Inverse Kernels for Fast Spatial Deconvolution

Li Xu[†] Xin Tao[‡] Jiaya Jia[‡]

[†] Image & Visual Computing Lab, Lenovo R&T
[‡] The Chinese University of Hong Kong

Abstract. Deconvolution is an indispensable tool in image processing and computer vision. It commonly employs fast Fourier transform (FFT) to simplify computation. This operator, however, needs to transform from and to the frequency domain and loses spatial information when processing irregular regions. We propose an efficient spatial deconvolution method that can incorporate sparse priors to suppress noise and visual artifacts. It is based on estimating inverse kernels that are decomposed into a series of 1D kernels. An augmented Lagrangian method is adopted, making inverse kernel be estimated only once for each optimization process. Our method is fully parallelizable and its speed is comparable to or even faster than other strategies employing FFTs.

Keywords: deconvolution, inverse kernels, numerical analysis, optimization

1 Introduction

Deconvolution has been an essential tool for solving many image/video restoration and computer vision problems. It was also used in astronomy imaging [24], medical imaging [9], signal decoding, etc. In recent years, it is extensively applied to systems in computational photography and image/video editing, including flutter shutter motion deblurring [19], general motion deblurring [6, 30, 22, 4, 14, 10, 28, 25, 29, 21], coded aperture and depth [13, 32], and image super-resolution [2, 23, 17], since many types of degradation can be partly modeled or approximated by convolution, where kernels are monotonically decaying low-pass filters.

While convolution is easy to apply, its inverse problem of properly deconvolving images is not that simple. Band-limited convolution kernels have incomplete coverage in the frequency domain, which makes inversion ill-conditioned, especially under the existence of unavoidable quantization errors and camera noise. Regularization can remedy this problem – see early work of Wiener filtering [27] and Tikhonov deconvolution [26]. Existing methods are in two streams, which have their respective characteristics.

Spatial Deconvolution Very few deconvolution methods are performed in the spatial domain, owing to the high computational cost. Richardson-Lucy method [20] does not involve regularization and thus may suffer from the noise and ringing problems. Progressive approach [31] suppresses ringings by operations in image pyramids. Good performance is yielded in sparse prior deconvolution [13], which requires to solve large linear systems. With the re-weighting numerical scheme, the coefficient matrix of the linear system is no-longer Toeplitz and cannot be accelerated using FFTs. This indicates that sparse-prior deconvolution, albeit useful for preserving structures and suppressing ringings, is not translation invariant.

Deconvolution in Frequency Domain The convolution theorem states that spatial convolution can be computed by point-wise multiplication in frequency domain, which brings out pseudo-inversion in the frequency domain [16]. Shan et al. [22] fitted the gradient distribution using two convex functions. The half-quadratic implementation [11] mathematically links general α-norms to a family of hyper-Laplacian distributions. These iterative methods employ a few FFTs in each pass. Each FFT is with complexity $O(n \log n)$ where n is the pixel number in the image. Although frequency domain deconvolution is fast, it is non-trivial for further speedup by parallelization. Nor is it suitable to handle irregular regions, which however are common in object motion blur [3] and focal blur [13].

Our Contribution In this paper, we analyze the main difficulty of spatial deconvolution and propose a new numerical scheme based on inverse kernels to fill the gap between recent frequency-domain fast deconvolution and spatial pseudo-inverse. They are inherently linked in our system by introducing kernels constructed according to regularized optimization. The new relationship enables empirical strategies to inherit the nice properties in these two streams of work and to significantly speed up spatial deconvolution.

Although several useful sparse gradient priors may not lead to translation invariant process for deconvolution. We found it is possible to approximate them with a series of operators that are indeed spatially translation invariant. Accordingly, we propose an effective numerical scheme based on the augmented Lagrangian multipliers [15, 1] and kernel decomposition [18]. The resulting operations are no more than estimation of a set of 1D kernels that can be repeatedly applied to images in iterations.

Unlike all previous *fast* robust deconvolution techniques, our method works spatially and has a number of advantages. 1) It is easy to implement and parallelize. 2) It runs comparably with or even faster than FFT-based deconvolution for high-resolution images. 3) This method can deal with arbitrarily irregular regions without much computation overhead. 4) Visual artifacts are much reduced.

We apply our method to applications of extended depth of field [12], motion deblurring [29], and image upscaling using back projection [8].

2 Motivation and Analysis

To understand the inherent difference between spatial and frequency domain deconvolution, we begin with the discussion of convolution expressed in the form

$$y = x * k + \epsilon,$$

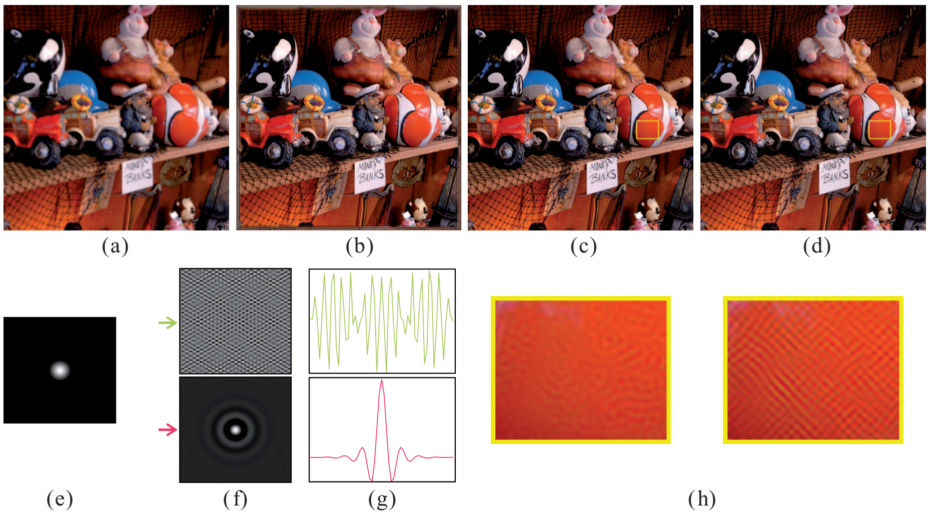


Fig. 1. Illustration of regularized inverse filters. (f). (a) is a Gaussian blurred image. (b)-(d) are the restored images by convolving the regularized inverse filter, Wiener deconvolution, and 1D separated Wiener deconvolution. (e) shows the Gaussian kernel. (f) shows the direct inverse filter, and regularized inverse filter from top down. (g) contains 1D scan lines of the two inverse filters in (f). (h) shows the close-ups of (c) and (d).

where k is the kernel, y is the degraded observation, x is the latent image, $*$ refers to the convolution operator, and ϵ indicates additive noise.

We first explain the inverse kernel problem using the simple Wiener deconvolution and then discuss the issues in designing a practical spatial solver using sparse gradient priors, which is effective to suppress noise and visual artifacts.

2.1 Spatial Inverse Kernels for Wiener Deconvolution

Wiener deconvolution introduces a pseudo-inverse filter in frequency domain, expressed as

$$W = \frac{\overline{F(k)}}{|F(k)|^2 + \frac{1}{SNR}}, \tag{1}$$

where $F(\cdot)$ denotes Fourier transform and $\overline{F(\cdot)}$ is its complex conjugate. SNR represents the signal to noise ratio that helps suppress the high frequency part of the inverse filter. The restored image is thus

$$x = F^{-1}(W \cdot F(y)), \tag{2}$$

where F^{-1} is the inverse Fourier transform.

Albeit efficient, restoration using FFTs loses the spatial information as discussed above and could be less favored in several applications. This motivates us to approximate this process using pseudo-inverse w in the spatial domain, expressed as

$$x = F^{-1}(W) * y = w * y, \tag{3}$$

where w is the latent (pseudo) spatial inverse kernel. It is known in signal processing that this task cannot always be accomplished given an arbitrary W. Taking the simple 2D Gaussian filter for example (Fig. 1(e)), its direct spatial inverse kernel is a 2D infinite impulse response (IIR) filter, as shown in the top of Fig. 1(f).

Contrarily, we found that the spatial counterpart of Wiener inversion, i.e. $F^{-1}(W)$, has a finite support, as shown in the bottom of Fig. 1(f). The difference is due to the involvement of regularization $1/SNR$. It is actually a general observation that *inverse filters with regularization are typically with decaying spatial responses.* An 1D visualization is given in Fig. 1(g). The kernel with regularization (bottom) decays quickly and thus has a compact support.

An image degraded by a Gaussian kernel (Fig. 1(e)) is shown in Fig. 1(a). The restored image using the spatial inverse kernel with compact support is given in Fig. 1(d), with visual artifacts near image border, which can be ameliorated by padding. To further increase the sharpness and suppress artifacts, we turn to a more advanced sparse gradient regularization.

2.2 Sparse Gradient Regularized Deconvolution

State-of-the-art deconvolution makes use of sparse gradient priors [13, 11], making the overall computation more complex than a Wiener one. In this paper, we propose a practical scheme to achieve spatial deconvolution even with these challenging highly non-convex sparse priors. We describe two issues in this process, which concern kernel size and non-separability of regularized deconvolution.

Kernel Size Spatial inverse kernels could be of considerable sizes. For a Gaussian kernel with variance $\sigma = 3$, the corresponding regularized inverse filter using Eq. (1) has a finite support of 51×51. Although it is independent of the input image size, it still lays a large computational burden to 2D convolution.

Kernel Non-separability Many kernels are inherently non-separable. Even for those that are separable, their inversions are not. For example, each Gaussian kernel can be decomposed into two 1D filters, applied in the horizontal and vertical directions respectively. However, its inversion is not separable due to regularization. The road to speeding up regularized deconvolution by simply performing 1D filtering is thus blocked.

The comparison in Fig. 1(c) and (d) illustrates the difference. There is a 2D inverse kernel of Gaussian created according to Eq. (1) and a separated approximation using outer product of two 1D filters, formed also following Eq.

(1). The restoration result using the re-combined 1D filters is shown in (d). It contains obvious oblique-line artifacts (see the close-ups in (h)).

We address these two issues using kernel decomposition with SVD, presented below.

3 Sparse Prior Robust Spatial Deconvolution

Sparse gradient regularized deconvolution works very well with a hyper Laplacian prior [11]. It minimizes the function of

$$E(x) = \sum_{i=1}^{n} \left(\frac{\lambda}{2}(x * k - y)_i^2 + |c_1 * x|_i^\alpha + |c_2 * x|_i^\alpha \right), \tag{4}$$

where i indexes image pixels. c_1 and c_2 are finite differential kernels in horizontal and vertical directions to approximate the first-order derivatives. α controls the shape of the prior with $0.5 \leq \alpha < 1$. A common way to solve this function is to employ a penalty decomposition

$$E(x; z_1, z_2) = \sum_{i=1}^{n} \left(\frac{\lambda}{2}(x * k - y)_i^2 + \sum_{j \in \{1,2\}} \frac{\beta}{2}(z_j - c_j * x)_i^2 + |z_j|_i^\alpha \right), \tag{5}$$

where z_1 and z_2 are auxiliary variables to approximate regularizers. The problem approaches the original one only if β is large enough. The solver is thus formed as iteratively updating variables as

$$z_j^{t+1} \leftarrow \mathrm{argmin}_z E(x^t, z_j, \beta^t), \tag{6}$$
$$x^{t+1} \leftarrow \mathrm{argmin}_x E(x, z_j^{t+1}, \beta^t), \tag{7}$$
$$\beta^{t+1} \leftarrow 2\beta^t. \tag{8}$$

t indexes iterations. Since z_j has an analytical solution (or can be found in look-up tables) [11], the main computation lies in the FFT inversion step to compute x, which gives

$$x = F^{-1}\left(\frac{\sum_j \overline{F(c_j)}F(z_j) + \frac{\lambda}{\beta}\overline{F(k)}F(y)}{\sum_j |F(c_j)|^2 + \frac{\lambda}{\beta}|F(k)|^2} \right). \tag{9}$$

It involves several FFTs. Basically, update of z_j is performed in spatial domain as it involves pixel-wise operations. So domain switch is unavoidable.

3.1 Penalty Decomposition Inverse Kernels

We expand Eq. (9) by decomposing the numerator and denominator and apply inverse FFT separately. It yields

$$x = F^{-1}\left(\frac{1}{\sum_j |F(c_j)|^2 + \frac{\lambda}{\beta}|F(k)|^2} \right) * \left(\sum_j c_j' * z_j + \frac{\lambda}{\beta} k' * y \right), \tag{10}$$

where c'_j and k' are adjoint kernels of c_j and k by rotating these kernels by 180 degree, and j indexes differential kernels c. The operations $c'_j * z_j$ and $k' * y$ are now in spatial domain. $k' * y$ is a constant independent of variables z and x.

$(\sum |F(c_j)|^2 + \frac{\lambda}{\beta}|F(k)|^2)^{-1}$ in Eq. (10) is the inversion in the frequency domain. Its domain switch to pixel values, in fact, corresponds to a spatial inverse kernel. The regularization makes its finite support exist. So it is possible to estimate spatial inverse kernels corresponding to this term, i.e.,

$$w_\beta = F^{-1}\left(\frac{1}{\sum_j |F(c_j)|^2 + \frac{\lambda}{\beta}|F(k)|^2}\right). \tag{11}$$

This process raises a technical challenge. Because β varies in iterations, w_β needs to be re-estimated in each pass. A series of spatial inverse filters thus should be produced, which are not optimal and waste much time.

3.2 Augmented Lagrangian Inverse Kernels

To fit the spatial processing framework, we adopt the *augmented Lagrangian* (AL) method [15, 5] to approximate deconvolution. AL was originally used to transform constrained optimization to an unconstrained one with the conventional *Lagrangian* and an additional augmented penalty term. Specifically, we transform Eq. (4) into

$$\begin{aligned} E(x; z_j, \gamma_j) = \sum_{i=1}^{n} \Bigg(& \frac{\lambda}{2}(x * k - y)_i^2 + \sum_{j\in\{1,2\}} |z_j|_i^\alpha \\ & + \sum_{j\in\{1,2\}} \frac{\beta}{2}(z_j - c_j * x)_i^2 - \langle \gamma_j, (z_j - c_j * x)\rangle_i \Bigg), \end{aligned} \tag{12}$$

where the term in the second row is the augmented Lagrangian multiplier specific for this problem. $\langle\cdot\rangle$ is the inner product of two vectors. The major difference from the original penalty decomposition optimization is that here the update of γ_j prevents β from varying while the optimization still proceeds nicely. The iterative solver is given by

$$z_j^{t+1} \leftarrow \text{argmin}_z E(x^t, z_j, \gamma_j^t), \tag{13}$$

$$x^{t+1} \leftarrow \text{argmin}_x E(x, z_j^{t+1}, \gamma_j^t), \tag{14}$$

$$\gamma_j^{t+1} \leftarrow \gamma_j^t - \beta(z_j^{t+1} - c_j * x^{t+1}). \tag{15}$$

From the convergence point of view, the AL method has basically no difference with penalty decomposition. But it is much more suitable for our deconvolution framework, in which β can be fixed, resulting in the same inverse kernel in all iterations.

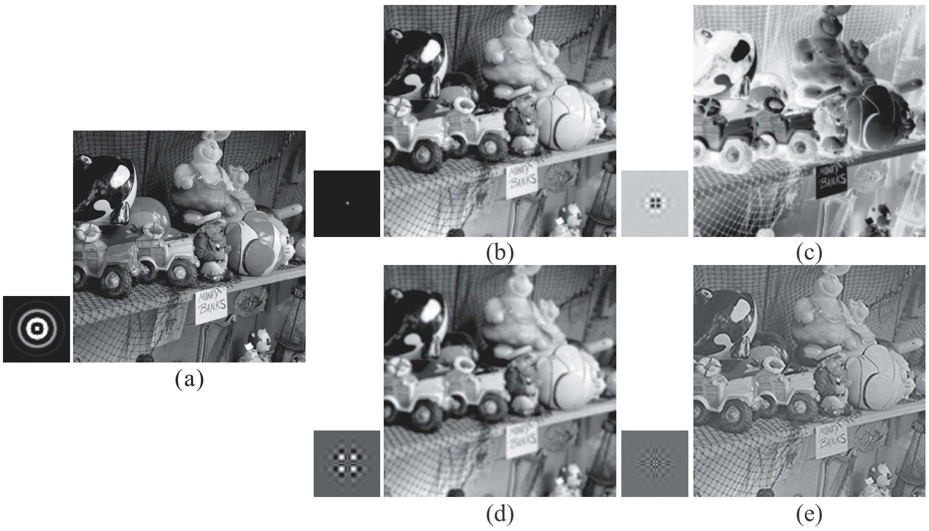


Fig. 2. Separating filters. A spatial inverse filter shown in (a) can be approximated as a linear combination of a few simpler ones as shown from (b)-(e). Each of them is separable. The finally restored image in (a) can be formed as a linear combination of images restored by these simple filters respectively.

By re-organizing the terms, we get an expression for the target image:

$$\begin{aligned} x = F^{-1}\left(\frac{1}{\sum_j |F(c_j)|^2 + \frac{\lambda}{\beta}|F(k)|^2}\right) * \\ F^{-1}\left(\sum_i \overline{F(c_j)}(F(z_j) - \frac{1}{\beta}F(\gamma_j)) + \frac{\lambda}{\beta}\overline{F(k)}F(y)\right), \\ = w_\beta * \left(\sum_{j\in\{1,2\}} c'_j * (z_j - \frac{1}{\beta}\gamma_j) + \frac{\lambda}{\beta}k' * y\right), \end{aligned} \tag{16}$$

where w_β denotes the same spatial inverse filter defined in Eq. (11). The difference is that β in this form no longer varies during iterations. $c'_j * (z_j - \frac{1}{\beta}\gamma_j)$ can be efficiently computed using forward/backword difference. $k' * y$ is a constant and can be computed only once before the iteration. w_β is a spatial inverse kernel that can also be pre-computed and stored.

It seems now we successfully produce workable inverse kernel without heavy computation spent to re-estimating it in each iteration. But there are still two aforementioned *size* the *separability* issues that may influence deconvolution efficiency. We further propose a decomposition procedure to address them.

Inverse Kernel Decomposition Kernel decomposition techniques have been widely explored. Steerable filters [7] decompose kernels into linear combination of a set of basis filters. Another kernel decomposition is based on the singular value decomposition (SVD) of w_β by treating it as a matrix [18]. Compared to steerable filter, it is a non-parametric decomposition for arbitrary filters.

Given our spatial kernel w_β, we decompose it as $w_\beta = USV'$, where U and V' are unitary orthogonal matrices, V' is the transpose of V, and the matrix S is a band-diagonal matrix with nonnegative real numbers in the diagonal. We use f_u^l and f_v^l to denote the l^{th} column vectors of U and V, which in essence are 1D filters. w_β is expressed as

$$w_\beta = \sum_l s_l f_u^l f_v^{l\prime}. \tag{17}$$

Convolving w_β with an image is now equivalent to convolving a set of 1D kernels f_u^l and f_v^l. It can be efficiently applied in spatial domain where the number of filters is controlled by the non-zero elements in the singular value matrix, in line with the rank of the kernel.

If a kernel is spatially smooth, which is common for natural images, the rank can be very small. It is thus allowed to use only a few 1D kernels to perform deconvolution. Note that we can even lower the approximation precision by dropping small non-zeros singular values for further acceleration. One example of the kernel and its decomposition is shown in Fig. 2. The filtered images are shown together with their separable filters. In this examples, 7 separable filters are used to approximate the inverse regularized Gaussian, which verifies that most inverse kernels are not originally separable.

3.3 More Discussions

Our spatial deconvolution is an iterative process. For each deconvolution process, we only need to use SVD to estimate w_β as several 1D kernels once. If the kernel was decomposed before, w_β is stored in our files for quick lookup. In this regard, common kernels, such as Gaussians, can be pre-computed to save computation during deconvolution.

The spatial support of the 1D inverse kernels depends on the amount of regularization,i.e. the weight λ. For noisy images, λ is set small, corresponding to strong regularization. Accordingly, the size of inverse kernels is small. In practice, the support of 1D kernels is estimated by thresholding insignificant values in the kernel and removing boundary zero values, which are determined automatically once λ is given.

The pseudo-code for inverse kernel deconvolution is provided in Alg. 1.

4 Experimental Validation

We evaluate the system performance with regard to running time and result quality. Our main objective is to handle focal, Gaussian or even sparse motion blur. In our implementation, primary parameters in Eq. (12) are set as follows: $\lambda \in [500, 3000]$, depending on the image noise level; β is fixed to 10 for all

Algorithm 1 $x = \textsc{FastSpatialDeconvolution}(y,\, k)$

```
1  $w_\beta \leftarrow \mathrm{real}\left(F^{-1}\left(\frac{1}{\sum_i |F(c_i)|^2 + \frac{\lambda}{\beta}|F(k)|^2}\right)\right)$
2  $\{s_l, f_u^l, f_v^l\} \leftarrow \mathrm{svd}(w_\beta)$
3  Discard $\{s_l, f_u^l, f_v^l\}$ pairs with $s_l$ below a threshold
4  $x^1 \leftarrow y,\ \gamma^1 \leftarrow 0$
5  for $t = 1$ to maxIters
6        do $z_i^{t+1} \leftarrow \mathrm{argmin}_z E(x^t, z_i, \gamma_i^t)$
7           $a \leftarrow \sum_{i \in \{1,2\}} c_i' * (z_i^{t+1} - \frac{1}{\beta}\gamma_i^t) + \frac{\lambda}{\beta} k' * y$
8           $x^{t+1} \leftarrow 0$
9           for $l = 1$ to $length(\{s_l\})$
10                do $x^{t+1} \leftarrow x^{t+1} + s_l \cdot a * f_u^l * {f_v^l}'$
11          $\gamma_i^{t+1} \leftarrow \gamma_i^t - \beta(z_i^{t+1} - c_i * x^{t+1})$
```

Table 1. Running time (in seconds) and PSNRs for different methods

Image Size	RL	IRLS	TVL1	Fast PD	Ours
325x365	0.91	85.83	7.50	0.59	0.57
1064x694	2.28	241.34	22.20	2.00	3.27
1251x1251	6.19	537.89	54.30	4.61	7.30
PSNRs	20.2	24.3	22.7	23.3	23.7

images; totally 5 iterations are enough in practice. We compared our method with others, including the spatial-domain Richardson-Lucy (RL) deconvolution, IRLS [13] (short for the iterative re-weighted least squares) approach, TVL1 deconvolution [28] and the fast deconvolution [11], denoted as PD for "penalty decomposition". The TVL1 method is implemented in C language and all the other four methods are implemented in MATLAB. We run 20 iterations for the standard RL. All other methods are based on the authors' implementation with default parameters.

Running time is obtained on different sizes of images. In total, we collect 10 natural images with different resolutions. They are blurred with Gaussian filters with variance $\sigma \in \{1, 2, 3, 4, 5\}$ respectively. Small Gaussian noise is added to each image. Running time for three resolutions is reported in Table 1. Our method is similarly fast as PD employing FFTs and is a magnitude faster than IRLS and TVL1. Our method updates z with analytical solutions. It can be further sped up by using a look-up table. As w_β is pre-computed, we do not include its estimation time in the table. In our experiments, a 51×51 kernel is computed in 0.1 second. The final PSNRs of all the 10 examples are included in Table 1.

We show in Fig. 3 a visual comparison along with close-ups for different methods. Our result is comparable with the sharpest one while not containing extra visual artifacts.

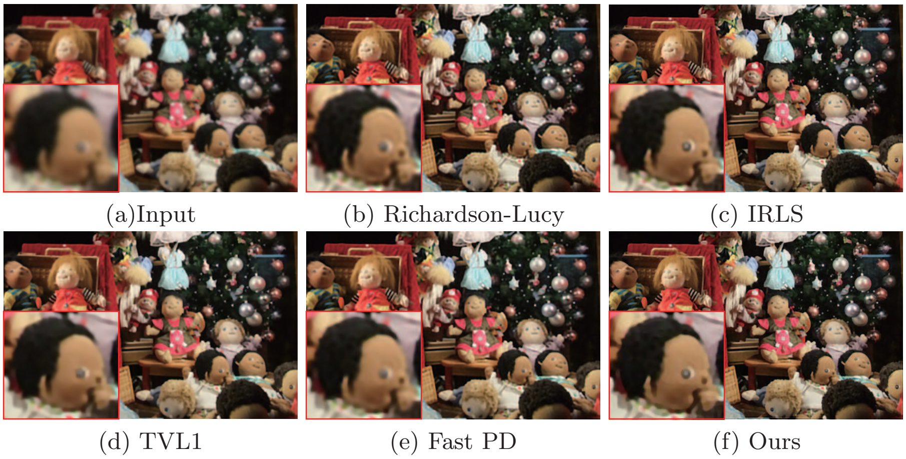


Fig. 3. Visual comparison. Similar quality results manifest that our method does not introduce additional visual artifacts.

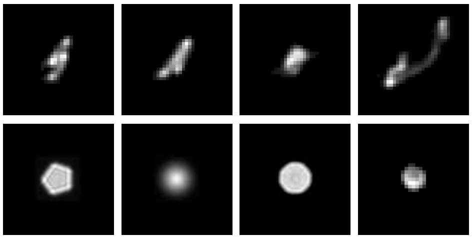


Fig. 4. Sample motion and focal blur kernels for validation.

Statistics of Filters We now present the statistics of the 1D filters w_β learned from different types of kernels. We collected a set of filters in real motion blur, representative Gaussian convolution, and natural out-of-focus. The 8 motion blur kernels are from [14]. The Gaussian blur kernels are with different scales, controlled by variance $\sigma \in \{1, 2, 3, 4, 5\}$. We also collect from internet the real focal blur kernels. We normalize all of them to size 35×35. A few examples are shown in Fig. 4.

The statistics in Table 2 indicate that motion deconvolution typically requires more 1D kernels to approximate the inverse filter than others, due primarily to large kernel variation and complex shapes. Convolving tens of kernels that approximate w_β is in fact a completely parallel process and can be easily accelerated using multiple-core CPU and GPU.

The number of 1D kernels is determined by thresholding the singular values and dropping out insignificant ones. Varying the threshold results in different numbers of 1D kernels and thus affects the performance. We show in Fig. 5 how the threshold affects the quality of restored images. One threshold can be applied to different types of kernels to generate reasonable results. We also note based

Table 2. Kernel decomposition statistics. "Average number" refers to the average number of non-zero singular values, i.e., the number of 1D filters used. "Average length" is the length of each 1D kernel.

Type	Avg. number	Avg. length
Motion	36.4	110.3
Gaussian	8.3	71.2
Out-of-focus	15.7	87.8

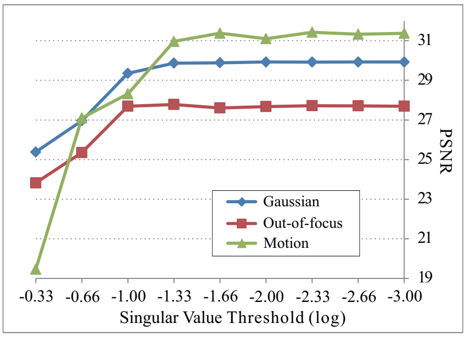


Fig. 5. PSNRs versus singular value thresholds for different types of kernels. The singular value thresholds are plotted in a logarithmic scale.

on Table 2 that one threshold may generate different numbers of 1D kernels depending on the structure and complexity of the original convolution kernels.

5 Applications

We apply our method to a few computer vision and computational photography applications.

5.1 Deconvolution-Intensive Super-Resolution

Iterative back-projection [8] is one effective scheme to upscale images and videos, and is fast in general. In this process, reconstruction errors are back projected into the high resolution image through interpolation and deconvolution, expressed as

$$h^{t+1} = h^t + (l - (h^t * G) \downarrow) \uparrow * p, \tag{18}$$

where G is a kernel that could be Gaussian [8] or non-Gaussian [17], h is the target high-resolution image and l is its low-resolution version. $\downarrow$ and $\uparrow$ are simple downscaling and upscaling with interpolation operations. p is the pseudo-inverse of the kernel. A good p positively influences high-quality image super-resolution. So we substitute our spatial deconvolution for p, which counts in regularization

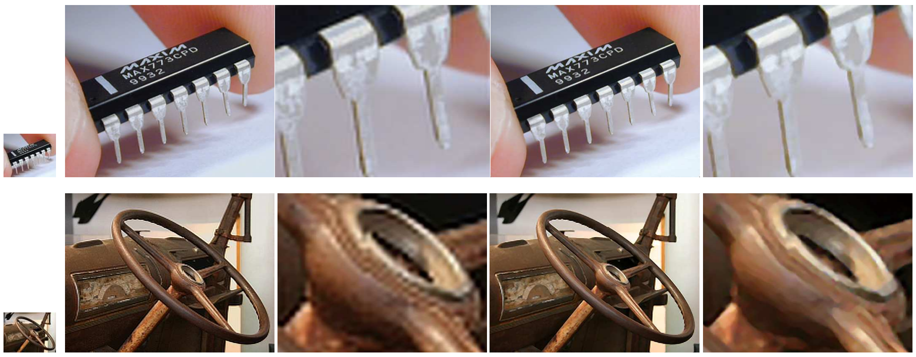


Fig. 6. Super-resolution by back-projection.

in deconvolution. It produces the results shown in Fig. 6. They demonstrate the usefulness of our inverse kernel scheme, as visual artifacts are suppressed.

5.2 Extended Depth of Field

The proposed method can be applied to removal of part of focal blur. We employ it in the extended depth of field photography [12], which generates a blurry image for each depth layer and restores it using deconvolution. Blurry image generation is achieved by controlling the motion of the detector during image integration or rotating the focus ring. Since the resulting blur PSFs belong to the generalized Gaussian family, they can be efficiently computed using our spatial scheme. Fig. 7 shows two examples. It takes 1.7s by our method on a single CPU core to produce the results shown in (b) with resolution 681×1032. In comparison, the fast deconvolution method [11] takes 2s to produce the results in (c). Our method can be fully parallelized to much speed up computation.

5.3 Motion Deblurring

Motion blur kernels are in general asymmetric, corresponding to a larger number of 1D kernels in our decomposition step. It reveals the non-separable nature of motion kernels. Our inverse kernel scheme is still applicable here thanks to the independence of each 1D filtering pass. We show in Fig. 8 the IRLS deconvolution results of [13] and our inverse filter results. The ground truth clear images and motion blur kernels are presented in the original paper [14]. While both approaches work in spatial domain, ours takes 0.5s to process the 255×255 images, compared to the 70 seconds by the IRLS method.

5.4 Real-time Partial Blur Removal

Our method directly helps partial image deconvolution. Fourier transform requires square inputs and any error produced after domain switch will be prop-

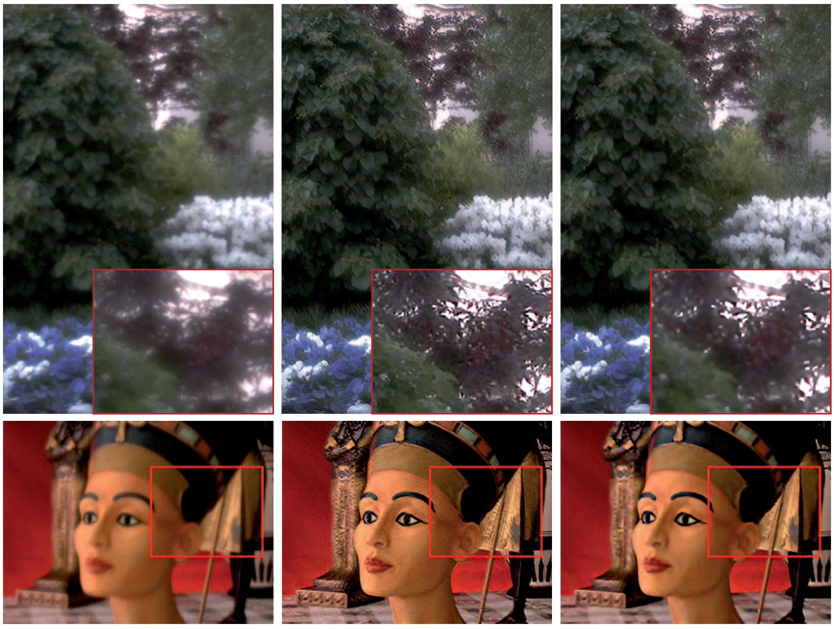


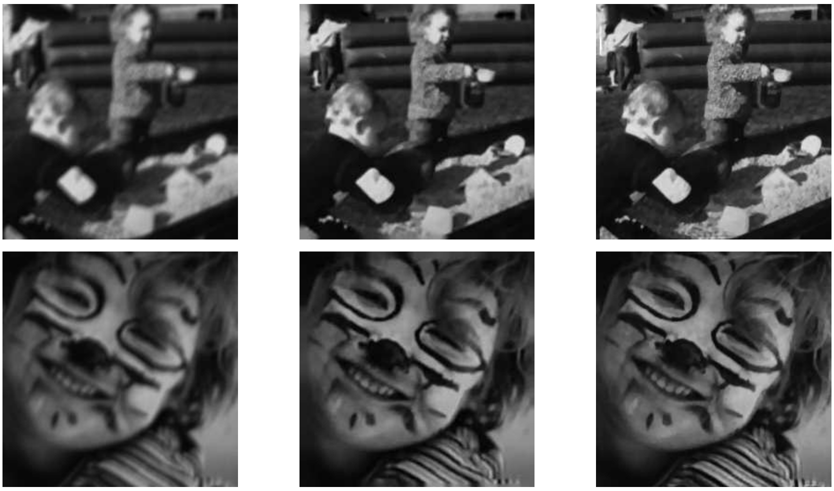
Fig. 7. Reconstructed pictures from extended depth of field cameras.

agated across pixels due to the lack of spatial consideration. Our method does not have these constraints. Our current implementation can achieve real-time performance on 130×130 patches on a single CPU core. It is notable that any shapes of regions can be handled in this system. Our empirically processed regions are slightly expanded from the user marked ones to include more pixels in optimization in order to avoid boundary visual artifacts.

One example is shown in Fig. 9, where a book is focal blurred. We restore a patch using our method, which does not introduce unexpected ringing artifacts. Our method takes only 0.07 second to process the content, compared with 0.4 second needed in the FFT-based method [11] to process all pixels within the tightest bounding box enclosing the selected region. The close-ups are shown in (c) and (d). The difference is caused by processing only the marked pixels by our method and processing all pixels in the rectangular bounding box by the FFT-involved method.

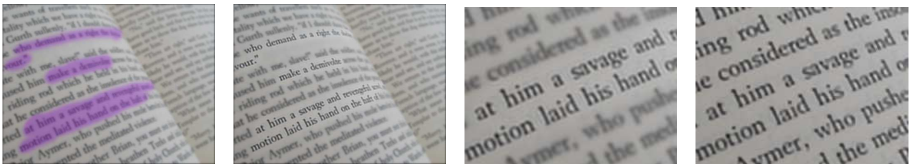
6 Conclusion

We have presented a spatial deconvolution method leveraging the pseudo-inverse spatial kernels under regularization. Fixed kernel estimation is achieved using the augmented Lagrangian method. Our framework is general and finds many applications. Its impact is the numerical bridge to connect fast frequency-domain



(a) Input (b) IRLS [13] (c) Ours

Fig. 8. Motion deblurring examples.



(a) Input (b) Our result (c) Close-up (Ours) (d) Close-up (PD)

Fig. 9. Partial Blur Removal. In (a), we mark a few pixels for deconvolution. The result is shown in (b) with the close-up in (c). The FFT-based method (PD) yields the result shown in (d) by devolving all pixels in the bounding box.

operations and robust local spatial deconvolution. Our method inherits the speed and location-sensitivity advantages in these two streams of work and opens up a new area for future exploration.

The method could be amazingly efficient if these 1D kernel bases involved in decomposition are handled by different threads in the parallel computing architecture. It works well for general Gaussian and other practical motion and focal blur kernels. One direction for future work is to investigate spatially varying inverse kernels for complex blur.

Acknowledgements

The work described in this paper was partially supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region (Project No. 413113). The authors would like to thank Shicheng Zheng for his help in implementing part of the algorithm.

References

1. Afonso, M.V., Bioucas-Dias, J.M., Figueiredo, M.A.: An augmented lagrangian approach to the constrained optimization formulation of imaging inverse problems. Image Processing, IEEE Transactions on 20(3), 681–695 (2011)
2. Agrawal, A.K., Raskar, R.: Resolving objects at higher resolution from a single motion-blurred image. In: CVPR (2007)
3. Chakrabarti, A., Zickler, T., Freeman, W.T.: Analyzing spatially-varying blur. In: CVPR. pp. 2512–2519 (2010)
4. Cho, S., Lee, S.: Fast motion deblurring. ACM Trans. Graph. 28(5) (2009)
5. Danielyan, A., Katkovnik, V., Egiazarian, K.: Image deblurring by augmented lagrangian with bm3d frame prior. In: Workshop on Information Theoretic Methods in Science and Engineering. pp. 16–18 (2010)
6. Fergus, R., Singh, B., Hertzmann, A., Roweis, S.T., Freeman, W.T.: Removing camera shake from a single photograph. ACM Trans. Graph. 25(3), 787–794 (2006)
7. Freeman, W.T., Adelson, E.H.: The design and use of steerable filters. IEEE Trans. Pattern Anal. Mach. Intell. 13(9), 891–906 (1991)
8. Irani, M., Peleg, S.: Motion analysis for image enhancement: Resolution, occlusion, and transparency. Journal of Visual Communication and Image Representation 4(4) (1993)
9. Jerosch-Herold, M., Wilke, N., Stillman, A., Wilson, R.: Magnetic resonance quantification of the myocardial perfusion reserve with a fermi function model for constrained deconvolution. Medical physics 25, 73 (1998)
10. Joshi, N., Zitnick, C.L., Szeliski, R., Kriegman, D.J.: Image deblurring and denoising using color priors. In: CVPR. pp. 1550–1557 (2009)
11. Krishnan, D., Fergus, R.: Fast image deconvolution using hyper-laplacian priors. In: NIPS (2009)
12. Kuthirummal, S., Nagahara, H., Zhou, C., Nayar, S.K.: Flexible depth of field photography. IEEE Trans. Pattern Anal. Mach. Intell. 33(1), 58–71 (2011)
13. Levin, A., Fergus, R., Durand, F., Freeman, W.T.: Image and depth from a conventional camera with a coded aperture. ACM Trans. Graph. 26(3), 70 (2007)
14. Levin, A., Weiss, Y., Durand, F., Freeman, W.T.: Understanding and evaluating blind deconvolution algorithms. In: CVPR. pp. 1964–1971 (2009)
15. Lin, Z., Chen, M., Ma, Y.: The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. UIUC Technical Report UILU-ENG-09-2215 (2010)
16. Mathews, J., Walker, R.L.: Mathematical methods of physics, vol. 271. WA Benjamin New York (1970)
17. Michaeli, T., Irani, M.: Nonparametric blind super-resolution. In: ICCV (2013)
18. Perona, P.: Deformable kernels for early vision. IEEE Trans. Pattern Anal. Mach. Intell. 17(5), 488–499 (1995)
19. Raskar, R., Agrawal, A.K., Tumblin, J.: Coded exposure photography: motion deblurring using fluttered shutter. ACM Trans. Graph. 25(3), 795–804 (2006)
20. Richardson, W.: Bayesian-based iterative method of image restoration. Journal of the Optical Society of America 62(1), 55–59 (1972)
21. Schmidt, U., Rother, C., Nowozin, S., Jancsary, J., Roth, S.: Discriminative non-blind deblurring. In: CVPR. pp. 604–611. IEEE (2013)
22. Shan, Q., Jia, J., Agarwala, A.: High-quality motion deblurring from a single image. ACM Trans. Graph. 27(3) (2008)

23. Shan, Q., Li, Z., Jia, J., Tang, C.K.: Fast image/video upsampling. ACM Trans. Graph. 27(5), 153 (2008)
24. Starck, J., Pantin, E., Murtagh, F.: Deconvolution in astronomy: A review. Publications of the Astronomical Society of the Pacific 114(800), 1051–1069 (2002)
25. Tai, Y.W., Lin, S.: Motion-aware noise filtering for deblurring of noisy and blurry images. In: CVPR. pp. 17–24 (2012)
26. Tikhonov, A., Arsenin, V., John, F.: Solutions of ill-posed problems (1977)
27. Wiener, N.: Extrapolation, interpolation, and smoothing of stationary time series: with engineering applications. Journal of the American Statistical Association 47(258) (1949)
28. Xu, L., Jia, J.: Two-phase kernel estimation for robust motion deblurring. In: ECCV (1). pp. 157–170 (2010)
29. Xu, L., Zheng, S., Jia, J.: Unnatural l0 sparse representation for natural image deblurring. In: CVPR. pp. 1107–1114 (2013)
30. Yuan, L., Sun, J., Quan, L., Shum, H.Y.: Image deblurring with blurred/noisy image pairs. ACM Trans. Graph. 26(3), 1 (2007)
31. Yuan, L., Sun, J., Quan, L., Shum, H.Y.: Progressive inter-scale and intra-scale non-blind image deconvolution. ACM Trans. Graph. 27(3) (2008)
32. Zhou, C., Lin, S., Nayar, S.K.: Coded aperture pairs for depth from defocus and defocus deblurring. International Journal of Computer Vision 93(1), 53–72 (2011)