CORRECTING CAMERA SHAKE BY INCREMENTAL SPARSE APPROXIMATION

Paul Shearer, Anna C. Gilbert, Alfred O. Hero III

University of Michigan, Ann Arbor

ABSTRACT

The problem of deblurring an image when the blur kernel is unknown remains challenging after decades of work. Recently there has been rapid progress on correcting irregular blur patterns caused by camera shake, but there is still much room for improvement. We propose a new blind deconvolution method using incremental sparse edge approximation to recover images blurred by camera shake. We estimate the blur kernel first from only the strongest edges in the image, then gradually refine this estimate by allowing for weaker and weaker edges. Our method competes with the benchmark deblurring performance of the state-of-the-art while being significantly faster and easier to generalize.

1. INTRODUCTION

In the problem of blind image deconvolution, we are given a blurry image y and challenged to determine an estimate x of the unknown sharp image x^{true} without knowledge of the blur kernel k^{true}. In the simplest model of blur, y is formed by convolving x^{true} with k^{true} and adding noise n:

$$y = k^{\text{true}} * x^{\text{true}} + n. \tag{1}$$

This convolution model assumes spatially uniform blur, which is frequently violated due to slight camera rotations and out-of-plane effects [1]. Still, the uniform model works surprisingly well and methods for it can be extended to handle nonuniform blur [2–4].

Even with uniform blur by a single kernel, the blind deconvolution problem is highly underdetermined and additional assumptions must be made to obtain a solution. These assumptions are often imposed most conveniently by moving the problem into a filter space. We define filters $\{f_\gamma\}_{\gamma=1}^L$ and set $y_\gamma = f_\gamma * y$ and $x_\gamma^{\text{true}} = f_\gamma * x^{\text{true}}$, so that

$$y_\gamma = k^{\text{true}} * x_\gamma^{\text{true}} + n_\gamma \tag{2}$$

for $\gamma \in [L] = \{1, \dots, L\}$. Defining $\mathbf{x}^{\text{true}} = \{x_\gamma^{\text{true}}\}_{\gamma=1}^L$, $\mathbf{y} = \{y_\gamma\}_{\gamma=1}^L$, and $(k * \mathbf{x}^{\text{true}})_\gamma = k * x_\gamma^{\text{true}}$, we can write the filter space problem compactly as

$$\mathbf{y} = k^{\text{true}} * \mathbf{x}^{\text{true}} + \mathbf{n}. \tag{3}$$

The simplest nontrivial filter space is gradient space, where $L = 2$ and $f_1 = [1, -1], f_2 = [1, -1]^T$, but there are many other possibilities. Determining x from a filter space representation $\mathbf{x}$ often does not work well, so typically one obtains an estimate k of k^{true} and deconvolves y with k to get x [1].

Bayesian inference is a convenient framework for imposing prior assumptions to regularize blind deconvolution [5]. By assuming some distribution of $\mathbf{n}$ we obtain a likelihood function $p(\mathbf{y} \mid k * \mathbf{x})$ which gives the probability that the data $\mathbf{y}$ arose from a given pair $(k, \mathbf{x})$. We then choose priors $p(k)$ and $p(\mathbf{x})$ and compute the posterior distribution

$$p(k, \mathbf{x} \mid \mathbf{y}) \propto p(\mathbf{y} \mid k * \mathbf{x})p(\mathbf{x})p(k). \tag{4}$$

Estimates of $\mathbf{x}$ and k may be obtained by summary statistics on $p(k, \mathbf{x} \mid \mathbf{y})$. We call the mode of $p(k, \mathbf{x} \mid \mathbf{y})$ the joint maximum *a posteriori* (MAP) estimator, while the mode of the marginal $p(k \mid \mathbf{y}) = \int p(k, \mathbf{x} \mid \mathbf{y})d\mathbf{x}$ is the kernel MAP estimator. Most blind deconvolution methods are nominally MAP estimators but do not actually find a global minimizer, as this is typically intractable and may even be counterproductive. We refer to any method organized around optimizing a posterior as a MAP method, while methods that actually find a global minimum will be called *ideal* MAP methods. Joint MAP methods typically attempt to minimize the cost function $F(k, \mathbf{x}) = -\log p(k, \mathbf{x} \mid \mathbf{y})$, which may be written (up to an irrelevant additive constant) as the sum of a data misfit and two regularization terms,

$$F(k, \mathbf{x}) = L(k * \mathbf{x}) + R_{\mathbf{x}}(\mathbf{x}) + R_k(k), \tag{5}$$

where each of these functions may take the value $+\infty$ to represent a hard constraint. Kernel MAP estimation is more difficult as it involves a high-dimensional marginalization, and it is typically approximated by variational Bayes or MCMC sampling [6].

Joint MAP estimation is the oldest, simplest, and most versatile approach to blind deconvolution [7–9], but initial joint MAP efforts on the camera shake problem met with failure [10], even when ℓ_p regularizers for $p < 1$ were used. In [1], Levin *et al* showed that the ℓ_p regularizer generally prefers blurry images to sharp ones: $\|\mathbf{y}\|_p^p < \|\mathbf{x}^{\text{true}}\|_p^p$, so that ideal joint MAP typically gives the trivial *no-blur* solution $(k, x) = (\delta_0, y)$, where δ_0 is the Kronecker delta kernel. The non-ideal joint MAP methods [11, 12] somewhat compensate

for the defects of ideal joint MAP by dynamic edge prediction and likelihood weighting, but benchmarking in [1, 13] showed that these heuristics sometimes fail.

In [10] Fergus *et al* developed a kernel MAP method with a sparse edge prior which was very effective for correcting camera shake. In [1] it was noted that marginalization over $\mathbf{x}$ seems to immunize ideal kernel MAP against the blur-favoring prior problem. More refined kernel MAP methods were recently reported in [13] and [14], and to our knowledge these two methods are the top performers on the benchmark 32 image test set from [1]. While these efforts have made kernel MAP much more tractable, it remains harder to understand and generalize than joint MAP, so it would be useful to find a joint MAP method that is competitive with kernel MAP on the camera shake problem.

In [15], Krishnan *et al* addressed the blur-favoring prior problem in joint MAP by changing the prior, proposing the scale-invariant ℓ_1/ℓ_2 ratio as a 'normalized' sparse edge penalty. The ℓ_2 normalization compensates for the way that blur reduces total ℓ_1 edge mass, causing the ℓ_1/ℓ_2 penalty to prefer sharp images and eliminating the need for additional heuristics. While their algorithm does not quite match the performance of [10] on the benchmark test set from [1], it comes fairly close while being significantly simpler, faster, and in some cases more robust. Other promising joint MAP methods include [16–18], but we are not aware of public code with full benchmark results for these methods.

1.1. Our approach

We propose a new approach to joint MAP blind deconvolution in which the kernel is estimated from a sparse approximation $\mathbf{x}$ of the sharp gradient map $\mathbf{x}^{\text{true}}$. Initially we constrain $\mathbf{x}$ to be very sparse, so it contains only the few strongest edges in the image, and we determine k such that $k * \mathbf{x} \approx \mathbf{y}$. Because $\mathbf{x}$ is so much sparser than $\mathbf{y}$, the solution $k = \delta_0$ is very unlikely. But generally this initial k overestimates k^{true}, so we refine k by letting weaker edges into $\mathbf{x}$.

To present this approach formally, we set $f_1 = [1, -1]$, $f_2 = [1, -1]^T$, so that $\mathbf{x}(p) = (x_1(p), x_2(p))$ is the discrete image gradient vector at each pixel p. We set $L(k, \mathbf{x}) = \frac{1}{2} \|k * \mathbf{x} - \mathbf{y}\|_2^2$ and impose the usual positivity and unit sum constraints on k. We measure gradient sparsity using the $\ell_{2,0}$ norm: $|\mathbf{x}(p)|$ is the ℓ_2 length of $\mathbf{x}(p)$ and $\|\mathbf{x}\|_{2,0} = \|\|\mathbf{x}\|\|_0$ the number of nonzero gradient vectors. The joint MAP optimization problem is then

$$\begin{aligned}
\underset{k, \mathbf{x}}{\text{minimize}} \quad & \tfrac{1}{2} \|k * \mathbf{x} - \mathbf{y}\|_2^2 \\
\text{subject to} \quad & k \geq 0, \ 1^T k = 1, \ \|\mathbf{x}\|_{2,0} \leq \tau,
\end{aligned} \tag{6}$$

where the expression $a^T b$ denotes the dot product of the arrays a and b when considered as vectors, and the 1 in $1^T k$ is an all-ones array.

We solve this problem with an iterative optimizer described in §2, and slowly increase τ as the iterations proceed. To initialize τ we use the ℓ_1/ℓ_2 ratio, a robust lower bound on a signal's sparsity [19]. The sharp $\mathbf{x}$ should be significantly sparser than $\mathbf{y}$, so initially we set $\tau = \beta_0 \tau_{\mathbf{y}}$, where $\tau_{\mathbf{y}} = \|\|\mathbf{y}\|\|_1 / \|\|\mathbf{y}\|\|_2$ and $\beta_0 < 1$ is a small constant. After an initial burn-in period of I_b iterations we multiply τ by a constant growth factor $\gamma > 1$, an action we repeat every I_s iterations thereafter.

We use a standard multiscale seeding technique to accelerate the kernel estimation step [10, 15]. We begin by solving (6) with a heavily downsampled $\mathbf{y}$, giving a cheap, low-resolution approximation to k and $\mathbf{x}$. We then upsample this approximation and use it as an initial guess to solve (6) with a higher resolution $\mathbf{y}$, repeating the upsample-and-seed cycle until we reach the full resolution $\mathbf{y}$. At each scale we use the same τ increase schedule. After kernel estimation we use non-blind deconvolution of y with k to get the sharp image x.

The easiest way to understand how our kernel estimation works is to watch k and $\mathbf{x}$ evolve as the iterations progress. In Fig. 1, the state of k and $\mathbf{x}$ is shown at iterations $2, 32$, and 150 of the final full-resolution scale, with k^{true} and $\mathbf{x}^{\text{true}}$ at far right. Initially $\mathbf{x}$ is quite sparse, so k cannot be a trivial kernel because the parts of $\mathbf{y}$ not in $\mathbf{x}$ must be attributed to blur. But this initial approximation is crude, so as τ increases with iteration, $\mathbf{x}$ is allowed to have more and more edges so that k can be refined.

1.2. Novelty and relations with existing methods

Direct ℓ_0 optimization is well-established in the compressed sensing community [20, 21] but we are not aware of any effective ℓ_0 approaches to blind deconvolution. In [15] the ℓ_1/ℓ_2 ratio was deliberately chosen over ℓ_0 because while both have the desired scale invariance, the graph of ℓ_1/ℓ_2 is smoother and looks more 'optimizable' than ℓ_0. We contend that ℓ_0 may be difficult to use as a cost function, but very effective as a constraint. Gradient and kernel thresholding are commonly used [11, 12] and these can be interpreted as ℓ_0 projections, but they are typically used as auxiliary heuristics, not as the central modeling idea. Our technique of slowly increasing the sparsity constraint τ is reminiscent of matching pursuit algorithms for sparse approximation [22, 23]. It is also related to the likelihood reweighting technique of [11], which may be seen by considering the Lagrangian of (6). However, our initialization strategy requires that we use the constrained formulation rather than the Lagrangian.

2. ALTERNATING PROJECTED GRADIENT METHOD

To solve problem (6) at a given scale, we use a standard alternating descent strategy: starting from some initial k and $\mathbf{x}$, we reduce the cost function by updating $\mathbf{x}$ with k fixed,

Fig. 1. Kernel estimation on an image from the test set of [1]; a small patch has been selected and rescaled for clarity. *Left:* Blurry edge map $|\mathbf{y}|$. *Center left to center right:* Evolution of the kernel k (inset) and edge map magnitude $|\mathbf{x}|$ in the final full-resolution stage. As τ increases in (6), the edge map becomes less sparse and the kernel is refined. *Right:* k^{true} and $|\mathbf{x}^{\text{true}}|$.

then k with $\mathbf{x}$ fixed, cycling until a stopping criterion is met. Each cycle, or outer iteration, consists of $I_{\mathbf{x}}$ inner iterations updating $\mathbf{x}$ and I_k inner iterations updating k. All updates are computed with a projected gradient method; given a smooth function $h(u)$ and a constraint set $\mathcal{U}$, projected gradient methods seek a solution of $\min_{u \in \mathcal{U}} h(u)$ by updates of the form $u \leftarrow \mathcal{P}_{\mathcal{U}}(u - \alpha_u g_u)$, where $g_u = \nabla h(u)$, α_u is a step size, and $\mathcal{P}_{\mathcal{U}}(w) = \operatorname{argmin}_{u \in \mathcal{S}} \|u - w\|_2^2$ is the Euclidean projection of w onto $\mathcal{U}$. Convergence of alternating descent and projected gradient methods to stationary points is proven in [24] under mild conditions.

We now describe how we compute the projected gradient iterations for the inner subproblems $\min_{k \in \mathcal{K}} L(k, \mathbf{x})$ and $\min_{\mathbf{x} \in \mathcal{X}} L(k, \mathbf{x})$, where $L(k, \mathbf{x}) = \frac{1}{2} \|k * \mathbf{x} - \mathbf{y}\|_2^2$, $\mathcal{K} = \{k \,|\, k \geq 0, 1^T k = 1\}$, and $\mathcal{X} = \{\mathbf{x} \,|\, \|\mathbf{x}\|_{2,0} \leq \tau\}$. Letting $\mathbf{r} = k * \mathbf{x} - \mathbf{y}$ denote the residual, we have $\nabla_k L = \sum_{\gamma} \bar{\mathbf{x}}_{\gamma} * \mathbf{r}_{\gamma}$ and $\nabla_{\mathbf{x}} L = \bar{k} * \mathbf{r}$, where the bar denotes $180°$ rotation about the origin. Assuming the nonzero elements of $|\mathbf{x}|$ are distinct, the projection $\mathcal{P}_{\mathcal{X}}(\mathbf{x})$ is the top-τ vector thresholding

$$P_{\mathcal{X}}(\mathbf{x})(i) = \mathbf{x}(i) \cdot \mathbf{1}\left(|\mathbf{x}(i)| \geq \theta(|\mathbf{x}|, \tau)\right), \qquad (7)$$

where $\mathbf{1}(A)$ is the indicator function for condition A and $\theta(|\mathbf{x}|, \tau)$ is the τth biggest element of $|\mathbf{x}|$. The set $\mathcal{K}$ is a canonical simplex with projection $\mathcal{P}_{\mathcal{K}}(k)$ given by

$$P_{\mathcal{K}}(k)(i) = \max(0, k(i) - \sigma), \qquad (8)$$

where σ is the unique solution of $1^T P_{\mathcal{K}}(k) = 1$. Both $P_{\mathcal{X}}$ and $P_{\mathcal{K}}$ can be computed in linear time using selection algorithms [25, 26].

The step sizes $\alpha_{\mathbf{x}}, \alpha_k$ are chosen by backtracking line search from an initial guess. In the $\mathbf{x}$ subproblem our initial guess is

$$\alpha_{\mathbf{x}} = \frac{(k * g_{\mathbf{x}})^T \mathbf{r}}{(k * g_{\mathbf{x}})^T (k * g_{\mathbf{x}})}, \qquad (9)$$

which is optimal in the sense that it solves the problem $\min_\alpha L(k, \mathbf{x} - \alpha g_{\mathbf{x}})$. This aggressive step size was chosen over several alternatives, as it was the most effective for securing good edge support estimates. In the k subproblem we use the spectral projected gradient (SPG) method [27]; in the first iteration $\alpha_k = 1$, and in subsequent iterations we use the Barzilai-Borwein step size

$$\alpha_k = \frac{(g_k - g_k^{old})^T (g_k - g_k^{old})}{(g_k - g_k^{old})^T (k - k^{old})} \qquad (10)$$

where g_k^{old} and k^{old} denote the values of g_k and k at the previous SPG iteration.

3. IMPLEMENTATION AND EXPERIMENTS

We implemented our method in MATLAB by modifying the code of [15], which uses a similar strategy of alternating minimization with multiscale seeding. The full-resolution kernel size was set to 35×35 for all experiments. The initial stage of the multiscale algorithm downsamples $\mathbf{y}$ by a factor of $5/35$ in each direction, so that the kernel is of size 5×5, and each upsample cycle increases the size of $k, \mathbf{x}$, and $\mathbf{y}$ by a factor of roughly $\sqrt{2}$ until full resolution is reached. The parameters of the core single-scale algorithm from §2 were set to $\beta_0 = 0.15$, $\gamma = 1.10$, $I_b = 30$, $I_s = 10$, $I_{\mathbf{x}} = 1$, $I_k = 6$. We do 30 iterations of the alternating projected gradient method for all scales except the final, full-resolution scale, which uses 180 iterations. Non-blind deconvolution with the estimated kernel was performed using the method of [28], using the parameter settings chosen in the code for [13]. The implementation is available online from the first author's website.

In [1] a test set of 32 blurry images with known ground truth was created for benchmarking blind deconvolution methods. Each blurry image was formed by taking a picture of a sharp image with a camera that shook in-plane, and bright points outside the image were used to obtain ground truth blur kernels. A total of 32 blurry images were formed by blurring 4 sharp images on 8 different shake trajectories. This test set has become the *de facto* standard for objectively comparing different methods.

We ran our algorithm on this test set and compared its performance against the methods of [13] and [14]. We compare against these methods because they have published implementations which match or exceed the performance of the
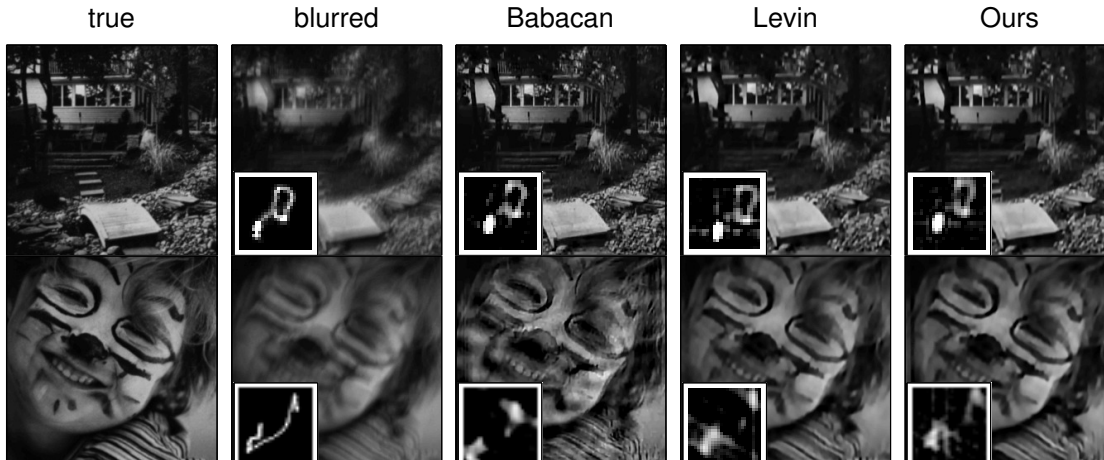
| true | blurred | Babacan | Levin | Ours |

Fig. 2. Sample results from our method, [13, 14] on the benchmark set of [1]. True and recovered kernels inset.
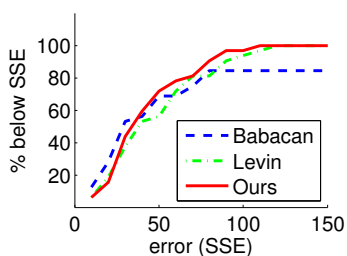

Fig. 3. Cumulative deblurring performance our method, [13], and [14] on the 32 image test set of [1]. The vertical axis is the percentage of the 32 runs having at most a given SSE.
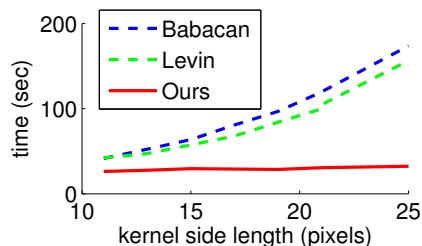

Fig. 4. Kernel estimation time as a function of the side length of the kernel bounding box. For each of the 8 kernels from the test set, we report the average kernel estimation time per image over the 4 images blurred by that kernel.

state-of-the-art methods in [10–12, 15], and we know of no methods that outperform [13] and [14] on this test. We use the squared error metric $\mathrm{SSE}(x) = \sum_i (x(i) - x^{\mathrm{true}}(i))^2$ to measure performance and note that results using the ratio metric of [1] are similar. Results for [13] were taken from files included with their published implementation, while results for [14] were generated by running their online test script using the log prior, which was the best in their experiments. Our experiments were performed in MATLAB 2011a on a quad-core Intel 2.6 GHz MacBook Pro.

Two sample images from the benchmark are shown in Fig. 2. Our method and [13] perform very similarly on both images. On the house image [14] is very sharp and by far the best, but it suffers from severe artifacts on the boy image. Fig. 3 summarizes the full-benchmark performance of the three methods using cumulative error histograms. The curves for our method and [13] are largely similar. The curve for [14] is above ours and [13] for about half of the images, but it flattens out below 85% while the others plateau at 100%. This is because method [14] struggled on the boy images.

It should be noted that the results reported in [14] for this test set are better than those obtained in our run of the code published online. This is because the code used to generate the reported results is an unpublished modification of the code base of [13] (*private communication*), while the online code is a simplified version. For reproducibility we have restricted our attention to published implementations.

Our method's kernel estimation step is up to 6 times faster than the others, whose computation time appears to depend superlinearly on kernel size (Fig. 4). The difference is mostly due to our use of cheap SPG iteration rather than quadratic programming in the k step, and because $\mathcal{P}_{\mathcal{K}}$ and $\mathcal{P}_{\mathcal{X}}$ make k and $\mathbf{x}$ sparse, enabling $k * \mathbf{x}$ to be computed faster.

4. CONCLUSION

We have proposed a blind deconvolution method in which the blur kernel is estimated by incremental sparse edge approximation. A rough global blur kernel is first estimated from only the strongest edges in the image, then it is refined as we allow the image edge map to gradually become less and less sparse. To our knowledge, this is the first simple, fast joint MAP method to match the state-of-the-art kernel MAP methods in [13, 14] on an objective benchmark. The results of [15] and this paper suggest that joint MAP remains viable if the concerns in [1] are appropriately addressed. The speed of our kernel estimation may make it useful as an input to high-quality non-blind methods.

5. REFERENCES

[1] A. Levin, Y. Weiss, F. Durand, and W.T. Freeman, "Understanding and evaluating blind deconvolution algorithms," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1964–1971, 2009.

[2] S. Cho, Y. Matsushita, and S. Lee, "Removing non-uniform motion blur from images," in *IEEE 11th International Conference on Computer Vision*, 2007, pp. 1–8.

[3] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce, "Non-uniform deblurring for shaken images," *International Journal of Computer Vision*, vol. 98, no. 2, pp. 168–186, 2012.

[4] M. Hirsch, C. J. Schuler, S. Harmeling, and B. Scholkopf, "Fast removal of non-uniform camera shake," in *IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2011, pp. 463–470.

[5] P. Campisi and K. Egiazarian, *Blind Image Deconvolution: Theory and Applications*, CRC Press, 2007.

[6] C.M. Bishop, *Pattern Recognition and Machine Learning*, Springer-Verlag New York, 2006.

[7] G.R. Ayers and J.C. Dainty, "Iterative blind deconvolution method and its applications," *Optics Letters*, vol. 13, no. 7, pp. 547–549, 1988.

[8] Y. L. You and M. Kaveh, "A regularization approach to joint blur identification and image restoration.," *IEEE Transactions on Image Processing*, vol. 5, no. 3, pp. 416–28, 1996.

[9] T. F. Chan and C.-K. Wong, "Total variation blind deconvolution," *IEEE Transactions on Image Processing*, vol. 7, pp. 370–375, Mar. 1998.

[10] R. Fergus, B. Singh, A. Hertzmann, S.T. Roweis, and W.T. Freeman, "Removing camera shake from a single photograph," in *ACM Transactions on Graphics*, 2006, vol. 25, pp. 787–794.

[11] Q. Shan, J. Jia, and A. Agarwala, "High-quality motion deblurring from a single image," in *ACM Transactions on Graphics*, 2008, vol. 27, p. 73.

[12] S. Cho and S. Lee, "Fast motion deblurring," in *ACM Transactions on Graphics*, 2009, vol. 28, p. 145.

[13] A. Levin, Y. Weiss, F. Durand, and W.T. Freeman, "Efficient marginal likelihood optimization in blind deconvolution," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 2657–2664.

[14] S. D. Babacan, R. Molina, M. N. Do, and A. K. Katsaggelos, "Bayesian blind deconvolution with general sparse image priors," in *European Conference on Computer Vision (ECCV)*, Firenze, Italy, October 2012, Springer.

[15] D. Krishnan, T. Tay, and R. Fergus, "Blind deconvolution using a normalized sparsity measure," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 233–240.

[16] J.F. Cai, H. Ji, C. Liu, and Z. Shen, "Framelet-based blind motion deblurring from a single image," *IEEE Transactions on Image Processing*, vol. 21, no. 2, pp. 562–572, 2012.

[17] C. Wang, L.F. Sun, Z.Y. Chen, J.W. Zhang, and S.Q. Yang, "Multi-scale blind motion deblurring using local minimum," *Inverse Problems*, vol. 26, no. 1, pp. 015003, 2009.

[18] L. Xu and J. Jia, "Two-phase kernel estimation for robust motion deblurring," *European Conference on Computer Vision*, pp. 157–170, 2010.

[19] M.E. Lopes, "Estimating unknown sparsity in compressed sensing," *arXiv preprint arXiv:1204.4227*, 2012.

[20] T. Blumensath and M.E. Davies, "Normalized iterative hard thresholding: Guaranteed stability and performance," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 298–309, 2010.

[21] R. Garg and R. Khandekar, "Gradient descent with sparsification: an iterative algorithm for sparse recovery with restricted isometry property," in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009, pp. 337–344.

[22] J.A. Tropp and A.C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4655–4666, 2007.

[23] D. Needell and J.A. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 301–321, 2009.

[24] H. Attouch, J. Bolte, and B. Svaiter, "Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward–backward splitting, and regularized gauss–seidel methods," *Mathematical Programming*, vol. 137, no. 1-2, pp. 91–129, February 2013.

[25] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, The MIT Press, 3rd edition, 2009.

[26] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra, "Efficient projections onto the ℓ_1-ball for learning in high dimensions," in *Proceedings of the 25th International Conference on Machine Learning*, 2008, pp. 272–279.

[27] E.G. Birgin, J.M. Martínez, and M. Raydan, "Nonmonotone spectral projected gradient methods on convex sets," *SIAM Journal on Optimization*, vol. 10, no. 4, pp. 1196–1211, 2000.

[28] A. Levin, R. Fergus, F. Durand, and W. T. Freeman, "Deconvolution using natural image priors," *ACM Transactions on Graphics (TOG)*, vol. 26, no. 3, pp. 0–2, 2007.