International Journal of Signal Processing, Image Processing and Pattern Recognition
Vol.9, No.6 (2016), pp.177-190
http://dx.doi.org/10.14257/ijsip.2016.9.6.16

A Non-convex Approximating ℓ^p Norm Regularization Algorithm for Image Deconvolution

Weijian Liu[1], Xingwei Zhong[1], Michael Jiang[2] and Ruohe Yao [1,1]

[1]School of Electronic and Information Engineering,South China University of Technology, No. 381, Wushan Road, Tianhe District, Guangzhou, China, 510640
[2]R&D Center, VTRON technology Company, No. 233 Kezhu Road, Guangzhou Hi-Tech Industrial Development Zone, Guangzhou, China, 510670
phrhyao@scut.edu.cn

Abstract

Up to now, the non-convex ℓ^p ($0 < p < 1$) norm regularization function has shown good performance for sparse signal processing. Indeed, it benefits from a significantly heavier-tailed hyper-Laplacian model, which is desirable in the context of image gradient distributions. Both $\ell^{1/2}$ and $\ell^{2/3}$ regularization methods have been given analytic solutions and fast closed-form thresholding formulae in recent image deconvolution methods. However, the methods with the other p-value norm penalty term still suffer difficulties in getting the analytic solution and fast closed-form thresholding algorithm. In this paper, to deal with these issues, we propose an approximation of ℓ^p regularization terms with $0.5 \leq p < 1$ using a linear combination of two ℓ^p terms (that is ℓ^1 and $\ell^{1/2}$) with closed form thresholding formulae. We develop an alternating minimization method to solve the image deconvolution problems involving the constructed approximating function. We derive theoretical analytic solutions and fast closed-form thresholding formulae. We perform extensive numerical experiments to demonstrate the versatility and effectiveness of the proposed method, through a comparison with the recent non-convex ℓ^p regularization dealing with the special p-value term, with an application to image deconvolution.

Keywords: *image deconvolution; non-convex regularization; ℓ^p norm*

1. Introduction

Image deconvolution is an active research topic in image processing and computer vision, aimed at yielding high-quality results from degraded images. Mathematically, image deconvolution is an ill-posed inverse problem. Regularization is an effective way to stabilize the inversion of the ill-posed problem.

The natural image marginal distribution prior used to be described by a Gaussian or Laplacian model, and various convex regularization methods can be deployed, such as ℓ^2, ℓ^1 and related TV-norm methods [1-4]. They can reconstruct High Resolution (HR) images with convex optimization methods, but often generate overly smooth images with jagged artifacts caused by the above assumption model.

Some recent researchers have shown that real-world images' gradients have the significant property of heavier-tailed distributions with the hyper-Laplacian model $P(x) \propto e^{-k|x|^p}$ ($0.5 \leq p \leq 0.8$) rather than a Gaussian or Laplacian model [5, 20]. Therefore, non-convex ℓ^p norm regularization algorithms for image deconvolution have attracted considerable attention, which continues to increase. Levin, *et al.*, [6-8, 18] used ℓ^p-norms ($0.7 \leq p \leq 1$), to reflect the statistics of natural images. Their method yields good results

[1] Corresponding Author

[9], but the calculation is complicated. To improve the speed, Krishnan, *et al.*, [5] proposed the analytic solution of image deconvolution based on ℓ^p (p = 1/2; p = 2/3) regularization. Xu, *et al.*, [10-12] deduced the closed-form thresholding formula of the $\ell^{1/2}$ regularization model, and exhibited a fast, iterative, half-thresholding algorithm. Cao, *et al.*, [13] presented the closed-form thresholding formula of the $\ell^{2/3}$ regularization model, which not only got better results, but also sped up the algorithm of image deconvolution.

From these existing techniques, for some specific values of p (p = 1/2; p = 2/3) researchers have successfully derived analytic solutions and fast closed-form thresholding formulae through polynomial root-finding methods [17]. However, in algebra, the Abel–Ruffini theorem (also known as Abel's impossibility theorem) states that there is no general algebraic solution to polynomial equations of degree five or higher with arbitrary coefficients[19]. In this way, according to the Abel-Ruffini theorem, a non-convex regularization algorithm with the other p-value penalty functions still suffers some difficulties in getting a closed-form thresholding formula.

In this paper, for a non-convex regularization algorithm with the ℓ^p norm penalty functions, we propose an approximation for general ℓ^p ($0.5 \leq p<1$) norm using a linear combination of norms with known analytic solutions. We have developed an alternating minimization method to solve the image deconvolution problems involving the constructed approximating function. Additionally, we have derived theoretical analytic solutions and fast closed-form thresholding formulae.

The rest of the paper is organized as follows. In Section 2, we show the introduction of the theory of non-convex ℓ^p regularized image deconvolution. In Section 3, we propose our algorithm, the analytic solutions and explore the closed-form thresholding formulae. In Section 4, we present the experimental results. Finally, Section 5 provides our conclusions.

2. Non-convex ℓ^p Regularization Image Deconvolution

Image deconvolution based on sparse regularization uses a sparsity prior to recover the image. Suppose that x is an original, uncorrupted, gray-scale image with N pixels here; y is an image degraded by blur kernel k or noise. $f_1 = [1,-1]$, $f_2 = [1,-1]^T$ are two first-order derivative filters. The Hyper-Laplacian distribution priors are often utilized as non-convex ℓ^p regularization. Then the image deconvolution problem is equivalent to the following optimization problem [5, 13]:

$$x^* = \arg\min_x \sum\nolimits_{i=1}^{N} \left(\frac{\lambda}{2}(y - x \otimes k)_i^2 + \sum\nolimits_{j=1}^{2} \left|(x \otimes f_j)_i\right|^p\right) \tag{2.1}$$

where a weighting term λ controls the strength of the regularization, and $\otimes$ is the 2-dimensional convolution operator. 0<p<1, $\|z\|_p = (\sum_i |z_i|^p)^{1/p}$.

If assume that $F^j x \equiv (x \otimes f_j)$ for j = 1,2;the problem (2.1) can be represented equivalently as:

$$x^* = \arg\min_x \left(\sum_i \frac{\lambda}{2}\left|(y - x \otimes k)_i\right|^2 + \left|F_i^1 x^p + F_i^2 x^p\right)\right. = \arg\min_x \left(\frac{\lambda}{2}\left\|y - x \otimes k\right\|_2^2 + \left\|F^1 x\right\|_p^p + \left|F^2 x\right\|_p^p\right) \tag{2.2}$$

Using the alternating minimization method and introducing auxiliary variables w^1 and w^2 (together denoted as *w*), Krishnan, *et al.,* [5] converted the problem (2.2) to the following optimization problem:

$$x^* = \arg\min_x \left(\frac{\lambda}{2}\left\|y - x \otimes k\right\|_2^2 + \frac{\beta}{2}\left(\left\|F^1 x - w^1\right\|_2^2 + \left\|F^2 x - w^2\right\|_2^2\right) + \left\|w^1\right\|_p^p + \left\|w^2\right\|_p^p\right) \tag{2.3}$$

where β is a control parameter.

As β tends to infinity, the solution of problem (2.3) converges to that of (2.2). Utilizing the alternating minimization approach, Equation (2.3) can be solved by alternating between steps: the *x* sub-problem, to solve *x*, given *w*; and the *w* sub-problem, to solve *w*, given *x*.

2.1. *x* Sub-problem

Given a fixed value of w to solve the x sub-problem. The optimal x is thus:

$$(F^{1^T} F^1 + F^{2^T} F^2 + \frac{\lambda}{\beta} K^T K)x = F^{1^T} w^1 + F^{2^T} w^2 + \frac{\lambda}{\beta} K^T y \tag{2.4}$$

$$x = IFFT\left(\frac{FFT(F^1)^* \circ FFT(w^1) + FFT(F^2)^* \circ FFT(w^2) + (\lambda/\beta)FFT(K)^* \circ FFT(y)}{FFT(F^1)^* \circ FFT(F^1) + FFT(F^2)^* \circ FFT(F^2) + (\lambda/\beta)FFT(K)^* \circ FFT(K)}\right) \tag{2.5}$$

where * denotes the complex conjugate, and $\circ$ denotes component-wise multiplication.

2.2. *w* Sub-problem

Given a fixed value of *x* to solve *w* sub-problem,

$$w^* = arg\min_w \left(\left|w\right|^p + \frac{\beta}{2}\left(w - v\right)^2\right) \tag{2.6}$$

where $v = F^j x$.

Krishnan, *et al.*, [5] proposed some discriminant rules to find the global optimal solution by comparing and selecting from roots of the first-order derivative of the cost function in Equation (2.6) especially for *p* = 1/2 or 2/3 cases. Although it accelerated the optimization procedure without the need of numerous iterations as required by the Newton-Raphson method, it still needed to compute and compare multiple roots using some discriminant rules. Cao, Xu, *et al.*, [10, 11, 13] deduced the closed-form thresholding formulae in Equation (2.6) especially for *p* = 1/2 or 2/3 cases, which had a significant acceleration over Krishnan, et al.'s analytic solution. But if we try the same manipulation to *p* = 4/5, this results in a 6th order polynomial, which can only be solved numerically. In some situations such as the case of *p* = 0.7, it lacks even the polynomial and root-finding method. So we need to find a universal method for solving p in the range [0.5, 1). To solve this problem, we next propose an image deconvolution algorithm using a mixed norm based on the Least-Squares Curve Fitting Method for approximating ℓ^p norm regularization.

3. Our Algorithm

3.1. Approximation of the ℓ^p Regularization Term

We take *p* = 4/5 as an example; other p-values are similar. When we use the $\ell^{4/5}$ norm, we can not derive the analytic solution and the closed-form thresholding formulae as those in previous works [5,13]. But if we can construct an approximation of ℓ^p regularization term using a linear combination of $\ell^1/\ell^{1/2}$ norms which has the same effect as the $\ell^{4/5}$ norm, the problem of solving higher-degree polynomials will be converted to that of solving lower-degree polynomials, and the fast root-finding algorithm can be used.

From Ref. [5, 13], we can derive the analytic solutions and the closed-form thresholding formulae of the $\ell^{1/2}$ norm. The solution to the ℓ^{1} norm has been relatively simple. So when we solve the complex ℓ^{p} norm function, we can use a linear combination of $\ell^{1}/\ell^{1/2}$ norms to replace the original ℓ^{p} norm. Then Equation (2.1) can be modified to:

$$x^{*} = \arg\min_{x} \sum_{i=1}^{N} (\frac{\lambda}{2}(y - x \otimes k)_{i}^{2} + \sum_{j=1}^{2} (k_{1} \left|(x \otimes f_{j})_{i}\right|^{1} + k_{2} \left|(x \otimes f_{j})_{i}\right|^{1/2})) \tag{3.1}$$

where $k1$ and $k2$ are weight coefficients ($k_1, k_2 \geq 0$). When p=4/5, we get

$|f|^{4/5} = k_1|f|^{1} + k_2|f|^{1/2}$ where f ∈ [-255,255].

3.2. Determining the Weight Coefficients of the Approximation of ℓ^{p} Norm

We use the Least-Squares Curve Fitting Method to determine the weight coefficients. For 8-bit gray-level images, the step length was set to 0.001 when the internal value ranged from -255 to 255 to make the fitting error (residual sum of squares) less than 5x10^{5}, which is acceptable. For different p values, we use ℓ^{1} / $\ell^{1/2}$ (L1+1/2) to approximate the ℓ^{p} norm. According to the fitting method and fitting error, we figure out the weight coefficients $k1$, $k2$, which are used to approximate the p norm, listed in Table 1. Next, we will show our approximating effect. First, our goal is to approximate $\ell^{4/5}$ with a linear combination of $\ell^{1}/\ell^{1/2}$, so the fitting curve should be closer to p=4/5 rather than p = 1, p = 2/3, or p = 1/2. The results shown in Figure 1 clearly demonstrate this. Second, we will compare some other p values with their fitting curves to better illustrate the effectiveness of the method. The results are shown in Figure 2.

Table 1 Weight Coefficients and Fitting Error

p	1/2	0.55	0.6	2/3	3/4	4/5	0.85	0.9	0.95
k_1	0	0.0119	0.0299	0.0682	0.1517	0.2307	0.3414	0.4953	0.7079
k_2	1	1.1357	1.2753	1.4528	1.6146	1.6372	1.5576	1.3183	0.8374
Fitting error	0	1.99x10^{3}	9.61x10^{3}	3.26x10^{4}	8.47x10^{4}	1.20x10^{5}	1.43x10^{5}	1.29x10^{5}	6.36x10^{4}

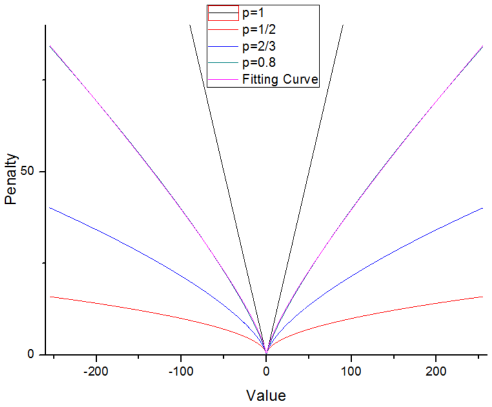

Figure 1. Prior Function Curves and *p*=4/5 Fitting Curve

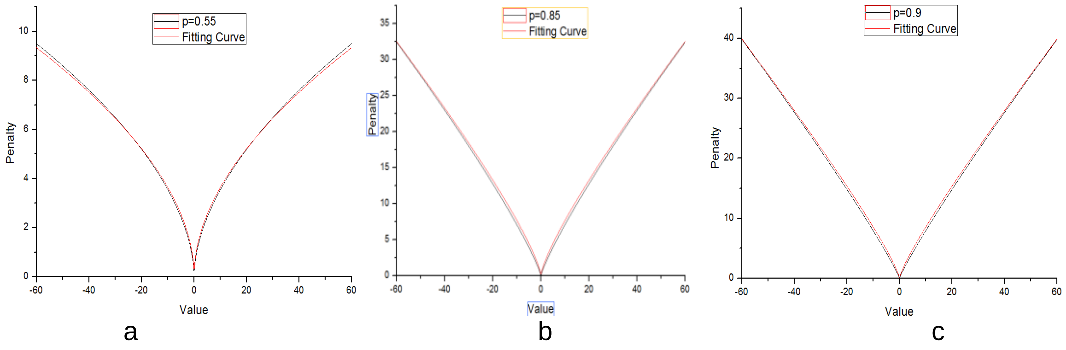

Figure 2. *p* Norm Curves and their Corresponding Fitting Curves

a p=0.55, k_1=0.0119, k_2=1.1357

b p=0.85, k_1=0.3414, k_2=1.5576

c p=0.9, k_1=0.4953, k_2=1.3183

3.3. Solution to the Algorithm

By optimizing Equation (3.1), we can get:

$$x^* = \arg\min_x \left(\frac{\lambda}{2}\|y - x\otimes k\|_2^2 + k_1\left(\|F^1 x\|_1^1 + \|F^2 x\|_1^1\right) + k_2\left(\|F^1 x\|_{1/2}^{1/2} + \|F^2 x\|_{1/2}^{1/2}\right)\right) \tag{3.2}$$

Using the alternating minimization method, we introduce auxiliary variables, giving a new cost function*:*

$$x^* = \arg\min_x \left(\frac{\lambda}{2}\|y - x\otimes k\|_2^2 + \frac{\beta}{2}\left(\|F^1 x - w^1\|_2^2 + \|F^2 x - w^2\|_2^2\right) + k_1\left(\|w^1\|_1^1 + \|w^2\|_1^1\right) + k_2\left(\|w^1\|_{1/2}^{1/2} + \|w^2\|_{1/2}^{1/2}\right)\right) \tag{3.3}$$

Utilizing the alternating minimization approach, Equation (3.3) can be performed by alternating between steps:

Solving the x sub-problem, to solve x, given w; which is similar to Equation (2.4) and Equation (2.5).

Solving the w sub-problem, to solve w, given x:

$$w^* = \arg\min_w \left(k_1|w|^1 + k_2|w|^{1/2} + \frac{\beta}{2}(w - v)^2\right) \tag{3.4}$$

Next we deduce the analytic solutions and the closed-form thresholding formulae of Equation (3.4).

3.3.1. Analytic Solution

The method of solving (3.4) is similar to Krishman's analytic solution in Ref. [5].

For non-zero w, setting the derivative of (3.4) w.r.t w to zero gives the following:

$$w^3 + \frac{2}{\beta}\left(k_1 sign(w) - \beta v\right)w^2 + \frac{1}{\beta^2}\left(k_1^2 + \beta^2 v^2 - 2\beta v k_1 sign(w)\right)w - \frac{1}{4\beta^2}k_2^2 sign(w) = 0 \tag{3.5}$$

Numerical root-finding methods can be utilized to solve the w sub-problem because its solutions can be characterized as roots of cubic polynomials [6].

For w$\in R$, we can select the correct roots by picking between w = 0 and a real root that is between 0 and v.

3.3.2. Closed-form Thresholding Formula

Cao, Xu, *et al.*, [10-13] deduced the closed-form thresholding formulae to solve $w^* = \arg\min_w \{f(w) = (w-v)^2 + \gamma |w|^{1/2}\}$ and accelerated the algorithm. Based on their work, we extend the thresholding representation theory established on $\ell^{1/2}$ regularization term to the cases with the combination of two norms, and deduce the closed-form thresholding formulae of the proposed algorithm in this paper.

We rewrite Equation (3.4) to the following equivalent expression,

$$w^* = \arg\min_w \{f(w) = (w-v)^2 + \gamma(\mathrm{k}_1 |w| + \mathrm{k}_2 |w|^{1/2})\} \tag{3.6}$$

where $\gamma = 2/\beta$.

We can obtain the corresponding closed-form thresholding formula in a similar fashion.

When $w \neq 0$, we get the first derivative of $f(w)$ in Equation (3.6) to find the minimum point:

$$h(t) = (w - v) + \gamma k_1 \frac{sign(w)}{4\sqrt{|w|}} + \gamma k_2 sign(w) = 0 \tag{3.7}$$

Case I: When $w > 0$, assuming that $t = \sqrt{|w|}$, we can obtain $w = t^2 > 0$.

So (3.7) can be rearranged as:

$$t^2 - v + \gamma k_2 \frac{1}{4t} + \frac{\gamma k_1}{2} = 0 ,$$

$$f(\mathrm{t}) = \frac{1}{t}(t^3 - (v - \frac{\gamma k_1}{2})t + \frac{\gamma k_2}{4}) = 0 .$$

Case II: When $w < 0$, let $t = \sqrt{|w|} > 0$,thus we can obtain $w = -t^2 < 0$.

So (3.7) can be rearranged as:

$$-t^2 - v - \gamma k_2 \frac{1}{4t} - \frac{\gamma k_1}{2} = 0 ,$$

$$f(\mathrm{t}) = -\frac{1}{t}(t^3 + (v + \frac{\gamma k_1}{2})t + \frac{\gamma k_2}{4}) = 0 .$$

Assuming that $g_1(t) = t^3 - (v - \frac{\gamma k_1}{2})t + \frac{\gamma k_2}{4}$ (3.8),

$$g_2(t) = t^3 + (v + \frac{\gamma k_1}{2})t + \frac{\gamma k_2}{4} \quad (3.9),$$

$$\theta_1 = v - \frac{1}{2}\gamma k_1 \quad \textit{and} \quad \theta_2 = v + \frac{1}{2}\gamma k_1 ,$$

thus, (3.8), (3.9) can be rearranged as

$$g_1(t) = t^3 - \theta_1 t + \frac{\gamma k_2}{4} \quad (3.10), \text{ and}$$

$$g_2(t) = t^3 + \theta_2 t + \frac{\gamma k_2}{4} \quad (3.11).$$

After the variable substitution, according *to* [11, 12], we can get the solution to (3.6) as follows:

Solution to 3.6: $f(w)$ in Equation (3.6) has the following closed-form thresholding formula when $w \in$ R:

$$w^* = \begin{cases} \frac{2}{3}|v - \frac{\gamma k_1}{2}|(1+\cos(\frac{2\pi}{3} - \frac{2\varphi_1}{3})) & if \quad v > p(\gamma) \\ 0 & if \quad |v| \le p(\gamma) \\ -\frac{2}{3}|v + \frac{\gamma k_1}{2}|(1+\cos(\frac{2\pi}{3} - \frac{2\varphi_2}{3})) & if \quad v < -p(\gamma) \end{cases} \tag{3.12}$$

where

$$\varphi_1 = \arccos(\frac{\gamma k_2}{8}(\frac{1}{3}(v - \frac{\gamma k_1}{2}))^{-3/2})$$

$$\varphi_2 = \arccos(\frac{\gamma k_2}{8}(\frac{1}{3}(v + \frac{\gamma k_1}{2}))^{-3/2})$$

$$p(\gamma) = \frac{\sqrt[3]{54}}{4}(\gamma u k_2)^{2/3} + \frac{\gamma u k_1}{2}$$

u is any fixed positive real number that satisfies $0 < u \le \| K \|^{-2}$ (see Ref.[11, 12]).

Proof: More details please refer to the Ref. [11,12].

4. Experiments and Analysis

Our experiments were executed using Matlab 7.8 on a computer with an Intel(R) Core(TM) i5 CPU 2.60GHz (quad-core). The natural test images are collected from the web site of http://www.flickr.com/ and from the Berkeley segmentation database [14,21]. All the test images are blurred by real-world camera shake kernels from Ref. [15], 1% Gaussian noise is added, followed by quantization to 255 discrete values. We use the PSNR defined as $PSNR = 10\log_{10}\frac{255^2}{MSE(x)}$ to evaluate the image deconvolution performance, where *x* is the deconvolution result, and MSE(*x*) denotes the mean square error between *x* and the real image. For the images in Ref. [14], we use the improved signal-to-noise ratio (ISNR) [16], that is ISNR = PSNR (recon) − PSNR (blurry), to measure the quality of restoration results.

The algorithm adopts a linear combination of $\ell^1 / \ell^{1/2}$ to approximate these *p* values (1/2, 0.55, 0.6, 2/3, 0.75, 0.8, 0.85, 0.9, 0.95) in image deconvolution.

After using the approximation of ℓ^p norm based on the Least-Squares Curve Fitting Method, we can get coefficients such as *k*1 and *k*2. By substituting *k*1 and *k*2 to the following algorithm block diagram, we can achieve fast image deconvolution.

```
Algorithm 1: A non-convex $\ell^p$ norm regularization algorithm for image deconvolution

Input: Blurred image $y$; kernel $k$; $p$;
    regularization weight $\lambda$; $\beta_0$, $\beta_{inc}$, $\beta_M$;
The estimation of the coefficients $k_1$, $k_2$ according to $p$;
Maximal number of outer iterations T;
Number of inner iterations J.
Initialize iter=0, $x=y$, $\beta=\beta_0$, pre-compute
        constant terms $F^1, F^2, K$ in Equation (2.5).
While ($\beta \le \beta_M$ && iter $\le$ T)
    for i=1 to J Do
        X-subproblem: optimize $x$ according to Equation (2.5)
        W-subproblem: optimize $w1$,$w2$ according to Equation (3.12)
    End for
        $\beta = \beta_{inc} * \beta$
        iter=iter+1
end while
Output：$x$
```

In the experiment, we set $\beta_0 = 1$, $\beta_{inc} = \sqrt{2}$, $\beta_{max} = 256$ and $\lambda = 2/\beta$ to get the best PSNR performance, comparing our methods to the ℓ^1, $\ell^{1/2}$, and $\ell^{2/3}$ methods in Ref. [5, 13].

To compare the performance of different methods we conduct experiments for 8 kernels. For each kernel we choose 10 test images to evaluate the deconvolution performance. We list recovery results in Table 2. We can find the best PSNR performance of each image appearing at different p values. To further test the performance of our algorithm, we evaluate the deconvolution results over 8 different kernels; the results are shown in Table 3. We also show one of the deconvolution results in Figure 3 and Figure 4.

Table 2. PSNR (dB) Performances for Kernel 4 (19 × 19)

p	blurry	1/2	0.55	0.6	2/3	0.75	0.80	0.85	0.90	0.95	1
lena	27.14	32.52	32.54	32.57	32.58	32.60	**32.74**	32.70	32.53	32.38	31.99
barbara	22.79	28.01	28.06	28.08	28.15	28.35	28.42	**28.47**	28.41	28.16	27.03
boats	25.36	31.40	31.50	31.52	31.91	31.82	31.72	31.78	31.91	**32.11**	30.98
couple	23.32	29.56	29.61	29.65	29.63	29.86	29.92	**29.98**	29.85	29.36	29.31
goldhill	26.37	30.86	30.96	30.98	31.10	**31.23**	31.04	30.98	30.96	30.94	30.80
man	24.48	30.23	30.33	30.35	30.44	30.48	**30.63**	30.52	30.49	30.33	30.19
dsc_0085[5]	22.68	30.23	30.26	30.30	30.34	30.42	30.49	**30.54**	30.26	30.18	29.52
crowd	23.86	30.16	30.30	30.35	30.19	30.46	**30.59**	30.48	30.39	29.92	29.88
bridge	21.64	27.43	27.45	27.49	27.72	27.88	27.90	**27.92**	27.89	27.72	26.32
lake	23.28	29.53	29.61	29.61	29.70	29.95	**30.12**	29.92	29.87	29.37	29.01

Table 3. Comparison for 8 Kernels with Image dsc_0085 [5]

Kernel	blurry	1/2	0.55	0.6	2/3	0.75	0.80	0.85	0.90	0.95	1
ker1(13*13)	23.77	31.33	31.40	31.45	31.54	31.67	**31.85**	31.83	31.69	31.22	30.56
ker2(15*15)	23.07	30.32	30.36	30.43	30.57	30.78	30.93	**30.98**	30.83	29.92	29.67
ker3(17*17)	21.74	30.06	30.10	30.18	30.26	30.19	**30.45**	30.38	30.11	30.07	29.74
ker4(19*19)	22.68	30.23	30.26	30.30	30.34	30.42	30.49	**30.54**	30.26	30.18	29.52
ker5(21*21)	17.75	31.05	31.18	31.25	31.29	31.34	31.51	**31.53**	31.28	30.98	30.06
ker6(23*23)	19.01	30.37	30.49	30.52	30.59	30.95	31.13	**31.18**	30.97	30.44	29.82
ker7(25*25)	18.87	29.93	29.95	30.29	30.12	30.33	**30.48**	30.45	30.35	29.91	28.48
ker8(27*27)	17.35	29.45	29.55	29.62	29.65	29.66	**29.87**	29.85	29.62	29.55	28.23

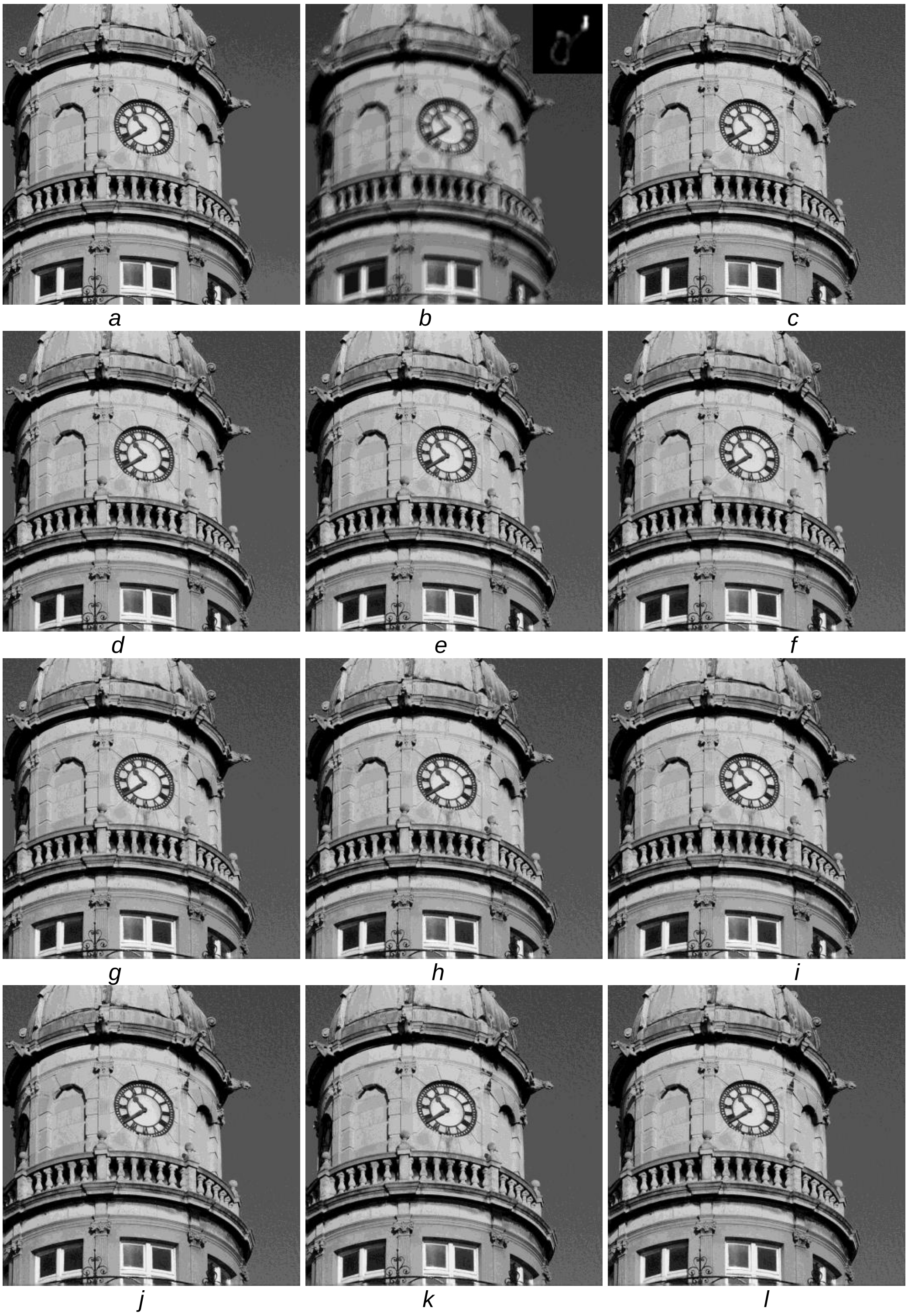

Figure 3. The Deconvolution Results for Different *p* with Image dsc_0085 [5] using Kernel6(23 x 23),1% Gaussian Noise

a Original image; b Blurry image PSNR=19.01dB; c *p*=1 PSNR=29.82dB;
d *p*=1/2 PSNR=30.37dB; e *p*=0.55 PSNR=30.49dB; f *p*=0.6 PSNR=30.52dB;
g *p*=2/3 PSNR=30.59dB; h *p*=0.75 PSNR=30.95dB; i *p*=0.8 PSNR=31.13dB;
j *p*=0.85 PSNR=31.18dB; k *p*=0.9 PSNR=30.97dB; l *p*=0.95 PSNR=30.44dB

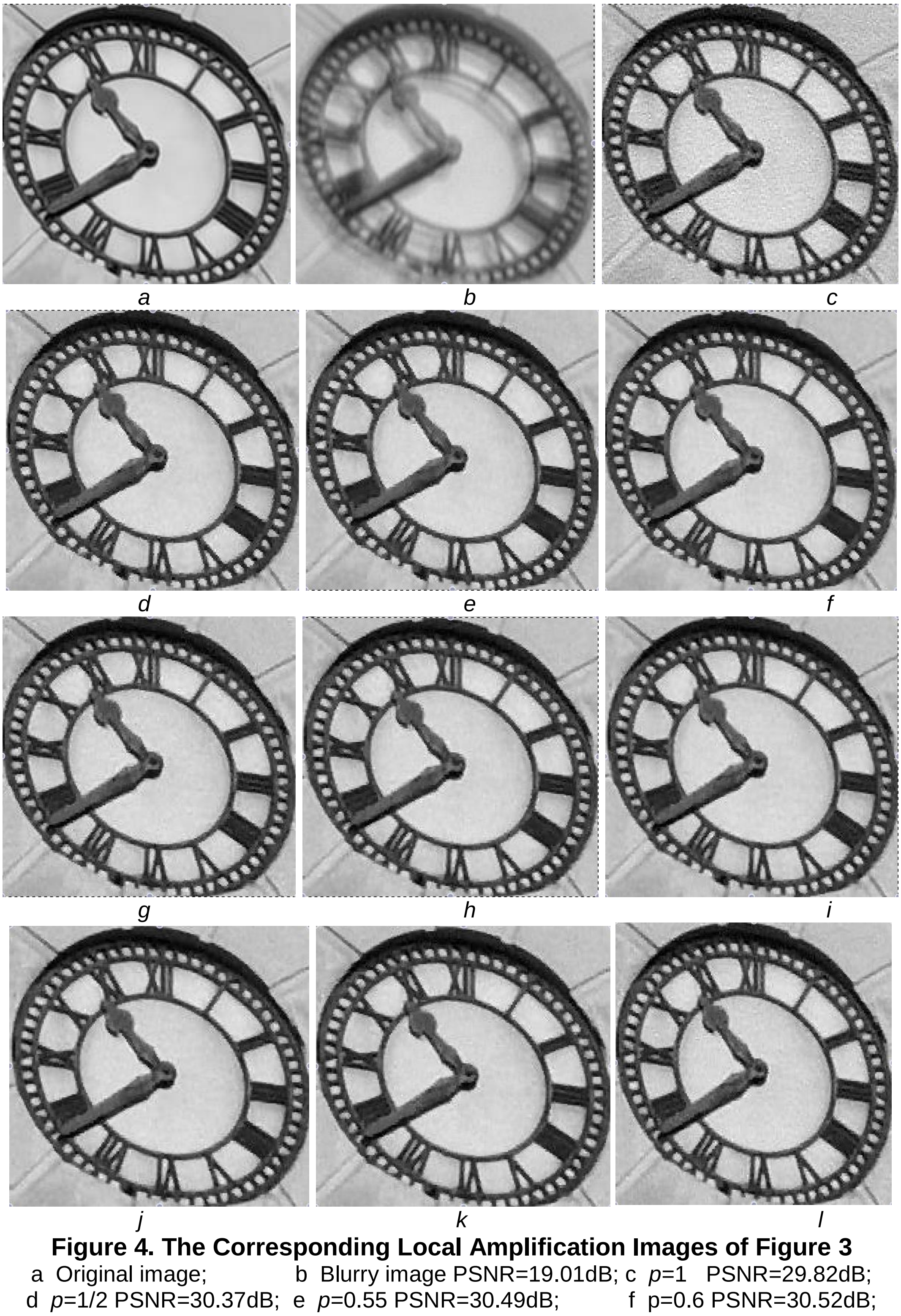

Figure 4. The Corresponding Local Amplification Images of Figure 3

a Original image; b Blurry image PSNR=19.01dB; c *p*=1 PSNR=29.82dB;
d *p*=1/2 PSNR=30.37dB; e *p*=0.55 PSNR=30.49dB; f p=0.6 PSNR=30.52dB;
g *p*=2/3 PSNR=30.59dB; h *p*=0.75 PSNR=30.95dB; i *p*=0.8 PSNR=31.13dB;
j *p*=0.85 PSNR=31.18dB; k *p*=0.9 PSNR=30.97dB; l *p*=0.95 PSNR=30.44dB

Besides, we conduct the experiment of 100 gray test images in Berkeley segmentation database [14,21] for 8 kernels and present the average ISNR values in Table 4. The comparison of average ISNR at different p values is listed in Figure 5.

From the simulation results of the approximating *p* norm compared to those in Ref.[5, 13], we can draw the conclusion that the best PSNR performance appears at different *p* values for different types of images. In this paper, the proposed algorithm can get deconvolution results of different *p* values effectively, and use different weight coefficients for approximating different *p* values, therefore, the image deconvolution algorithm becomes more universally adaptable.

Table 4. The Average ISNR Values Of 100 Images for 8 Kernels

Kernel	1/2	0.55	0.6	2/3	0.75	0.80	0.85	0.90	0.95	1
ker1(13*13)	5.92	6.01	6.04	6.36	6.55	6.60	**6.61**	6.59	6.29	5.29
ker2(15*15)	5.40	5.41	5.38	5.79	5.91	5.93	5.94	**5.95**	5.73	4.64
ker3(17*17)	6.06	6.08	6.04	6.25	6.44	6.47	**6.48**	6.43	6.05	4.93
ker4(19*19)	5.99	6.04	6.02	6.12	6.32	6.37	**6.38**	6.30	5.90	4.98
ker5(21*21)	10.06	10.15	10.19	10.33	10.57	10.62	**10.64**	10.59	10.17	9.07
ker6(23*23)	8.28	8.30	8.33	8.68	8.77	8.80	8.81	**8.82**	8.67	7.93
ker7(25*25)	7.98	8.00	7.98	8.29	8.41	8.43	**8.45**	8.44	8.21	7.15
ker8(27*27)	9.04	9.08	9.05	9.14	9.35	9.40	**9.41**	9.34	8.89	7.64

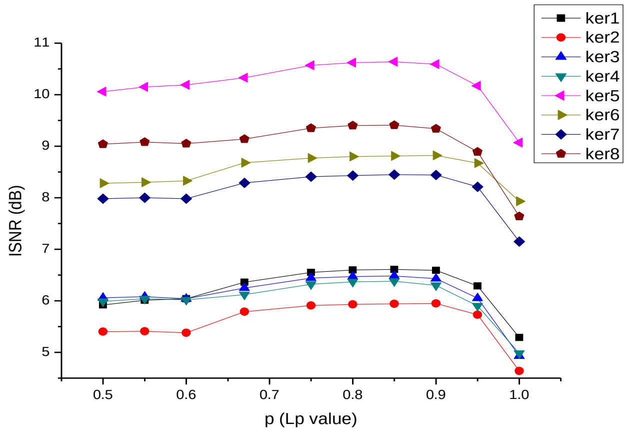

Figure 5. Comparison of average ISNR at Different *p* Values

5. Conclusion

In this paper, we propose an image deconvolution algorithm using a linear combination of ℓ^{1} *and* $\ell^{1/2}$ terms to approximate ℓ^{p} norm minimization based on a least-squares fitting method, and derive the analytic solutions and the closed-form thresholding formulae. Extensive experiments demonstrate that our algorithm becomes more universally adaptable compared to existing specific *p*-values of the ℓ^{p} norm (*p* = 1/2; *p* = 2/3) regularization method.

More generally, the method proposed in this paper can also be extended to the application of solving a kind of ℓ^{p} regularization problem which is very hard to solve. We can use a convex regularization and a non-convex regularization for approximating any *p* value in ℓ^{p} (1/2 $\leq$ *p* < 1) regularization problems. We can also further extend this method to other areas, such as compressed sensing. remote sensing and machine learning. Then we can utilize the objective image quality method[22] to evaluate them.

Acknowledgments

The research is supported by the Science and Technology Program of Guangzhou, China (Grant No: 2014Y2-00211).

References

[1] A. N. Tikhonov and V. Y Arsenin, “Solutions of ill posed problems”, VH Winston and Sons, Washington, DC, **(1977)**.

[2] L. Rudin, S. Osher and E. Fatemi, “Nonlinear total variation based noise removal algorithms”, Physica D, vol. 60, **(1992)**, pp. 259–268

[3] T. Chan and C. Wong, “Total variation blind deconvolution”, IEEE Transactions on Image Processing, vol. 7, no. 3, **(1998)**, pp. 370–375

[4] Y. Wang, J. Yang, W. Yin and Y. Zhang, “A new alternating minimization algorithm for total variation image reconstruction”, SIAM J. Imaging Sciences, vol. 1, no. 3, **(2008)**, pp. 248–272.

[5] D.Krishnan and R. Fergus, “Fast image deconvolution using hyperlaplacian priors”, Proc. IEEE Conf. Neutral Information Processing System, Vancouver, BC, Canada, **(2009)**, pp. 1033–1041

[6] A. Levin, R. Fergus, F. Dorand and W.T. Freeman, “'Image and depth from a conventional camera with a coded aperture”, ACM Transactions on Graphics, SIGGRAPH, vol. 26, no. 3. **(2007)**.

[7] A. Levin and Y. Weiss, “User assisted separation of reflections from a single image using a sparsity prior”, PAMI, vol. 29, no. 9, **(2007)**, pp. 1647–1654.

[8] A. Gupta, N. Joshi, C. L. Zitnick, M. Cohen and B. Curless, “Curless. Single image deblurring using motion density functions”, ECCV, no. 10, **(2010)**.

[9] D. Krishnan. T. Tay and R. Fergus, “Blind deconvolution using a normalized sparsity measure”, Proc. IEEE Conf. Computer Vision and Pattern Recognition, Washington, DC, USA, **(2011)**, pp. 233– 240

[10] Z. Xu, “Data modeling: visual psychology approach and L1/2 regularization theory”, Proceedings of the International Congress of Mathematicians, Hyderabad, India, **(2010)**, pp. 1–34

[11] Z. Xu, X. Chang, F. Xu and H. Zhang, “L1/2 Regularization: A Thresholding Representation Theory and a Fast Solver”, IEEE Transactions on Neural Networks and Learning Systems, vol. 23, no. 7, **(2012),** pp. 1013-1027.

[12] J. Zeng, S. Lin, Y. Wang and Z.Xu, “L1/2 Regularization: Convergence of Iterative Half Thresholding Algorithm”, IEEE Transactions on Signal Processing, vol. 62, no. 9, **(2014)**, pp. 2317-2329.

[13] W. Cao, J. Sun and Z. Xu,“Fast image deconvolution using closed-form thresholding formulas of Lq(q=1/2,2/3) regularization”, Journal of Visual Communication and Image Representation, vol. 24, no. 1, **(2013)**, pp. 31-41.

[14] D. Martin, C. Fowlkes, D. Tal and J. Malik, “A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics”, ICCV, Vancouver, BC, **(2001)**, pp. 416-423.

[15] A. Levin, Y. Weiss, F. Durant and W.T. Freeman, “Understanding and evaluating blind deconvolution algorithms”, CVPR, Miami, F L, **(2009)**, pp. 1964-1971.

[16] C. He, C. Hu, W. Zhang, B. Shi and X. Hu, “Fast Total-Variation Image Deconvolution with Adaptive Parameter Estimation via Split Bregman Method”, Mathematical Problems in Engineering, **(2014)**, pp. 1-9.

[17] S. Tang, L. Xiao, P. Liu, J. Zhang and L.Huang, “Edge and color preserving single image superresolutio”, Journal of Electronic Imaging, vol. 23, no. 3, **(2014)**.

[18] A. Levin, R. Fergus, F.Durand and W. T. Freeman, “Deconvolution using natural image priors”, Technical report, MIT, **(2007)**.

[19] “Abel–Ruffini theorem”, https://en.wikipedia.org/wiki/Abel%E2%80%93Ruffini_theorem, accessed, no. 15 **(2015)** June.

[20] S. Tao, W. Dong, H. Feng, Z. Xu and Q. Li, “Non-blind image deconvolution using natural image gradient prior', Optik, vol. 124, **(2013),** pp. 6599– 6605.

[21] “Berkeley Segmentation Dataset: Images”, http://www.eecs.berkeley.edu/Research/ Projects/ CS/vision/grouping/segbench/BSDS300/html/dataset/images.html, accessed, **(2015)** May10.

[22] Z. Qu and T. Si, “A kind of method of color image quality assessment”, Journal of Harbin University Of Science And Technology, **(2013).**