can preserve not only edges, but also the high-frequency details on non-edge regions. The proposed method is completely single image super-resolution method without using training data.

The organization of this paper is as follows. In Section 2, we review Heaviside function and show why we can use it for image super-resolution. In Section 3, we present the proposed method, including the corresponding new model, new algorithm and other strategies. Visual and quantitative experiments are given in Section 4. Finally, we draw conclusions in Section 5.

2 The Heaviside function

The Heaviside function, or the Heaviside step function, is defined as follows (see Fig. 1(a))

$$\phi(x) = \begin{cases} 0, & x < 0, \\ 1, & x \geq 0, \end{cases} \quad (1)$$

which is singular at $x = 0$. In practice, we usually use its approximation, called approximated Heaviside function (AHF), such as $\frac{1}{1+e^{-2x/\xi}}$ and $\frac{1}{2} + \frac{1}{\pi} \arctan(\frac{x}{\xi})$ which approximate to $\phi(x)$ when $\xi \rightarrow 0$. In this paper, we use the following approximated Heaviside function,

$$\psi(x) = \frac{1}{2} + \frac{1}{\pi} \arctan\left(\frac{x}{\xi}\right), \quad (2)$$

where $\xi \in \mathcal{R}$ actually controls the smoothness. The smaller ξ , the sharper edge (see Fig. 1(b)).

From [33], we know that any function in $\mathcal{L}_p([0, 1]^d)$, $p \in [1, \infty)$, has a best approximation by linear combinations of m characteristic functions of half-space, and m is any positive integer. Let H_d be a set of functions on $[0, 1]^d$ defined as

$$H_d = \{f : [0, 1]^d \rightarrow \mathcal{R} : f(\mathbf{x}) = \psi(\mathbf{v} \cdot \mathbf{x} + c), \mathbf{v} \in \mathcal{R}^d, c \in \mathcal{R}\}, \quad (3)$$

where ψ is an approximated Heaviside function. H_d is the set of characteristic function of closed half-spaces of R^d .

Theorem 1 (see [33]) *For any positive integer d , define $\text{span}_m H_d$ as $\{\sum_{i=1}^m \omega_i \psi(\mathbf{v}_i \cdot \mathbf{x} + c_i)\}$, where $\omega_i \in \mathcal{R}$ and $\mathbf{v}_i \in \mathcal{R}^d$ and $c_i \in \mathcal{R}$, then it is known that $U_{m \in \mathcal{N}^+} \text{span}_m H_d$ is dense in $(\mathcal{L}_p([0, 1]^d), \|\cdot\|_p)$, $p \in [1, \infty)$.*

Theorem 2 (see [33]) *For every positive integers m, d and every $p \in [1, \infty)$, $\text{span}_m H_d$ is approximately a compact subset of $(\mathcal{L}_p([0, 1]^d), \|\cdot\|_p)$.*

In practical computing, one only can afford to use $\text{span}_m H_d$ for a finite m .

In our work, we focus on two-dimensional image super-resolution, so $d = 2$. Here, we assume the underlying image intensity function f is defined on $[0, 1]^2$ and $f \in \mathcal{L}_p([0, 1]^2)$ with $p \in [1, \infty)$, by Theorem 1 and 2, f can be approximated by the following equation,

$$f(\mathbf{z}) = \sum_{j=1}^m \omega_j \psi(\mathbf{v}_j \cdot \mathbf{z} + c_j), \quad (4)$$

where $\omega_j \in \mathcal{R}$, $\mathbf{v}_j \in \mathcal{R}^2$ and $c_j \in \mathcal{R}$. We discretize $\mathbf{v}_j = \{(\cos\theta_t, \sin\theta_t)', t = 1, 2, \dots, k\}$ to denote k different directions, and $c_j = \{\frac{1}{q}, \frac{2}{q}, \frac{3}{q}, \dots, 1\}$ to denote discrete positions, $m = kq$, $\mathbf{z} = (x, y)'$. We fix q as the total number of pixels of the input image. Furthermore, $\{\psi(\mathbf{v}_j \cdot \mathbf{z} + c_j)\}_{j=1}^m$ with a specific ξ is called a class of AHFs. In particular, we can describe edges of different orientations θ_t at the

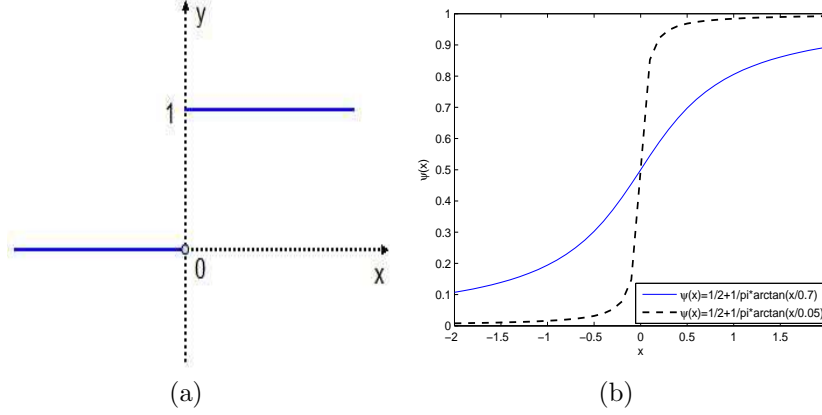


Figure 1: (a) Heaviside function; (b) Two approximated Heaviside functions with $\xi = 0.7$ (blue solid line) and $\xi = 0.05$ (black dash line), the smaller ξ the sharper edge (color images are better visualized in the pdf file).

locations c_j when setting a small ξ . For an image $L \in \mathcal{R}^{n_1 \times n_2}$, we assume it is a discretization of intensity function f on $[0, 1]^2$, i.e., $L_{i,j} = f(x_i, y_j)$, $x_i = \frac{i}{n_1}$, $y_j = \frac{j}{n_2}$, $i = 1, 2, \dots, n_1$, $j = 1, 2, \dots, n_2$. We can rewrite Eq. (4) as matrix-vector form, $L \approx \Psi\omega$, where $L \in \mathcal{R}^n$, $\Psi \in \mathcal{R}^{n \times m}$, $\omega \in \mathcal{R}^m$ with $n = n_1 n_2$, $m = kq$. Once we have computed coefficient ω , we can make image super-resolution with upscaling factor s possible by the equation $\tilde{\Psi}\omega$, where $\tilde{\Psi} \in \mathcal{R}^{N \times m}$ is obtained similarly with Ψ , $N = s^2 n_1 n_2$. Note that the approximation accuracy of Eq. (4) depends on the value of ξ . In the next section, we will address how to choose it.

3 The proposed model and its solution

3.1 The proposed ℓ_1 model and its solution

From Fig. 2 and Fig. 3, it is easy to know that the corresponding AHFs with bigger ξ represent smooth components well, and the corresponding AHFs with smaller ξ depict non-smooth components such as edges well. An image is generally consisted of smooth components and non-smooth components, thus we represent an image with two classes of AHFs, one with a bigger parameter ξ_1 to represent smooth components (forming Ψ_1) and the other with a smaller ξ_2 to depict non-smooth components (forming Ψ_2). The two classes of AHFs form a redundant representation for the underlying intensity function. By this strategy, the vector-form image L can be approximated by the following discrete formula,

$$L \approx \Psi_1 \beta_1 + \Psi_2 \beta_2, \quad (5)$$

where $\Psi_1, \Psi_2 \in \mathcal{R}^{n \times m}$, and $\beta_1, \beta_2 \in \mathcal{R}^{m \times 1}$ are representation coefficients. Once the coefficients β_1, β_2 are computed, one can apply the computed coefficients to get high-resolution images at finer grids by $\hat{f} = \tilde{\Psi}_1 \beta_1 + \tilde{\Psi}_2 \beta_2$, where $\tilde{\Psi}_1, \tilde{\Psi}_2 \in \mathcal{R}^{N \times m}$ are generated similarly as $\tilde{\Psi}$ using finer grids. Note that, in a recent work of ourself [13], we have used Heaviside function to describe non-smooth components only but in this work both smooth and non-smooth components are described using approximated Heaviside functions.

For an image, non-smooth components such as edges are more sparsely distributed than smooth components, we thus enforce ℓ_1 sparsity on β_2 while ℓ_2 regularity on β_1 . The optimization model is

$$\min_{\beta_1, \beta_2} \|L - \Psi_1 \beta_1 - \Psi_2 \beta_2\|_2^2 + \lambda_1 \|\beta_1\|_2^2 + \lambda_2 \|\beta_2\|_1, \quad (6)$$

where λ_1, λ_2 are regularization parameters, L represents the low-resolution input.

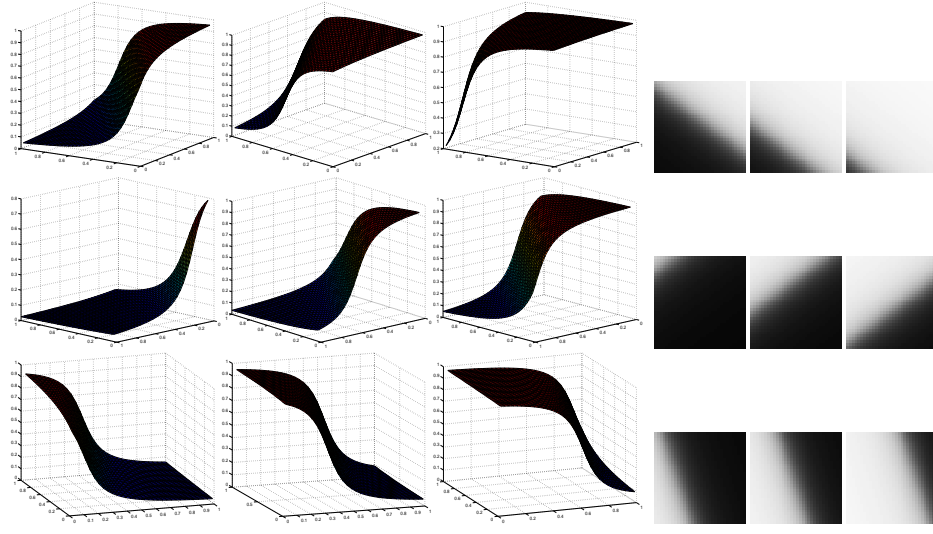


Figure 2: Left panel: surface visualization of ψ under $\xi = 0.1$ and nine random parameter pairs (θ, c) ; Right panel: the corresponding 2D intensity images. From left to right and then from top to bottom: $(\frac{4\pi}{5}, \frac{51}{1024})$, $(\frac{4\pi}{5}, \frac{25}{64})$, $(\frac{4\pi}{5}, \frac{175}{256})$, $(\frac{6\pi}{5}, \frac{135}{1024})$, $(\frac{6\pi}{5}, \frac{1}{2})$, $(\frac{6\pi}{5}, \frac{25}{32})$, $(\frac{8\pi}{5}, \frac{5}{64})$, $(\frac{8\pi}{5}, \frac{75}{256})$, $(\frac{8\pi}{5}, \frac{75}{128})$ (for better visualization, we make slightly rotation to some surfaces).

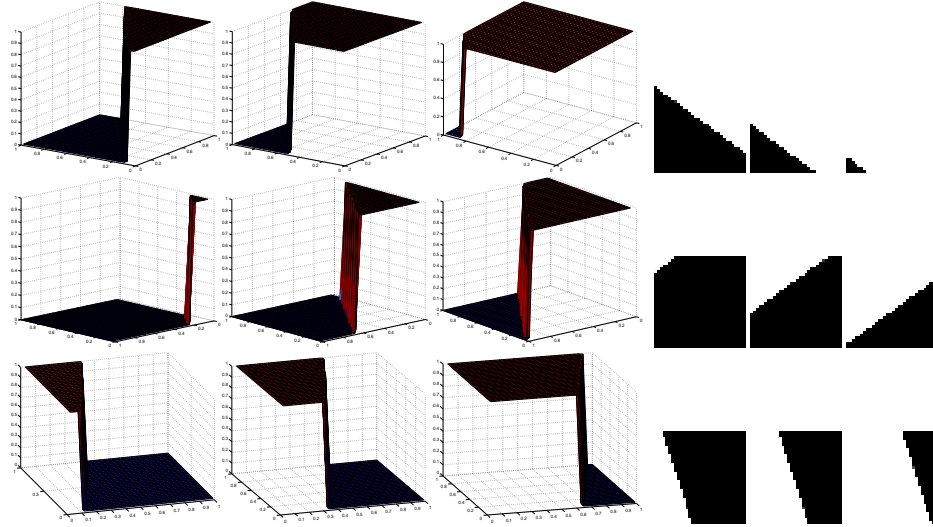


Figure 3: Left panel: surface visualization of ψ under $\xi = 10^{-4}$ and the same nine parameter pairs (θ, c) as in Fig. 2; Right panel: the corresponding 2D intensity images. (for better visualization, we make slightly rotation to some surfaces).

Since ℓ_1 term is not differentiable, we make a variable substitution for β_2 , and rewrite model (6) as

$$\min_{\beta_1, \beta_2} \|L - \Psi_1 \beta_1 - \Psi_2 \beta_2\|_2^2 + \lambda_1 \|\beta_1\|_2^2 + \lambda_2 \|u\|_1, \quad \text{s.t.}, \quad u = \beta_2, \quad (7)$$

where u is the substitution variable. Eq. (7) can be rewritten as

$$\min_{\beta} \|L - \Psi \beta\|_2^2 + \lambda_1 \|A\beta\|_2^2 + \lambda_2 \|u\|_1, \quad \text{s.t.}, \quad u = B\beta, \quad (8)$$

where $\Psi = (\Psi_1, \Psi_2)$, $\beta = (\beta_1, \beta_2)'$, $A = (I, \mathbf{0})$ and $B = (\mathbf{0}, I)$. The optimization problem (8) is separable w.r.t β and u . There are many methods to solve (8). We select alternating direction method of multipliers (ADMM)[26, 55, 29].

The augmented Lagrangian of Eq. (8) is

$$\mathcal{L}(\beta, u, b) = \|L - \Psi \beta\|_2^2 + \lambda_1 \|A\beta\|_2^2 + \lambda_2 \|u\|_1 + \frac{\rho}{2} \|u - B\beta + b\|_2^2, \quad (9)$$

where b is Lagrangian multiplier with proper size. The problem of minimizing $\mathcal{L}(\beta, u, b)$ is solved by iteratively and alternatively solving the following two subproblems:

$$\beta\text{-subproblem} : \min_{\beta} \|L - \Psi \beta\|_2^2 + \lambda_1 \|A\beta\|_2^2 + \frac{\rho}{2} \|u - B\beta + b\|_2^2, \quad (10)$$

$$u\text{-subproblem} : \min_u \lambda_2 \|u\|_1 + \frac{\rho}{2} \|u - B\beta + b\|_2^2. \quad (11)$$

The following ADMM algorithm is used to solve Eq. (8).

Algorithm 1

Input: Given low-resolution image L , Ψ_1 , Ψ_2 , λ_1 , λ_2 , ρ

Output: β

1. $k \leftarrow 0$, $\beta^{(k)} \leftarrow \mathbf{0}$, $u^{(k)} \leftarrow \mathbf{0}$, $b^{(k)} \leftarrow \mathbf{0}$
 2. *while not converged do*
 3. $k \leftarrow k + 1$
 4. $\beta^{(k)} \leftarrow$ solve subproblem (10) for $u = u^{(k-1)}$, $b = b^{(k-1)}$,
 5. $u^{(k)} \leftarrow$ solve subproblem (11) for $\beta = \beta^{(k)}$, $b = b^{(k-1)}$,
 6. $b^{(k)} \leftarrow b^{(k-1)} + (u^{(k)} - B\beta^{(k)})$.
 7. *End while.*
-

The β -subproblem (10) can be solved by least squares method:

$$\beta = K^{-1}r, \quad (12)$$

where $\beta \in \mathcal{R}^{2m \times 1}$, $K = (\Psi^T \Psi + \lambda_1 A^T A + \frac{\rho}{2} B^T B) \in \mathcal{R}^{2m \times 2m}$, $r = \frac{\rho}{2}(u + b) + \Psi^T L \in \mathcal{R}^{2m \times 1}$.

The u -subproblem (11) has a closed form solution for each u_i (see [55])

$$u_i = \text{shrink}((B\beta)_i - b_i, \frac{\lambda_2}{\rho}), \quad (13)$$

where $\text{shrink}(a, b) = \text{sign}(a) \max(|a - b|, 0)$ and $0.(0/0) = 0$ is assumed.

3.2 The iterative AHF method based on the proposed model

Even though model (6) takes different behaviors of smooth components and non-smooth components into consideration, we still see some blur in the high-resolution image. We find that the difference image $L - DH^{(1)}$, where $H^{(1)}$ is the resulted high-resolution output by model (6) and D is a down-sampling operator, contains some residual edges. To pick up more edges and details and thus to make results less blurry, we consider the difference $L - DH^{(1)}$ as a new low-resolution input L of model (6) to recompute a residual high-resolution image $H^{(2)}$. This process is repeated until the residual edges are small enough. The sum of $H^{(1)}$ and its residual high-resolution images is the resulted

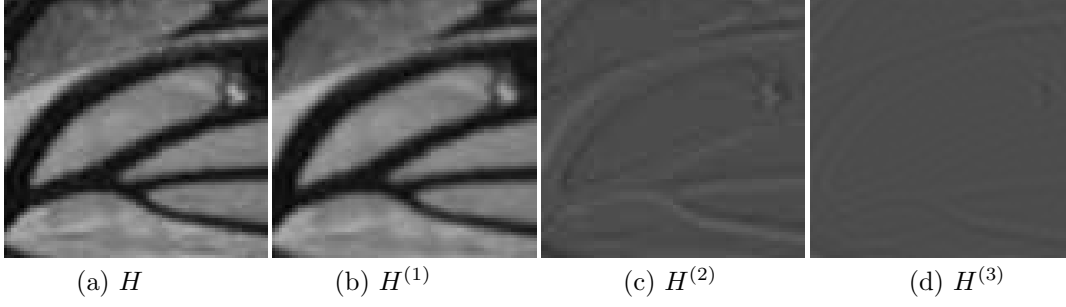


Figure 4: Illustration of step 2 of Algorithm 2 using “wing” image. (a) The sum image of $H^{(i)}, i = 1, 2, 3$ (we set 3 iterations here); (b) the computed image for the first iteration. For better visualization, we add 0.3 to the intensities of $H^{(2)}$ and $H^{(3)}$ to brighten them. From the last two images, we can see that $H^{(2)}$ and $H^{(3)}$ pick up image details.

super-resolution image. This iterative strategy can recover more image details (see Fig. 4). The following Algorithm 2 is the proposed iterative AHF algorithm for single image super-resolution. We use bicubic downsampling D in our experiments although Algorithm 2 can work for general downsampling.

Algorithm 2 (Single image super-resolution via iterative AHF method)

Input: one low-resolution image (vector-form): $L \in \mathcal{R}^{n \times 1}$, $\lambda_1 > 0$, $\lambda_2 > 0$, s : upscaling factor.
 τ : maximum number of iterations.

Output: high-resolution image (vector-form): $\hat{H} \in \mathcal{R}^{N \times 1}$

1. Construct matrices $\Psi_1, \Psi_2 \in \mathcal{R}^{n \times m}$ on coarse grids and $\tilde{\Psi}_1, \tilde{\Psi}_2 \in \mathcal{R}^{N \times m}$ on fine grids (see Section 3.1), where $N = s^2 n$.
2. Initialization: $L^{(1)} = L$.
for $k = 1$: τ
 - a. Compute the coefficients: $(\beta_1^{(k)}, \beta_2^{(k)}) = \text{argmin} \|L^{(k)} - \Psi_1 \beta_1 - \Psi_2 \beta_2\|_2^2 + \lambda_1 \|\beta_1\|_2^2 + \lambda_2 \|\beta_2\|_1$.
 - b. Update the high-resolution image: $H^{(k)} = S^{(k)} + E^{(k)}$ where $S^{(k)} = \tilde{\Psi}_1 \beta_1^{(k)}, E^{(k)} = \tilde{\Psi}_2 \beta_2^{(k)}$.
 - c. Downsampling $H^{(k)}$ to the coarse grid: $\tilde{L} = DH^{(k)}$.
 - d. Compute residual: $L^{(k+1)} = L^{(k)} - \tilde{L}$.**end**
3. Assemble the high-resolution outputs: $S = \sum_{i=1}^{\tau} S^{(i)}, E = \sum_{i=1}^{\tau} E^{(i)}$.
4. Compute the final high-resolution image:

$$\hat{H} = S + \mathbf{c}(E, p),$$

where \mathbf{c} represents a convolution operator, and p is a Gaussian kernel with small size.

Note that step 2.a in Algorithm 2 is solved by Algorithm 1. As shown in Fig. 5, E represents some edges. In particular, we use a Gaussian kernel p with size 5×5 and standard derivation 1 to make a convolution to reduce the oversharpening information of the non-smooth component E on non-edge regions. In addition, although we introduce some parameters in the algorithm, they are not sensitive and easy to select. We will give a remark on parameter selection in Section 4.

3.3 Apply the proposed method to image patches

We have proposed the iterative AHF method for single image super-resolution, but it can be very expensive if we apply it to a whole image. For instance, if one low-resolution image has size 100×100 and the upscaling factor is 2, then Ψ_1, Ψ_2 with 10 angles should have size $10,000 \times 100,000$. It is very expensive to implement Algorithm 2 with large-scale non-sparse matrices. In our work, we apply the iterative AHF method to image patches, so that we can reduce computation and storage significantly. We set patch size to be 6×6 and overlap to be 2 pixels in our work. We take the mean as the intensity

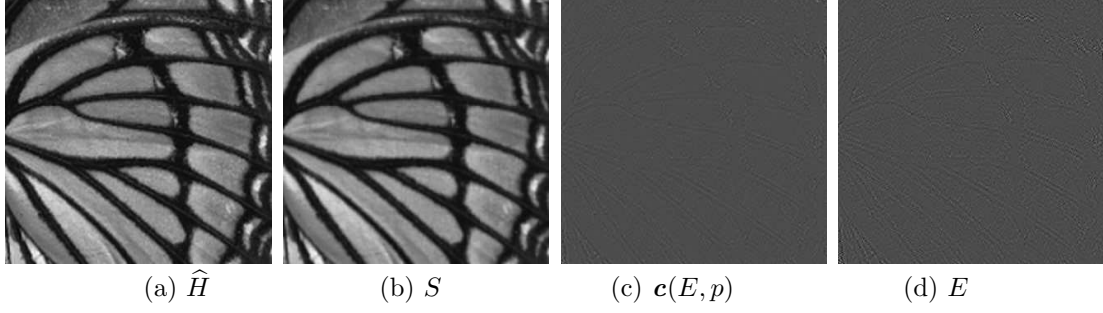


Figure 5: Illustration for the step 4 of Algorithm 2. (a) The sum image $S + c(E, p)$; (b) the image S ; (c) the image $c(E, p)$ which contains some edges and high-frequency information away from edges; (d) the image E without using the convolution operator. Note that the image E contains more oversharp information on non-edge regions than the image $c(E, p)$. For better visualization, we add 0.3 to the intensities of $c(E, p)$ and E to brighten them.

of pixels at the overlapped region. Moreover, the intensity of pixels on the boundary will be handled by bicubic interpolation.

In summary, we use a set of Heaviside functions with sharp edges to represent sharp image edges which can be viewed as a kind of image high-frequency information. In particular, applying the proposed method to each image patches may make image high-frequency information more significant. This is the reason why the proposed method can pick up high-frequency information.

4 Numerical experiments

In this section, two kinds of test images are utilized to illustrate the performance of different methods. One is low-resolution images without high-resolution ground-truth. The other is low-resolution images downsampled from known high-resolution images, for which we can make quantitative comparisons. All experiments are implemented in MATLAB(R2010a) on a laptop of 3.25Gb RAM and Intel(R) Core(TM) i3-2370M CPU: @2.40 GHz.

We mainly compare the proposed method with some competitive image super-resolution methods: nearest-neighbor interpolation, bicubic interpolation, a kernel regression method (denoted as “07’TIP” [49]), a fast upsampling method (denoted as “08’TOG” [44]), TV-based super-resolution method (denoted as “11’JMIV” [8]), two state-of-the-art learning-based methods (“10’TIP” [59] and “12’BMVC” [3]) and three state-of-the-art interpolation-based methods (denoted as “11’IPOL” [25], “11’SIAM” [24] and “14’TIP” [54]).

For gray images, we can apply the proposed Algorithm 2 directly. For color images such as RGB, we first transform from RGB color space to “Ycbcr” color space which is very popular in image/video processing. “Y” represents luma component, “Cb” and “Cr” are blue-difference and red-difference components, respectively. We only apply the proposed algorithm to the illuminance channel because humans are more sensitive to luminance changes. In addition, we interpolate the color layers (Cb, Cr) using bicubic interpolation. After getting the upsampled images in “Ycbcr” color space, we transform them back to the original RGB color space for visualization.

In the experiments, we use three measures, root-mean-square error (RMSE), peak signal-noise ratio (PSNR) and the structural similarity (SSIM) index¹ [56], to evaluate quantitative performance.

Remark on parameter selection: For the parameters in Algorithm 1 and Algorithm 2, we set $\rho = 10^{-4}$, $\lambda_1 = 10^{-2}$, $\lambda_2 = 10^{-6}$, $\tau = 3$. Matrices $\Psi_1, \tilde{\Psi}_1$ with $\xi = 10^{-1}$ and $\Psi_2, \tilde{\Psi}_2$ with $\xi = 10^{-4}$. We set 12 angles evenly distributed on $[0, 2\pi]$, i.e., $\theta = \{0, \frac{\pi}{6}, \frac{\pi}{3}, \frac{\pi}{2}, \frac{2\pi}{3}, \frac{5\pi}{6}, \pi, \frac{7\pi}{6}, \frac{4\pi}{3}, \frac{3\pi}{2}, \frac{5\pi}{3}, \frac{11\pi}{6}\}$. We fix q as the total number of pixels of the low-resolution patch, i.e., $q = 36$ for 6×6 patches. Note that,

¹<https://ece.uwaterloo.ca/~z70wang/research/ssim/>

fine tuning of parameters for some images may get better results. However, we unify the parameter selection to illustrate the stability of the proposed method.

4.1 Super-resolution for generic images

First, we test on low-resolution images without known high-resolution images (see Fig. 6). We compare the proposed method with bicubic interpolation, two state-of-the-art learning-based methods (“10’TIP” [59] and “12’BMVC” [3]), a fast upsampling method (“08’TOG” [44]) and three state-of-the-art interpolation-based methods (“11’IPOL” [25], “11’SIA” [24] and “14’TIP” [54]). The upscaling factors are all 3. From the resulted images, super-resolution images by bicubic interpolation generate blur effect. “08’TOG” performs well on image edges, but it smooths the intensities on non-edge regions. The two learning-based methods (“10’TIP” and “12’BMVC”) perform comparable visually. In addition, “11’IPOL” and “11’SIA” also show similar performance. In particular, “14’TIP” appears to be better on image edges, but it smooths out image details on non-edge regions. The proposed method outperforms all the others, especially for image details on non-edge regions.

In Fig. 7 and Fig. 8, low-resolution test images are generated by downsampling known high-resolution via bicubic interpolation. Since ground-truth images are known, we can compute quantitative measures such as RMSE, PSNR and SSIM. The upscaling factors are set to be 4 for “baboon” and “forest”, 3 for “field” and 2 for “leopard”. From these figures, one can see that the results of “08’TOG” generate sharp edges but smooth the intensities on non-edge regions. TV-based method “11’JMIV” [8] also obtains sharp edges but has oil-painting effect (e.g., see the close-up of “baboon” in Fig. 7). Nearest-neighbor interpolation in Fig. 8 generates significant jaggy effect. Results of bicubic interpolation and the kernel regression method “07’TIP” [49] show significant blur. Because the proposed method can be viewed as an interpolation method, we compare our method with three state-of-the-art interpolation-based methods (“11’IPOL” [25], “11’SIA” [24] and “14’TIP” [54]). In addition, we also compare the proposed method with two modern learning-based methods (“10’TIP” [59] and “12’BMVC” [3]). As shown in Fig. 7 and Fig. 8, the proposed method obtains the best quantitative results for almost all test examples (see captions of the two figures), and generates high-resolution images with more image details. Actually, learning-based methods need extra training data that can be viewed as extra prior information while our method does not require any extra data. Note that the interpolation-based method “14’TIP” [54] keeps very sharp image edges and runs quite fast, but it does not obtain good quantitative results and smooths image details significantly on non-edge regions (see close-ups for better visualization). Furthermore, more quantitative results can be found in Tab. 2 that also demonstrates the effectiveness of the proposed method.

4.2 Super-resolution for images with smooth background

The proposed method performs well for natural images. It is a better method than the compared methods. However, the proposed method encounters a drawback: ring artifacts along large scale edges of images with smooth background (see Fig. 9(b)). The images with smooth background are generally obtained from single lens reflex (SLR) camera or other specific scenes. The ring artifacts mainly come from the added non-smooth components (see step 4 of Algorithm 2). In this section, we utilize a method without ring artifacts to make a mask, aiming to discard the ring artifacts of non-smooth components E . We learn that bicubic interpolation will not cause ring artifacts (see the resulted images in section 4.1) and run very fast. Thus we use bicubic interpolation as our intermediate method to make the mask. Actually, we only need to update the E in the step 4 of Algorithm 2 by E_{new} which is obtained from the following equations, then we can reduce the ring artifacts significantly.

$$E_{new} = M * E, \quad (14)$$

where M is the mask matrix defined as follows,

$$M = \begin{cases} 0, & \text{if } 0 \leq G_{i,j} \leq t, \\ 1, & \text{otherwise,} \end{cases} \quad (15)$$

where $G_{i,j}$ is the vector norm of gradient at the location (i,j) of image B ; and the image B is a high-resolution image generated by bicubic interpolation with an upscaling factor s ; notation “ $*$ ” here



Figure 6: Results of “butterflywing” and “landscape”, upscaling factors are all 3. From left to right and top to bottom: low-resolution images, results of bicubic interpolation, “10’TIP” [59], “08’TOG” [44], “11’IPOL” [25], “11’SIAM” [24], “12’BMVC” [3], “14’TIP” [54] and the proposed method (color images are better visualized in the pdf file).

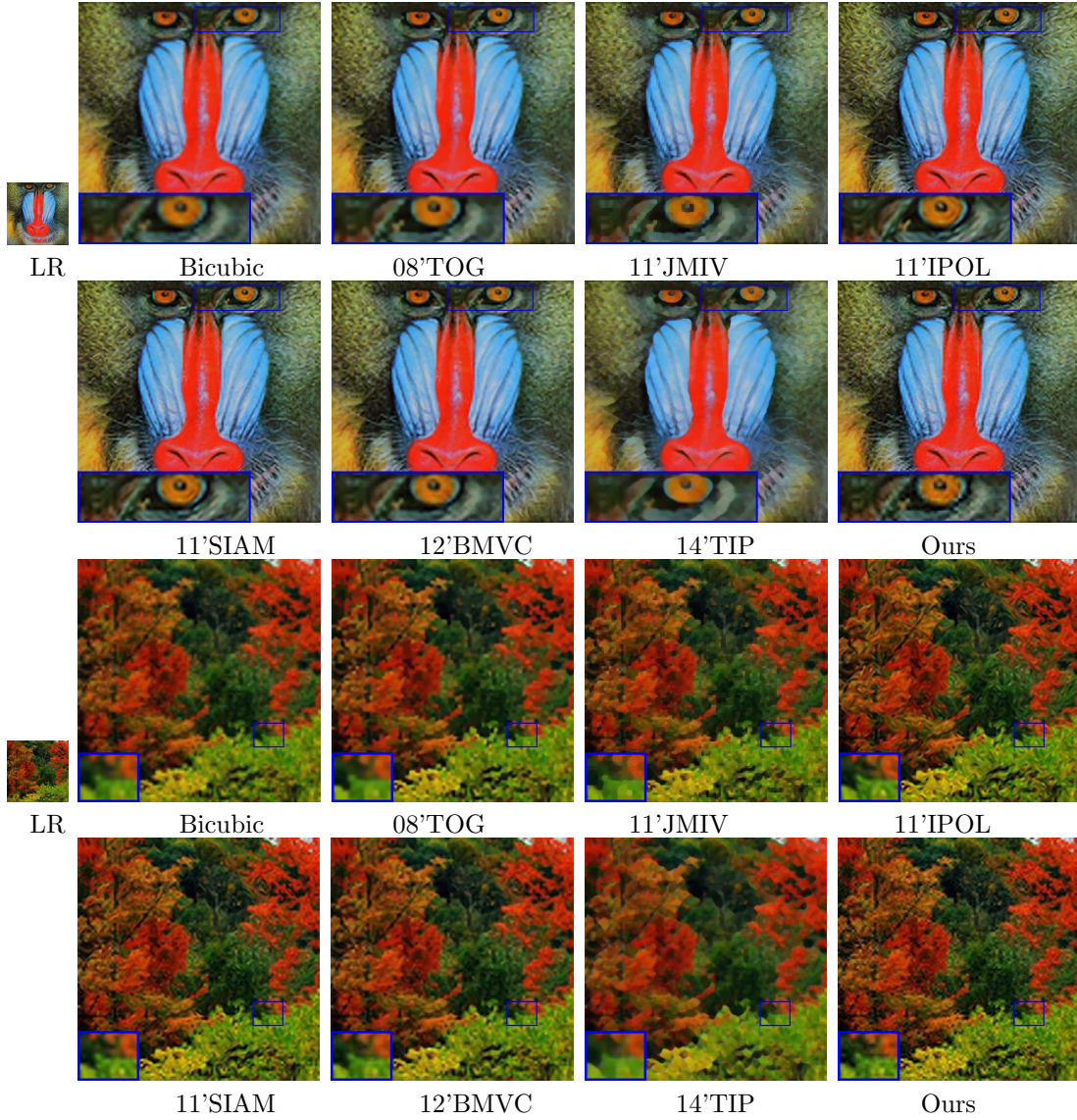


Figure 7: Visual results and the corresponding (RMSE, PSNR, SSIM) of “baboon” and “forest” by the upscaling factor of 4. *First example:* bicubic interpolation (19.69, 22.25, 0.704), “08”TOG” (19.54, 22.31, 0.714), “11”JMIV” (19.81, 22.19, 0.720), “11”IPOL” (19.32, 22.41, 0.760), “11”SIAM” (19.28, 22.43, **0.765**), “12”BMVC” (19.28, 22.43, 0.749), “14”TIP” (21.51, 21.48, 0.591) and the proposed method (**19.20**, **22.45**, 0.760). *Second example:* bicubic interpolation (18.41, 22.83, 0.687), “08”TOG” (18.10, 22.98, 0.701), “11”JMIV” (18.56, 22.76, 0.700), “11”IPOL” (18.23, 22.92, 0.739), “11”SIAM” (18.02, 23.01, 0.750), “12”BMVC” (17.99, 23.03, 0.735), “14”TIP” (20.55, 21.88, 0.553) and the proposed method (**17.81**, **23.09**, **0.758**) (Bold: the best one; Underline: the second best).

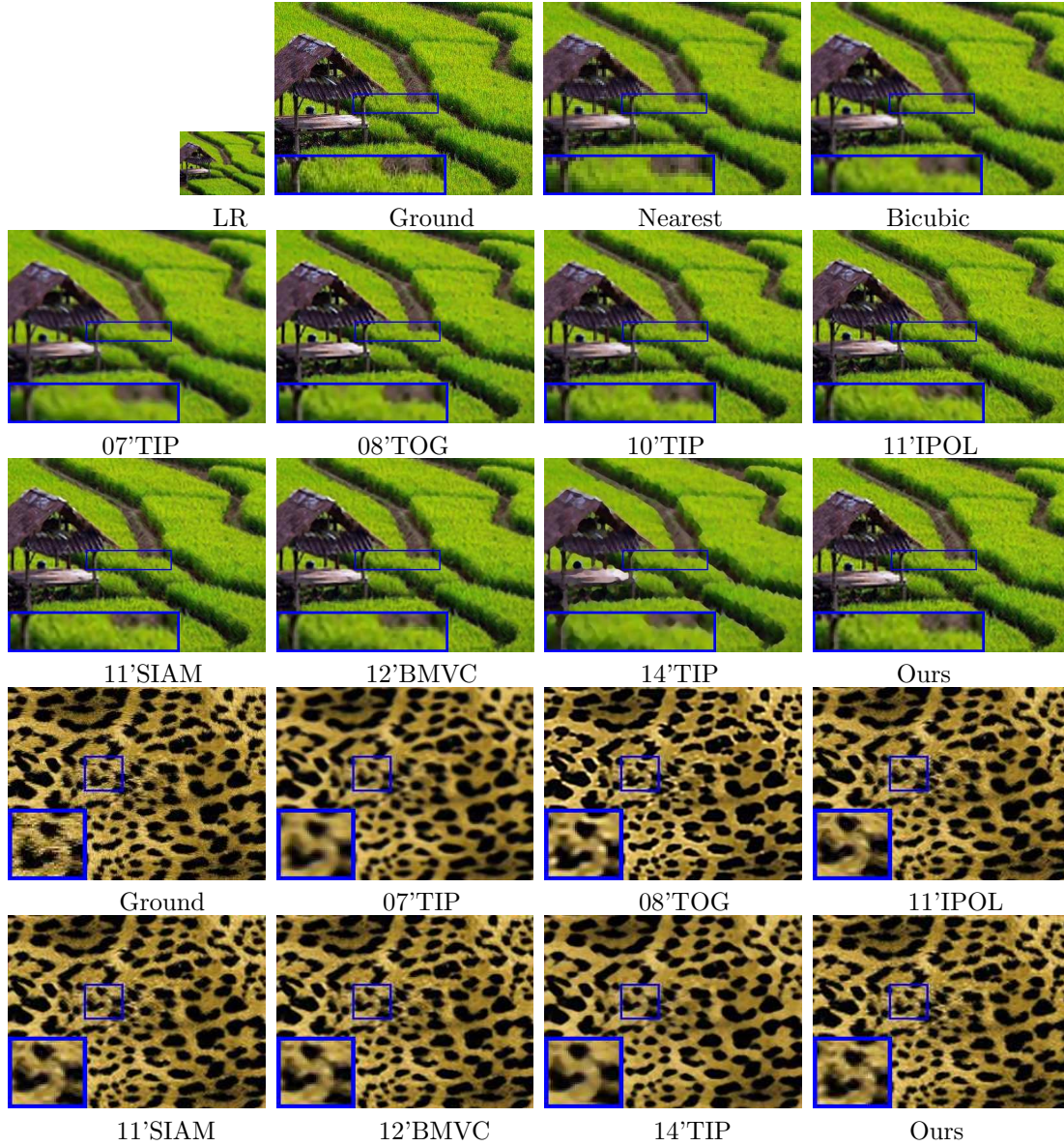


Figure 8: Visual results of “field” (upscaling factor 3) and “leopard” (upscaling factor 2), and the corresponding (RMSE, PSNR, SSIM). *First example:* nearest-neighbor interpolation (13.01, 25.52, 0.579), bicubic interpolation (12.56, 26.20, 0.599), “07’TIP” (14.70, 25.12, 0.551), “08’TOG” (12.37, 26.32, 0.595), “10’TIP” (12.33, 26.36, 0.615), “11’IPOL” (12.28, 26.39, 0.638), “11’SIAM” (12.16, 26.50, **0.646**), “12’BMVC” (12.17, 26.48, 0.631), “14’TIP” (13.86, 25.37, 0.559) and the proposed method (**12.08**, **26.53**, 0.642). *Second example:* “07’TIP” (20.25, 22.51, 0.772), “08’TOG” (16.44, 23.84, 0.832), “11’IPOL” (14.65, 24.81, 0.850), “11’SIAM” (14.40, 25.00, 0.855), “12’BMVC” (14.28, 25.01, 0.849), “14’TIP” (18.46, 22.84, 0.779) and the proposed method (**14.23**, **25.11**, **0.858**).

Table 2: RMSE, PSNR and SSIM for more test examples (Bold: the best one; Underline: the second best).

Image(factor)	Index	Bicubic	07'TIP [49]	08'TOG [44]	10'TIP [59]	Ours
lena(X2)	RMSE	5.10	9.91	6.98	4.09	3.84
	PSNR	33.52	28.11	31.07	<u>36.02</u>	37.51
	SSIM	0.961	0.920	0.941	<u>0.982</u>	0.995
landscape(X2)	RMSE	10.75	13.55	11.74	<u>10.09</u>	9.63
	PSNR	27.57	25.47	27.25	<u>27.84</u>	28.55
	SSIM	0.905	0.690	0.893	<u>0.916</u>	0.929
purplebutterfly(X2)	RMSE	13.03	16.85	14.19	<u>11.69</u>	11.43
	PSNR	25.14	23.71	25.30	<u>27.26</u>	27.35
	SSIM	0.802	0.759	0.781	<u>0.865</u>	0.871
leaf(X3)	RMSE	17.51	18.23	17.55	<u>17.39</u>	17.28
	PSNR	22.12	21.11	22.04	<u>22.85</u>	23.01
	SSIM	0.726	0.715	0.720	<u>0.733</u>	0.741
dog(X3)	RMSE	2.12	3.76	1.95	1.93	1.85
	PSNR	41.88	38.13	42.51	<u>42.57</u>	43.00
	SSIM	0.997	0.996	<u>0.998</u>	<u>0.998</u>	0.999
fruits(X3)	RMSE	4.13	9.09	<u>3.55</u>	3.75	3.53
	PSNR	36.11	29.55	<u>38.61</u>	38.00	38.66
	SSIM	0.992	0.948	0.996	<u>0.995</u>	0.996
babyface(X3)	RMSE	5.12	8.44	<u>4.86</u>	4.87	4.74
	PSNR	33.60	29.43	<u>33.91</u>	<u>33.91</u>	33.98
	SSIM	0.957	0.925	<u>0.962</u>	<u>0.962</u>	0.964
train(X4)	RMSE	12.98	15.85	<u>12.64</u>	12.85	12.62
	PSNR	25.29	23.28	<u>26.07</u>	25.55	26.14
	SSIM	0.811	0.763	<u>0.829</u>	0.818	0.831

stands for dot product; t is a thresholding value, and we set $t = 0.05$ in our work. Obviously, this strategy reduces the ring artifacts of super-resolution images (see Fig. 9(c)).

In Fig. 10, we apply the proposed method to images “comic” and “face” which have smooth background. The upscaling factors are set to be 2 and 4, respectively. We can find more details from the close-ups in Fig. 10. The proposed method reduces ring artifacts significantly and performs competitive, both visually and quantitatively.

In Fig. 11, we compare the proposed method with some competitive methods. The upscaling factor is 2. The test image “butterfly” has obvious smooth background. The results of bicubic interpolation show significant blur effect. Although the results by “08'TOG” and “14'TIP” keep sharp edges well, they make image details of non-edge regions oversmoothing. The two learning-based methods “10'TIP” and “12'BMVC” obtain competitive visual and quantitative results (see the figure caption for quantitative results). In addition, the two interpolation methods “11'IPOL” and “11'SIAM” generate excellent visual results and run very fast. In particular, the proposed method preserves image details of non-edge regions well and overcomes ring artifacts significantly. Furthermore, our method also gets the competitive quantitative results. Please note that the quantitative results by the proposed method in Fig. 10 and Fig. 11 are not the best, since the updating strategy in Eq. (14) can reduce ring artifacts significantly, it however might discard some crucial information in E , e.g., image details away from edges. This will lead to the not so good quantitative results.

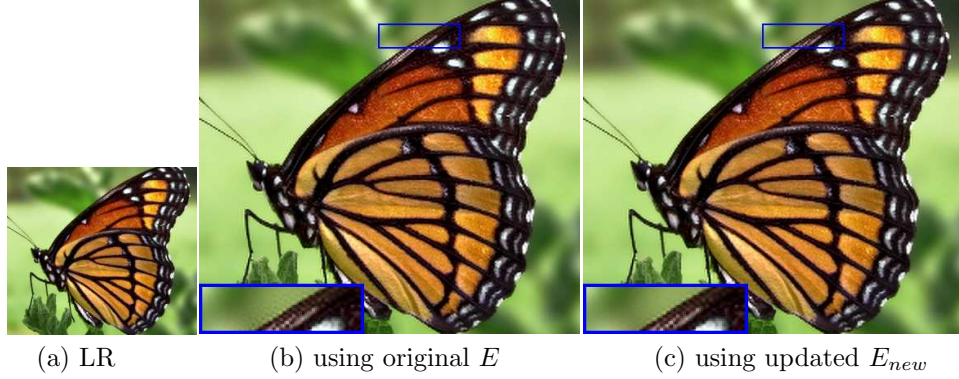


Figure 9: Results of “butterfly” by an upscaling factor of 2. Image “butterfly” is an image with the smooth background; (a) low-resolution image; (b) super-resolution image by the proposed method using original E ; (c) super-resolution image by the proposed method using updated E_{new} .

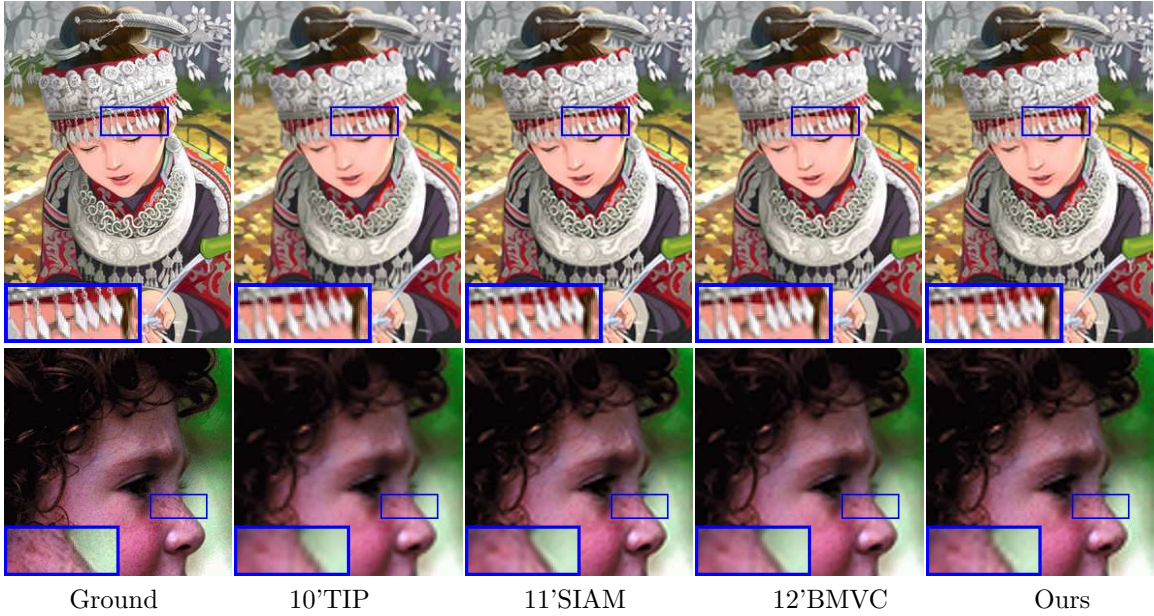


Figure 10: Results of “comic” (first row, upscaling factor 2) and “face” (second row, upscaling factor 4), and the corresponding (RMSE, PSNR, SSIM). *First example*: “10’TIP” (11.11, 27.15, 0.886), “11’SIAM” (10.85, **27.41**, **0.896**), “12’BMVC” (**10.83**, **27.41**, 0.891) and the proposed method (11.01, 27.25, 0.888). *Second example*: “10’TIP” (6.59, 31.76, 0.759), “11’SIAM” (**6.22**, **32.26**, **0.780**), “12’BMVC” (6.39, 32.15, 0.772) and the proposed method (6.36, 32.17, 0.773).

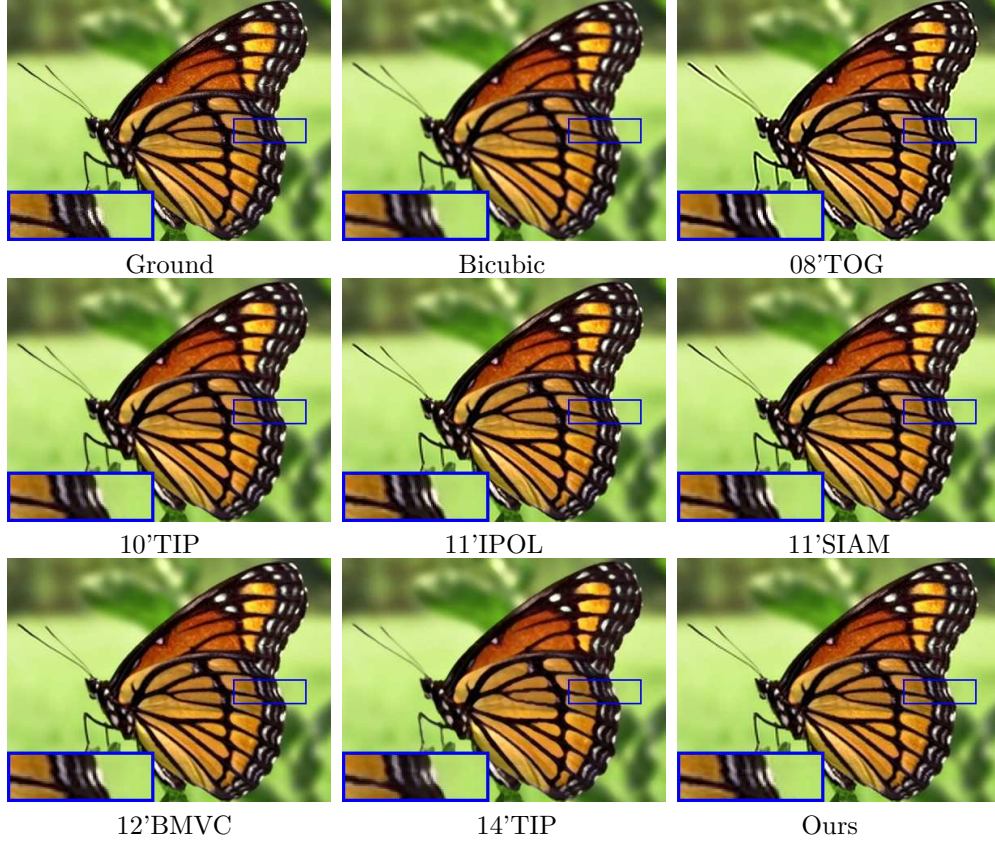


Figure 11: Results of “butterfly” by an upscaling factor of 2. First row: ground-truth, results of bicubic interpolation (6.36, 32.05, 0.956), “08'TOG” (8.08, 29.97, 0.942). Second row: results of “10'TIP” (5.14, 33.86, 0.964), “11'IPOL” (5.32, 33.61, 0.966) and “11'SIAM” (5.22, 33.78, 0.967). Third row: results of “12'BMVC” (**4.90**, **34.33**, **0.969**), “14'TIP” (9.47, 28.58, 0.928) and the proposed method (5.15, 33.81, 0.967).



Figure 12: Results of “girl” (first row) and “animal” (second row) with the upscaling factor of 3. From left to right: low-resolution images, results of “09’ICCV” [11], “13’ICCV” [51], “14’ECCV” [15] and the proposed method.

4.3 Discussions

More comparison: We further compare the proposed method with three recent state-of-the-art methods, i.e., “09’ICCV” [11], “13’ICCV” [51] and “14’ECCV” [15]. The proposed method actually can be viewed as interpolation-based method. It does not need any extra information for training, and is a completely single image super-resolution method. The methods “13’ICCV and “14’ECCV are two competitive learning based methods. In particular, “14’ECCV utilizes the recent most efficient technique: deep learning. They first utilize extra training data to learn dictionaries, and then apply the dictionaries to get the final super-resolution images. Furthermore, the method “09’ICCV” is a very novel and efficient single image super-resolution method. From Fig. 12, we can observe that the proposed method performs comparably with the others, including image edges and the details on non-edge regions.

More on parameters selection: The proposed method involves many parameters, e.g., λ_1 , λ_2 , ξ_1 , ξ_2 , patch size, etc. However, they are easy to select because the results are not sensitive to the selection of parameters (see ξ_1 , ξ_2 in Tab. 3 and λ_1 , λ_2 in Tab. 4). Actually, choosing suitable parameters is always a difficulty to many algorithms. Tuning empirically is a popular way for determining parameters. In our work, we obtain the parameters empirically. For instance, for λ_1 and λ_2 , we first fix one parameter λ_1 , and then tune λ_2 with 10 times change. For instance, we fix $\lambda_1 = 10^{-2}$, then tune λ_2 by 10^{-3} , 10^{-4} , 10^{-5} , 10^{-6} , 10^{-7} , etc. When we find $\lambda_2 = 10^{-6}$ is the best one for λ_1 , then tune λ_2 in the rang of $[10^{-5}, 10^{-7}]$, e.g., 5×10^{-6} and 5×10^{-7} . Actually, the results around $\lambda_1 = 10^{-2}$ and $\lambda_2 = 10^{-6}$ also do not show obvious difference (see Tab. 4). Fine tuning of parameters might lead to slightly better results (see Tab. 5). But to make it simple, we set $\lambda_1 = 10^{-2}$, $\lambda_2 = 10^{-6}$ in all test

examples. Under this setting, the results are already good enough for comparisons. We choose values for other parameters in a similar way.

Table 3: RMSE around $\xi_1 = 10^{-1}$ and $\xi_2 = 10^{-4}$. Test example is “field” in Fig. 8.

$\xi_2 \backslash \xi_1$	0.8×10^{-1}	1×10^{-1}	1.2×10^{-1}
0.8×10^{-4}	12.09	12.08	12.07
1×10^{-4}	12.10	12.08	12.07
1.2×10^{-4}	12.09	12.08	12.08

Table 4: RMSE around $\lambda_1 = 10^{-2}$ and $\lambda_2 = 10^{-6}$.

$\lambda_2 \backslash \lambda_1$	0.8×10^{-2}	1×10^{-2}	1.2×10^{-2}
0.8×10^{-6}	12.09	12.08	12.08
1×10^{-6}	12.09	12.08	12.08
5×10^{-6}	12.10	12.09	12.08

Table 5: RMSE of the proposed method when setting λ_1 from 5×10^{-1} to 1×10^{-3} and λ_2 from 1×10^{-3} to 1×10^{-7} (Bold: the best one; Underline: the second best).

$\lambda_2 \backslash \lambda_1$	5×10^{-1}	1×10^{-1}	5×10^{-2}	1×10^{-2}	5×10^{-3}	1×10^{-3}
1×10^{-3}	12.16	12.14	12.13	12.11	12.13	12.18
1×10^{-4}	12.13	12.12	12.11	12.10	12.12	12.17
1×10^{-5}	12.11	12.10	12.10	12.09	12.11	12.15
5×10^{-6}	12.09	12.07	<u>12.08</u>	<u>12.08</u>	12.11	12.14
1×10^{-6}	12.09	12.07	12.07	<u>12.08</u>	12.10	12.14
5×10^{-7}	12.10	<u>12.08</u>	<u>12.08</u>	12.10	12.10	12.14
1×10^{-7}	12.11	12.10	12.09	12.11	12.11	12.17

Computation time: From Fig. 13(a), we know that the computation time does not increase significantly when the upscaling factor is increased from 2 to 9. Using our non-optimized Matlab code, it is about below 10 seconds for a low-resolution image 60×60 . From Fig. 13(b), when the size of low-resolution image is from 30×30 to 110×110 , the computation time is increased from 3 seconds to 34 seconds. We can conclude that the computation time mainly depends on the size of low-resolution image but not so significantly on the upscaling factor, since the main computation is to compute coefficients β_1 and β_2 via the low-resolution image. Note that, there is a lot of room to speed up the speed. We can use cmex in Matlab to speed up the code that involves a lot of loops. We can also use parallel computing as the computation is done patch by patch.

About θ and c : The proposed method is based on the approximated Heaviside function that mainly depends on two factors θ and c (see Eq. (4)). Here, we tend to study the relations on RMSE, computation time and parameter pairs. From the first row of Fig. 14, the RMSE of the proposed method decreases smoothly (only about 0.05 RMSE reduction) with q increasing from 36 to 90 when setting

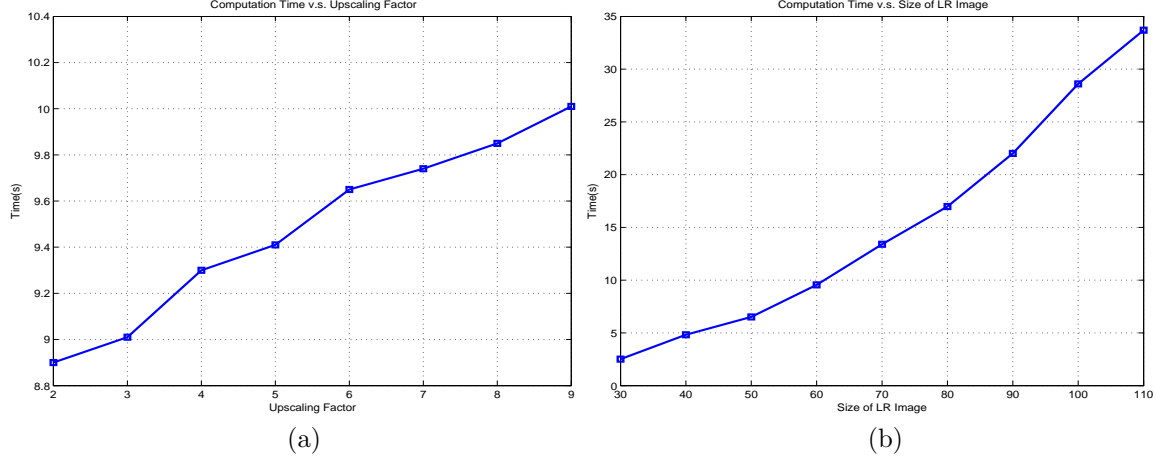


Figure 13: (a) Computation time v.s. upscaling factor for low-resolution image with the size of 60×60 . (b) Computation time v.s. size of low-resolution image, the size of low-resolution image is increased from 30×30 to 110×110 and the upscaling factor is always set to be 5. The experimental computer is with 3.47GB RAM, Inter(R) Core(TM) i3-2130 CPU, @3.40GHz. Note that each value reported in the figure is the average of 10 runs.

$k = 12$, while computation time increases significantly from about 20 seconds to 60 seconds. Thus more parameter pairs (k, q) lead to better approximation but more computation as the matrices Ψ_1 and Ψ_2 will become larger. In particular, we also can get similar conclusions for the second row of Fig. 14. In the experiments, to balance computation efficiency and quality, we set $k = 12$, $q = 36$ empirically (see the parameter selection before section 4.1).

The relation between model (6) and model (6) combined with iterative idea: It is necessary to illustrate the relation between model (6) and model(6) combined with our iterative idea. Actually, there is only fine visual difference, especially in image details and edges, between model (6) and model (6) combined with our iterative idea. Due to small magnitude of the difference, there is no significant visual difference between the two methods. However, from Fig. 15, it is easy to know that the proposed model (6) combined with the iterative AHF method performs smaller RMSE than the proposed model (6). In particular, the model (6) with the iterative strategy will result in more computation.

5 Conclusions

In this paper, we casted image super-resolution problem as an intensity function estimation problem. We assumed the underlying intensity function, defined on a continuous domain and belonging to a space with redundant basis, could be approximately represented by two classes of approximated Heaviside functions (AHFs). The representation coefficients were computed by the proposed iterative AHF method using only one low-resolution image. We then applied the coefficients to get high-resolution images. To reduce computation and storage, we applied the proposed iterative AHF method to image patches. For images with smooth background, we designed a strategy utilizing image gradient to reduce ring artifacts along large scale edges. The method can be applied to any upscaling factors and needs no extra data for learning. In particular, we also discussed the parameter selection and computation time. Many experiments show that the proposed approach outperforms existing competitive methods, both visually and quantitatively.

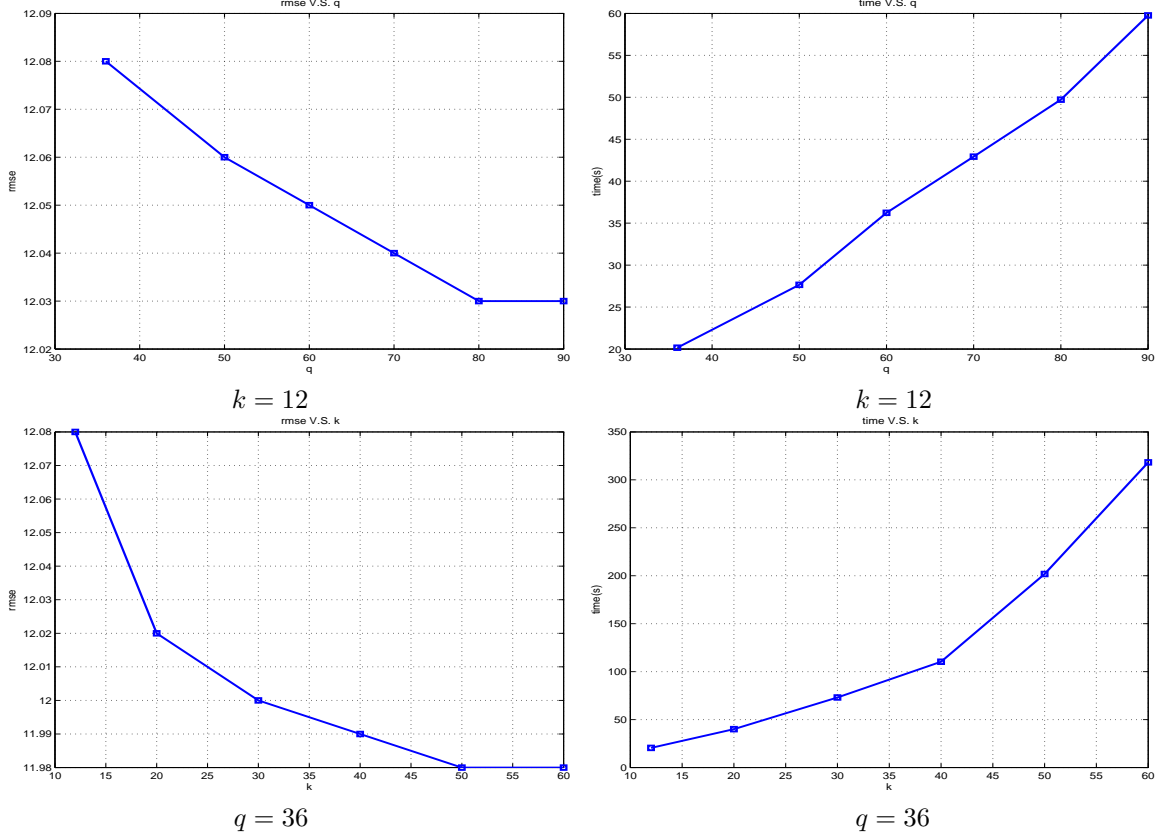


Figure 14: The investigation on RMSE, computation time and the number of parameter pairs (k, q) . First row: $k = 12$, the RMSE of the proposed method decreases smoothly with the number of pixels q increasing from 36 to 90 (left); whereas the computation time increases significantly (right). Second row: $q = 36$, the RMSE of the proposed method decreases smoothly with the number of directions k increasing from 12 to 60 (left); whereas the computation time increases significantly (right). It demonstrate that there is no significant RMSE reduction with more parameter pairs (k, q) , but increasing computation time apparently.

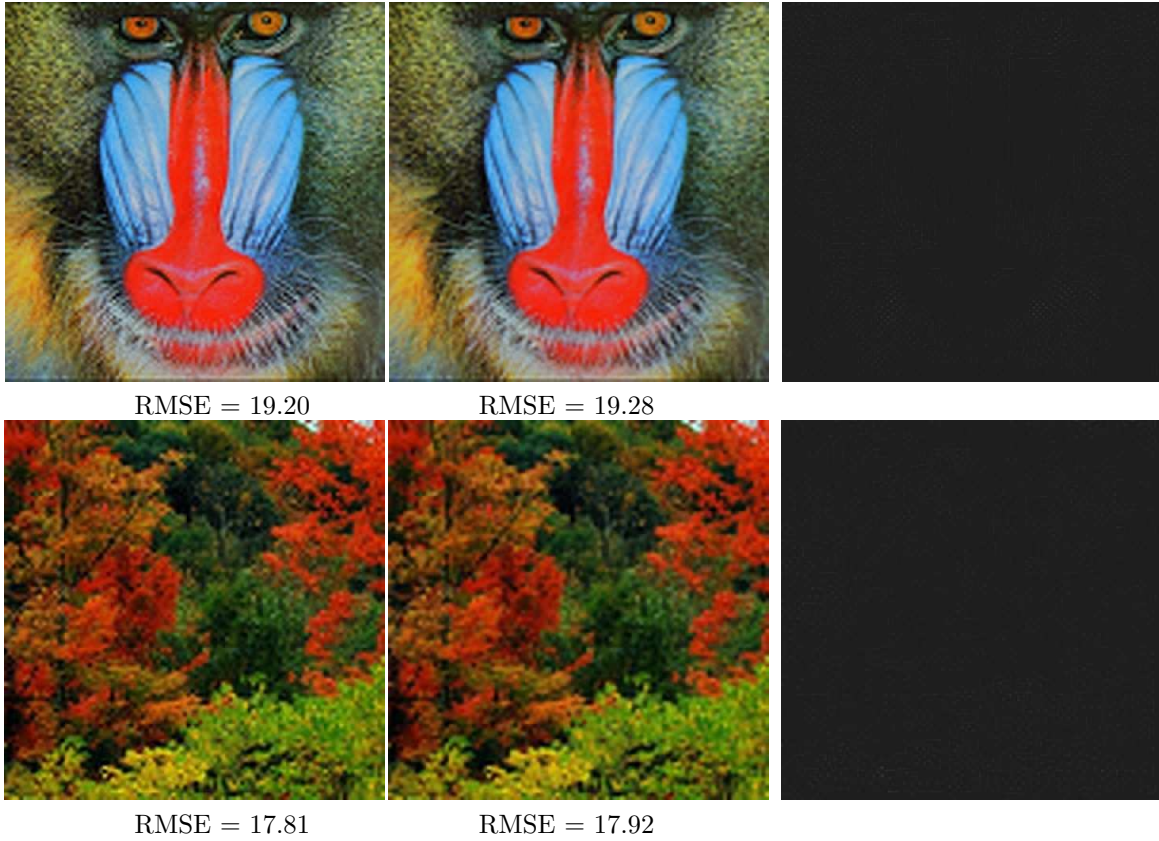


Figure 15: First column: results of model (6) with iterative AHF method. Second column: results of model (6) without iterative AHF method. Third column: error maps (for better vision, we add extra intensity 30/255 to true error maps). Upscaling factors: 4.

Acknowledgement

The first and third authors thank the support by 973 Program (2013CB329404), NSFC (61370147, 61170311), Sichuan Province Sci. & Tech. Research Project (2012GZX0080). The first author is also supported by NSFC (61402082), Fundamental Research Funds for the Central Universities and Outstanding Doctoral Students Academic Support Program of UESTC. The second author thanks US NIH 1R21EB016535-01 for partial support.

References

- [1] C. B. Atkins, C. A. Bouman, and J. P. Allebach. Tree-based resolution synthesis. *International Conference on Image Processing (ICIP)*, pages 405–410, 1999.
- [2] C. B. Atkins, C. A. Bouman, and J. P. Allebach. Optimal image scaling using pixel classification. *International Conference on Image Processing (ICIP)*, pages 864–867, 2001.
- [3] M. Bevilacqua, A. Roumy, C. Guillemot, and M.L.A. Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. *BMVC*, 2012.
- [4] S. Borman and R. L. Stevenson. Super-resolution from image sequences - a review. *Midwest Symposium on Circuits and Systems*, pages 374–378, 1998.
- [5] E. Candès and C. Fernandez-Granda. Towards a mathematical theory of super-resolution. *To appear in Communications on Pure and Applied Mathematics*.
- [6] D. Capel and A. Zisserman. Super-resolution enhancement of text image sequences. *International Conference on Pattern Recognition (ICPR)*, 1:600–605, 2000.
- [7] Y. Cha, G. Lee, and S. Kim. Image zooming by curvature interpolation and iterative refinement. *SIAM Journal on Imaging Sciences*, 7:1284–1308, 2014.
- [8] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40:120–145, 2011.
- [9] H. Chang, D. Yeung, and Y. Xiong. Super-Resolution through neighbor embedding. *Computer Vision and Pattern Recognition (CVPR)*, 1, 2004.
- [10] P. Chatterjee, S. Mukherjee, S. Chaudhuri, and G. Seetharaman. Application of Papoulis-Gerchberg method in image super-resolution and inpainting. *The Computer Journal*, 52:80–89, 2007.
- [11] G. Daniel, S. Bagon, and M. Irani. Super-Resolution from a single image. *ICCV*, pages 349–356, 2009.
- [12] L.-J. Deng, H. Guo, and T.-Z. Huang. A fast image recovery algorithm based on splitting deblurring and denoising. *Journal of Computational and Applied Mathematics*, 287:88–97, 2015.
- [13] L.-J. Deng, W. Guo, and T.-Z. Huang. Single image super-resolution via an iterative reproducing kernel Hilbert space method. *IEEE Trans. on Circuits and Systems for Video Technology*, 2015, doi:10.1109/TCSVT.2015.2475895.
- [14] L.-J. Deng, T.-Z. Huang, and X.-L. Zhao. Wavelet-based two-level methods for image restoration. *Communications in Nonlinear Science and Numerical Simulation*, 17:5079–5087, 2012.
- [15] C. Dong, C. Loy, K. He, and X. Tang. Learning a deep convolutional network for image super-resolution. *ECCV*, 2014.
- [16] W. Dong, G. Shi, L. Zhang, and X. Wu. Superresolution with nonlocal regularized sparse representation. *Proceeding of SPIE*, 2010.
- [17] S. Farsiu, M. D. Robinson, M. Elad, and P. Milanfar. Fast and robust multiframe super resolution. *IEEE transactions on image processing*, 13:1327–1344, 2004.
- [18] R. Fattal. Image upsampling via imposed edge statistics. *ACM Transactions on Graphics*, 26, 2007.
- [19] C. Fernandez-Granda and E. Candès. Super-resolution via transform-invariant group-sparse regularization. *ICCV*, 2013.
- [20] G. Freedman and R. Fattal. Image and video upscaling from local self-examples. *ACM Trans. on Graphics (TOG)*, 30, 2011.
- [21] W. T. Freeman, T. R. Jones, and E. C. Pasztor. Example-based super-resolution. *IEEE Computer Graphics and Applications*, 22:56–65, 2002.

- [22] W. T. Freeman and E. C. Pasztor. Markov networks for super-resolution. *Proceedings of 34th Annual Conference on Information Sciences and Systems*, 2000.
- [23] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael. Learning low-level vision. *International Journal of Computer Vision*, 40:25–47, 2000.
- [24] P. Getreuer. Contour stencils: Total Variation along curves for adaptive image interpolation. *SIAM Journal on Imaging Sciences*, 4:954–979, 2011.
- [25] P. Getreuer. Image interpolation with contour stencils. *Image Processing On Line*, 1, 2011.
- [26] T. Goldstein and S. Osher. The split bregman method for l1-regularized problems. *SIAM Journal on Imaging Sciences*, 2:323–343, 2009.
- [27] W. Gong, L. Hu, J. Li, and W. Li. Combining sparse representation and local rank constraint for single image super-resolution. *Information Sciences*, 325:1–19, 2015.
- [28] E. Gur and Z. Zalevsky. Single-Image digital super-resolution a revised Gerchberg-Papoulis algorithm. *IAENG International Journal of Computer Science*, 34:251–255, 2007.
- [29] B. He, M. Tao, and X. Yuan. Alternating direction method with gaussian back Substitution for separable Convex programming. *SIAM Journal on Optimization*, 22:313–340, 2012.
- [30] L. He, H. Qi, and R. Zaretzki. Beta process joint dictionary learning for coupled feature spaces with application to single image super-resolution. *CVPR*, pages 345–352, 2013.
- [31] M. Irani and S. Peleg. Super resolution from image sequence. *Proceedings of 10th International Conference on Pattern Recognition (ICPR)*, pages 115–120, 1990.
- [32] M. Irani and S. Peleg. Motion analysis for image enhancement: resolution, occlusion and transparency. *Journal of Visual Communication and Image Representation*, 4:324–335, 1993.
- [33] P. C. Kainen, V. Kůrková, and A. Vogt. Best approximation by linear combinations of characteristic functions of half-space. *Journal of Approximation Theory*, 122:151–159, 2003.
- [34] C. Kim, K. Choi, K. Hwang, and J. B. Ra. Learning-based super-resolution using a multi-resolution wavelet approach. *International workshop on Advance Image Technology (IWAIT)*, 2009.
- [35] C. Kim, K. Choi, and J. B. Ra. Improvement on learning-based super-resolution by adopting residual information and patch reliability. *IEEE International Conference on Image Processing (ICIP)*, pages 1197–1200, 2009.
- [36] K. I. Kim and Y. Kwon. Single-image super-resolution using sparse regression and natural image prior. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 32:1127–1133, 2010.
- [37] K. Komatsu, T. Igarashi, and T. Saito. Very high resolution imaging scheme with multiple different-aperture cameras. *Signal Processing: Image Communication*, 5:511–526, 1993.
- [38] J. Li, W. Gong, and W. Li. Dual-sparsity regularized sparse representation for single image super-resolution. *Information Sciences*, 298:257–273, 2015.
- [39] X. Li and M. Orchard. New edge-directed interpolation. *IEEE Trans. Image Processing*, 10:1521–1527, 2001.
- [40] J. Liu, T.-Z. Huang, I. W. Selesnick, X.-G. Lv, and P.-Y. Chen. Image restoration using total variation with overlapping group sparsity. *Information Sciences*, 295:232–246, 2015.
- [41] Liyakathunisa and V. K. Ananthashayana. Super resolution blind reconstruction of low resolution images using wavelets based fusion. *International Journal of Computer and Information Engineering*, 2:106–110, 2008.
- [42] T.-H. Ma, T.-Z. Huang, and X.-L. Zhao. Group-based image decomposition using 3D cartoon and texture priors. *Information Sciences*, 328:510–527, 2016.
- [43] N. Mueller, Y. Lu, and M. Do. Image interpolation using multiscale geometric representations. *SPIE proceedings*, 2007.
- [44] Q. Shan, Z. Li, J. Jia, and C. Tang. Fast image/video upsampling. *ACM Transactions on Graphics (TOG)*, 27, 2008.
- [45] H. Shen, L. Zhang, B. Huang, and P. Li. A MAP Approach for Joint Motion Estimation, Segmentation, and Super Resolution. *IEEE Trans. Image Processing*, 16:479–490, 2007.
- [46] J. Sun, J. Sun, Z. Xu, and H.-Y. Shum. Image super-resolution using gradient profile prior. *CVPR*, pages 1–8, 2008.

- [47] J. Sun, N. N. Zheng, H. Tao, and H. Shum. Image hallucination with primal sketch priors. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2:729–736, 2003.
- [48] Y.-W. Tai, S. Liu, M. Brown, and S. Lin. Super resolution using edge prior and single image detail synthesis. *CVPR*, pages 2400–2407, 2010.
- [49] H. Takeda, S. Farsiu, and P. Milanfar. Kernel regression for image processing and reconstruction. *IEEE Transactions on Image Processing*, 16:349–366, 2007.
- [50] M. F. Tappen, B. C. Russell, and W. T. Freeman. Exploiting the sparse derivative prior for super-resolution and image demosaicing. *IEEE Workshop on Statistical and Computational Theories of Vision*, 2003.
- [51] R. Timofte, V. De Smet, and L. Van Gool. Anchored neighborhood regression for fast example-based super-resolution. *ICCV*, pages 1920–1927, 2013.
- [52] R. Timofte, V. De Smet, and L. Van Gool. A+: adjusted anchored neighborhood regression for fast super-resolution. *ACCV*, 2014.
- [53] F. Viola, A. W. Fitzgibbon, and R. Cipolla. A unifying resolution-independent formulation for early vision. *CVPR*, pages 494–501, 2012.
- [54] L. Wang, H. Wu, and C. Pan. Fast image upsampling via the displacement field. *IEEE Trans. Image Processing*, 23:5123–5135, 2014.
- [55] Y. Wang, J. Yang, W. Yin, and Y. Zhang. A new alternating minimization algorithm for total variation image reconstruction. *SIAM Journal on Imaging Sciences*, 1:248–272, 2008.
- [56] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Processing*, 13:600–612, 2004.
- [57] Q. Xie, H. Chen, and H. Cao. Improved example-based single-image superresolution. *International Congress on Image and Signal Processing (CISP)*, 3:1204–1207, 2010.
- [58] J. Yang, Z. Wang, L. Zhe, and T. Huang. Coupled dictionary training for image super-resolution. *IEEE transactions on image processing*, 21:3467–3478, 2011.
- [59] J. Yang, J. Wright, T. Huang, and Y. Ma. Image super-resolution via sparse representation. *IEEE transactions on image processing*, 19:2861–1873, 2010.
- [60] J. Yang, J. Wright, Y. Ma, and T. Huang. Image super-resolution as sparse representation of raw image patches. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008.
- [61] J. Yu, Y. Rui, Y. Tang, and D. Tao. High-Order distance-based multiview stochastic learning in image classification. *IEEE Trans. Cybernetics*, 44:2431–2442, 2014.
- [62] J. Yu, Y. Rui, and D. Tao. Click prediction for web image reranking using multimodal sparse coding. *IEEE Trans. Image Processing*, 23:2019–2032, 2014.
- [63] J. Yu, D. Tao, M. Wang, and Y. Rui. Learning to rank using user clicks and visual features for image retrieval. *IEEE Trans. Cybernetics*, 45:767–779, 2015.
- [64] Q. Yuan, L. Zhang, and H. Shen. Regional Spatially Adaptive Total Variation Super-resolution with Spatial Information Filtering and Clustering. *IEEE Trans. Image Processing*, 22:2327–2342, 2013.
- [65] Q. Yuan, L. Zhang, H. Shen, and P. Li. Adaptive Multiple-Frame Image Super-resolution Based on U-curve. *IEEE Trans. Image Processing*, 19:3157–3170, 2010.
- [66] L. Yue, H. Shen, Q. Yuan, and L. Zhang. A locally adaptive L1-L2 norm for multi-frame super-resolution of images with mixed noise and outliers. *Signal Processing*, 105:156–174, 2014.
- [67] R. Zeyde, M. Elad, and M. Protter. On single image scale-up using sparse-representations. *Curves and Surfaces, Lecture Notes in Computer Science*, 6920:711–730, 2012.
- [68] D. Zhang, P.-M. Jodoin, C. Li, Y. Wu, and G. Cai. Novel graph cuts method for multi-frame super-resolution. *IEEE Signal Processing Letters*, 22:2279–2283, 2015.
- [69] L. Zhang and X. Wu. An edge-guided image interpolation algorithm via directional filtering and data fusion. *IEEE Trans. Image Processing*, 15:2226–2238, 2006.
- [70] L. Zhang, Q. Zhang, L. Zhang, D. Tao, X. Huang, and B. Du. Ensemble manifold regularized sparse low-rank approximation for multiview feature embedding. *Pattern Recognition*, 48:3102–3112, 2015.
- [71] Y. Zhang, Y. Du, F. Ling, and X. Li. Improvement of the example-regression-based super-resolution land cover mapping algorithm. *IEEE Geoscience and Remote Sensing Letters*, 12:1740–1744, 2015.

- [72] Y. Zhang, J. Liu, W. Yang, and Z. Guo. Image super-resolution based on structure-modulated sparse representation. *IEEE Trans. Image Processing*, 24:2797–2810, 2015.
- [73] X.-L. Zhao, F. Wang, and M. K. Ng. A new convex optimization model for multiplicative noise and blur removal. *SIAM Journal on Imaging Sciences*, 7:456–475, 2014.
- [74] X.-L. Zhao, W. Wang, T.-Y. Zeng, T.-Z. Huang, and M. K. Ng. Total variation structured total least squares method for image restoration. *SIAM Journal on Scientific Computing*, 35:1304C1320, 2013.
- [75] Y. Zhao, J. Yang, Q. Zhang, S. Lin, Y. Cheng, and Q. Pan. Hyperspectral imagery superresolution by sparse representation and spectral regularization. *EURASIP Journal on Advances in Signal Processing*, 2011.
- [76] H. Zheng, A. Bouzerdoun, and S. L. Phung. Wavelet based nonlocal-means superresolution for video sequences. *IEEE International Conference on Image Processing (ICIP)*, pages 2817–2820, 2010.