

RESEARCH

Open Access



# Super-resolution via a fast deconvolution with kernel estimation

Han Yu, Ting-Zhu Huang\* , Liang-Jian Deng and Xi-Le Zhao

## Abstract

Image super-resolution has wide applications in biomedical imaging, computer vision, image recognition, etc. In this paper, we present a fast single-image super-resolution method based on deconvolution strategy. The deconvolution process is implemented via a fast total variation deconvolution (FTVd) method that runs very fast. In particular, due to the inaccuracy of kernel, we utilize an iterative strategy to correct the kernel. The experimental results show that the proposed method can improve image resolution effectively and pick up more image structures. In addition, the speed of the proposed method is fast.

**Keywords:** Super-resolution, Kernel estimation, Fast total variation deconvolution (FTVd)

## 1 Introduction

The process of estimating a high-resolution (HR) image from one or multiple low-resolution (LR) images is often referred to as image super-resolution. According to the number of low-resolution images, image super-resolution can be divided into two categories: one is single-image super-resolution, and the other is multiple-image super-resolution. Based on image sequence, multiple-image super-resolution uses overlapping information between multiple low-resolution images to estimate details of the high-resolution image [1–3]. Due to multiple-image super-resolution needs more than one input image, it cannot deal with the situation when only one image is inputted. In this paper, we mainly focus on single-image super-resolution.

Interpolation-based methods are one of classical image super-resolution methods. To determine pixel values of each position in the high-resolution image, interpolation-based methods need to construct a rational interpolation function. The conventional interpolation-based methods contain bicubic interpolation method, cubic interpolation method [4], cubic spline interpolation method, nearest-neighbor interpolation method, etc. These methods usually run very fast but always produce blurring or jagged artifacts. Recently, many contributions in terms of

interpolation-based methods have been proposed [5–11]. In [6], Zhang et al. present an edge-guided interpolation algorithm through directional filtering and data fusion. This method can preserve sharp edges. In [9], Wang et al. propose a fast image upsampling method within a two-scale framework. They use interpolation method to recover the low-frequency image and reconstruction technique to recover the local high-frequency structures.

Reconstruction-based methods are another class of image super-resolution approaches. Through studying the low-resolution image generating process, reconstruction-based methods use a certain model to depict the mapping relationship between the high-resolution image and the low-resolution image. There are three main types of reconstruction-based methods: frequency domain techniques, spatial domain techniques, and other techniques. Frequency domain techniques [12–14] solve the problem in frequency domain, and the observation model is based on displacement characteristics of Fourier transform. Spatial domain techniques, such as non-uniform interpolation within samples method [15], convex set projection method [16], statistical recovery method (maximum a posteriori and maximum likelihood estimation) [2, 17, 18], solve the problem in spatial domain. In addition, there are some other reconstruction-based methods [19–24]. In [22], Shan et al. propose an efficient upsampling method which lies in a feedback-control framework. This method runs very fast and can preserve the essential structural information.

\*Correspondence: tingzhuhuang@126.com

Research Center for Image and Vision Computing/School of Mathematical Sciences, University of Electronic Science and Technology of China, Xiyuan Ave, 611731 Chengdu, China

Learning-based methods are the third category of image super-resolution methods. Learning-based methods [25–32] need to train two dictionaries for low-resolution and high-resolution images or patches. When given a low-resolution image, learning-based methods can get a high-resolution image through using the learnt mapping relationship between the two dictionaries. Although these methods obtain good visual results, they rely on the two training dictionaries and cannot change the magnification factor arbitrarily.

In this paper, we propose a new single image super-resolution method based on deconvolution strategy. According to Efrat et al. [33], an accurate kernel is more important than a sophisticated prior for image super-resolution. Thus, we take into account the error of blur kernel in our method. We develop an iterative strategy to adjust the blur kernel and then estimate the final high-resolution image via a reconstruction method. The proposed method is based on the framework of [22]. However, it has two main contributions. First, the proposed method can get faster speed than [22], since we employ a fast total variation deconvolution (FTVd) method in our work. Second, the proposed method estimates the deconvolution kernel iteratively to get better visual and quantitative results than [22].

The rest of this paper is organized as follows. Section 2 introduces image super-resolution problem, reviews FTVd method and a fast image upsampling method. In Section 3, we detail the proposed model and algorithm. Numerical results are shown in Section 4. Finally, we draw some conclusions in Section 5.

## 2 Problem description and related works

### 2.1 Image super-resolution problem

Let  $H$  be a high-resolution image and let  $f$  be the camera's point spread function (PSF) which is approximated by a Gaussian filter. According to [1], the low-resolution image can be modeled as

$$L = (f \otimes H) \downarrow^d, \quad (1)$$

where  $\downarrow^d$  is a downsampling operator with factor  $d$ . This equation can be broken down into two steps,

$$B = f \otimes H, L = B \downarrow^d, \quad (2)$$

where  $B$  is a linearly filtered high-resolution image. Image super-resolution is to recover the high-resolution image  $H$  from an input low-resolution image  $L$ .

### 2.2 Image super-resolution problem

FTVd method is a popular way to solve the problem of image restoration [34]. Image restoration is a process of recovering images from blurring and noise observation. This process can be modeled as

$$g = Au + n, \quad (3)$$

where  $g$  represents the observed image,  $A$  represents a convolution matrix,  $u$  is an original image, and  $n$  is random noise.

A degraded image  $u$  can be recovered from the following simple model:

$$\min_u \sum_i \|D_i u\| + \frac{\mu}{2} \|Au - g\|_2^2, \quad (4)$$

where  $D_i u$  is the discrete gradient of  $u$  at pixel  $i$ ,  $\sum_i \|D_i u\|$  is the discrete total variation (TV) of  $u$ , and  $\mu$  is a regularization parameter.

Actually, similar cases with the above problems have been studied by many works [34–39]. In particular, FTVd method [34] is one of effective methods for solving Eq. (4). In [34], Wang et al. utilize an auxiliary variable  $w_i$  to substitute  $D_i u$  to generate the following problem:

$$\min_{w,u} \sum_i \|w_i\|_2 + \frac{\beta}{2} \sum_i \|w_i - D_i u\|_2^2 + \frac{\mu}{2} \|Au - g\|_2^2, \quad (5)$$

where  $\beta$  is a penalty parameter.

This model is a half-quadratic model, which can be applied to total variation discretization with anisotropic or isotropic form. In [34], Wang et al. use a fast total variation deconvolution (FTVd) method to solve Eq. (5) (see Algorithm 1). This algorithm can be applied to image deblurring with different blurring kernels and different noise.

---

#### Algorithm 1 Fast total variation deconvolution (FTVd) method

---

**Input:**  $g, A, \mu > 0, \beta_0 > 0$  and  $\beta_{\max} > \beta_0$ .

**Output:**  $w$  and  $u$ .

**Initialize:**  $u = g, \beta = \beta_0$

**while**  $\beta \leq \beta_{\max}$  **do**

**while** "not converged" **do**

**Step 1:** Computing  $w$  by

$$w_i = \max \left\{ \|D_i u\| - \frac{1}{\beta}, 0 \right\} \frac{D_i u}{\|D_i u\|}; \quad (6)$$

**Step 2:** Solving  $u$  via

$$\left( \sum_i D_i^T D_i + \frac{\mu}{\beta} A^T A \right) u = \sum_i D_i^T w_i + \frac{\mu}{\beta} A^T g; \quad (7)$$

Note that Eq. (7) can be solved by Fast Fourier transforms (see more details in [34]).

**end while**

**Step 3:** Computing  $\beta$  by

$$\beta = 2\beta; \quad (8)$$

**end while**

---

### 2.3 Fast image upsampling method

In [22], Shan et al. introduce a new single image super-resolution method. This method can enhance image resolution automatically and preserve essential structural information. A key feature of this method is a feedback-control framework that contains three parts: deconvolution, reconvolution, and pixel substitution.

In the deconvolution process, Shan et al. [22] take a non-blind deconvolution method. The non-blind deconvolution method is to solve the following energy function:

$$E(H) \propto \|f \otimes H - B\|_2^2 + \lambda(\|\Phi(\partial_x H)\|_1 + \|\Phi(\partial_y H)\|_1), \quad (9)$$

where  $\partial_x H$  and  $\partial_y H$  are the values of the  $x$  and  $y$  direction gradients, respectively.  $\lambda$  is a regularization parameter. After the deconvolution process, the output image is refiltered in the reconvolution stage. In the process of pixel substitution, pixels of the low-resolution image are utilized to replace the pixels at the corresponding locations of the high-resolution image. There are two advantages for using pixel substitution. First, it can utilize the accurate low-resolution image pixels. Second, it can approximate the image, output from the reconvolution process, as a Gaussian-filtered image with a feedback-control loop. This method does not depend on the quality and quantity of the selected examples. Besides, the running time of this method is very fast.

## 3 The proposed method

### 3.1 The proposed framework

In this section, we give the proposed method which is consisted of four parts: deconvolution, estimating blur kernel, reconvolution, and pixel substitution.

Figure 1 shows the diagram of our framework. In our scheme, the input is a low-resolution image  $L$ . We first transform the low-resolution image from RGB color space to YUV color space. Next, we upsample the low-resolution image to an ideal size by bicubic interpolation method and only conduct at Y space. We take an iterative strategy to achieve the image upsampling process (see Algorithm 2). Our strategy contains four parts: deconvolution, estimating blur kernel, reconvolution, and pixel substitution. We take deconvolution process to eliminate the effect of the linear filtering. For instance, there are some visual artifacts around the image “wheel” after bicubic interpolation in the Y space (see Fig. 1). Besides, because the accurate blur kernel can not be known exactly, the further estimated high-resolution image will become more inaccurate. Thus, we take account of the error of blur kernel. Furthermore, the same as [22], we take reconvolution and pixel substitution process to control the image upsampling. By applying our strategy iteratively at the initial

high-resolution image  $B^{(0)}$ , we can obtain the estimated high-resolution image at Y space. The final estimated high-resolution image  $H^*$  is acquired by transforming the high-resolution image from YUV color space to RGB color space. We will show more details about the four steps of the proposed method (deconvolution, estimating blur kernel, reconvolution, and pixel substitution) as follows.

---

**Algorithm 2** Single image super-resolution via a fast deconvolution with kernel estimation (see the flow chart of the framework from Fig. 1)

---

**Input:** a low-resolution image:  $L$ , upsampling factor:  $d$ , iteration number:  $\tau$ , an initial kernel:  $\tilde{f}^{(0)}$ , regularization parameter:  $\alpha$ .

**Output:** the high-resolution image:  $H^*$ .

**Step 1:** Getting the initial high-resolution image  $B^{(0)}$  by bicubic interpolation with the upsampling factor  $d$ :

$$B^{(0)} = \text{Bicubic}(L, d);$$

**Step 2:** Computing the estimating high-resolution image  $H^{(i)}$ :

**for**  $i = 1 : \tau$  **do**

a. Computing the high-resolution image by FTVd method (see Algorithm 1):

$$H^{(i)} = \text{FTVd}(B^{(i-1)}, \tilde{f}^{(i-1)});$$

where  $H^{(i)}$  is equal to  $u$ ,  $B^{(i-1)}$  is equivalent to  $g$  and the blur kernel  $\tilde{f}^{(i-1)}$  can generate  $A$  in Algorithm 1.

b. Estimating the error of blur kernel  $e^{(i)}$  via Eq. (11):

$$e^{(i)} = \text{Estimate}(H^{(i)}, B^{(0)}, \tilde{f}^{(i-1)}, \alpha);$$

where  $H^{(i)}$  is the intermediate high-resolution image and  $\tilde{f}^{(i-1)}$  is the estimated blur kernel.

c. Updating the blur kernel:  $\tilde{f}^{(i)} = \tilde{f}^{(i-1)} + e^{(i)}$ ;

d. Reconvoluting the image  $H^{(i)}$  by the initial kernel  $\tilde{f}^{(0)}$ :

$$\text{Reblur} = \tilde{f}^{(0)} \otimes H^{(i)};$$

e. Updating the initial high-resolution image by pixel substitution (see details in Section 3.1):

$$B^{(i)} = \text{pixelsubs}(L, \text{Reblur});$$

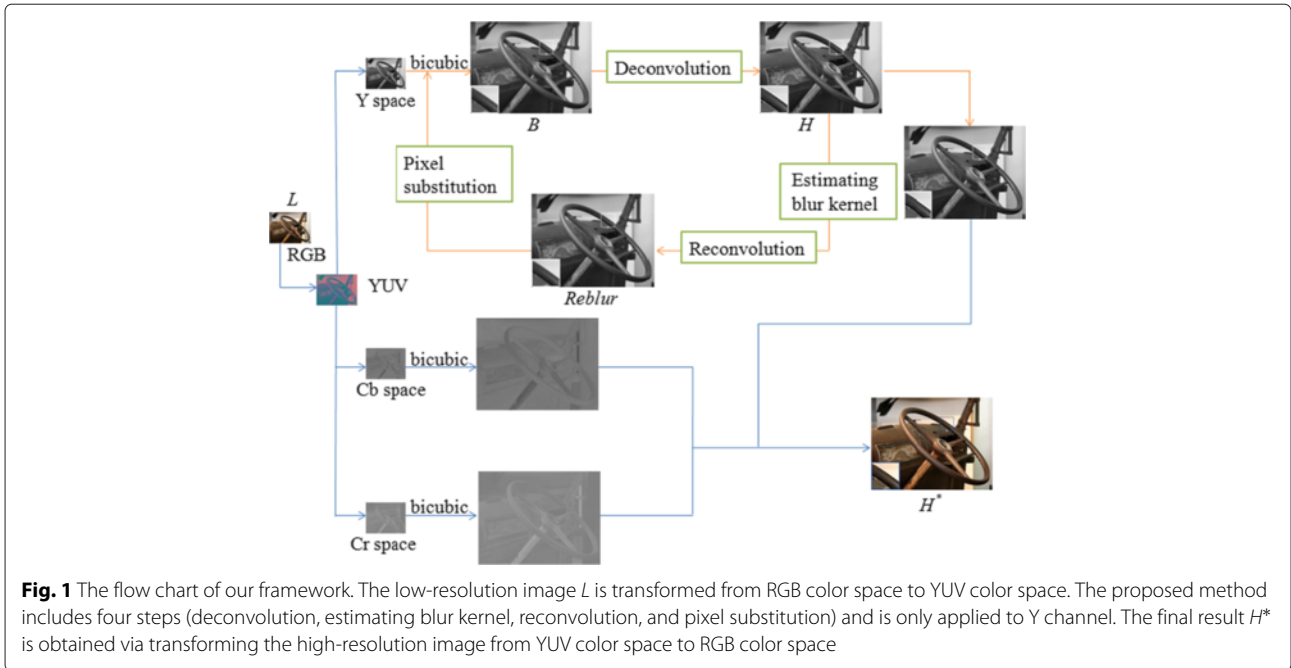
**end for**

**Step 3:** Computing the final high-resolution image  $H^* = H^{(i)}$ .

---

#### 3.1.1 Deconvolution

Let  $B^{(i)}$  as a high-resolution that is gotten at iteration  $i, i \geq 0$ , and  $B^{(0)}$  is obtained by bicubic interpolation method. In particular,  $B^{(0)}$  is obtained by bicubic interpolation method. The deconvolution process, estimating the high-resolution image  $H^{(i)}$ , can be regarded as solving the first part of Eq. (2). This problem can be modeled as  $\|f \otimes H - B\|$ , where  $\otimes$  is a convolution operator. Let  $W$  be a convolution matrix, then  $\|f \otimes H - B\|$  can be regarded as a



multiplication of  $W$  and  $H$ . Note that the problem of minimizing  $\|f \otimes H - B\|_2^2$  is hard to solve because the inverse of  $W$  does not always exist and sometimes  $W$  can be influenced by noise. In particular, taking the high-resolution image  $B$  as a blurred image, the deconvolution process can be considered as an image restoration problem. In this paper, we take FTVd method [34] in the deconvolution process, since FTVd method [34] is an effective way to deal with image restoration problem. The main steps of FTVd method are shown in Section 2.1.

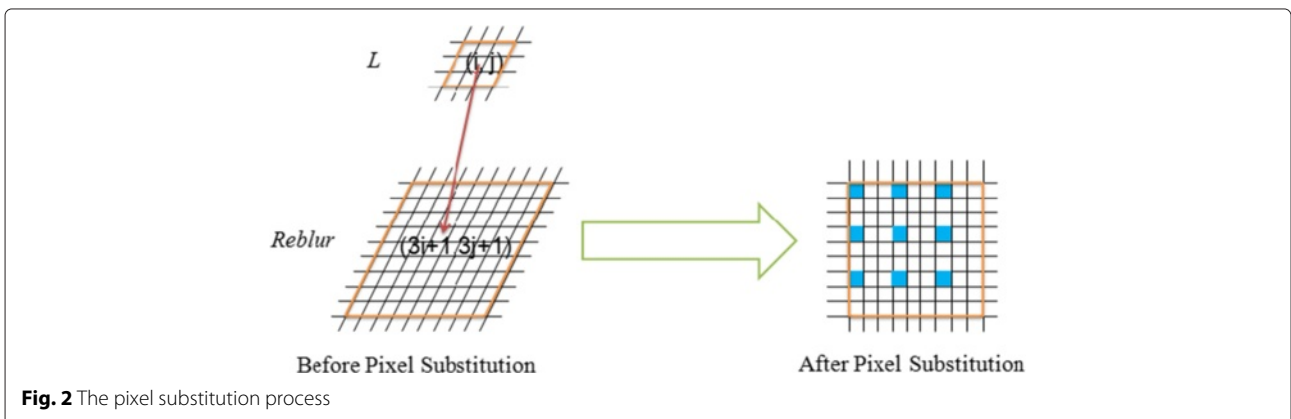
### 3.1.2 Estimating blur kernel

Because the blur kernel is not known exactly in the image formation process, it may have some errors:  $f = \tilde{f} + e$ , where  $f$  is the accurate blur kernel and  $\tilde{f}$  is the inaccurate blur kernel containing an error  $e$ . In order to get a

reasonable high-resolution image, we need to consider the error of blur kernel.

We use the method similar to that described in [40] to estimate the blur kernel. Considering a connected bounded domain  $\Omega \in R^2$  with compact Lipschitz boundary, we take the initial high-resolution image  $B^{(0)}$ , acquired by bicubic interpolation, as a blurred image and the intermediate high-resolution image at  $i$  times  $H^{(i)}$  as a real image.  $\tilde{f}$  be the blur kernel and  $e$  be the error of blur kernel. We study the following objective function to get the error of blur kernel  $e$ :

$$\min_e \int_{\Omega} [(\tilde{f} + e) \otimes H - B]^2 dx + \alpha \int_{\Omega} e^2 dx, \quad (10)$$



**Table 1** Parameter selection in terms of blur kernel and regularization parameter (case 1 is for the low-resolution image without ground truth; case 2 is for the low-resolution images acquired by downsampling the ground truth images)

	Upscale factor	Size	Deviation	$\alpha$
Case 1	2	5 × 5	1.25	10 <sup>3</sup>
	3	7 × 7	1.85	10 <sup>4</sup>
	4	9 × 9	2.5	10 <sup>5</sup>
Case 2	2	3 × 3	1.5	10 <sup>3</sup>
	3	5 × 5	1.8	10 <sup>4</sup>
	4	7 × 7	2.3	10 <sup>5</sup>

where  $\alpha$  is a positive regularization parameter. This problem can be solved by fast Fourier transform:

$$e = \mathcal{F}^{-1} \left( \frac{\mathcal{F}^{-1}(H)\mathcal{F}(B - \tilde{f} \otimes H)}{\mathcal{F}^{-1}(H)\mathcal{F}(H) + \alpha} \right), \quad (11)$$

where  $\mathcal{F}$  and  $\mathcal{F}^{-1}$  are the Fourier transform and the inverse Fourier transform, respectively. When we compute the error  $e$ , then the blur kernel can be estimated by  $f = \tilde{f} + e$ .

### 3.1.3 Reconvolution

Taking account of reconvoluting the output image  $H^{(i)}$  with the blur kernel  $\tilde{f}$ , the result should be close to  $B^{(i-1)}$ , where  $i \geq 1$ . If not, there must be some incorrect pixel values in  $B$ . So we need to modify the high-resolution image using the low-resolution image information, which leads

to pixel substitution in the next step. In particular, in each reconvolution step, we choose the initial blur kernel  $\tilde{f}^{(0)}$  to reconvolute the high-resolution image. If we choose the updated blur kernel in the reconvolution process, the high-resolution image cannot be well estimated due to the change of blur kernel.

In the pixel decimation process, a low-resolution image is acquired by subsampling the high-resolution image with a downsampling factor  $d$ . In addition, the subsampling process only keep one pixel in the high-resolution image. Thus, the corresponding pixels in the high-resolution image can be substituted for pixels in the low-resolution image. In this paper, we take the pixel substitution strategy the same as [22] (see Fig. 2). If we upscale the low-resolution image for  $d$  times, we use the pixel  $(i, j)$  in the low-resolution image  $L$  to replace the pixel  $(d \times i + 1, d \times j + 1)$  in the corresponding high-resolution image *Reblur*.

Then, we can use the pixel-replaced image to conduct the next iteration. After several iterations, the estimated high-resolution image  $H^*$  can be obtained. Our algorithm is given in Algorithm 2.

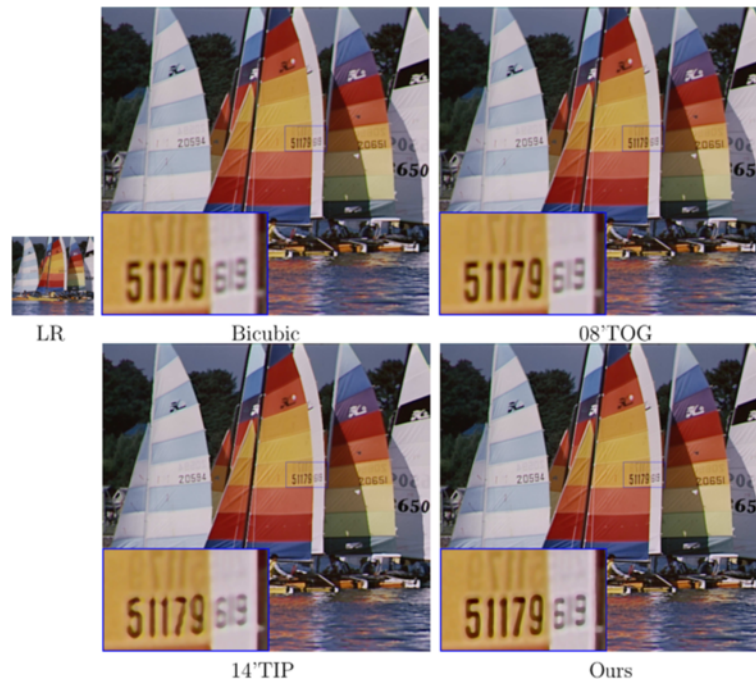
### 3.2 The difference between [22] and the proposed method

In [22], Shan et al. introduce a fast image/video upsampling method that involves a feedback-control framework. In particular, the proposed method has the similar feedback-control framework with the work in [22] (see Fig. 1). However, there are two main differences comparing with [22].

First, there are two different methods between [22] and the proposed method in the deconvolution process. Shan et al. [22] take account of a density distribution prior.



**Fig. 3** Visual results of “flower” with the upsampling factor of 4. From top to bottom and from left to right: the low-resolution image “flower,” bicubic result, “08'TOG” [22], “11'IPOL” [7], “11'SIAM” [8], “14'TIP” [9], and our result



**Fig. 4** Visual results of “yacht” with the upsampling factor of 4. From left to right and then from top to bottom: the low-resolution image “yacht,” bicubic result, “08'TOG” [22], “14'TIP” [9], and our result

However, the proposed method utilizes FTVd method in the deconvolution process. Since the main step of FTVd method is fast Fourier transforms (FFTs) for each iteration, we can control the iterative number to get the faster speed and more accurate results than the deconvolution method in [22].

Second, due to the inaccuracy of blur kernel in the deconvolution process, we employ a strategy of iterative kernel estimation to get more accurate kernel, aiming to make the resulted high-resolution image better. In contrast, Shan et al. [22] only utilize a fixed Gaussian kernel. However, the fixed kernel cannot get better results obviously.

**Table 2** Times of test images (unit: seconds)

Example(factor)	Bicubic	08'TOG [22]	14'TIP [9]	Ours
Flower(4)	0.063	8.268	0.641	4.512
Yacht(4)	0.145	74.948	3.919	29.716
Wheel(3)	0.052	3.886	0.323	3.529
Comic(3)	0.073	17.136	1.281	5.654
Chilies(3)	0.032	8.053	0.466	3.403
Number(3)	0.055	5.518	0.416	4.255
House(2)	0.069	14.267	0.598	5.472
Starfish(2)	0.072	13.289	0.665	5.706

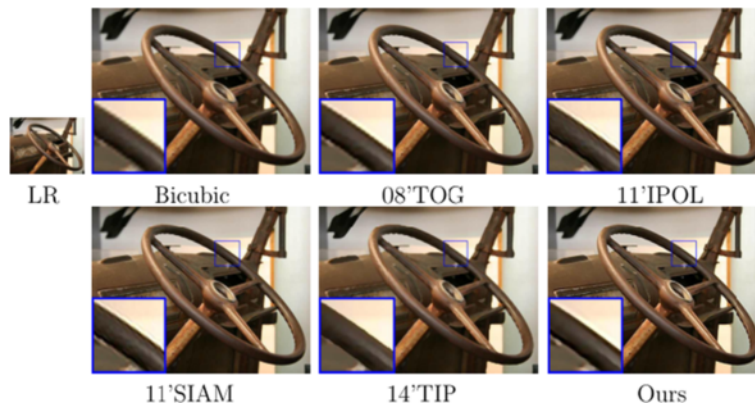
#### 4 Numerical experiments

In this section, we test the proposed method on two kinds of images. One is the low-resolution images without ground truth, and the other is the low-resolution images acquired by downsampling the ground truth images. All experiments are conducted in MATLAB(R2010a) on a laptop of 3.47 GB RAM and Intel(R) Core(TM) i3-2130M CPU: @3.40 GHz.

We make comparisons between the proposed method and some state-of-the-art image super-resolution methods, including bicubic interpolation, a fast upsampling method (“08'TOG” [22] <sup>1</sup>), a two-scale method (“14'TIP” [9] <sup>2</sup>) and two state-of-the-art interpolation methods (“11'IPOL” [7] <sup>3</sup> and “11'SIAM” [8] <sup>4</sup>).

For grayscale image, we apply the proposed algorithm directly. For colored image, we first transform the low-resolution image to YUV color space and then only conduct our algorithm on the Y channel. Images on the U and V channels are upsampled by bicubic method. After acquiring final upsampling image, we transform them from YUV space to RGB space for visual comparisons.

**Parameter selection:** There are four parameters in our algorithm: the size and the deviation of blur kernel, the regularization parameter  $\alpha$ , and the iteration number  $\tau$ . In fact, it is really difficult to choose appropriate parameters for many algorithms. Empirical adjustment is one of



**Fig. 5** Visual results of “wheel” with the upsampling factor of 3. From *top* to *bottom* and from *left* to *right*: the low-resolution image “wheel,” bicubic result, “08'TOG” [22], “11'IPOL” [7], “11'SIAM” [8], “14'TIP” [9], and our result

favorite ways for determining parameters. Thus, we select parameters empirically in our experiments. For the size and the deviation of blur kernel, the regularization parameter  $\alpha$ , we fix them according to different kinds of test images and different values of the upsampling factor (see details at Table 1). In particular, we use the Gaussian kernel similar to that described in “08'TOG” [22]. In addition, we estimate the errors on many images with different iteration numbers, and find that two or three iterations can get the best results. In particular, we set the iteration number  $\tau$  as 2 in all experiments to reduce the computation time.

#### 4.1 Results on the low-resolution images without ground truth

In this section, test images are the low-resolution images without ground truth; thus, it is not available to make quantitative comparisons. We compare the proposed method with bicubic interpolation method, “08'TOG” [22], “11'IPOL” [7], “11'SIAM” [8], and “14'TIP” [9]. The upsampling factors are 2, 3, and 4. In particular, if the dimensions of the estimated high-resolution images acquired by “11'IPOL” [7] and “11'SIAM” [8] exceed  $800 \times 800$ , the images will be cropped. Hence, we do not show these results in Fig. 4, Fig. 6, Fig. 9, and Fig. 10 for the consistency of results.

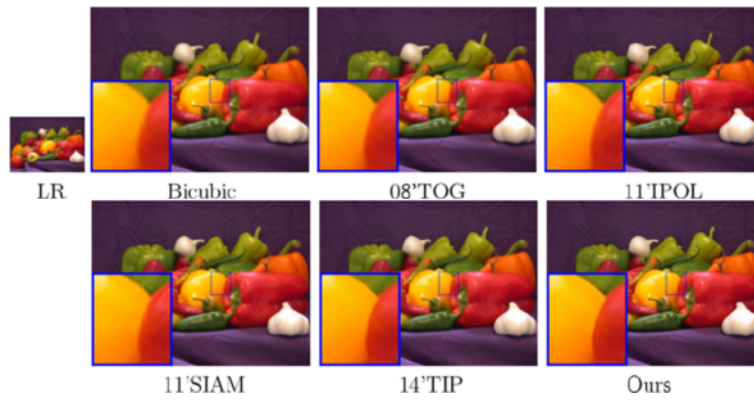
In Figs. 3 and 4, we test the proposed method on the image “flower” and “yacht.” The upsampling factor is 4. From Fig. 3, the results of bicubic interpolation, “11'IPOL” and “11'SIAM” show significant blur effects along the image edge. Although the result by “08'TOG” preserves sharp edges, its luminance is not good enough for visual effects. As shown in this figure, our result can generate sharp edges, while enjoying a better effect of brightness. From Fig. 4, the number of our result is clearer than other results. In addition, after upsampling the image “yacht” with the factor of 4, the resolution comes to  $2048 \times 1920$ . However, this procedure only takes less than 30 s (see

Table 2). It demonstrates that the proposed method can deal with large-scale image resolution problem effectively.

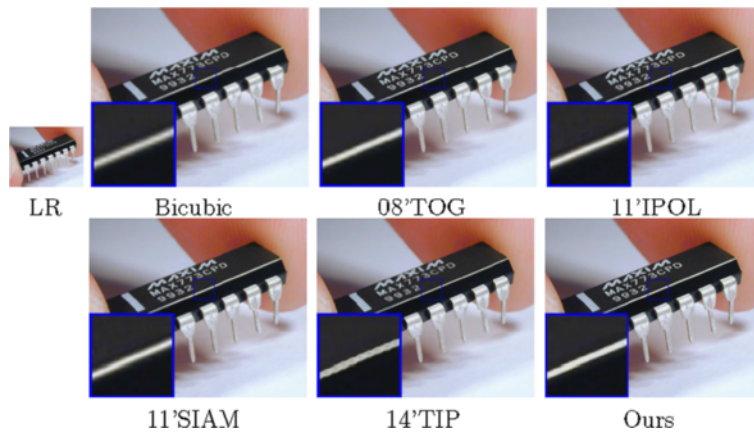
Figures 5, 6, 7 and 8 shows four example images with the upsampling factor of 3. In Fig. 5, there are some blur in the result of bicubic method and some staircases in the result of “08'TOG.” In the results of “11'IPOL” and “11'SIAM,” there are some visual artifacts. Despite preserving sharp edges and enjoying fast running time,



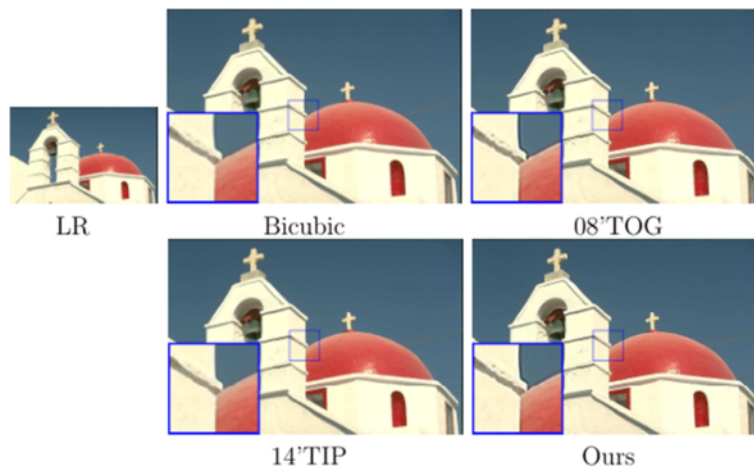
**Fig. 6** Visual results of “comic” with the upsampling factor of 3. From *left* to *right*: the low-resolution image “comic,” bicubic result, “08'TOG” [22], “14'TIP” [9], and our result



**Fig. 7** Visual results of “chilies” with the upsampling factor of 3. From *top to bottom* and from *left to right*: the low-resolution image “chilies,” bicubic result, “08'TOG” [22], “11'IPOL” [7], “11'SIAM” [8], “14'TIP” [9], and our result

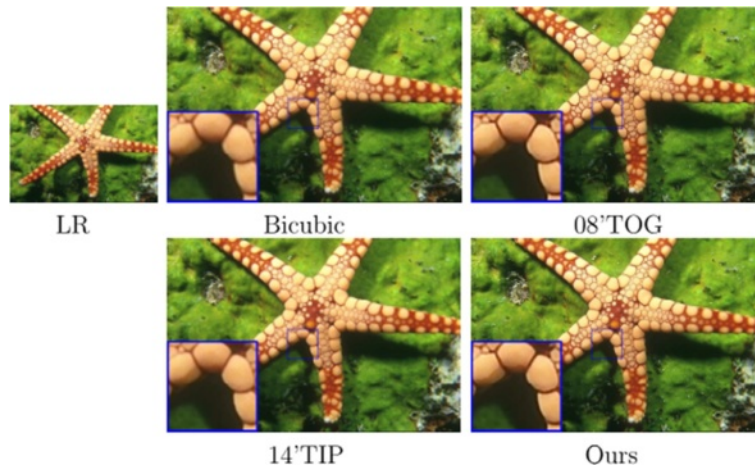


**Fig. 8** Visual results of “number” with the upsampling factor of 3. From *top to bottom* and from *left to right*: the low-resolution image “number,” bicubic result, “08'TOG” [22], “11'IPOL” [7], “11'SIAM” [8], “14'TIP” [9], and our result



**Fig. 9** Visual results of “house” with the upsampling factor of 2. From *top to bottom* and from *left to right*: the low-resolution image “house,” bicubic result, “08'TOG” [22], “14'TIP” [9], and our result





**Fig. 10** Visual results of “starfish” with the upsampling factor of 2. From *top to bottom* and from *left to right*: the low-resolution image “starfish”, bicubic result, “08'TOG” [22], “14'TIP” [9], and our result

the result of “14'TIP” makes the image over-flat. However, our result is able to introduce a better visual result. From Fig. 6, the proposed method provides sharp edges (see headdress and the green background). From Fig. 7, the result of the proposed method performs sharper edges than the results of bicubic method, “08'TOG,” and “11'IPOL.” In Fig. 8, the results of bicubic method, “08'TOG,” “11'IPOL,” and “11'SIAM” show blur effects significantly in the high-resolution images. For the result of “14'TIP” it presents a clear high-resolution image but introduces significant staircases. In particular, the proposed method performs the best visual result comparing to other methods. Furthermore, the proposed method is faster than “08'TOG.” For instance, for the image “comic,” “08'TOG” requires 17.136 s, while the proposed method only needs 5.654 s (see Table 2).

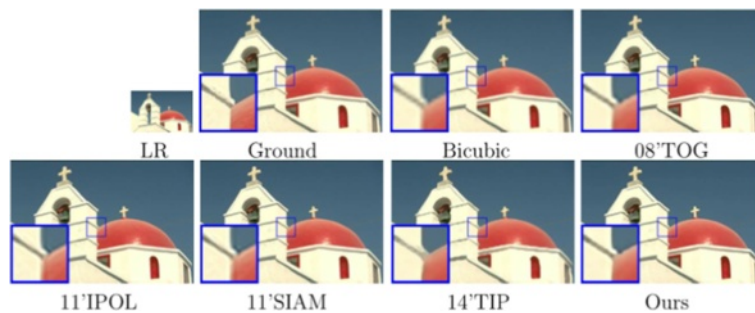
In Figs. 9 and 10, we test our method on images “house” and “starfish” with the upsampling factor of 2. As shown in these figures, the proposed method preserves sharp edges and keeps more details. In addition, the running

time of our method is also much faster than “08'TOG.” For instance, for the image “starfish,” “08'TOG” needs 13.289 s, but our method only need 5.706 s to complete the image upsampling procedure (see Table 2).

#### 4.2 Results on the low-resolution images acquired by downsampling the ground truth images

In this section, the low-resolution images are acquired by downsampling the ground truth images. We provide quantitative comparisons including root-mean-square error (RMSE), peak signal-noise ratio (PSNR), and structure similarity (SSIM) [41].

We mainly compare the proposed method with one reconstruction-based method “08'TOG” and four interpolation-based methods, including bicubic method, “11'IPOL,” “11'SIAM,” and “14'TIP.” In Fig. 11, the test image is “castle” and the upsampling factor is 3. Note that the results of bicubic method and “08'TOG” present blur effects. The results of “11'IPOL” and “11'SIAM” show



**Fig. 11** Visual results of “castle” with the upsampling factor of 3. From *top to bottom* and from *left to right*: the low-resolution image “castle,” the ground truth, bicubic result, “08'TOG” [22], “11'IPOL” [7], “11'SIAM” [8], “14'TIP” [9], and our result

**Table 3** Quantitative comparison in terms of RMSE, PSNR and SSIM

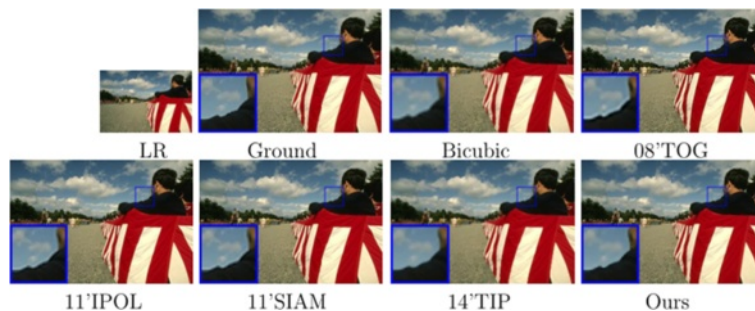
Example(factor)	Index	Bicubic	08'TOG [22]	11'IPOL [7]	11'SIAM [8]	14'TIP [9]	Ours
House(2)	RMSE	6.4742	6.0985	5.4645	5.3947	7.4008	5.1384
	PSNR	31.9071	32.4263	33.3798	33.4915	30.7452	33.9143
	SSIM	0.8831	0.8858	0.8961	0.8969	0.8761	0.8958
House(3)	RMSE	8.9995	7.2818	7.8816	7.8106	9.1071	7.1034
	PSNR	29.0465	30.886	30.1985	30.2771	28.9432	31.1015
	SSIM	0.847	0.8655	0.861	0.8615	0.8554	0.872
House(4)	RMSE	11.018	9.1549	9.8511	9.7658	11.7308	9.0703
	PSNR	27.2887	28.8977	28.2611	28.3366	26.7443	28.9784
	SSIM	0.8169	0.8418	0.8302	0.8311	0.8278	0.8444
Race(2)	RMSE	11.18	11.2381	10.3093	10.1244	12.5453	10.8124
	PSNR	27.1619	27.1169	27.8662	28.0234	26.1612	27.4524
	SSIM	0.6912	0.6911	0.7353	0.7386	0.6536	0.692
Race(3)	RMSE	13.9661	13.0151	13.0624	12.9765	14.1891	13.1136
	PSNR	25.2293	25.8418	25.8103	25.8676	25.0917	25.7764
	SSIM	0.6042	0.6078	0.6407	0.6429	0.5955	0.6171
Race(4)	RMSE	15.7714	14.6136	14.8819	14.8098	15.7557	14.6909
	PSNR	24.1734	24.8357	24.6776	24.7198	24.182	24.7898
	SSIM	0.5524	0.562	0.5804	0.5826	0.5575	0.5761

some visual artifacts. Although the result of “14’TIP” can generate sharp edges, it has worse quantitative results (see Table 3). Furthermore, in this example, the proposed method shows the best RMSE, PSNR, and SSIM.

Figure 12 exhibits the results of image “race” with the upsampling factor of 2. From this figure, we can see that the results of bicubic method and “08'TOG” show some blur effects. The results of “11'IPOL” and “11'SIAM” introduce some artificial contours. The result of “14'TIP” exhibits sharp edges but smoothens the image details. The proposed method provides sharp edges and enjoys competitive quantitative results. More quantitative results are shown in Table 3.

### 5 Conclusions

In this paper, we presented a simple and effective single image super-resolution method. Our method was motivated by a fast image upsampling method, but we differently studied the estimated error of the blur kernel in the proposed method. We believed that the point spread function is not known exactly in the process of image super-resolution. Through applying a fast total variation deconvolution (FTVd) strategy, we took an iterative updating strategy to update the blur kernel and the high-resolution image. In particular, the proposed method could be applied to any upscaling factors without any extra datasets. In addition, we analyzed the parameter



**Fig. 12** Visual results of “race” with the upsampling factor of 2. From top to bottom and from left to right: the low-resolution image “race,” the ground truth, bicubic result, “08'TOG” [22], “11'IPOL” [7], “11'SIAM” [8], “14'TIP” [9], and our result

selection and computation time. Extensive experiments were provided to illustrate the effectiveness of the proposed method.

## Endnotes

<sup>1</sup><http://www.cse.cuhk.edu.hk/~leojia/projects/upsampling/index.html>.

<sup>2</sup><http://www.escience.cn/people/LingfengWang/publication.html>.

<sup>3</sup>[http://demo.ipol.im/demo/g\\_image\\_interpolation\\_with\\_contour\\_stencils/](http://demo.ipol.im/demo/g_image_interpolation_with_contour_stencils/).

<sup>4</sup>[http://demo.ipol.im/demo/g\\_interpolation\\_geometric\\_contour\\_stencils/](http://demo.ipol.im/demo/g_interpolation_geometric_contour_stencils/).

## Competing interests

The authors declare that they have no competing interests.

## Authors' contributions

HY performed the main part of this manuscript. TZH modified the content of the manuscript. LJD carried out the numerical results, and XLZ participated in the discussion. All authors read and approved the final manuscript.

## Acknowledgements

This research is supported by 973 Program (2013CB329404), NSFC (61370147, 61170311), the Fundamental Research Funds for the Central Universities (ZYGX2013Z005).

Received: 27 September 2015 Accepted: 11 July 2016

Published online: 25 July 2016

## References

1. S Baker, T Kanade, Limits on super-resolution and how to break them. *IEEE Trans. Pattern Anal. Mach. Intell.* **24**, 1167–1183 (2002)
2. S Farsiu, MD Robinson, M Elad, P Milanfar, Fast and robust multiframe super resolution. *IEEE Trans. Image Process.* **13**, 1327–1344 (2004)
3. RY Tsai, TS Huang, Multiframe image restoration and registration. *IEEE Trans. Image Process.* **1**, 317–339 (1984)
4. R Keys, Cubic convolution interpolation for digital image processing. *IEEE Trans. Speech Signal Process.* **29**, 1153–1160 (1981)
5. LJ Deng, W Guo, TZ Huang, Single image super-resolution by approximated heaviside functions. *Inform. Sci.* **348**, 107–123 (2016)
6. D Zhang, X Wu, An edge-guided image interpolation algorithm via directional filtering and data fusion. *IEEE Trans. Image Process.* **15**, 2226–2238 (2006)
7. P Getreuer, Image interpolation with contour stencils. *Image Processing On Line* (2011), <http://www.ipol.im/>
8. P Getreuer, Contour stencils: total variation along curves for adaptive image interpolation. *SIAM J. Imaging Sci.* **4**, 954–979 (2011)
9. L Wang, H Wu, C Pan, Fast image upsampling via the displacement field. *IEEE Trans. Image Process.* **23**, 5123–5135 (2014)
10. X Li, MT Orchard, New edge-directed interpolation. *IEEE Trans. Image Process.* **10**, 1521–1527 (2001)
11. LJ Deng, W Guo, TZ Huang, Single image super-resolution via an iterative reproducing kernel hilbert space method. *IEEE Trans. Circuits and Systems for Video Technology* (2015). doi:10.1109/TCSVT.2015.2475895
12. S Borman, RL Stevenson, Super-resolution from image sequences—a review, *Midwest Symposium on Circuits and Systems (MWSCAS)*. Notre Dame, Indiana, pp. 374–378 (1998)
13. S Rhee, MG Kang, Discrete cosine transform based regularized high-resolution image reconstruction algorithm. *Optical Eng.* **38**, 1348–1356 (1999)
14. AK Katsaggelos, KT Lay, NP Galatsanos, A general framework for frequency domain multi-channel signal processing. *IEEE Trans. Image Process.* **2**, 417–420 (1993)
15. S Lertrattanapanich, NK Bose, High resolution image formation from low resolution frames using delaunay triangulation. *IEEE Trans. Image Process.* **11**, 1427–1441 (2002)
16. H Stark, P Oskoui, High-resolution image recovery from image-plane arrays, using convex projections. *J. Optical Soc. Am. A.* **6**, 1715–1726 (1989)
17. D Capel, A Zisserman, Super-resolution enhancement of text image sequences. *Intern. Conf. Pattern Recognit.* **1**, 600–605 (2000)
18. R Fattal, Image upsampling via imposed edge statistics. *ACM Trans. Graphics (TOG)*. **26**(3) (2007). Article No.95
19. T Komatsu, T Igarashi, K Aizawa, T Saito, Very high resolution imaging scheme with multiple different-aperture cameras. *Signal Process. Image Commun.* **5**, 511–526 (1993)
20. A Chambolle, T Pock, A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imaging Vis.* **40**, 120–145 (2011)
21. M Irani, S Peleg, Motion analysis for image enhancement: resolution, occlusion, and transparency. *Journal of visual communication and image representation. J. Visual Commun. Image Repres.* **4**, 324–335 (1993)
22. Q Shan, Z Li, J Jia, CK Tang, Fast image/video upsampling. *ACM Trans. Graphics (TOG)*. **27**, 32–39 (2008)
23. P Chatterjee, S Mukherjee, S Chaudhuri, G Seetharaman, Application of papoulis-gerchberg method in image super-resolution and inpainting. *Comput. J.* **52**, 80–89 (2009)
24. H Takeda, S Farsiu, P Milanfar, Kernel regression for image processing and reconstruction. *IEEE Trans. Image Process.* **16**, 349–366 (2007)
25. X Qinlan, C Hong, C Huimin, Improved example-based single-image super-resolution. *Intern. Congr. Image Signal Process. (CISP)*. **3**, 1204–1207 (2010)
26. C Kim, K Choi, K Hwang, JB Ra, Learning-based superresolution using a multi-resolution wavelet approach, *International Workshop Advance. Image Tech (IWAIT)*, Gangwon-Do, South Korea (2009)
27. C Dong, CC Loy, K He, X Tang, Learning a deep convolutional network for image super-resolution. *European Conference on Computer Vision (ECCV)*, Zurich, Switzerland, pp. 184–199 (2014)
28. J Yang, J Wright, TS Huang, Y Ma, Image super-resolution via sparse representation. *IEEE Trans. Image Process.* **19**, 2861–2873 (2010)
29. K Jia, X Wang, X Tang, Image transformation based on learning dictionaries across image spaces. *IEEE Trans. Pattern Anal. Mach. Intell.* **35**, 367–380 (2013)
30. J Yang, Z Wang, Z Lin, S Cohen, T Huang, Coupled dictionary training for image super-resolution. *IEEE Trans. Image Process.* **21**, 3467–3478 (2012)
31. WT Freeman, TR Jones, EC Pasztor, Example-based super-resolution. *IEEE Comput. Graph. Appl.* **22**, 56–65 (2002)
32. WT Freeman, EC Pasztor, OT Carmichael, Learning low-level vision. *Intern. J. Comput. Vis.* **40**, 25–47 (2000)
33. N Efrat, D Glasner, A Apartsin, B Nadler, A Levin, Accurate blur models vs. image priors in single image super-resolution. *IEEE Intern. Conf. Computer Vision (ICCV)*, Sydney, Australia (2013)
34. Y Wang, JC Yang, W Yin, Y Zhang, A new alternating minimization algorithm for total variation image reconstruction. *SIAM J. Imaging Sci.* **1**, 248–272 (2008)
35. XL Zhao, F Wang, TZ Huang, MK Ng, RJ Plemmons, Deblurring and sparse unmixing for hyperspectral images. *IEEE Trans. Geosci. Remote Sensing.* **51**, 4045–4058 (2013)
36. S Wang, TZ Huang, J Liu, XG Lv, An alternating iterative algorithm for image deblurring and denoising problems. *Commun. Nonlin. Sci. Numer. Simul.* **19**, 617–626 (2014)
37. G Liu, TZ Huang, J Liu, High-order tv l1-based images restoration and spatially adapted regularization parameter selection. *Comput. Math. App.* **67**, 2015–2026 (2014)
38. XL Zhao, F Wang, MK Ng, A new convex optimization model for multiplicative noise and blur removal. *SIAM J. Imaging Sci.* **7**, 456–475 (2014)
39. LJ Deng, H Guo, TZ Huang, A fast image recovery algorithm based on splitting deblurring and denoising. *J. Comput. Appl. Math.* **287**, 88–97 (2015)
40. XL Zhao, W Wang, TY Zeng, TZ Huang, MK Ng, Total variation structured total least squares method for image restoration. *SIAM J. Sci. Comput.* **35**, 1304–1320 (2013)
41. Z Wang, AC Bovik, HR Sheikh, EP Simoncelli, Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* **13**, 600–612 (2004)