



Contents lists available at SciVerse ScienceDirect

J. Vis. Commun. Image R.

journal homepage: [www.elsevier.com/locate/jvcir](http://www.elsevier.com/locate/jvcir)

# Fast image deconvolution using closed-form thresholding formulas of $L_q(q = \frac{1}{2}, \frac{2}{3})$ regularization

Wenfei Cao, Jian Sun, Zongben Xu\*

School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an 710049, China

## ARTICLE INFO

## Article history:

Received 3 April 2012

Accepted 29 October 2012

Available online xxx

## Keywords:

Sparsity

 $L_{\frac{1}{2}}$  regularization $L_{\frac{2}{3}}$  regularization

Variable splitting

Image deconvolution

 $L_0$  regularization $L_1$  regularization

Thresholding formula

## ABSTRACT

In this paper, we focus on the research of fast deconvolution algorithm based on the non-convex  $L_q(q = \frac{1}{2}, \frac{2}{3})$  sparse regularization. Recently, we have deduced the closed-form thresholding formula for  $L_{\frac{1}{2}}$  regularization model (Xu (2010) [1]). In this work, we further deduce the closed-form thresholding formula for the  $L_{\frac{2}{3}}$  non-convex regularization problem. Based on the closed-form formulas for  $L_q(q = \frac{1}{2}, \frac{2}{3})$  regularization, we propose a fast algorithm to solve the image deconvolution problem using half-quadratic splitting method. Extensive experiments for image deconvolution demonstrate that our algorithm has a significant acceleration over Krishnan et al.'s algorithm (Krishnan et al. (2009) [3]). Moreover, the simulated experiments further indicate that  $L_{\frac{2}{3}}$  regularization is more effective than  $L_0, L_{\frac{1}{2}}$  or  $L_1$  regularization in image deconvolution, and  $L_{\frac{1}{2}}$  regularization is competitive to  $L_1$  regularization and better than  $L_0$  regularization.

© 2012 Elsevier Inc. All rights reserved.

## 1. Introduction

Image blur is a common artifact in digital photography caused by camera shake or object movement. Recovering the un-blurred sharp image from the blurry image, which is generally called image deconvolution, has been a fundamental research problem in image processing and computational photography. Image deconvolution algorithms [4–6] can be categorized to blind deconvolution and non-blind deblurring, in which the blur kernel is unknown and known respectively. Tremendous methods have been proposed to estimate the blur kernel. In this work, we focus on the non-blind deblurring problem, i.e., recovering the sharp image from a blurry image given the blur kernel.

Mathematically, blurry image can be modeled as the convolution of an ideal sharp image with a blur kernel and then adding zero mean Gaussian white noise. The degraded process can be modeled as

$$Y = X \otimes k + n \quad (1.1)$$

where  $X$  is the sharp image,  $k$  is a blur kernel and  $n$  is the noise. Image deconvolution aims to recover a high quality image  $X$ , given a blurry image  $Y$ .

The ill-posed nature of this problem implies that additional assumption on  $X$  should be introduced. Recently, many kinds of image priors are discovered and utilized to regularize this ill-posed

inverse problem, such as the total variation [7], nonlocal self-similarity [8–10], sparse prior [11–13] and so on. Especially, the sparsity induced by nonconvex non-convex regularization or the hyper-Laplacian distribution from probabilistic point of view attracts a lot of attention in the community of computer vision [14], machine learning and compressive sensing [15–17]. These prior models give rise to surprising results. For example, Chartrand [17,18] applies non-convex regularization to the Magnetic Resonance Imaging (MRI) reconstruction task, bringing about promising results that only few samples in  $K$ -data space can effectively reconstruct the MRI image.

In this paper, we work on the fast image deconvolution algorithm with non-convex regularization to suppress ringing artifacts and noises. The idea is motivated by Krishnan's work in [3], in which hyper-Laplacian prior of natural image is imposed on the image non-blind deconvolution algorithm, which is equivalent to solving an inverse linear optimization problem with  $L_q$ -norm ( $0 < q < 1$ ) non-convex regularization. Using quadratic splitting framework, one sub-problem is to optimize the non-convex regularization problem:

$$x^* = \underset{x}{\operatorname{argmin}} \{ (x - a)^2 + \lambda |x|^q \}. \quad (1.2)$$

This sub-problem actually is a very special case of the problem proposed by Elad [25] in the context of sparse representation and by Chartrand [15–17] in the setting of compressive sensing. According to their work, from the geometric point of view, this solution is just the intersection point between a hyperplane and a  $L_q(0 < q < 1)$  ball, and when  $q$  goes closer to zero, the solution of this problem

\* Corresponding author.

E-mail addresses: [caowenf2006@163.com](mailto:caowenf2006@163.com) (W. Cao), [jiansun@mail.xjtu.edu.cn](mailto:jiansun@mail.xjtu.edu.cn) (J. Sun), [zbxu@mail.xjtu.edu.cn](mailto:zbxu@mail.xjtu.edu.cn) (Z. Xu).

becomes more sparse. However, from the algebraic point of view, how to fast solve this optimization problem is a challenge due to the non-convexity and the non-smoothness of this problem. In this work, our efforts will focus on two special values,  $\frac{1}{2}$  and  $\frac{2}{3}$ , over the interval  $(0, 1)$ . Traditionally, closed-form hard thresholding [19] and soft-thresholding formulas [20,21] have been proposed to solve this regularization problem when  $q = 0$  and  $q = 1$ . In [3], when  $q = \frac{1}{2}$  or  $\frac{2}{3}$ , Krishnan et al. [3] proposed to solve the above problem by presenting some clever discriminate conditions to compare and select optimal solution from the multiple roots of the first-order derivative equation of the cost function. Although this method makes this problem undertake a major breakthrough, multiple roots should be computed and compared to produce the final solution. A natural question is whether we could derive the closed-form thresholding formulas for non-convex regularization with  $q = \frac{1}{2}$  or  $\frac{2}{3}$  in  $0 < q < 1$ , in parallel to the well-known hard/soft thresholding formulas for  $q = 0$  or 1.

In this work, we will present the closed-form thresholding formulas for non-convex regularization problem in Eq. (1.2) with  $q = \frac{2}{3}$  or  $\frac{1}{2}$ , and apply them to solve the image deconvolution problem. It has been found that the gradients of natural images are distributed as heavy-tailed hyper-Laplacian distribution  $p(x) \propto e^{-k|x|^q}$  with  $0.5 \leq q \leq 0.8$ . In the Bayesian framework, this prior will impose the  $L_q$  norm non-convex regularization for the inverse problem with the formulation in Eq. (1.2). Therefore, developing a fast algorithm for  $L_{\frac{1}{2}}$  or  $L_{\frac{2}{3}}$  regularization problem in the range of  $0.5 \leq q \leq 0.8$  can be expected to be promising in image deconvolution task. The contribution of this work can be summarized as:

- We deduce the closed-form thresholding formula for linear inverse model with  $L_{\frac{2}{3}}$  regularization by deeply analyzing the distribution of roots of the first-order derivative equation of the cost function. Together with our previous work on the close-form thresholding formula for  $L_{\frac{1}{2}}$  regularization in [1,2], these thresholding formulas enable fast and efficient image deconvolution algorithm in the framework of half-quadratic splitting strategy.
- We conduct extensive experiments over a set natural images blurred by eight real blur kernels. The results demonstrate that our algorithm enables a significantly faster speed over Krishnan et al.'s method; Moreover,  $L_{\frac{2}{3}}$  regularization is more effective over  $L_0, L_{\frac{1}{2}}$  or  $L_1$  regularization for image deconvolution, and  $L_{\frac{1}{2}}$  regularization is competitive to  $L_1$  regularization and better than  $L_0$  regularization.

We believe that the closed-form thresholding formulas for  $L_{\frac{2}{3}}$  or  $L_{\frac{1}{2}}$  non-convex regularization are important to machine learning and computer vision communities beyond the application of image deconvolution. That is because this linear inverse problem with non-convex regularization is a general model with wide applications for compressive sensing [16,17], image demosaicing [14], image super-resolution [14], etc. Moreover, theoretically, the closed-form formulas make the theoretical analysis of the non-convex regularization problem possible or easier, which deserves to be investigated in our future work.

The remainder of this paper can be organized as follows. Section 2 will describe the image deconvolution model based on non-convex regularization and its optimization using half-quadratic splitting scheme; In Section 3, we will deduce the thresholding formula for  $L_q (q = \frac{2}{3})$  regularization problem and also introduce our previously proposed thresholding formula for  $L_q (q = \frac{1}{2})$  regularization problem. Then we will present our deconvolution algorithm with  $L_q (q = \frac{1}{2}, \frac{2}{3})$  regularization; In Section 4, we will report the experimental results in both speed and quality; Finally, this paper is concluded in Section 5.

## 2. Image deconvolution based on non-convex regularization

### 2.1. Formulation

Assuming that  $X$  is the original uncorrupted grayscale image with  $N$  pixels;  $Y$  is an image degraded by blur kernel  $k$  and noise  $n$ :  $Y = X \otimes k + n$  (2.1)

Non-blind deconvolution aims to restore the real image  $X$  given the known or estimated blur kernel  $k$ . Due to the ill-posedness of this task, prior information of natural images should be utilized to regularize the inverse problem [13,3]. In this work, we utilize the sparse hyper-Laplacian distribution prior of nature image in the gradient domain [3], i.e.,

$$p(X) \propto e^{-\tau \sum_{j=1}^2 \|X \otimes f_j\|_q^q}, \quad (2.2)$$

where  $\|z\|_q = (\sum_i |z_i|^q)^{\frac{1}{q}}$ ,  $\otimes$  denotes convolution,  $f_1 = [1, -1]$  and  $f_2 = [-1, 1]$  are two first-order derivative filters and  $0 < q < 1$ . From the probabilistic perspective, we seek the MAP (maximum-a-posteriori) estimate of  $X$  in Bayesian framework:  $p(X|Y, k) \propto p(Y|X, k)P(X)$ , the first term is the Gaussian likelihood and the second term is the hyper-Laplacian image prior. Maximizing  $p(X|Y, k)$  is equivalent to minimizing

$$X^* = \underset{X}{\operatorname{argmin}} \left\{ \frac{\lambda}{2} \|X \otimes k - Y\|_F^2 + \sum_{j=1}^2 \|X \otimes f_j\|_q^q \right\} \quad (2.3)$$

where  $\|A\|_F = \sqrt{\sum_{i=1}^M \sum_{j=1}^N a_{ij}^2}$  indicates the Frobenious norm. If assume that  $\mathbf{x}$  and  $\mathbf{y}$  are vectors stretched from  $X$  and  $Y$  column by column, and  $K, F_1$  and  $F_2$  are the matrix form of the filters  $k, f_1$  and  $f_2$  for image convolution, then problem (2.3) can be equivalently represented as

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}} \left\{ \frac{\lambda}{2} \|K\mathbf{x} - \mathbf{y}\|_2^2 + \|F_1\mathbf{x}\|_q^q + \|F_2\mathbf{x}\|_q^q \right\} \quad (2.4)$$

where  $\lambda$  makes a trade-off between the fidelity term and the regularization term. When  $0 \leq q < 1$ ,  $\|F\mathbf{x}\|_q = (\sum_i (F\mathbf{x})_i^q)^{\frac{1}{q}}$  imposes non-convex regularization on the image gradients.

### 2.2. Half-quadratic splitting algorithm

Using the half-quadratic splitting method, Krishnan et al., [3] introduced two auxiliary variables  $\mathbf{u}_1$  and  $\mathbf{u}_2$  and the problem (2.4) can be converted to the following optimization problem

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}} \left\{ \frac{\lambda}{2} \|K\mathbf{x} - \mathbf{y}\|_2^2 + \frac{\beta}{2} \|F_1\mathbf{x} - \mathbf{u}_1\|_2^2 + \frac{\beta}{2} \|F_2\mathbf{x} - \mathbf{u}_2\|_2^2 + \|\mathbf{u}_1\|_q^q + \|\mathbf{u}_2\|_q^q \right\} \quad (2.5)$$

where  $\beta$  is a control parameter. As  $\beta \rightarrow \infty$ , the solution of problem(2.5) converges to that of Eq. (2.4). Minimizing Eqn. (2.5) for a fixed  $\beta$  can be performed by alternating two steps: one sub-problem is to solve  $\mathbf{x}$ , given  $u_1$  and  $u_2$ , which is called  $\mathbf{x}$ -subproblem; the other sub-problem is to solve  $u_1, u_2$ , given  $\mathbf{x}$ , which is called  $\mathbf{u}$ -subproblem.

#### 2.2.1. $\mathbf{x}$ -Subproblem

Given  $\mathbf{u}_1$  and  $\mathbf{u}_2$ , the  $\mathbf{x}$ -subproblem aims to obtain the optimal  $\mathbf{x}$  by optimizing the energy function Eq. (2.5), which is to optimize:

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}} \{ \lambda \|K\mathbf{x} - \mathbf{y}\|_2^2 + \beta \|F_1\mathbf{x} - \mathbf{u}_1\|_2^2 + \beta \|F_2\mathbf{x} - \mathbf{u}_2\|_2^2 \}$$

The subproblem can be optimized by setting the first derivative of the cost function to zero:

$$\left(F_1^T F_1 + F_2^T F_2 + \frac{\lambda}{\beta} K^T K\right) \mathbf{x} = F_1^T \mathbf{u}_1 + F_2^T \mathbf{u}_2 + \frac{\lambda}{\beta} K^T \mathbf{y} \quad (2.6)$$

where  $K\mathbf{x} = X \otimes k$ . Assuming circular boundary conditions, we can apply 2D FFT to efficiently obtain the optimal solution  $\mathbf{x}$  as:

$$IFFT\left(\frac{FFT(F_1)^* \odot FFT(\mathbf{u}_1) + FFT(F_2)^* \odot FFT(\mathbf{u}_2) + \frac{\lambda}{\beta} FFT(K)^* \odot FFT(\mathbf{y})}{FFT(F_1)^* \odot FFT(F_1) + FFT(F_2)^* \odot FFT(F_2) + \frac{\lambda}{\beta} FFT(K)^* \odot FFT(K)}\right) \quad (2.7)$$

where  $*$  denotes the complex conjugate,  $\odot$  denotes the component-wise multiplication, and the division is also performed in component-wise fashion. The fast fourier transform of  $F_1, F_2, K$  can be pre-computed, therefore solving Eq. (2.7) only requires 3 FFTs at each iteration, i.e.,  $FFT(\mathbf{u}_1), FFT(\mathbf{u}_2), IFFT(*)$ .

### 2.2.2. $\mathbf{u}$ -Subproblem

Given a fixed  $\mathbf{x}$ , finding the optimal  $\mathbf{u}_1, \mathbf{u}_2$  can be achieved by optimizing

$$\mathbf{u}_i^* = \underset{\mathbf{u}_i}{\operatorname{argmin}} \left\{ \frac{\beta}{2} \|F_i \mathbf{x} - \mathbf{u}_i\|_2^2 + \|\mathbf{u}_i\|_q^q \right\}$$

where  $i = 1, 2$ . This optimization problem can be decomposed into  $2N$  independent one-dimension  $L_q$  regularization problems:

$$(\mathbf{u}_i^*)_j = \underset{u}{\operatorname{argmin}} \left\{ |u|^q + \frac{\beta}{2} (u - (F_i \mathbf{x})_j)^2 \right\} \quad (2.8)$$

where  $(\cdot)_j$  ( $j = 1, \dots, N$ ) denotes the  $j$ -th component of a vector. It has been derived that the formulation of the closed-form solutions of the above problem are hard thresholding and soft-thresholding when  $q = 0$  and  $1$  respectively.

For  $0 < q < 1$ , it is challenging to derive the closed-form solution of this optimization problem. Krishnan et al. [3] utilized Newton-Raphson method to optimize this problem for  $0 < q < 1$ . Especially for  $q = \frac{1}{2}$  or  $\frac{2}{3}$  cases, some discriminant rules are proposed to find global optimal solution by comparing and selecting from roots of the first order derivative of the cost function in Eq. (2.8). Although it accelerated the optimization procedure without the need of numerous iterations as done by Newton-Raphson method, it still needs to compute and compare multiple roots using some discriminant rules.

In the next section, we will present the closed-form thresholding formulas for the global optimal solution of Eq. (2.8) with  $q = \frac{1}{2}, \frac{2}{3}$ . These formulas not only further speed up the deconvolution algorithm, but also can be easily extended to other applications in signal/image processing, e.g., denoising or super-resolution, since Eq. (2.8) is a general non-convex regularization model in these applications.

## 3. Image deconvolution based on closed-form thresholding formulas of $L_{\frac{1}{2}}, L_{\frac{2}{3}}$ regularization

In this section, we will firstly present the thresholding formulas for  $L_{\frac{1}{2}}, L_{\frac{2}{3}}$  regularization problems. And then, by combining the thresholding formulas and half-quadratic splitting strategy, we propose a fast algorithm for image deconvolution.

We first review our previous work on the closed-form thresholding formula for  $L_{\frac{1}{2}}$  regularization problem [1,2], i.e., to solve:

$$x^* = \underset{x}{\operatorname{argmin}} \{ (x - a)^2 + \lambda |x|^{\frac{1}{2}} \} \quad (3.1)$$

where the variables is scalar values instead of vectors. This optimization problem has the closed-form thresholding formula:

$$x^* = \begin{cases} \frac{2}{3}|a| \left( 1 + \cos\left(\frac{2\pi}{3} - \frac{2\varphi_2(a)}{3}\right) \right) & \text{if } a > p(\lambda) \\ 0 & \text{if } |a| \leq p(\lambda) \\ -\frac{2}{3}|a| \left( 1 + \cos\left(\frac{2\pi}{3} - \frac{2\varphi_2(a)}{3}\right) \right) & \text{if } a < -p(\lambda) \end{cases} \quad (3.2)$$

where

$$\varphi_2(a) = \arccos\left(\frac{\lambda}{8} \left(\frac{|a|}{3}\right)^{-\frac{2}{3}}\right), \quad p(\lambda) = \frac{\sqrt[3]{54}}{4} (\lambda)^{\frac{2}{3}}.$$

Next, we will derive the closed-form thresholding formula for the  $L_{\frac{2}{3}}$  regularization problem.

### 3.1. The thresholding formula for $L_{\frac{2}{3}}$ regularization

The  $L_{\frac{2}{3}}$  regularization model is

$$x^* = \underset{x}{\operatorname{argmin}} \{ f(x) = (x - a)^2 + \lambda |x|^{\frac{2}{3}} \} \quad (3.3)$$

Our aim is to seek the minimum point of  $f(x)$ , denoted as  $x^*$ . In the following, we will first present two lemmas, Lemma 3.1 and Lemma 3.2; And then, based on the two lemmas, we derive the minimum point of  $f(x)$  when  $x \neq 0$  in Lemma 3.3; Finally, by the Lemma 3.1 and Lemma 3.3, we derive the minimum point of  $f(x)$  when  $x \in \mathfrak{R}$  in Theorem 3.4.

**Lemma 3.1.** The minimum point  $x^*$  of  $f(x)$  in Eq. (3.3) satisfies the following properties:

1. If  $a \geq 0$ , then  $x^* \in [0, a]$ ;
2. If  $a < 0$ , then  $x^* \in [a, 0]$ .

**Proof.** We only prove the case (1), and case (2) can be proved in the same way. If assuming  $x^* < 0$ , without the loss of generality, let  $x^* = -M$  ( $M > 0$ ), then we have  $f(x^*) = f(-M) = (-M - a)^2 + \lambda |(-M)|^{\frac{2}{3}} = (M + a)^2 + \lambda |M|^{\frac{2}{3}} > a^2 = f(0)$ . This obviously contradicts the fact that  $x^*$  is a minimum point. On the other hand, if assuming  $x^* > a$  and let  $x^* = a + \Delta$  ( $\Delta > 0$ ), then we have

$$f(x^*) = f(a + \Delta) = \Delta^2 + \lambda |a + \Delta|^{\frac{2}{3}} > \lambda |a|^{\frac{2}{3}} = f(a).$$

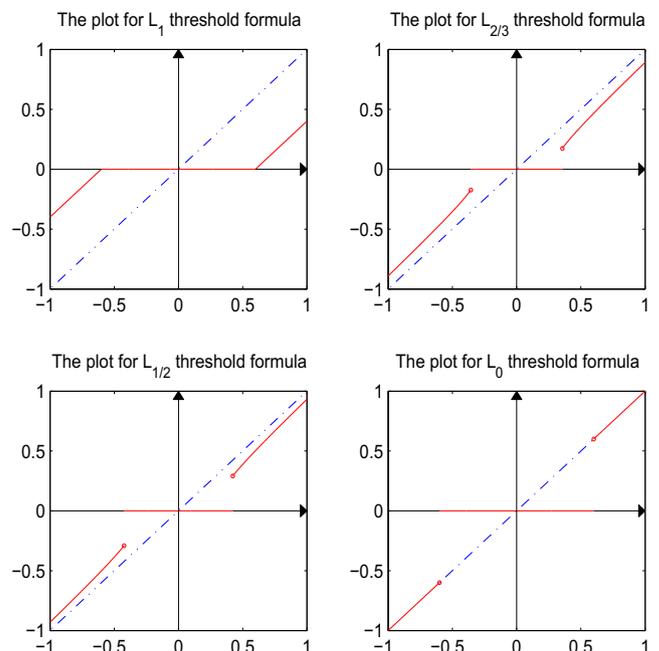


Fig. 1. The plots of the different threshold formulas.



Fig. 2. Test images.

This also leads to contradiction with the fact that  $x^*$  is a minimum point. Hence,  $x^* \in [0, a]$  for case (1).  $\square$

**Lemma 3.2.** Assume that  $\delta_1 = -(A^2 + 2\frac{a}{A})$  and  $\delta_2 = -(A^2 - 2\frac{a}{A})$ , where

$$|A| = \sqrt{\left(\frac{a^2}{2} + \sqrt{\frac{a^4}{4} - \frac{4^3}{27^2}\lambda^3}\right)^{\frac{1}{3}} + \left(\frac{a^2}{2} - \sqrt{\frac{a^4}{4} - \frac{4^3}{27^2}\lambda^3}\right)^{\frac{1}{3}}}$$

and  $|a| > \frac{4}{\sqrt{27}}\lambda^{\frac{3}{4}}$ , then  $\delta_1 * \delta_2 < 0$ .

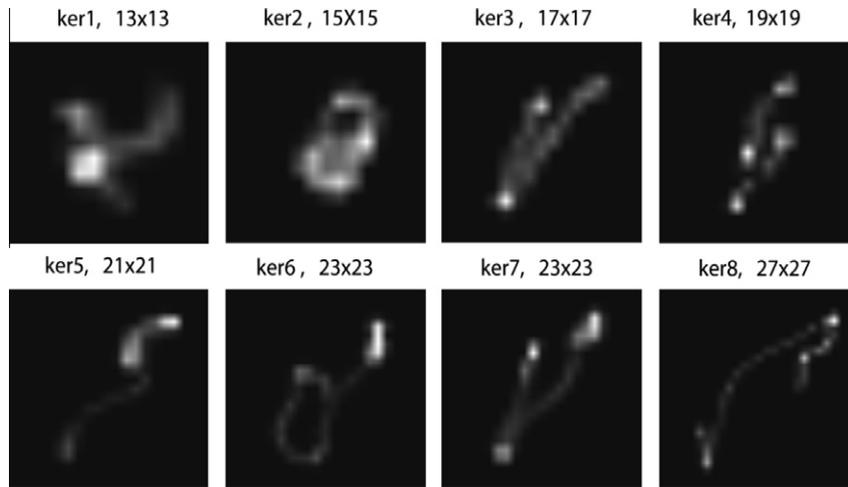


Fig. 3. Test blur kernels.

Table 1  
Comparison for running time with kernel 4(19 × 19).

Size	$DL_{\frac{1}{2}}$ (Ave./Std.)	$OurL_{\frac{1}{2}}$ (Ave./Std.)	Ratio	$DL_{\frac{2}{3}}$ (Ave./Std.)	$OurL_{\frac{2}{3}}$ (Ave./Std.)	Ratio
256 × 256	0.7269/0.0340	0.1452/0.0134	5.0062	0.9009/0.0464	0.2278/0.0192	3.9548
512 × 512	3.5615/0.0084	0.6923/0.0061	5.1444	4.2966/0.0048	1.1865/0.0077	3.6212
1024 × 1024	14.1109/0.0168	2.5477/0.0166	5.5387	17.0178/0.0085	4.4434/0.0110	3.8299
2048 × 2048	55.0141/0.0503	9.6791/0.0284	5.6838	66.9524/0.0730	17.0195/0.0485	3.9339
3000 × 3000	109.8524/0.1082	18.7584/0.0948	5.8562	126.5699/0.2687	34.4123/0.1194	3.6766
3500 × 3500	150.1106/0.3826	25.9526/0.2777	5.7840	172.3184/0.3806	46.8686/0.0987	3.9304
4000 × 4000	196.3966/0.3407	33.5712/0.1605	5.8502	225.7459/1.0865	60.7211/0.1459	3.7178
Ave.	75.6819/0.1344	13.0495/0.0854	<b>5.5519</b>	87.6860/0.2669	23.5542/0.0643	<b>3.7732</b>

Note: Bold value indicates a ratio for the average consuming time of different methods.

**Proof.** The proof of  $\delta_1 * \delta_2 < 0$  is equivalent to the proof of  $A^6 < 4a^2$ . We now prove that  $A^6 < 4a^2$ . Let  $s(t) = t^{1/3}(t \geq 0)$ . It is easy to verify that  $s(t)$  is concave, then we can obtain the inequality  $s(\frac{t_1+t_2}{2}) > \frac{s(t_1)+s(t_2)}{2}$ . By setting  $t_1 = \frac{a^2}{2} + \sqrt{\frac{a^4}{4} - \frac{4^3}{27^2}\lambda^3}$ ,  $t_2 = \frac{a^2}{2} - \sqrt{\frac{a^4}{4} - \frac{4^3}{27^2}\lambda^3}$  (obviously,  $t_1 \neq t_2$  and  $t_1, t_2 \geq 0$ ), we can easily prove that  $A^6 < 4a^2$ , i.e.,  $\delta_1 * \delta_2 < 0$ .  $\square$

**Lemma 3.3.** For  $f(x)$  in Eq. (3.3) where  $\lambda > 0$  and  $a \in \mathfrak{R}$ , if  $x \neq 0$ , then the minimum point of  $f(x)$  can be represented as:

$$\hat{x} = \begin{cases} \left( \frac{|a| + \sqrt{\frac{2|a|}{|a|} - |a|^2}}{2} \right)^3 & \text{if } a > s(\lambda) \\ - \left( \frac{|a| + \sqrt{\frac{2|a|}{|a|} - |a|^2}}{2} \right)^3 & \text{if } a < -s(\lambda) \end{cases}$$

where

$$|a| = \sqrt{\left( \frac{a^2}{2} + \sqrt{\frac{a^4}{4} - \frac{4^3}{27^2}\lambda^3} \right)^{\frac{1}{3}} + \left( \frac{a^2}{2} - \sqrt{\frac{a^4}{4} - \frac{4^3}{27^2}\lambda^3} \right)^{\frac{1}{3}}}, \quad s(\lambda) = \frac{4}{\sqrt{27}}\lambda^{\frac{3}{4}}$$

**Proof.** Please refer to Appendix A for the proof.  $\square$

**Theorem 3.4.** The minimum point of  $f(x)$  in Eq. (3.3) has the following closed-form thresholding formula when  $x \in \mathfrak{R}$ :

$$x^* = \begin{cases} \left( \frac{|a| + \sqrt{\frac{2|a|}{|a|} - |a|^2}}{2} \right)^3 & \text{if } a > p(\lambda) \\ 0 & \text{if } |a| \leq p(\lambda) \\ - \left( \frac{|a| + \sqrt{\frac{2|a|}{|a|} - |a|^2}}{2} \right)^3 & \text{if } a < -p(\lambda) \end{cases} \quad (3.4)$$

where

$$|a| = \frac{2}{\sqrt{3}}\lambda^{\frac{1}{4}} \left( \cosh\left(\frac{\phi}{3}\right) \right)^{\frac{1}{2}}, \quad \phi = \operatorname{arccosh}\left(\frac{27a^2}{16}\lambda^{-\frac{3}{2}}\right), \quad p(\lambda) = \frac{2}{3}(3\lambda^{\frac{3}{4}})^{\frac{1}{3}}$$

**Proof.** Please refer to Appendix B for the proof.  $\square$

In Fig. 1, we plot the closed-form thresholding formulas for the optimal solutions of  $L_q$  regularization problem  $x^* = \operatorname{argmin}\{(x-a)^2 + \lambda|x|^q\}$  when  $q = 0, \frac{1}{2}, \frac{2}{3}, 1$  respectively. The  $x$ -coordinate and  $y$ -coordinate in these sub-figures correspond to  $a$  and  $x^*$  respectively. We can observe that the thresholding curves of  $L_{\frac{1}{2}}, L_{\frac{2}{3}}$  regularization problems lie between the curves of the traditional soft thresholding ( $L_1$  regularization) and hard thresholding ( $L_0$  regularization).

### 3.2. Image deconvolution algorithm

Given the closed-form thresholding formulas for  $L_{\frac{1}{2}}, L_{\frac{2}{3}}$  regularization problems, the optimal solutions of Eq. (2.8) with  $q = \frac{1}{2}, \frac{2}{3}$  in  $u$ -subproblem can be efficiently computed by the thresholding formulas in Eqs. (3.2) and (3.4) with  $x = u$ ,  $\lambda = \frac{2}{\beta}$ ,  $a = (F_i \mathbf{x})_j$ .

Now both the  $x$ -subproblem and  $u$ -subproblem in the quadratic splitting algorithm in Section 2.2 can be efficiently computed in

**Table 2**  
Comparison for running time with kernel 8(27 × 27).

Size	$DL_{\frac{1}{2}}$ (Ave./Std.)	$OurL_{\frac{1}{2}}$ (Ave./Std.)	Ratio	$DL_{\frac{2}{3}}$ (Ave./Std.)	$OurL_{\frac{2}{3}}$ (Ave./Std.)	Ratio
256 × 256	0.7264/0.0389	0.1365/0.0118	5.3216	0.9096/0.0369	0.2350/0.0233	3.8706
512 × 512	3.5751/0.0033	0.6935/0.0043	5.1552	4.3552/0.0551	1.1877/0.0024	3.6669
1024 × 1024	14.1563/0.0165	2.5433/0.0124	5.5661	17.0849/0.0377	4.4553/0.0134	3.8347
2048 × 2048	55.2309/0.0637	9.6818/0.0253	5.7046	67.1039/0.0482	17.0219/0.0325	3.9422
3000 × 3000	110.3661/0.1105	18.7466/0.0501	5.8873	126.8374/0.1049	34.5268/0.1721	3.6736
3500 × 3500	150.4214/0.2632	25.8423/0.0951	5.8207	172.8680/0.2739	47.0590/0.0900	3.6734
4000 × 4000	196.7216/0.3144	33.5025/0.1168	5.8718	225.7242/0.7428	61.1651/0.1594	3.6904
Ave.	75.8854/0.0704	13.0209/0.0451	<b>5.6182</b>	87.8405/0.1856	23.6644/0.0704	<b>3.7646</b>

Note: Bold value indicates a ratio for the average consuming time of different methods.

**Table 3**  
Comparison for different images with kernel 7.

Images	Blurry	$L_1$	$L_{\frac{1}{2}}$	$L_{\frac{2}{3}}$	$L_0$
a(512 × 512)	20.68	26.88	<b>26.90</b>	26.77	25.38
b(512 × 512)	20.47	30.54	<b>30.63</b>	30.44	28.96
c(512 × 512)	20.43	29.56	<b>29.70</b>	29.57	27.81
d(512 × 512)	22.87	32.41	<b>32.42</b>	32.19	30.54
e(512 × 512)	22.40	<b>30.55</b>	30.51	30.34	29.56
f(512 × 512)	21.79	30.37	<b>30.45</b>	30.31	28.94
g(1608 × 1624)	13.67	25.39	<b>25.45</b>	25.27	21.90
h(1554 × 1383)	26.93	34.58	<b>34.59</b>	34.50	33.80
i(1362 × 1263)	23.67	32.64	<b>33.14</b>	33.09	30.31
j(1024 × 1341)	15.39	26.12	<b>26.43</b>	26.35	22.14
k(1308 × 1197)	18.47	26.03	<b>26.24</b>	26.21	24.14
l(1000 × 667)	18.34	27.28	<b>27.66</b>	27.67	25.08
m(886 × 886)	22.80	30.35	<b>30.70</b>	30.64	28.69
n(1246 × 1119)	17.57	26.36	<b>26.48</b>	26.35	23.89
o(1413 × 1413)	20.38	30.82	<b>31.13</b>	30.98	28.31
p(1284 × 1380)	18.94	31.62	<b>31.89</b>	31.63	28.23
q(1600 × 1200)	14.16	23.54	<b>23.75</b>	23.67	20.18
r(1693 × 1084)	17.77	27.91	<b>28.14</b>	28.02	25.36
s(1024 × 675)	17.36	27.16	<b>27.63</b>	27.65	24.50
t(1600 × 1200)	18.89	25.98	<b>26.07</b>	25.97	24.45

Note: Bold values indicate the highest PSNR value.

closed-form formulation, then the final algorithm for image deconvolution is shown in Algorithm 1.

#### 4. Experiments

In this section, we will conduct several groups of experiments to demonstrate that the proposed deconvolution algorithm enables significantly faster speed over Krishnan et al's algorithm [3]. Moreover, by extensive experiments, we will show that  $L_{\frac{2}{3}}$  regularization is more effective for image deconvolution than  $L_0$ ,  $L_{\frac{1}{2}}$  or  $L_1$  regularization, and  $L_{\frac{2}{3}}$  regularization is competitive to  $L_1$  regularization and better than  $L_0$  regularization.

##### 4.1. Experiment setting

Our test natural images are collected from two sources: (1) the standard test images for image processing with size of 521 × 512; (2) the high-resolution images from web site of <http://www.flickr.com/>. The images from the second source are commonly with larger resolutions to test the ability for our algorithm to handle large images. We list all the test images in Fig. 2. All the test images are blurred by real-world camera shake kernels from [22], and the blur kernels are shown in Fig. 3 (the images are scaled for better illustration). To better simulate the real-captured blurry image, we also add Gaussian noises with standard deviation of 0.01 to the blurry image and followed by quantization to 255 discrete values. The PSNR defined as  $10\log_{10}\frac{255^2}{MSE(\mathbf{x})}$  is employed to evaluate the deconvolution performance, where  $\mathbf{x}$  is the deconvolution result and  $MSE(\mathbf{x})$  denotes the mean square error between  $\mathbf{x}$  and the

**Table 4**  
Comparison for different images with kernel 8.

Images	Blurry	$L_1$	$L_{\frac{1}{2}}$	$L_{\frac{2}{3}}$	$L_0$
a(512 × 512)	19.05	26.39	<b>26.42</b>	26.33	25.18
b(512 × 512)	19.39	29.29	<b>29.47</b>	29.38	27.78
c(512 × 512)	19.23	28.69	<b>28.89</b>	28.83	27.05
d(512 × 512)	20.33	30.83	<b>30.94</b>	30.84	29.25
e(512 × 512)	20.93	29.55	<b>29.59</b>	29.47	28.60
f(512 × 512)	20.08	29.44	<b>29.59</b>	29.49	28.05
g(1608 × 1624)	12.75	24.43	<b>24.64</b>	24.58	21.00
h(1554 × 1383)	25.65	<b>33.81</b>	33.73	33.54	33.04
i(1362 × 1263)	21.86	32.35	33.05	<b>33.07</b>	30.01
j(1024 × 1341)	14.46	25.97	26.43	<b>26.44</b>	21.65
k(1308 × 1197)	17.43	26.17	<b>26.46</b>	<b>26.46</b>	23.92
l(1000 × 667)	17.64	27.58	28.10	<b>28.15</b>	24.69
m(886 × 886)	22.20	30.12	<b>30.52</b>	30.46	28.34
n(1246 × 1119)	17.05	26.09	<b>26.31</b>	26.25	23.44
o(1413 × 1413)	19.54	29.61	<b>30.12</b>	30.10	27.08
p(1284 × 1380)	17.59	30.12	<b>30.71</b>	30.68	26.95
q(1600 × 1200)	13.58	23.55	<b>23.82</b>	23.80	19.93
r(1693 × 1084)	17.23	27.30	27.72	<b>27.73</b>	24.49
s(1024 × 675)	16.50	27.45	28.14	<b>28.26</b>	23.97
t(1600 × 1200)	18.54	26.15	<b>26.29</b>	26.22	24.23

Note: Bold values indicate the highest PSNR value.

ground-truth high quality image. In our implementation, edge tapering operation is utilized to reduce the possible boundary artifacts. To compare the best potential performance of different regularization algorithms, we set  $\beta_{inc} = \sqrt{2}$  and  $\lambda = \frac{2}{\beta}$  to the optimal value in a range of values with best PSNR performance as in [3]. Our experiments are executed using Matlab software on desktop computer with 2.51 GHz AMD CPU (dual core) and 1.87 GB RAM.

#### Algorithm 1. Fast Image Deconvolution Using Closed-Form Thresholding Formulas of $L_q(q = \frac{1}{2}, \frac{2}{3})$ Regularization

**Input:** Blurred image  $\mathbf{y}$ ; blur kernel  $k$ ; regularization weight

$$\lambda; q = \frac{1}{2} \text{ or } \frac{2}{3}; \beta_0, \beta_{inc}, \beta_M;$$

maximal number of outer iterations  $T$ ;

number of inner iterations  $J$ .

Step 1: Initialize iter = 0,  $\mathbf{x} = \mathbf{y}$  and  $\beta = \beta_0$ , pre-compute constant terms in Eq. (2.7).

Step 2: **repeat**

iter = iter + 1.

**for**  $i = 1$  to  $J$  **do**

x-subproblem: optimize  $\mathbf{x}$  according to Eq. (2.7).

u-subproblem: optimize  $\mathbf{u}_1, \mathbf{u}_2$  according to Eq. (3.2)

or (3.4) when  $q = \frac{1}{2}$  or  $\frac{2}{3}$ .

**endfor**

$$\beta = \beta_{inc} * \beta.$$

**until**  $\beta > \beta_M$  or iter >  $T$ .

**Output:**  $\mathbf{x}$ .

**Table 5**  
Average results for different regularization.

Ave.	Blurry	$L_1$	$L_{\frac{2}{3}}$	$L_{\frac{1}{2}}$	$L_0$
ker1(13 × 13)	22.5621	30.5138	<b>30.5250</b>	30.3146	28.2054
ker2(15 × 15)	22.2905	29.1895	<b>29.3143</b>	29.1567	26.9743
ker3(17 × 17)	21.7455	29.0075	<b>29.2680</b>	29.1865	26.6380
ker4(19 × 19)	22.2555	29.3000	<b>29.6125</b>	29.5735	26.7215
ker5(21 × 21)	18.8620	30.6030	<b>30.7085</b>	30.5335	28.0740
ker6(23 × 23)	19.7125	29.4790	<b>29.5740</b>	29.4305	27.4645
ker7(23 × 23)	19.6490	28.8050	<b>28.9955</b>	28.8810	26.6085
ker8(27 × 27)	18.5515	28.2445	<b>28.5470</b>	28.5040	25.9325

Note: Bold values indicate the highest PSNR value.

**Table 6**  
Comparison for 8 kernels with image o.

Image o	Blurry	$L_1$	$L_{\frac{2}{3}}$	$L_{\frac{1}{2}}$	$L_0$
ker1(13 × 13)	23.97	33.38	<b>33.41</b>	33.10	30.47
ker2(15 × 15)	23.39	31.57	<b>31.83</b>	31.62	28.96
ker3(17 × 17)	22.26	30.87	<b>31.21</b>	31.08	28.34
ker4(19 × 19)	23.34	30.87	<b>31.37</b>	31.34	28.03
ker5(21 × 21)	19.57	33.14	<b>33.31</b>	33.04	30.12
ker6(23 × 23)	20.51	31.21	<b>31.51</b>	31.38	28.86
ker7(23 × 23)	20.38	30.82	<b>31.13</b>	30.98	28.31
ker8(27 × 27)	19.54	29.59	<b>30.10</b>	30.08	27.08

Note: Bold values indicate the highest PSNR value.

**Table 7**  
Comparison for 8 kernels with image p.

Image p	Blurry	$L_1$	$L_{\frac{2}{3}}$	$L_{\frac{1}{2}}$	$L_0$
ker1(13 × 13)	23.64	<b>34.23</b>	34.09	33.66	30.76
ker2(15 × 15)	22.87	32.19	<b>32.36</b>	32.07	28.94
ker3(17 × 17)	21.63	30.97	<b>31.44</b>	31.33	27.84
ker4(19 × 19)	22.60	31.23	<b>31.81</b>	31.78	27.73
ker5(21 × 21)	17.91	33.56	<b>33.71</b>	33.38	29.91
ker6(23 × 23)	19.07	32.39	<b>32.56</b>	32.29	29.38
ker7(23 × 23)	18.94	31.62	<b>31.89</b>	31.63	28.23
ker8(27 × 27)	17.59	30.14	<b>30.72</b>	30.66	26.96

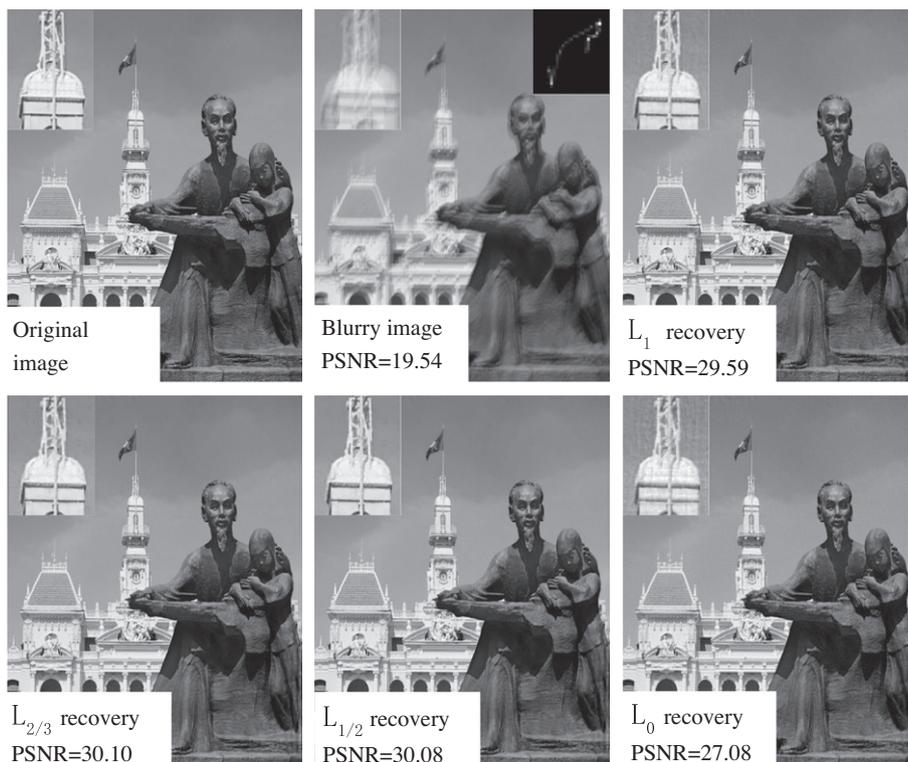
Note: Bold values indicate the highest PSNR value.

#### 4.2. Comparison for speed

In this experiment, we will evaluate the speed of our algorithm compared to the Krishnan's algorithm (without the look-up table technique) [3]. It has been shown in [3] that the speed of other methods such as re-weighted method is slower than Krishnan's algorithm. We test the algorithms on images with varying resolutions. Our proposed deconvolution algorithms using  $L_{\frac{1}{2}}$  and  $L_{\frac{2}{3}}$  regularization are denoted as Our $L_{\frac{1}{2}}$  and Our $L_{\frac{2}{3}}$  respectively whereas the corresponding algorithms proposed in [3] are denoted as  $DL_{\frac{1}{2}}$  (without the look-up table technique) and  $DL_{\frac{2}{3}}$  (without the look-up table technique) respectively. Table 1 exhibits for kernel 4 the average result and the standard deviation of ten times experiment with different resolutions, and Table 2 for kernel 8. From Table 1 and Table 2, we find that our  $L_{\frac{1}{2}}$  algorithm is roughly 5.5 times faster than Krishnan's  $L_{\frac{1}{2}}$  algorithm, and our  $L_{\frac{2}{3}}$  algorithm is roughly 3.7 times faster than Krishnan's  $L_{\frac{2}{3}}$  algorithm in average, indicating that the closed-form thresholding formula significantly speed up the deconvolution algorithm in the framework of splitting quadratic algorithm. Of course, we can further exploit other engineering technologies like look-up table or high performance computing platform like GPU (Graphic Processing Units) to further accelerate our algorithm. Moreover, from Table 1 and Table 2, the small standard deviations and the similar acceleration results for two different sizes of kernels manifest that the acceleration speed is stable.

#### 4.3. Evaluation for different regularization

To compare the performance of different regularization algorithms, we conduct eight groups of experiments for different kernels. For each kernel, we evaluate the deconvolution performance over the 20 test images shown in Fig. 2. Because of the space restriction, we only list two groups of recovery results, as showed in Table 3 and Table 4. From Table 3 and Table 4, we find that, first, the deconvolution results using  $L_{\frac{2}{3}}$  regularization outperform over those by  $L_0$ ,  $L_{\frac{1}{2}}$  or  $L_1$  regularization in terms of PSNR values; Second,



**Fig. 4.** The deconvolution results by different regularization algorithms for image o.



Fig. 5. The deconvolution results by different regularization algorithms for image p.

Table 8  
The triangle and hyperbolic expression for the roots of cubic equation

$x^3 + 3px + 2q = 0, \quad p \neq 0, \quad \text{let } r = \text{sgn}(q)\sqrt{ p }$		
$p < 0$		$p > 0$
$q^2 + p^3 < 0$	$q^2 + p^3 > 0$	
$\cos \varphi = \frac{q}{p^{\frac{3}{2}}}$	$\cosh \varphi = \frac{q}{p^{\frac{3}{2}}}$	$\sinh \varphi = \frac{q}{p^{\frac{3}{2}}}$
$x_1 = -2r \cos \frac{\varphi}{3}$	$x_1 = -2r \cosh \frac{\varphi}{3}$	$x_1 = -2r \sinh \frac{\varphi}{3}$
$x_2 = 2r \cos(\frac{\varphi}{3} - \frac{\varphi}{3})$	$x_2 = r \cosh \frac{\varphi}{3} + i\sqrt{3}r \sinh \frac{\varphi}{3}$	$x_2 = r \sinh \frac{\varphi}{3} + i\sqrt{3}r \cosh \frac{\varphi}{3}$
$x_3 = 2r \cos(\frac{\varphi}{3} + \frac{\varphi}{3})$	$x_3 = r \cosh \frac{\varphi}{3} - i\sqrt{3}r \sinh \frac{\varphi}{3}$	$x_3 = r \sinh \frac{\varphi}{3} - i\sqrt{3}r \cosh \frac{\varphi}{3}$

the  $L_{\frac{2}{3}}$  regularization is competitive to  $L_1$  regularization and better than  $L_0$  regularization for image deconvolution. We also present the average PSNR results over all the test images for each kernel in Table 5. From Table 5, we can also draw the consistent conclusions though the blur kernels are different in both shape and resolution.

To further test the performance of our algorithm, we evaluate the deconvolution results for each image over eight different kernels. Similarly, we only list two groups of experiment over two test images o and p, the results are shown in Table 6 and Table 7. From Table 6 and Table 7, we can derive the same conclusions. We show two of the deconvolution results in Fig. 4 and Fig. 5, it is shown that the deblurred images using  $L_q (q = \frac{1}{2}, \frac{2}{3})$  regularization algorithm is clearly with higher visual quality with less noises/ringing artifacts compared to  $L_0$  or  $L_1$  regularization algorithm.

In summary, extensive experiments demonstrate that our deconvolution algorithm with  $L_q (q = \frac{1}{2}, \frac{2}{3})$  regularization enables significantly faster speed over Krishnan's algorithm [3], while  $L_{\frac{2}{3}}$  regularization outperforms over  $L_0, L_{\frac{1}{2}}$  or  $L_1$  regularization and  $L_{\frac{1}{2}}$  regularization is competitive to  $L_1$  regularization and better than  $L_{\frac{2}{3}}$  regularization.

### 5. Conclusion and future work

In this paper, we derived the closed-form thresholding formula for  $L_{\frac{2}{3}}$  regularization problem. Based on this thresholding formula together with our previously derived thresholding formula for  $L_{\frac{1}{2}}$  regularization problem, we proposed a fast deconvolution algorithm using half quadratic splitting strategy. Extensive experiments demonstrate that our algorithm significantly speeds up the previous deconvolution algorithm in the same framework. And we also justified that  $L_{\frac{2}{3}}$  regularization is more powerful than  $L_1, L_{\frac{1}{2}}$  or  $L_0$  regularization, and  $L_{\frac{1}{2}}$  regularization is competitive to  $L_1$  regularization and better than  $L_0$  regularization for image deconvolution.

The derived thresholding formula in this work provides an effective way to optimize the non-convex regularization problem using closed-form formulation. The  $L_{\frac{2}{3}}$  regularization problem has wide applications beyond the image deconvolution, e.g., compressive sensing, super-resolution, denoising, etc. On the other hand, the derived thresholding formula can be extended to solve more complex regularization problem

$$\vec{x}^* = \underset{\vec{x}}{\text{argmin}} \{ \|A\vec{x} - \vec{y}\|^2 + \lambda \|\vec{x}\|_q^q \} \tag{5.1}$$

where  $A$  is a matrix commonly composed of a set of basis in its columns, and  $\vec{x}$  and  $\vec{y}$  are vectors of variables. This optimization problem is different to the problem in Eq. (1.2) that matrix  $A$  is introduced which makes vector of variables in  $\vec{x}$  dependent on each other. This optimization problem has wide applications in image/signal processing such as dictionary learning [23], image restoration [12], etc. It can be fast optimized by iterative thresholding algorithm:  $\vec{x}_{k+1} = T_{\lambda t}(\vec{x}_k - 2tA^T(A\vec{x}_k - \vec{y}))$ ,  $t$  is an appropriate stepsize,  $T$  is the hard, soft and half thresholding operator when  $q = 0, 1, \frac{1}{2}$  respectively [19,21,1,2]. Obviously, the proposed thresholding formula for  $L_{\frac{2}{3}}$  regularization can be incorporated to the iterative thresholding algorithm as the thresholding operator to solve the Eq. (5.1) when  $q = \frac{2}{3}$ .

In the future work, we are interested in the analysis of the convergence of our deconvolution algorithm and plan to extend the applications of the thresholding formulas of  $L_{\frac{1}{3}}$  and  $L_{\frac{2}{3}}$  regularization to other related problems in image processing and machine learning.

**Acknowledgements**

We would like to thank Doctor J. X. Jia and Q. Zhao for many helpful suggestions. This work was supported by the National 973 Programming (2013CB329404), the Key Program of National Natural Science Foundation of China (Grant No. 11131006), and the National Natural Science Foundations of China (Grant No. 61075054).

**Appendix A. The proof of Lemma 3.3**

**Proof.** We will find the minimum point of  $f(x)$  on  $x \neq 0$  by seeking and analyzing the roots of equation  $f'(x) = 0$  on  $x \neq 0$ . Since  $f'(x) = 2(x - a) + \frac{2}{3}\lambda \frac{\text{sign}(x)}{|x|^{\frac{1}{3}}}$ , we obtain that:

$$x|x|^{\frac{1}{3}} - a|x|^{\frac{1}{3}} + \frac{\lambda}{3}\text{sign}(x) = 0 \quad (x \neq 0) \tag{A.1}$$

Assuming that  $|x| = y^3$ , (A.1) can be reduced as the following two cases:

- case1:** for  $x > 0, y^4 - ay + \frac{\lambda}{3} = 0$ ;
- case2:** for  $x < 0, -y^4 - ay - \frac{\lambda}{3} = 0$ .

Let  $g(x) = x|x|^{\frac{1}{3}} - a|x|^{\frac{1}{3}} + \frac{\lambda}{3}\text{sign}(x)$ ,  $h_1(y) = y^4 - ay + \frac{\lambda}{3}$  and  $h_2(y) = y^4 + ay + \frac{\lambda}{3}$ . In the following, for case 1, we seek the positive minimum point of  $f(x)$  by exploring the zero point distribution of  $h_1(y)$ . And for case 2, we seek the negative minimum point of  $f(x)$  by the symmetry relationship of  $h_2(y)$  and  $h_1(y)$ , i.e.,  $h_2(y) = h_1(-y)$ .

1. We first analyze case 1. By Lemma 3.1, in order to seek the minimum point of  $f(x)$  on  $x > 0$ , it suffices to consider the case of  $a \geq 0$ . Since for  $x > 0, g(x) = g(y^3) = h_1(y)$ , we just need to seek the positive root  $\hat{y}$  of  $h_1(y)$  satisfying:

$$h_1(\hat{y}) = 0; h_1(y) < 0 \text{ when } y < \hat{y}; h_1(y) > 0 \text{ when } y > \hat{y}. \tag{A.2}$$

where  $y$  is near to  $\hat{y}$ . We next investigate the root distribution of  $h_1(y)$  by analyzing its derivative. Since  $h_1'(y) = 4y^3 - a$ , it is easy to verify that when  $y < \sqrt[3]{\frac{a}{4}}, h_1(y)$  monotonically decreases and when  $y > \sqrt[3]{\frac{a}{4}}, h_1(y)$  monotonically increases, which means that  $y = \sqrt[3]{\frac{a}{4}}$  is the unique minimum point of  $h_1(y)$ . Since  $h_1''(y) = 12y^2 \geq 0, h_1(y)$  is a convex function. The root distribution for  $h_1(y)$  has three cases:

- case(a):**  $h_1(y) = 0$  has no root. This means that  $h_1(\sqrt[3]{\frac{a}{4}}) > 0$ , then we can get  $(\sqrt[3]{\frac{a}{4}})^4 - a(\sqrt[3]{\frac{a}{4}}) + \frac{\lambda}{3} > 0$ , i.e.,  $a < \frac{4}{\sqrt{27}}\lambda^{\frac{3}{2}}$ ;
- case(b):**  $h_1(y) = 0$  has one unique real root. This means that  $h_1(\sqrt[3]{\frac{a}{4}}) = 0$ , therefore  $a = \frac{4}{\sqrt{27}}\lambda^{\frac{3}{2}}$ . In this case, however,  $\hat{y} = \sqrt[3]{\frac{a}{4}}$  does not satisfy Eq. (A.2), so  $\hat{y} = \sqrt[3]{\frac{a}{4}}$  is a saddle point rather than a minimum point;
- case(c):**  $h_1(y) = 0$  has two real roots. This means that  $h_1(\sqrt[3]{\frac{a}{4}}) < 0$ , therefore  $a > \frac{4}{\sqrt{27}}\lambda^{\frac{3}{2}}$ . In this case,  $h_1(y)$  has two different roots  $y_1, y_2 (y_1 < y_2)$ , and only  $y_2$  satisfies Eq. (A.2) that corresponds to minimum point of  $f(x)$ . In the following, we will seek  $y_2$  by method of undetermined coefficients. Assume that

$$h_1(y) = y^4 - ay + \frac{\lambda}{3} = (y^2 + Ay + B)(y^2 + Cy + D) \text{ where } A, B, C, D \in \mathfrak{R}. \tag{A.3}$$

By expansion and comparison, we get

$$A + C = 0 \tag{A.4}$$

$$B + D + AC = 0 \tag{A.5}$$

$$AD + BC = -a \tag{A.6}$$

$$BD = \frac{\lambda}{3} \tag{A.7}$$

From (A.4), we get  $C = -A$ . By substituting  $C = -A$  into (A.5), (A.6), we get

$$B + D - A^2 = 0 \tag{A.8}$$

$$AD - AB = -a \tag{A.9}$$

(i) when  $A = 0$ , from (A.4) and (A.6), we get  $C = A = a = 0, a = 0$  obviously contradicts  $a > \frac{4}{\sqrt{27}}\lambda^{\frac{3}{2}}$ , so (i) never occurs.

(ii) when  $A \neq 0$ ,

$$B + D = A^2 \tag{A.10}$$

$$-B + D = -\frac{a}{A} \tag{A.11}$$

we can further obtain that,

$$B = \frac{(A^2 + \frac{a}{A})}{2} \tag{A.12}$$

$$D = \frac{(A^2 - \frac{a}{A})}{2} \tag{A.13}$$

By substituting (A.12), (A.13) to (A.7), we get  $\frac{(A^2 + \frac{a}{A})}{2} * \frac{(A^2 - \frac{a}{A})}{2} = \frac{\lambda}{3}$ . And still, by reduction and rearrangement, we obtain  $A^6 - \frac{4}{3}\lambda A^2 - a^2 = 0$ . Let  $M = A^2$ , we get

$$M^3 - \frac{4}{3}\lambda M - a^2 = 0 \tag{A.14}$$

From the root discriminant formula of the triple equation, we get

$$\Delta = \left(\frac{q}{2}\right)^2 + \left(\frac{p}{3}\right)^3 = \left(-\frac{a^2}{2}\right)^2 + \left(-\frac{4}{9}\lambda\right)^3 = \left(\frac{a^4}{4} - \frac{4^3}{9^3}\lambda^3\right)$$

where  $q = -a^2, p = -\frac{4}{3}\lambda$ . Since  $a > \frac{4}{\sqrt{27}}\lambda^{\frac{3}{2}}, \Delta > 0$ . Hence, Eq. (A.14) only have one real root. According to the Cardan formula for cubic equation, we get the root of Eq. (A.14) as  $M = A^2 =$

$$\left(\frac{a^2}{2} + \sqrt{\frac{a^4}{4} - \frac{4^3}{27^2}\lambda^3}\right)^{\frac{1}{3}} + \left(\frac{a^2}{2} - \sqrt{\frac{a^4}{4} - \frac{4^3}{27^2}\lambda^3}\right)^{\frac{1}{3}}.$$

Right now, by (A.3), (A.4), (A.12), (A.13) and  $h_1(y) = 0$ , we get

$$y^2 + Ay + \frac{(A^2 + \frac{a}{A})}{2} = 0 \tag{A.15}$$

$$y^2 - Ay + \frac{(A^2 - \frac{a}{A})}{2} = 0 \tag{A.16}$$

From the root discriminant formula of the quadratic equation, we get

$$\delta_1 = -(A^2 + 2\frac{a}{A}) \tag{A.17}$$

$$\delta_2 = -(A^2 - 2\frac{a}{A}) \tag{A.18}$$

The following is to seek the real root of equation (A.15), (A.16) by Lemma 3.2.

When  $A > 0$ , due to  $a > \frac{4}{\sqrt{27}}\lambda^{\frac{3}{2}}$ , we can obtain  $\delta_1 < 0$ . By Lemma 3.2,  $\delta_1 * \delta_2 < 0$ , hence,  $\delta_2 > 0$ . Therefore, (A.15) has no real roots, and (A.16) has two different real roots:  $y_1 = \frac{A - \sqrt{\frac{2a}{A} - A^2}}{2}, y_2 = \frac{A + \sqrt{\frac{2a}{A} - A^2}}{2}$ . Because we need the root of  $h_1(y)$  that satisfies Eq. (A.2),  $y_2$  is the root what we seek and  $y_1$  is discarded.

When  $A < 0$ , due to  $a > \frac{4}{\sqrt{27}}\lambda^{\frac{3}{2}}$ , we can obtain  $\delta_2 < 0$ . since  $\delta_1 * \delta_2 < 0$ , hence  $\delta_1 > 0$ . In this case, (A.16) has no real roots, and (A.15) has two different real roots,  $y_3 = \frac{-A - \sqrt{\frac{2a}{A} - A^2}}{2}, y_4 = \frac{-A + \sqrt{\frac{2a}{A} - A^2}}{2}$ . What we need is the maximal root, so  $y_4$  is kept and  $y_3$  is discarded.

In both cases, the obtained roots  $y_2$  and  $y_4$  can be unified as:

$$\hat{y} = \frac{|A| + \sqrt{\frac{2a}{|A|} - A^2}}{2} \quad (a > \frac{4}{\sqrt{27}} \lambda^{\frac{3}{4}}).$$

Hence, the minimum point of  $f(x)$  on  $x > 0$  is  $\hat{x} = \hat{y}^3$  ( $a > \frac{4}{\sqrt{27}} \lambda^{\frac{3}{4}}$ ).

II. For case 2, we give a simple argumentation by symmetry. Our goal is to seek the minimum point of  $f(x)$  on  $x < 0$ . Due to  $g(x) = g(-y^3) = -h_2(y)$  for  $x < 0$ , we need to seek the root  $\hat{y}$  of  $h_2(y)$  satisfying:

$$h_2(\hat{y}) = 0; h_2(y) > 0 \text{ when } y < \hat{y}; h_2(y) < 0 \text{ when } y > \hat{y}. \quad (A.19)$$

where  $y$  is near to  $\hat{y}$ . Actually we can seek the required root of  $h_2(y)$  in the similar way as case 1. However, the deductions in case 2 can be simplified by the symmetry between  $h_2(y)$  and  $h_1(y)$ . Since  $h_2(y) = h_1(-y)$ , therefore  $h_2(y)$  and  $h_1(y)$  are symmetric with respect to the vertical axis. Thus the minimal root of  $h_2(y)$  corresponds to the minus of the maximal root of  $h_1(y)$ :  $\hat{y} = \frac{|A| + \sqrt{\frac{2a}{|A|} - A^2}}{2}$ . Hence the minimum root of  $h_2(y)$ , i.e., the minimum point of  $f(x)$  on  $x < 0$  is  $\hat{x} = -\hat{y}^3$  ( $a < \frac{4}{\sqrt{27}} \lambda^{\frac{3}{4}}$ ).

In summary, the minimal point of  $f(x)$  for  $x \neq 0$  is:

$$\hat{x} = \begin{cases} \left( \frac{|A| + \sqrt{\frac{2a}{|A|} - A^2}}{2} \right)^3 & \text{if } a > s(\lambda) \\ - \left( \frac{|A| + \sqrt{\frac{2a}{|A|} - A^2}}{2} \right)^3 & \text{if } a < -s(\lambda) \end{cases}$$

where

$$|A| = \sqrt{\left( \frac{a^2}{2} + \sqrt{\frac{a^4}{4} - \frac{4^3}{27^2} \lambda^3} \right)^{\frac{1}{3}} + \left( \frac{a^2}{2} - \sqrt{\frac{a^4}{4} - \frac{4^3}{27^2} \lambda^3} \right)^{\frac{1}{3}}}, \quad s(\lambda) = \frac{4}{\sqrt{27}} \lambda^{\frac{3}{4}}. \quad \square$$

**Appendix B. The proof of Theorem 3.4**

**Proof.** Note that we have already derived the minimum point  $\hat{x}$  of  $f(x)$  when  $x \neq 0$  in Lemma 3.3. We now derive the minimum point  $x^*$  of  $f(x)$  when  $x \in \mathfrak{R}$ . From the previous analysis, we can easily get

$$x^* = \begin{cases} \hat{x} & f(\hat{x}) < f(0) = a^2 \\ 0 & f(\hat{x}) \geq f(0) = a^2 \end{cases}$$

However, our aim is not to seek  $x^*$  by comparison between  $f(\hat{x})$  and  $f(0)$ , but to seek  $x^*$  in a closed-form thresholding formula, i.e.,

$$x^* = \begin{cases} \hat{x} & |a| > t^*(\lambda) \\ 0 & |a| \leq t^*(\lambda) \end{cases}$$

Next our task is to explicitly compute the expression of  $t^*(\lambda)$ . When  $f(\hat{x}) \leq a^2$ , we get  $(\hat{x} - a)^2 + \lambda|\hat{x}|^{\frac{2}{3}} \leq a^2$ . By reducing the equation, we further get

$$2a\hat{x} \geq \hat{x}^2 + \lambda|\hat{x}|^{\frac{2}{3}} \quad (B.1)$$

When  $a > 0$ , by Lemma 3.1,  $\hat{x} \in (0, a)$ ,  $a \geq \frac{\hat{x}^2 + \lambda|\hat{x}|^{\frac{2}{3}}}{2\hat{x}}$ ; when  $a < 0$ , by

Lemma 3.1,  $\hat{x} \in (a, 0)$ , (B.1) can be reduced as  $-a \geq \frac{\hat{x}^2 + \lambda|\hat{x}|^{\frac{2}{3}}}{-2\hat{x}}$ . Hence we unify these two cases as follows

$$|a| \geq \frac{\hat{x}^2 + \lambda|\hat{x}|^{\frac{2}{3}}}{2|\hat{x}|} \quad \text{or} \quad |a| \geq \frac{m_\lambda(a)^2 + \lambda|m_\lambda(a)|^{\frac{2}{3}}}{2|m_\lambda(a)|} \quad (|a| > \frac{4}{\sqrt{27}} \lambda^{\frac{3}{4}}) \quad (B.2)$$

where  $\hat{x} = m_\lambda(a) := \left( \frac{|A| + \sqrt{\frac{2a}{|A|} - A^2}}{2} \right)^3 \text{sign}(a)$ . Let  $u(a) = |a| - \frac{m_\lambda(a)^2 + \lambda|m_\lambda(a)|^{\frac{2}{3}}}{2|m_\lambda(a)|}$ . Obviously, the threshold  $t^*(\lambda)$  can be computed from the roots of  $u(a)$ . Since  $m_\lambda(a)$  is the minimum point of the equation  $f(x) = (x - a)^2 + \lambda|x|^{\frac{2}{3}}$  ( $x \neq 0$ ),  $m_\lambda$  satisfies:  $|m_\lambda(a)| - a + \frac{\lambda}{3} \frac{\text{sign}(m_\lambda(a))}{|m_\lambda(a)|^{1/3}} = 0$ . According to the equation, we can obtain

$$|m_\lambda(a)|^2 = \left| a|m_\lambda(a)| - \frac{\lambda}{3} \right| |m_\lambda(a)|^{\frac{2}{3}} \quad (B.3)$$

we substitute (B.3) into  $u(a)$ ,  $u(a)$  can be further reduced as

$$u(a) = \frac{|a|}{2} - \frac{\lambda}{3|m_\lambda(a)|^{1/3}} \quad (B.4)$$

From (B.4), we can learn that  $u(a) = u(-a)$ , so  $u(a)$  is symmetric with respect to the vertical axis. In  $a \in [\frac{4}{\sqrt{27}} \lambda^{3/4}, \infty)$ ,  $u(a)$  is monotonically increasing. Moreover,  $\lim_{a \rightarrow \frac{4}{\sqrt{27}} \lambda^{3/4}} u(a) = \frac{8}{\sqrt{27}} \lambda^{3/4} - \frac{2}{3} \frac{\lambda}{m_\lambda(\frac{4}{\sqrt{27}} \lambda^{3/4})} < 0$  and  $\lim_{a \rightarrow +\infty} u(a) = +\infty$ . Therefore,  $u(a)$  on  $[\frac{4}{\sqrt{27}} \lambda^{3/4}, \infty)$  has a unique root  $t^*(\lambda)$ . By the symmetry of  $u(a)$ , it has another unique root  $-t^*(\lambda)$  on  $(-\infty, -\frac{4}{\sqrt{27}} \lambda^{3/4}]$ . Thus, we have

$$f(\hat{x}) < a^2 = f(0) \iff |a| > t^*(\lambda)$$

And still, from inequality (B.2), we obtain

$$\begin{aligned} |a| &\geq \frac{|m_\lambda(a)|}{2} + \frac{\lambda}{2|m_\lambda(a)|^{1/3}} \\ &= \frac{|m_\lambda(a)|}{2} + \frac{\lambda}{2 * 3 * |m_\lambda(a)|^{1/3}} + \frac{\lambda}{2 * 3 * |m_\lambda(a)|^{1/3}} + \frac{\lambda}{2 * 3 * |m_\lambda(a)|^{1/3}} \\ &\geq 4 \sqrt[4]{\frac{|m_\lambda(a)|}{2} * \frac{\lambda^3}{2^3 3^3 |m_\lambda(a)|}} = \frac{2}{3} \sqrt[4]{3\lambda^3} \end{aligned}$$

From above inequality, we can learn that when  $\frac{|m_\lambda(a)|}{2} = \frac{\lambda}{2 * 3 * |m_\lambda(a)|^{1/3}}$ , i.e.,  $|m_\lambda(a)| = (\frac{\lambda}{3})^{\frac{3}{4}}$ , the equality holds, i.e.,  $|a| = \frac{2}{3} \sqrt[4]{3\lambda^3}$ . By substituting  $|m_\lambda(a)| = (\frac{\lambda}{3})^{\frac{3}{4}}$  and  $|a| = \frac{2}{3} \sqrt[4]{3\lambda^3}$  into Eq. (B.4), we can get  $u(\frac{2}{3} \sqrt[4]{3\lambda^3}) = 0$ . Hence  $t^*(\lambda) = \frac{2}{3} \sqrt[4]{3\lambda^3}$ .

In summary, the minimum point of  $f(x) = (x - a)^2 + \lambda|x|^{\frac{2}{3}}$ ,  $x \in \mathfrak{R}$ , satisfies

$$x^* = \begin{cases} \left( \frac{|A| + \sqrt{\frac{2a}{|A|} - A^2}}{2} \right)^3 & \text{if } a > \frac{2}{3} \sqrt[4]{3\lambda^3} \\ 0 & \text{if } |a| \leq \frac{2}{3} \sqrt[4]{3\lambda^3} \\ - \left( \frac{|A| + \sqrt{\frac{2a}{|A|} - A^2}}{2} \right)^3 & \text{if } a < -\frac{2}{3} \sqrt[4]{3\lambda^3} \end{cases}$$

By utilizing the triangle and hyperbolic expression for the roots of cubic equation,  $|A|$  can be reduced as

$$|A| = \frac{2}{\sqrt{3}} \lambda^{\frac{1}{4}} \left( \cosh\left(\frac{\phi}{3}\right) \right)^{\frac{1}{2}} \quad \text{where } \phi = \text{arccosh}\left(\frac{27a^2}{16} \lambda^{-\frac{3}{2}}\right).$$

Thus, the proof of Theorem 3.4 is complete.  $\square$

**Appendix C. The triangle and hyperbolic expression for the roots of cubic equation**

The triangle and hyperbolic expressions for the roots of cubic equation are presented in Table 8 [24]. According to the second column of Table 8, we can derive the hyperbolic expression for the unique real root of our cubic Eq. (A.14), i.e.,

$$|A| = \frac{2}{\sqrt{3}} \lambda^{\frac{1}{2}} \left( \cosh \left( \frac{\phi}{3} \right) \right)^{\frac{1}{2}} \quad \text{where} \quad \phi = \operatorname{arccosh} \left( \frac{27a^2}{16} \lambda^{-\frac{3}{2}} \right).$$

## References

- [1] Z.B. Xu, Data modeling: visual psychology approach and  $L_{\frac{1}{2}}$  regularization theory, in: Proceedings of the International Congress of Mathematicians, 2010, pp. 1–34.
- [2] Z.B. Xu, X.Y. Chang, F.M. Xu, H. Zhang,  $L_{\frac{1}{2}}$  regularization: an iterative half thresholding algorithm, *IEEE Trans. Neural Networks Learning Syst.* 23 (7) (2012) 1013–1027.
- [3] D. Krishnan, R. Fergus, Fast image deconvolution using hyper-Laplacian priors, *Adv. Neural Inf. Process. Syst. (NIPS)* (2009).
- [4] R. Fergus, B. Singh, A. Hertzmann, S.T. Roweis, W.T. Freeman, Removing camera shake from a single photograph, *ACM Trans. Graphics* 25 (3) (2006) 787–794.
- [5] S. Cho, S. Lee, Fast motion deblurring, *ACM Trans. Graphics* 28 (5) (2009) 1–8.
- [6] L. Yuan, J. Sun, L. Quan, H.-Y. Shum, Progressive inter-scale and intra-scale non-blind image deconvolution, *ACM Trans. Graphics* 27 (3) (2008) 1–8.
- [7] T. Chan, S. Esedoglu, F. Park, A. Yip, *Math. Models Comput. Vision* 17 (2005).
- [8] A. Buades, B. Coll, J.M. Morel, A Non-local Algorithm for Image Denoising, *CVPR*, vol.2, IEE, 2005, pp. 60–65.
- [9] K. Dabov, A. Foi, V. Katkovnik, K. Egiazarian, Image denoising by sparse 3-D transform-domain collaborative filtering, *IEEE Trans. Image Process.* 16 (8) (2007) 2080–2095.
- [10] J. Mairal, F. Bach, J. Ponce, G. Sapiro, A. Zisserman, Non-local sparse models for image restoration, in: 2009 IEEE 12th International Conference on Computer Vision, 2009, pp. 2272–2279.
- [11] M. Elad, M. Aharon, Image denoising via sparse and redundant representations over learned dictionaries, *IEEE Trans. Image Process.* 15 (12) (2006) 3736–3745.
- [12] J. Mairal, M. Elad, G. Sapiro, Sparse representation for color image restoration, *IEEE Trans. Image Process.* 17 (1) (2008) 53–69.
- [13] W. Dong, L. Zhang, G. Shi, X. Wu, Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization, *IEEE Trans. Image Process.* 20 (7) (2011) 1838–1857.
- [14] M.F. Tappen, B.C. Russell, W.T. Freeman, Exploiting the sparse derivative prior for super-resolution and image demosaicing, in: Third International Workshop on Statistical and Computational Theories of Vision at ICCV, 2003.
- [15] R. Chartrand, Exact reconstruction of sparse signals via nonconvex minimization, *IEEE Signal Process. Lett.* 14 (10) (2007) 707–710.
- [16] R. Chartrand, V. Staneva, Restricted isometry properties and nonconvex compressive sensing, *Inverse Prob.* 24 (3) (2008) 1–14.
- [17] R. Chartrand, W. Yin, Iteratively reweighted algorithms for compressive sensing, in: IEEE International Conference on Acoustics, Speech and Signal Processing, 2008, pp. 3869–3872.
- [18] R. Chartrand, Fast algorithms for nonconvex compressive sensing: MRI reconstruction from very few data, in: IEEE International Symposium on Biomedical Imaging: From Nano to Macro, 2009, pp. 262–265.
- [19] T. Blumensath, M. Yaghoobi, M.E. Davies, Iterative hard thresholding and  $L_0$  regularisation, *IEE Trans. Acoust. Speech Signal Process.* vol. 3 (2007).
- [20] D.L. Donoho, Denoising by soft thresholding, *IEEE Trans. Inf. Theory* 41 (3) (1995) 613–627.
- [21] I. Daubechies, M. Defrise, C. De Mol, An iterative thresholding algorithm for linear inverse problems with a sparsity constraint, *Commun. Pure and Appl. Math.* 57 (11) (2004) 1413–1457.
- [22] A. Levin, Y. Weiss, F. Durand, W.T. Freeman, Understanding and evaluating blind deconvolution algorithms, in: CVPR, 2009, pp. 1964–1971.
- [23] J. Yang, J. Wright, T.S. Huang, Y. Ma, Image super-resolution via sparse representation, *IEEE Trans. Image Process.* 19 (11) (2010) 2861–2873.
- [24] F.C. Xing, Investigation on solutions of cubic equations with one unknown, *J. Central Univ. Nationalities (Natural Sciences Edition)* 12 (3) (2003) 207–218.
- [25] M. Elad, *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*, Springer, 2010.