

Contents

<i>List of contributors</i>	<i>page</i> 2
2 Mathematical models and practical solvers for uniform motion deblurring Jiaya Jia	1
2.1 Non-blind deconvolution	1
2.1.1 Regularized approaches	3
2.1.2 Iterative approaches	6
2.1.3 Recent advancement	7
2.1.4 Variable splitting solver	9
2.1.5 A few results	12
2.2 Blind deconvolution	13
2.2.1 Maximum marginal probability estimation	15
2.2.2 Alternating energy minimization	16
2.2.3 Implicit edge recovery	18
2.2.4 Explicit edge prediction for very large PSF estimation	19
2.2.5 Results and running time	28

List of contributors

Jiaya Jia

The Chinese University of Hong Kong

2 Mathematical models and practical solvers for uniform motion deblurring

Jiaya Jia

Recovering an unblurred image from a single motion-blurred picture has long been a fundamental research problem. If one assumes that the blur kernel – or point spread function (PSF) – is shift-invariant, the problem reduces to that of image deconvolution. Image deconvolution can be further categorized to the blind and non-blind cases.

In non-blind deconvolution, the motion blur kernel is assumed to be known or computed elsewhere; the task is to estimate the unblurred latent image. The general problems to address in non-blind deconvolution include reducing possible unpleasing ringing artifacts that appear near strong edges, suppressing noise, and saving computation. Traditional methods such as Weiner deconvolution (Wiener 1949) and Richardson-Lucy (RL) method (Richardson 1972, Lucy 1974) were proposed decades ago and find many variants thanks to their simplicity and efficiency. Recent development involves new models with sparse regularization and proposal of effective linear and non-linear optimization to improve result quality and further reduce running time.

Blind deconvolution is a much more challenging problem, since both the blur kernel and latent image are unknown. One can regard non-blind deconvolution as one inevitable step in blind deconvolution during the course of PSF estimation or after PSF has been computed. Both blind and non-blind deconvolution is practicably very useful, which is studied and employed in a variety of disciplines, including, but not limited to, image processing, computer vision, medical and astronomic imaging and digital communication.

This chapter discusses shift-invariant single image motion deblurring methods, which assume that the image is uniformly blurred with only one PSF, which may not be known in prior. This set of problems have a long history in theoretical and empirical research and are notably advanced in recent 5-10 years with a few remarkably effective models and solvers.

2.1 Non-blind deconvolution

Ideally, a blur observation is modeled as a linearly filtered version of the latent unblurred signal. This process can be expressed as

$$\mathbf{b} = \mathbf{l} \otimes \mathbf{f}, \quad (2.1)$$

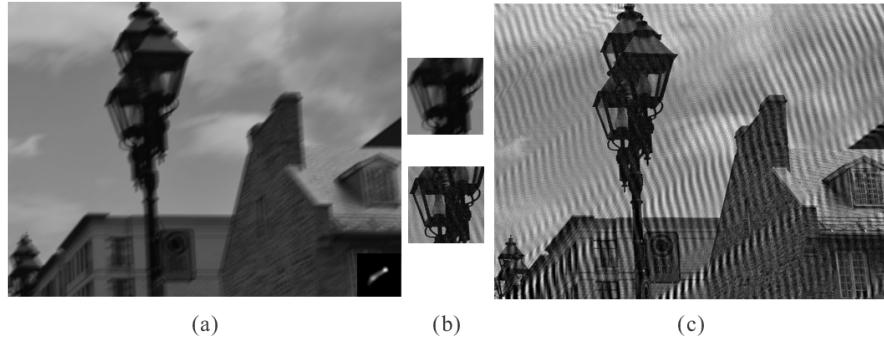


Figure 2.1 Visual artifacts caused by inverse filter. (a) Blurred image and PSF. (b) Close-ups. (c) Output of inverse filtering.

where \mathbf{b} , \mathbf{l} and \mathbf{f} are the blurred image, latent unblurred image, and PSF (or blur kernel) respectively. In the frequency domain,

$$\mathcal{F}(\mathbf{b}) = \mathcal{F}(\mathbf{l}) \cdot \mathcal{F}(\mathbf{f}), \quad (2.2)$$

where \mathcal{F} is the Fourier transform.

If $\mathcal{F}(\mathbf{f})$ does not contain zero or very small values and the blurred image is noise-free, the latent image \mathbf{l} can be obtained simply by inverting the convolution process using inverse filtering, the simplest method that solves for \mathbf{l} . This process is expressed as

$$\mathcal{F}(\mathbf{l}) = \mathcal{F}(\mathbf{b}) / \mathcal{F}(\mathbf{f}). \quad (2.3)$$

This strategy practically may produce severe visual artifacts, such as ringings, with the following reasons. First, inversion of \mathbf{f} may not exist, especially for low-pass filters. Second, motion PSFs caused by object or camera motion are typically band-limited and their spectrums have zero or near-zero values at high frequency. Third, image formation causes problems including image noise, quantization error, color saturation, and non-linear camera response curve. They make blur violate the ideal convolution model and lead to a more flexible form

$$\mathbf{b} = \mathbf{l} \otimes \mathbf{f} + \mathbf{n}, \quad (2.4)$$

where \mathbf{n} denotes error in the blurred image, which we call image noise in general. One deconvolution result by direct inverse filter is shown in Figure 2.1.

Development of more advanced non-blind deconvolution methods dated back to 1970's. Early representative approaches include Wiener Deconvolution (Wiener 1949), Least Square Filtering (Miller 1970, Hunt 1973, Tikhonov & Arsenin 1977), Richardson-Lucy method (Richardson 1972, Lucy 1974) and recursive Kalman Filtering (Woods & Ingle 1981). Readers are referred to (Andrews & Hunt 1977) for a review of these early approaches.

Simply put, many algorithms minimize an energy consisting of two terms, i.e., the *data* term E_{data} (corresponding to *likelihood* in probability) and *regularization*

(also known as *prior*) E_{prior} . E_{data} measures the difference between the convolved image and the blur observation, and is written as

$$E_{\text{data}} = \Phi(\mathbf{l} \otimes \mathbf{f} - \mathbf{b}), \quad (2.5)$$

where Φ is a distance function. A common definition is $\Phi(\cdot) = \|\cdot\|^2$ (Wiener 1949), representing the L2-norm of all elements. It is also called Gaussian likelihood. E_{prior} is denoted as a function $\Psi(\mathbf{l})$, which has different specifications in existing approaches. Given E_{data} and E_{prior} , latent image \mathbf{l} can be estimated by minimizing the energy incorporating these two terms, expressed as

$$\min_{\mathbf{l}} \|\mathbf{l} \otimes \mathbf{f} - \mathbf{b}\|^2 + \lambda \Psi(\mathbf{l}), \quad (2.6)$$

where λ is a weight. In what follows, we discuss a few representative non-blind deconvolution methods with respect to model design and solver construction. Their respective strength, disadvantage, and relation are also presented.

2.1.1 Regularized approaches

A number of early methods incorporated square regularization constraints. Two representative forms are $\Psi(\mathbf{l}) = \|\mathbf{l}\|^2$ and $\Psi(\mathbf{l}) = \|\nabla \mathbf{l}\|^2$, where ∇ is the gradient operator. They enforce smoothness on image values and image gradients, and are called Tikhonov and Gaussian regularizers respectively. Substituting them into Eq. (2.6) yields

$$\min_{\mathbf{l}} \|\mathbf{l} \otimes \mathbf{f} - \mathbf{b}\|^2 + \lambda \|\mathbf{l}\|^2 \quad \text{and} \quad (2.7)$$

$$\min_{\mathbf{l}} \|\mathbf{l} \otimes \mathbf{f} - \mathbf{b}\|^2 + \lambda \|\nabla \mathbf{l}\|^2, \quad (2.8)$$

for overall energy minimization. Weight λ is typically a small value.

The main advantage of these constrained least square methods is on the simplicity of formation, which results in a solver similar to inverse filter. Taking the Tikhonov method as an example, there exists a closed form solution \mathbf{l}^* for Eq. (2.7) by setting its first order derivative to zero with respect to \mathbf{l} . Rearranging Eq. (2.7) in a matrix form and denoting by E the total energy yield

$$\begin{aligned} E &= \|F\nu(\mathbf{l}) - \nu(\mathbf{b})\|^2 + \lambda \|\nu(\mathbf{l})\|^2 \\ &= \nu(\mathbf{l})^T F^T F \nu(\mathbf{l}) - 2\nu(\mathbf{l})^T F \nu(\mathbf{l}) + \nu(\mathbf{l})^T \nu(\mathbf{l}) + \lambda \nu(\mathbf{l})^T \nu(\mathbf{l}), \end{aligned}$$

where F is a sparse convolution matrix generated from \mathbf{f} , and ν is the operator that transforms the image into its vector form. The partial derivative is

$$\frac{\partial E}{\partial \nu(\mathbf{l})} = 2F^T F \nu(\mathbf{l}) - 2F^T \nu(\mathbf{b}) + 2\lambda \nu(\mathbf{l}). \quad (2.9)$$

By setting the above equation to zero, the optimal solution \mathbf{l}^* is

$$\nu(\mathbf{l}^*) = \frac{F^T}{F^T F + \lambda \Lambda} \nu(\mathbf{b}), \quad (2.10)$$

where Λ is an identity matrix, the same size as $F^T F$.

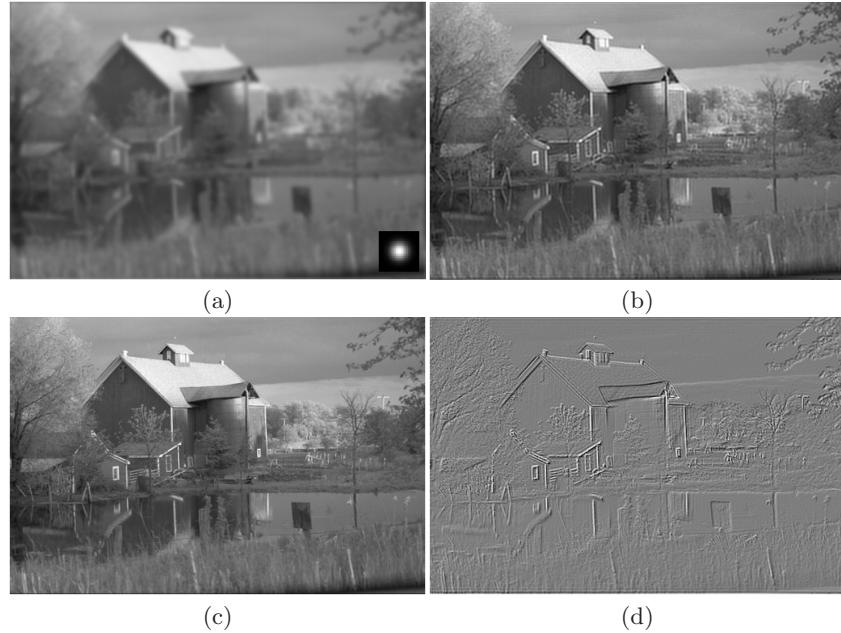


Figure 2.2 Error introduced when using the Tikhonov regularization term. (a) Blurred image with the ground truth PSF. (b) Deblurred image with Tikhonov regularization. (c) Ground truth latent image. (d) Map of $\delta\mathbf{l}$ computed using Eq. (2.13).

Regularization bias

If there is neither kernel error nor image noise and the kernel matrix F is invertible, the ground truth latent image $\hat{\mathbf{l}}$ is simply the reversion of convolution, expressed as

$$\nu(\hat{\mathbf{l}}) = F^{-1}\nu(\mathbf{b}) = \frac{F^T\nu(\mathbf{b})}{F^TF}. \quad (2.11)$$

The difference between Eqs. (2.10) and (2.11) makes it possible to analyze how the regularization term introduces bias in deconvolution in an ideal noise-free situation. It serves as guidance for future deconvolution model design.

We denote the error map of the recovered image as

$$\delta\mathbf{l} = \mathbf{l}^* - \hat{\mathbf{l}}, \quad (2.12)$$

where $\delta\mathbf{l}$ is the error introduced in deconvolution. Eqs. (2.11) and (2.12) together lead to

$$\nu(\delta\mathbf{l}) = \nu(\mathbf{l}^*) - \nu(\hat{\mathbf{l}}) = -\frac{\lambda\nu(\mathbf{b})}{(F^TF + \lambda\Lambda)F} = -\frac{\lambda}{F^TF + \lambda\Lambda}\nu(\hat{\mathbf{l}}). \quad (2.13)$$

Because $\frac{\lambda}{F^TF + \lambda\Lambda}$ can be regarded as a weight fixed by blur, this equation indicates that $\delta\mathbf{l}$ generally appears as a high frequency map dependant of image structures in $\nu(\hat{\mathbf{l}})$ as shown in Figure 2.2(d). Intuitively, a large λ makes the re-

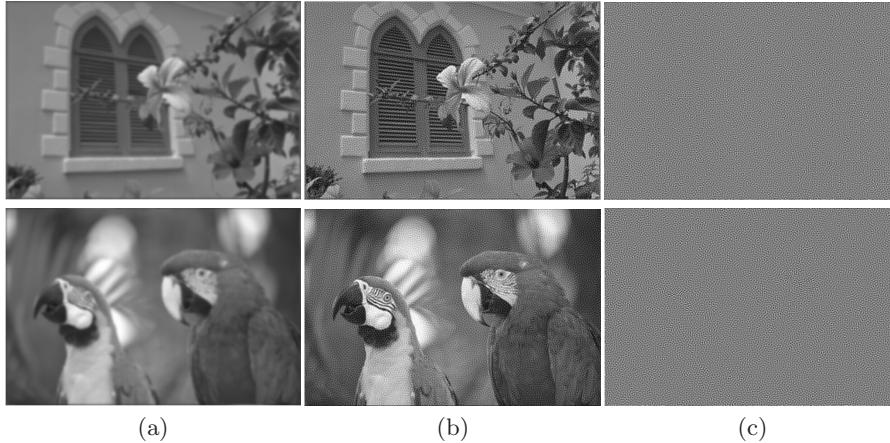


Figure 2.3 Influence of image noise in deconvolution using Tikhonov Regularization. (a) Input images with additive noise. (b) Deconvolution results. Artifacts are primarily the amplified noise. (c) Difference maps between (b) and the ground truth unblurred noise-free images.

sult lose details. If we consider inevitable image noise and PSF error, Tikhonov regularizer actually enhances the stability of deconvolution, discussed below.

Noise amplification

Now consider the case that image noise $\delta\mathbf{b}$ is presented, which is common in natural images. With derivation similar to Eq. (2.9), which takes derivatives and sets them to zeros, expression is obtained as

$$\nu(\delta\mathbf{l}) = \frac{F^T \nu(\delta\mathbf{b})}{F^T F + \lambda \Lambda} + \frac{-\lambda \nu(\hat{\mathbf{l}})}{F^T F + \lambda \Lambda}. \quad (2.14)$$

We have explained the second term given the same expression in Eq. (2.13). It produces a map that contains high frequency structure in general.

In the first term, setting $\kappa = \frac{F^T}{F^T F + \lambda \Lambda}$ to represent a coefficient matrix, the expression simplifies to $\kappa \nu(\delta\mathbf{b})$. It actually functions as adding noise with a ratio κ , which makes results still noisy. Summing up the effects of the two terms in Eq. (2.14), it is concluded that the deconvolution results contain noise while lacking an amount of structural details compared to the ground truth image. Two examples are shown in Figure 2.3.

Relation to Wiener deconvolution

Wiener filter is a method having been widely used in non-blind deconvolution (Wiener 1949). Its specialty is on the use of image and noise power spectra to suppress noise, expressed as

$$\mathcal{F}(\mathbf{l}) = \frac{1}{\mathcal{F}(\mathbf{f})} \left(\frac{|\mathcal{F}(\mathbf{f})|^2}{|\mathcal{F}(\mathbf{f})|^2 + \frac{1}{\text{SNR}(\mathbf{f})}} \right) \cdot \mathcal{F}(\mathbf{b}), \quad (2.15)$$

where $\text{SNR}(\mathbf{f})$ is the signal-to-noise ratio and $|\mathcal{F}(\mathbf{f})|^2$ denotes autocorrelation.

It can be proven that the Tikhonov regularized method is equivalent to Wiener filter with a proper λ . First, Eq. (2.10) can be rewritten as

$$(F^T F + \lambda \Lambda) \nu(\mathbf{l}) = F^T \nu(\mathbf{b}). \quad (2.16)$$

Because it holds that

$$\begin{aligned} F \nu(\mathbf{l}) &= \nu(\mathbf{f} \otimes \mathbf{l}), \\ F^T \nu(\mathbf{l}) &= \nu(\mathbf{f} \oplus \mathbf{l}) = \nu(\mathbf{f}' \otimes \mathbf{l}), \\ \mathcal{F}(\mathbf{f}) \cdot \mathcal{F}(\mathbf{f}') &= |\mathcal{F}(\mathbf{f})|^2, \end{aligned}$$

where \mathbf{f}' is the flipped version of \mathbf{f} , \oplus denotes correlation, and \cdot is an element-wise multiplication operator, Eq. (2.16) finds the solution in image domain as

$$\nu(\mathbf{f}' \otimes (\mathbf{f} \otimes \mathbf{l})) + \lambda \mathbf{l} = \nu(\mathbf{f}' \otimes \mathbf{b}). \quad (2.17)$$

Taking the Fourier transform on both sides of Eq. (2.17) yields

$$\mathcal{F}(\mathbf{f}') \cdot \mathcal{F}(\mathbf{f}) \cdot \mathcal{F}(\mathbf{l}) + \lambda \mathcal{F}(\mathbf{l}) = \mathcal{F}(\mathbf{f}') \cdot \mathcal{F}(\mathbf{b}),$$

which can be further expressed as

$$\mathbf{l} = \mathcal{F}^{-1} \left(\frac{\mathcal{F}(\mathbf{f}') \cdot \mathcal{F}(\mathbf{b})}{\mathcal{F}(\mathbf{f}') \cdot \mathcal{F}(\mathbf{f}) + \lambda \Lambda} \right). \quad (2.18)$$

Eq. (2.18) is the same as Eq. (2.15) when $\lambda \Lambda = \frac{1}{\text{SNR}(\mathbf{f})}$. This equivalence implies that Wiener deconvolution has similar noise amplification and structure information loss properties as Tikhonov regularized deconvolution.

2.1.2 Iterative approaches

Iterative computation was also used in several methods. The VanCittert (Van Cittert 1931) solver can be applied to iteratively estimate the deconvolved image as

$$\mathbf{l}^{t+1} = \mathbf{l}^t + \beta(\mathbf{b} - \mathbf{l}^t \otimes \mathbf{f}), \quad (2.19)$$

where β is a parameter automatically or manually adjustable, controlling the convergence speed. t and $t + 1$ index iterations. Eq. (2.19) converges ideally to a result close to that produced by inverse filter expressed in Eq. (2.3), which does not incorporate any prior or regularization.

The widely employed Richardson-Lucy (RL) deconvolution (Richardson 1972, Lucy 1974) can be expressed as

$$\mathbf{l}^{t+1} = \mathbf{l}^t \left(\mathbf{f}' \otimes \left(\frac{\mathbf{b}}{\mathbf{l}^t \otimes \mathbf{f}} \right) \right), \quad (2.20)$$

where \mathbf{f}' is the flipped version of \mathbf{f} , used in correlation instead of convolution. How Eq. (2.20) is constructed is explained in quite a number of papers and tutorials available online, and is thus omitted here. Different from direct inversion

(Eq. (2.3)), RL deconvolution is iterative and can be stopped halfway, which *empirically* alleviates in part noise amplification. Performing it for many iterations or making it converge, contrarily, could yield less satisfactory results. The following derivation shows that the RL method is equivalent to Poisson maximum likelihood, without imposing any image or kernel prior.

When assuming independent and identically distributed (i.i.d.) Gaussian noise $\mathbf{n} = \mathbf{b} - \mathbf{l} \otimes \mathbf{f}$, maximum likelihood estimation of \mathbf{l} is generally expressed as

$$p(\mathbf{b}|\mathbf{l}) \propto \prod_i \exp\left(-\frac{(\mathbf{b}_i - (\mathbf{l} \otimes \mathbf{f})_i)^2}{2\sigma^2}\right), \quad (2.21)$$

where $p(\mathbf{b}|\mathbf{l})$ is the conditional probability (also known as likelihood), i indexes pixels, and σ^2 is the Gaussian variance. Similarly, assuming that noise $\mathbf{n} = \mathbf{b} - \mathbf{l} \otimes \mathbf{f}$ follows a Poisson distribution yields

$$p(\mathbf{b}|\mathbf{l}) \propto \prod_i \frac{(\mathbf{l} \otimes \mathbf{f})_i^{\mathbf{b}_i} \exp(-(\mathbf{l} \otimes \mathbf{f})_i)}{\mathbf{b}_i!}, \quad (2.22)$$

where i indexes pixels. Its logarithmic energy is

$$\log(p(\mathbf{b}|\mathbf{l})) \propto \sum_i (\mathbf{b}_i \log(\mathbf{l} \otimes \mathbf{f})_i - (\mathbf{l} \otimes \mathbf{f})_i), \quad (2.23)$$

where the constant $\mathbf{b}_i!$ term is omitted. Taking partial derivative w.r.t. each pixel on the log energy and setting them to zeros yield

$$\mathbf{f}' \otimes \left(\frac{\mathbf{b}}{\mathbf{l} \otimes \mathbf{f}} - 1 \right) = 0. \quad (2.24)$$

Since \mathbf{f} is a PSF, its elements amount to 1, making $\mathbf{f}' \otimes 1 = 1$. Eq. (2.24) thus can be approximated by Richardson-Lucy deconvolution in iterations as

$$\mathbf{l}^{t+1} = \mathbf{l}^t \left(\mathbf{f}' \otimes \left(\frac{\mathbf{b}}{\mathbf{l}^t \otimes \mathbf{f}} \right) \right). \quad (2.25)$$

The above derivation shows that the RL method is equivalent to the Poisson maximum likelihood estimator in theory. Because there is no prior on the latent image \mathbf{l} , the algorithm should be stopped halfway to reduce noise and other visual artifacts. There has been research to improve RL. For example, Yuan, Sun, Quan & Shum (2008), in the multi-scale refinement scheme, applied edge-preserving bilateral filtering to the RL result. This nonlocal regularizer makes the iterative method a bit more robust against noise.

2.1.3 Recent advancement

Effective non-blind deconvolution needs to deal with noise and suppress ringing artifacts introduced by incorrect blur kernel estimates and sometimes by compression or tone management in image formation. Understanding of these issues lead to better means to *regularize* the deconvolution process in recent years, giving prior E_{prior} (denoted as $\Psi(\mathbf{l})$) a number of new forms. A general principle

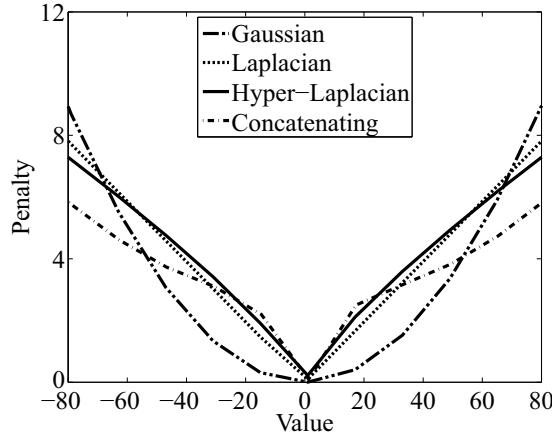


Figure 2.4 Different prior functions penalize values differently. The Gaussian prior increases energy most quickly for large absolute values.

is that the prior should not penalize excessively estimation outliers in order not to wrongly deviate final results. In what follows, without special mention, the overall objective function is still the one expressed in Eq. (2.6):

$$\min_{\mathbf{l}} \|\mathbf{l} \otimes \mathbf{f} - \mathbf{b}\|^2 + \lambda \Psi(\mathbf{l}). \quad (2.26)$$

Chan & Wong (1998) used a total variation regularizer, which is also known as Laplacian prior, by setting

$$\Psi(\mathbf{l}) = \|\nabla \mathbf{l}\|^1, \quad (2.27)$$

where ∇ denotes the first-order derivative operator, i.e., $\nabla \mathbf{l} = (\partial_x \mathbf{l}, \partial_y \mathbf{l})$, concatenation of the two gradient images. $\|\cdot\|^1$ is the L_1 -norm operator for all image gradients. This prior, illustrated in Figure 2.4 by solid lines, has a stronger effect to reduce the influence of large errors compared to the Gaussian prior used in Eq. (2.8) (dashed curve in Figure 2.4).

There are other ways to define $\Psi(\mathbf{l})$. Shan, Jia & Agarwala (2008) constructed a natural prior for the latent image as concatenating two piece-wise continuous convex functions, plotted as the “Concatenating” curve in Figure 2.4. The expression is

$$\Psi(\mathbf{l}_i) = \begin{cases} a|\nabla \mathbf{l}_i| & |\nabla \mathbf{l}_i| \leq \xi \\ b(\nabla \mathbf{l}_i)^2 + c & |\nabla \mathbf{l}_i| > \xi \end{cases} \quad (2.28)$$

where i indexes pixels and $\nabla \mathbf{l}_i$ represents partial derivative for \mathbf{l}_i in either x or y direction. ξ is the value on which the linear and quadratic functions are concatenated. a , b , and c are three parameters. $\Psi(\mathbf{l})$ actually can be used to approximate natural image statistics when a is large and b is very small.

To make the resulting structure be less smoothed, Levin, Fergus, Durand &

	Likelihood	Prior
Wiener (Wiener 1949)	Gaussian	Gaussian
L-S (Tikhonov et al. 1977)	Gaussian	Gaussian
RL (Richardson 1972, Lucy 1974)	Poisson	/
Chan & Wong 1998	Gaussian	Laplacian
Wang et al. 2008	Gaussian	Laplacian
Shan et al. 2008	Gaussian	Concatenating
Yuan et al. 2008	Poisson	Non-local (Bilateral)
Krishnan and Fergus 2009	Gaussian	Hyper-Laplacian
Yang et al. 2009	Laplacian	Laplacian
Xu and Jia 2010	Laplacian	Laplacian

Table 2.1 Comparison of a few non-blind deconvolution methods with respect to the employed prior and likelihood. “L-S” stands for least square methods.

Freeman (2007) suggested a hyper-Laplacian prior, written as

$$\Psi(\mathbf{l}) = \|\nabla\mathbf{l}\|^\alpha, \quad (2.29)$$

where $\alpha < 1$, representing a norm corresponding to a sparser distribution.

Additionally, methods of Yang, Zhang & Yin (2009) and Xu & Jia (2010) suppress noise via a TV- L_1 objective, which uses the Laplacian data term, i.e., $E_{\text{data}} = \|\mathbf{l} \otimes \mathbf{f} - \mathbf{b}\|^1$, making the objective function be expressed as

$$\min_{\mathbf{l}} \|\mathbf{l} \otimes \mathbf{f} - \mathbf{b}\|^1 + \|\nabla\mathbf{l}\|^1. \quad (2.30)$$

This function can suppress strong Gaussian and, particularly, impulse image noise with the robust constraint on the data term.

The likelihood and prior forms of different recent methods are listed in Table 2.1. Gaussian likelihood and Laplacian prior are most frequently used thanks to their simplicity in expression and reasonable ability to resist noise and error.

Albeit not quadratic, objective functions incorporating the Laplacian prior in Eq. (2.27), concatenating term in Eq. (2.28), hyper-Laplacian prior in Eq. (2.29), and the robust data term in Eq. (2.30) as a TV- L_1 energy can be solved efficiently through half-quadratic splitting, which decomposes the original problem into a quadratic minimization one and a simple single-variable optimization process. Details are provided in Section 2.1.4.

2.1.4 Variable splitting solver

An effective scheme to solve sparsely constrained non-blind deconvolution is variable splitting, implemented by half-quadratic penalty methods (Geman & Reynolds 1992, Geman & Yang 1995). This scheme has been used in many recent methods (Shan et al. 2008, Wang, Yang, Yin & Zhang 2008, Krishnan & Fergus 2009, Xu & Jia 2010). In what follows, we discuss the half-quadratic

penalty solver for minimizing

$$E = \|\mathbf{l} \otimes \mathbf{f} - \mathbf{b}\|^2 + \lambda \|\nabla \mathbf{l}\|^\alpha, \quad (2.31)$$

with a (hyper) Laplacian prior, where $0.5 \leq \alpha \leq 1$. Objective functions with the concatenating prior expressed in Eq. (2.28) and the TV- L_1 function in Eq. (2.30) can be solved similarly.

The basic idea is to separate variables involved in convolution from those in other terms, so that they can be estimated quickly and reliably using Fourier transform. It is realized by using a set of auxiliary variables $\psi = (\psi_x, \psi_y)$ for $\nabla \mathbf{l} = (\partial_x \mathbf{l}, \partial_y \mathbf{l})$, and adding extra condition $\psi \approx \nabla \mathbf{l}$. Eq. (2.31) is accordingly updated to

$$E_{\mathbf{L}} = \|\mathbf{l} \otimes \mathbf{f} - \mathbf{b}\|^2 + \lambda \|\psi_x\|^\alpha + \lambda \|\psi_y\|^\alpha + \gamma \|\psi_x - \partial_x \mathbf{l}\|^2 + \gamma \|\psi_y - \partial_y \mathbf{l}\|^2, \quad (2.32)$$

given the isotropic implementation of the α -norm. γ is a weight. When its value is infinitely large, the desired conditions $\psi_x = \partial_x \mathbf{l}$ and $\psi_y = \partial_y \mathbf{l}$ can be satisfied. In this case, minimizing $E_{\mathbf{L}}$ converges to minimizing E .

Given this variable substitution, it is possible now to iterate between optimizing ψ and \mathbf{l} . This process is efficient and is able to converge to an optimal point, since, in each iteration, the global optimum of ψ is reached in a closed form, while fast Fourier transform can be used to update \mathbf{l} .

Updating ψ

With estimated \mathbf{l} in previous pass, Eq. (2.32) is simplified to

$$E'_{\psi} = \lambda \|\psi_x\|^\alpha + \lambda \|\psi_y\|^\alpha + \gamma \|\psi_x - \partial_x \mathbf{l}\|^2 + \gamma \|\psi_y - \partial_y \mathbf{l}\|^2. \quad (2.33)$$

With a few algebraic operations to decompose ψ into the set containing all elements $\psi_{i,x}$ and $\psi_{i,y}$ corresponding to all pixels i , E'_{ψ} can be written as a sum of sub-energy terms

$$E'_{\psi} = \sum_i \left(E'_{\psi_{i,x}} + E'_{\psi_{i,y}} \right), \quad (2.34)$$

where each $E'_{\psi_{i,v}}$, $v \in \{x, y\}$ only contains a single variable $\psi_{i,v} \in \psi_v$, given by

$$E'_{\psi_{i,v}} = \lambda |\psi_{i,v}|^\alpha + \gamma (\psi_{i,v} - \partial_v \mathbf{l}_i)^2, \quad (2.35)$$

where \mathbf{l}_i is pixel i in \mathbf{l} . Each $E'_{\psi_{i,v}}$ contains only one variable $\psi_{i,v}$; so it can be optimized independently. For any α smaller than 1, minimizing Eq. (2.35) depends on two variables, i.e., joint weight γ/λ and image-dependent $\partial_v \mathbf{l}_i$. By sampling values from them, a 2D lookup table can be constructed off-line, based on which optimal results are obtained efficiently. Possible errors caused by the discrepancy of actual values and nearest samples are controllable (Krishnan & Fergus 2009). For the special cases that $\alpha = 1/2$, $\alpha = 2/3$ and $\alpha = 1$, analytic solutions are available. We discuss the case that $\alpha = 1$, where $\psi_{i,v}$ is expressed as

$$\psi_{i,v} = \text{sign}(\partial_v \mathbf{l}_i) \max \left\{ \left| \partial_v \mathbf{l}_i \right| - \frac{\lambda}{2\gamma}, 0 \right\}, \quad (2.36)$$

which is the 1D shrinkage formula (Wang et al. 2008). Its computation is efficient.

Updating \mathbf{l}

With Ψ estimated above, \mathbf{l} is updated by minimizing

$$E'_\mathbf{l} = \|\mathbf{l} \otimes \mathbf{f} - \mathbf{b}\|^2 + \gamma \|\psi_x - \partial_x \mathbf{l}\|^2 + \gamma \|\psi_y - \partial_y \mathbf{l}\|^2. \quad (2.37)$$

Since the major computation is on convolution, frequency domain operation is applied. Denoting the Fourier transform operator and its inverse as \mathcal{F} and \mathcal{F}^{-1} respectively, $E'_\mathbf{l}$ is updated to

$$\begin{aligned} E'_{\mathcal{F}(\mathbf{l})} &= \|\mathcal{F}(\mathbf{l}) \cdot \mathcal{F}(\mathbf{f}) - \mathcal{F}(\mathbf{b})\|^2 + \gamma \|\mathcal{F}(\psi_x) - \mathcal{F}(\mathbf{l}) \cdot \mathcal{F}(\partial_x)\|_2^2 + \\ &\quad \gamma \|\mathcal{F}(\psi_y) - \mathcal{F}(\mathbf{l}) \cdot \mathcal{F}(\partial_y)\|_2^2, \end{aligned} \quad (2.38)$$

where $\mathcal{F}(\partial_v)$, $v \in \{x, y\}$, is the filter ∂_v in the frequency domain. It can be obtained by the Matlab function ‘‘psf2otf’’.

According to Plancherel theorem in harmonic analysis, which states that the sum of square of a function equals to the sum of square of its Fourier transform, the energy equivalence $E'_\mathbf{l} = E'_{\mathcal{F}(\mathbf{l})}$ can be established for all possible values of \mathbf{l} . It further follows that the optimal \mathbf{l}^* that minimizes $E'_\mathbf{l}$ corresponds to the counterpart $\mathcal{F}(\mathbf{l}^*)$ in frequency domain that minimizes $E'_{\mathcal{F}(\mathbf{l})}$:

$$E'_\mathbf{l}|_{\mathbf{l}^*} = E'_{\mathcal{F}(\mathbf{l})}|_{\mathcal{F}(\mathbf{l}^*)}. \quad (2.39)$$

Accordingly, the optimal \mathbf{l}^* is given by

$$\mathbf{l}^* = \mathcal{F}^{-1}(\arg \min_{\mathcal{F}(\mathbf{l})} E'_{\mathcal{F}(\mathbf{l})}). \quad (2.40)$$

Since $E'_{\mathcal{F}(\mathbf{l})}$ is a sum of quadratic energies of unknown $\mathcal{F}(\mathbf{l})$, it is a convex function and can be solved by simply setting the partial derivatives $\partial E'_{\mathcal{F}(\mathbf{l})}/\partial \mathcal{F}(\mathbf{l})$ to zeros. The solution of \mathbf{l}^* can be expressed as

$$\mathbf{l}^* = \mathcal{F}^{-1} \left(\frac{\overline{\mathcal{F}(\mathbf{f})} \cdot \mathcal{F}(\mathbf{b}) + \gamma \overline{\mathcal{F}(\partial_x)} \cdot \mathcal{F}(\psi_x) + \gamma \overline{\mathcal{F}(\partial_y)} \cdot \mathcal{F}(\psi_y)}{\overline{\mathcal{F}(\mathbf{f})} \cdot \mathcal{F}(\mathbf{f}) + \gamma \overline{\mathcal{F}(\partial_x)} \cdot \mathcal{F}(\partial_x) + \gamma \overline{\mathcal{F}(\partial_y)} \cdot \mathcal{F}(\partial_y)} \right), \quad (2.41)$$

where $\overline{(\cdot)}$ denotes the conjugate operator. The division is an element-wise one.

The above two steps respectively update ψ and \mathbf{l} until convergence. Note that γ in Eq. (2.32) controls how strongly ψ is constrained to be similar to $\nabla \mathbf{l}$, and its value can be set with the following consideration. If γ is too large initially, the convergence is quite slow. On the other hand, if γ is overly small before convergence, the optimal solution of Eq. (2.32) must not be the same as that of (2.31). A general rule (Shan et al. 2008, Wang et al. 2008) is to adaptively adjust γ in iterations. In early stages, γ is set small to stimulate significant gain for each step. Its value increases in every or every a few iterations, making ψ gradually approach $\nabla \mathbf{l}$. γ should be sufficiently large at convergence.

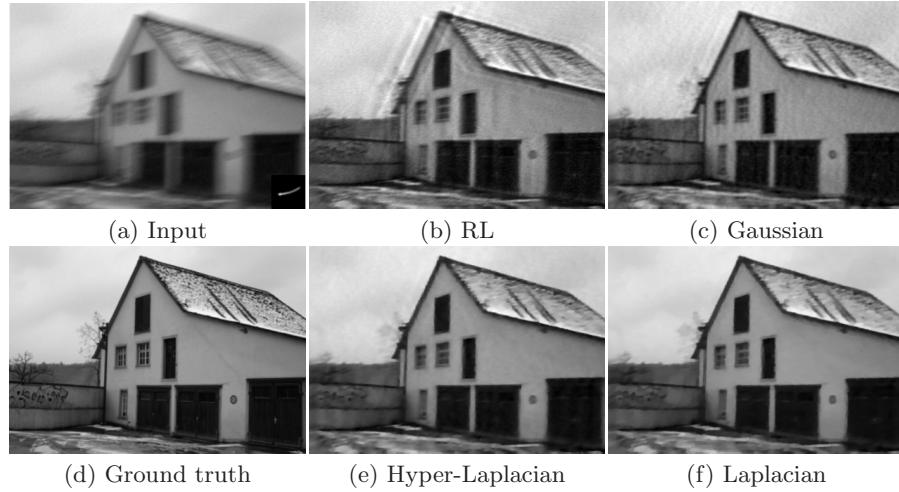


Figure 2.5 Visual comparison of non-blind deconvolution methods with the data term defined in Eq. (2.5) and priors set as none (Richardson-Lucy), Gaussian, hyper-Laplacian and Laplacian respectively.

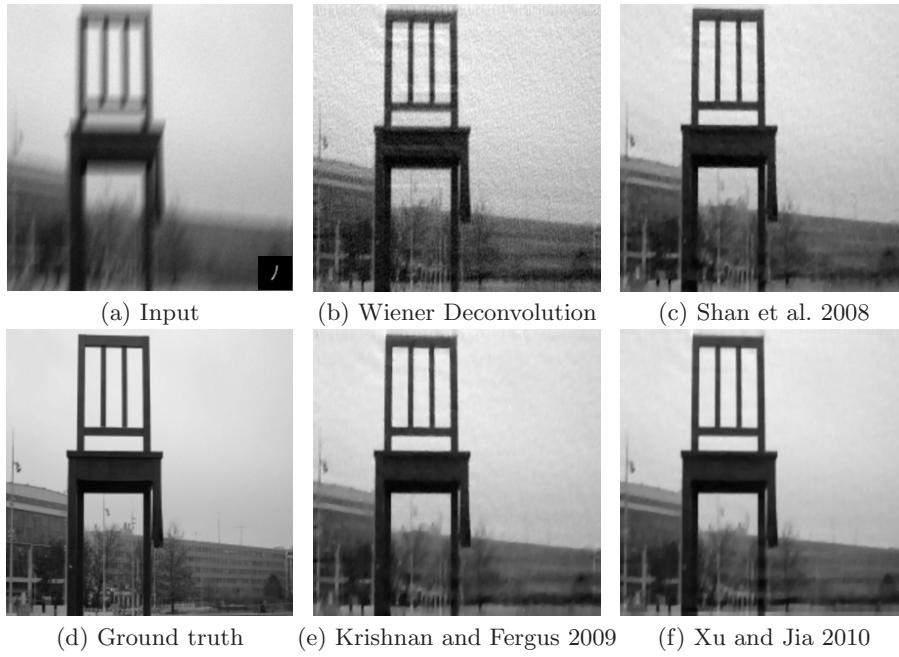


Figure 2.6 Visual comparison of results produced by a few efficient non-blind deconvolution methods.

2.1.5 A few results

We show a few examples to visually compare the aforementioned methods and models. In Figure 2.5, results from approaches incorporating different prior terms

are shown. The input blurred image and PSF contains noise, making RL and the Gaussian prior method produce a level of ringing artifacts and image noise. Laplacian and hyper-Laplacian priors, in comparison, perform better in terms of robustness against these problems. Based on publicly available executables or codes, we deconvolve another input image in Figure 2.6(a) similarly containing noise and PSF errors. The results shown in (c), (e) and (f) are visually pleasing. Laplacian and hyper-Laplacian priors used to produce the results are effective to suppress a medium level of image noise.

In terms of computation complexity, Wiener deconvolution involves simplest operation and thus runs fastest. Methods incorporating concatenating and Laplacian priors can produce higher quality results; their corresponding algorithm is also efficient when written in optimized C. The method of Krishnan & Fergus (2009) makes use of a lookup table, which is constructed offline with respect to parameters. This table much speeds up the solver.

2.2 Blind deconvolution

Blind deconvolution solves shift-invariant (uniform) motion deblurring

$$\mathbf{b} = \mathbf{l} \otimes \mathbf{f} + \mathbf{n} \quad (2.42)$$

by estimating both \mathbf{f} and \mathbf{l} . \mathbf{n} represents inevitable additive noise.

There have been many blind deconvolution methods. Approaches proposed before year 2005 mainly use the strategy to separately estimate the blur PSF and latent image, which results in alternating optimization. For example, Ayers & Dainty (1988) iterated between updating the blur PSF and latent image in a style similar to Wiener filter; Fish, Brinicombe, Pike & Walker (1995) performed blind deconvolution in a maximum likelihood fashion, using the Richardson-Lucy iteration; Chan & Wong (1998) applied the Total Variation regularizer to both the PSF and image. These methods are not elaborated in this book because they have respective limitations in handling natural image blur especially when noise and complex-structure PSFs present. The remaining of this chapter will focus on recent understanding and more advanced development of models and solvers.

The major difficulty for successful natural image motion blur blind deconvolution is on the high dimension of solution space. Any PSF \mathbf{f} can be fitted into Eq. (2.42) to find corresponding \mathbf{l} and \mathbf{n} , making it challenging to define proper criteria for optimization. Figure 2.7 shows an example. Two solutions in the right two rows indicate huge ambiguity for PSF and image estimation. A small change on estimation steps could significantly deviate the solution.

Modern objective functions can generally be expressed as

$$\min_{\mathbf{l}, \mathbf{f}} \Phi(\mathbf{l} \otimes \mathbf{f} - \mathbf{b}) + \lambda_1 \Psi(\mathbf{l}) + \lambda_2 \Upsilon(\mathbf{f}), \quad (2.43)$$

similar to the one shown in Eq. (2.6), where λ_1 and λ_2 are two weights. Φ , Ψ , and

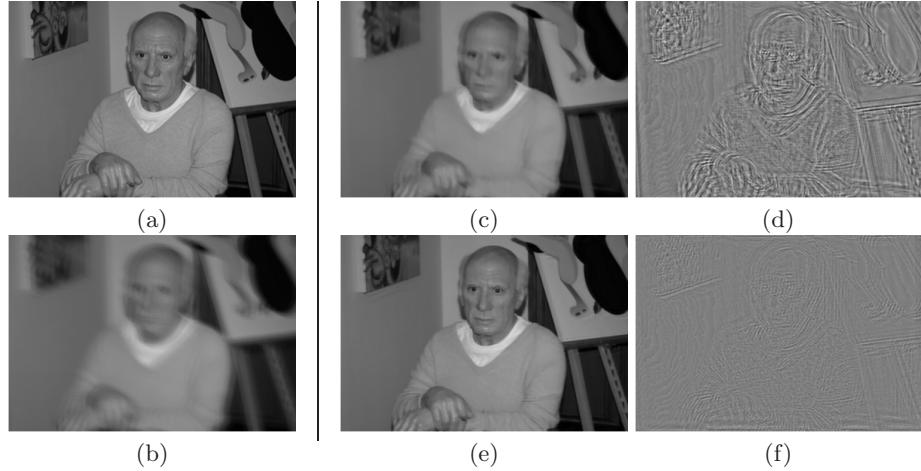


Figure 2.7 Ambiguity of solution. (a) Ground truth latent image. (b) Blurred input. (c)-(d) One latent image estimate and the corresponding noise map \mathbf{n} . (e)-(f) Another latent image result and corresponding noise map. (d) and (f) are normalized for visualization.

Υ are different functions to constrain noise, latent image and PSF respectively. Among them, Φ and Ψ can use the same expression introduced above in non-blind deconvolution. Generally, $\Phi(\mathbf{l} \otimes \mathbf{f} - \mathbf{b})$ is set to $\|\mathbf{l} \otimes \mathbf{f} - \mathbf{b}\|^2$ (or $\|\nabla \mathbf{l} \otimes \mathbf{f} - \nabla \mathbf{b}\|^2$), which is a quadratic cost on pixel values (or the derivatives of them). $\Psi(\mathbf{l})$ can be set the same way as Eqs. (2.27)-(2.29) to follow sparse gradient distributions. The new Υ is ideally a sparse function since a motion PSF tends to have most elements close to zero. Its L_1 -norm form is

$$\Upsilon(\mathbf{f}) = \|\mathbf{f}\|^1.$$

With these three functions, the objective function can be written as

$$\min_{\mathbf{l}, \mathbf{f}} \|\mathbf{l} \otimes \mathbf{f} - \mathbf{b}\|^2 + \lambda_1 \|\nabla \mathbf{l}\|^\alpha + \lambda_2 \|\mathbf{f}\|^1, \quad (2.44)$$

where α setting to 2, 1, and a value between 0 and 1 corresponds to respectively quadratic, Laplacian, and hyper-Laplacian functions. Note that different methods may alter these terms or use extra ones in the above objective, but, overall, the constraints are enough for blind deconvolution.

This objective also corresponds to a posterior probability

$$\begin{aligned} p(\mathbf{l}, \mathbf{f} | \mathbf{b}) &\propto p(\mathbf{b} | \mathbf{l}, \mathbf{f}) p(\mathbf{l}) p(\mathbf{f}), \\ &\propto \exp(-\Phi(\mathbf{l} \otimes \mathbf{f} - \mathbf{b})) \cdot \exp(-\lambda_1 \Psi(\mathbf{l})) \cdot \exp(-\lambda_2 \Upsilon(\mathbf{f})), \end{aligned} \quad (2.45)$$

in the probability framework (Shan et al. 2008, Fergus, Singh, Hertzmann, Roweis & Freeman 2006).

Solving Eq. (2.44) by simply estimating the PSF and latent image iteratively cannot produce correct results. Trivial solutions or local-minima can, contrarily,

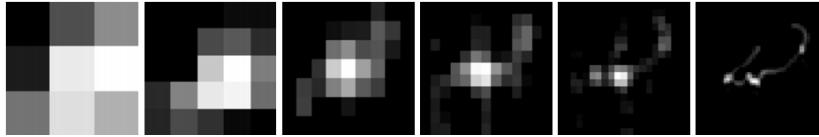


Figure 2.8 Coarse-to-fine PSF estimation in several levels.

be obtained. The *trivial* solution is the delta-function PSF, which contains a one in the center and zeros for all other elements, and exactly the blurred image as the latent image estimate (Levin, Weiss, Durand & Freeman 2009). Without any deblurring, the resulting energy in Eq. (2.44) could be even lower than that with correct deblurring for many images. In addition, simple iterative optimization is easily stuck in poor local minima.

To tackle the blind deconvolution problem, there are mainly two streams of research work. They are respectively full posterior distribution approximation $p(\mathbf{l}, \mathbf{f}|\mathbf{b})$ using maximum marginal probability estimation and energy minimization directly in Eq. (2.44). Most existing methods estimate PSFs in a multi-scale framework where the PSF is first estimated on the small-resolution image in an image pyramid. The estimate is then propagated to the next level as initialization to refine the result in a higher resolution. This process repeats for a few passes, which improves numerical stability, avoids many local minima, and even saves computation by reducing the total number of iterations. An illustration of multi-level PSFs is shown in Figure 2.8. The following discussion is based on estimation in one image level.

2.2.1 Maximum marginal probability estimation

Theoretically, the blur PSF can be perfectly obtained by maximizing the following marginalized probability, expressed as

$$p(\mathbf{f}|\mathbf{b}) = \int p(\mathbf{l}, \mathbf{f}|\mathbf{b}) d\mathbf{l}, \quad (2.46)$$

where $p(\mathbf{l}, \mathbf{f}|\mathbf{b})$ is the full posterior distribution defined in Eq. (2.45). Empirically, a huge difficulty exists regarding computational tractability of integration on the latent image \mathbf{l} . Even if one treats the latent image as discrete, marginalization still involves summing all possible image values, which is prohibitively costly.

To address this problem, Fergus et al. (2006) approximated the posterior distribution using parametric factorization, written as

$$p(\mathbf{l}, \mathbf{f}|\mathbf{b}) \approx q(\mathbf{f}, \mathbf{l}) = q(\mathbf{f})q(\mathbf{l}) \quad (2.47)$$

$$= \prod_i q(\mathbf{f}_i) \prod_j q(\mathbf{l}_j), \quad (2.48)$$

where the first equation in Eq. (2.47) assumes independence between \mathbf{f} and \mathbf{l} , making marginalization, i.e., $\int q(\mathbf{f}, \mathbf{l}) d\mathbf{l} = q(\mathbf{f})$, valid. The second equation in

Eq. (2.48) assumes pixel independence, to ease parameter estimation. Also by assuming Gaussian distributions for both the kernel and latent image, the optimal kernel that maximizes the marginalized probability becomes mean of the Gaussian distribution, i.e., $\mathbf{f}_i^* = \mathbb{E}_{q(\mathbf{f}_i)}(\mathbf{f}_i)$, where $\mathbb{E}_{q(\mathbf{f}_i)}$ is the expectation w.r.t. distribution $q(\mathbf{f}_i)$. This process corresponds to the *mean field* approach.

The final approximated distributions, in this case, are Gaussian moments, obtained by minimizing a function representing the Kullback-Leibler divergence $\text{KL}(q(\mathbf{l}, \mathbf{f}) \| p(\mathbf{l}, \mathbf{f} | \mathbf{b}))$ between the approximating distribution and the true posterior, following the variational Bayesian framework (Jordan, Ghahramani, Jaakkola & Saul 1999, Miskin & MacKay 2000). More details of this approach, e.g., use of gradients and unknown noise variance, can be found in the original papers. Note that iteratively minimizing the KL divergence cost function is very time consuming. For reference, the publicly available Matlab code takes 10 minutes to process a 255×255 image on a PC with an Intel i3 2.13GHz CPU.

Following this line, Levin, Weiss, Durand & Freeman (2011) proposed approximating the conditional distribution $p(\mathbf{l} | \mathbf{b}, \mathbf{f}) \approx q(\mathbf{l})$ instead of the joint distribution $p(\mathbf{l}, \mathbf{f} | \mathbf{b})$. It leads to an Expectation-Maximization (EM) framework that treats \mathbf{l} as a latent variable and computes expectation on it instead of integration over all possible configurations. The M-step minimizes the log-likelihood

$$\mathbb{E}_{q(\mathbf{l})}(-\ln p(\mathbf{l}, \mathbf{b} | \mathbf{f})) = \mathbb{E}_{q(\mathbf{l})}(\|\mathbf{l} \otimes \mathbf{f} - \mathbf{b}\|^2). \quad (2.49)$$

The M-step leads to quadratic programming and can be efficiently solved using frequency-domain acceleration.

The E-step, which use $q(\mathbf{l})$ to approximate the conditional distribution, is analogues to minimization of KL divergence in (Fergus et al. 2006). If a Gaussian prior on the latent image is imposed, the E-step $q(\mathbf{l}) = p(\mathbf{l} | \mathbf{b}, \mathbf{f})$ has a closed-form solution. Another difference compared to (Fergus et al. 2006) is that, instead of considering the distribution of \mathbf{f} , i.e., $q(\mathbf{f})$, Levin et al. (2011) counted in only a single \mathbf{f} estimation in the M-step, which also makes it more efficient than the maximum marginal probability implementation. It improves running time to around 1.2 minutes for a 255×255 image based on the author released Matlab code. Approximating $p(\mathbf{l} | \mathbf{b}, \mathbf{f})$ with the general sparse image priors is still costly, especially compared to methods employing explicit edge recovery or prediction, discussed below.

2.2.2 Alternating energy minimization

Energy minimization from Eq. (2.44) is another common way for uniform blind deconvolution. It has reached great success based on a few milestone techniques proposed in recent a few years. Alternating minimization can now be applied to many natural images that are blurred with very large PSFs and/or with significant noise. The process is also efficient. For example, top performing methods (Cho & Lee 2009, Xu & Jia 2010, Xu, Zheng & Jia 2013) written in optimized C++, or even Matlab, take around 5 seconds to process an 800×800 image.

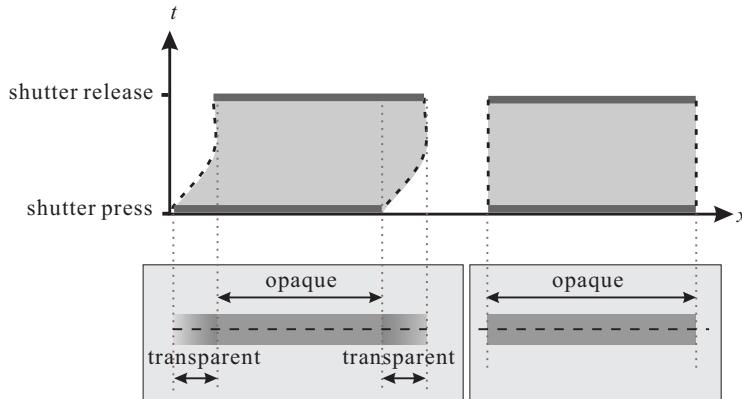


Figure 2.9 1D example showing how transparency is produced on a motion blurred object. Left: motion blurred bar, whose two ends are blended to the background and cause semi-transparency. Right: binary transparency map without object motion blur.

Additionally, this set of methods is flexibly expandable, and has been employed as key steps in many non-uniform (resp., spatially-variant) motion deblurring approaches.

The most important empirical strategy to make the solver avoid the trivial solution is to generate an intermediate sharp-edge representation. This idea was introduced by Jia (2007), who selected object boundaries for transparency estimation and performed PSF estimation only in these regions. It is based on the observation that an opaque object with sharp edges has its boundary blended to background after motion blur, as illustrated in Figure 2.9.

With this finding, the original energy function (2.44) can be updated to estimation of the PSF and transparency map, instead of the latent natural image. The transparency value for a blurred pixel is denoted as α^i . It ranges in $[0, 1]$. Its latent unblurred value is α^o . Ideally for solid objects, the α^o map is a binary one, i.e., $\alpha^o(i) = \{0, 1\}$ for any pixel i . These variables update the original convolution model (2.42) to

$$\alpha^i = \alpha^o \otimes \mathbf{f} + \mathbf{n}. \quad (2.50)$$

The corresponding objective function is updated too. Note that this model does not cause the trivial solution problem. Thanks to value binarization in the latent transparency map α^o , direct optimization can lead to satisfactory results if the input transparency values α^i are accurate enough.

Later on, Joshi, Szeliski & Kriegman (2008), instead of generating the transparency map, directly detected edges and predicted step ones. These pixels are used to guide PSF estimation, also avoiding the trivial solution.

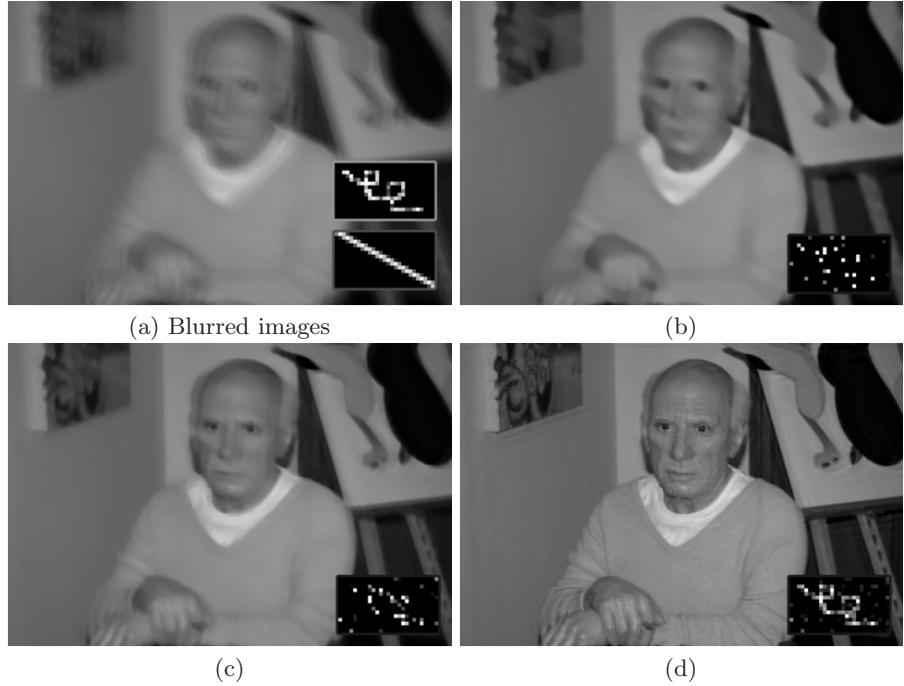


Figure 2.10 Illustration of optimization in iterations. (a) Blurred image. The ground truth blur kernel and simple initial kernel are shown in the two rectangles. (b)-(d) Restored images and kernels in three iterations.

2.2.3 Implicit edge recovery

Following this line, several other methods also implicitly or explicitly predict edges from the blurred image to guide PSF estimation. A general procedure employed in (Shan et al. 2008) iterates between PSF estimation and latent image recovery. PSF estimation is achieved by converting Eq. (2.44) to

$$\min_{\mathbf{f}} \|\mathbf{l} \otimes \mathbf{f} - \mathbf{b}\|^2 + \lambda_2 \|\mathbf{f}\|^1. \quad (2.51)$$

Latent image \mathbf{l} estimation, accordingly, is obtained by a non-blind deconvolution process, expressed as

$$\min_{\mathbf{l}} \|\mathbf{l} \otimes \mathbf{f} - \mathbf{b}\|^2 + \lambda_1 \|\nabla \mathbf{l}\|^1. \quad (2.52)$$

Its solver has been presented in Section 2.1.4. Eqs. (2.51) and (2.52) iterate until convergence.

To solve Eq. (2.51), writing it as matrix multiplication yields

$$\min_{\nu(\mathbf{f})} \|A\nu(\mathbf{f}) - \nu(\mathbf{b})\|^2 + \lambda_2 \|\nu(\mathbf{f})\|^1, \quad (2.53)$$

where A is a matrix computed from the convolution operator whose elements depend on the estimated latent image \mathbf{l} . $\nu(\mathbf{f})$ and $\nu(\mathbf{b})$ are the vectorized \mathbf{f} and

\mathbf{b} respectively. Eq. (2.53) is of a standard L_1 -regularized minimization form, and can be solved by transforming optimization to its dual problem and computing the solution via an interior point method (Kim, Koh, Lustig & Boyd 2007) or by iterative reweighted least squares (IRLS).

This algorithm can produce reasonable kernel estimates and avoids the trivial solution because it adopts a special mechanism to set parameters λ_1 and λ_2 in Eqs. (2.51) and (2.52), which control how strong image and kernel regularization is. At the beginning of blind image deconvolution, the input kernel is not accurate; the weight λ_1 is therefore set large, encouraging the system to produce an initial latent image with mainly strong edges and few ringing artifacts, as shown in Figure 2.10(b). This also helps guide PSF estimation in the following steps to eschew the trivial delta kernel. Then, after each iteration of optimization, the λ values decrease to reduce the influence of regularization on the latent image and kernel estimate, allowing for recovering more details. Figure 2.10 shows intermediate results produced in this process, where the PSF is gradually shaped and image details are enhanced in iterations.

Normalized L_1 regularization

An algorithm similar to that of Shan et al. (2008) was afterwards proposed by Krishnan, Tay & Fergus (2011). It incorporates a normalized L_1 regularization term on image gradients, written as $\|\nabla \mathbf{l}\|^1 / \|\nabla \mathbf{l}\|^2$, where $\nabla \mathbf{l}$ denotes gradients of \mathbf{l} . Normalized L_1 modifies traditional L_1 regularization $\|\nabla \mathbf{l}\|^1$ by weight $1 / \|\nabla \mathbf{l}\|^2$, which makes the resulting $\|\nabla \mathbf{l}\|^1 / \|\nabla \mathbf{l}\|^2$ value generally smaller than that of $\|\nabla \mathbf{b}\|^1 / \|\nabla \mathbf{b}\|^2$. This means the trivial blurred image solution is not favored by regularization. In this algorithm, blind deconvolution can be achieved by iteratively solving

$$\min_{\nabla \mathbf{l}} \|\nabla \mathbf{l} \otimes \mathbf{f} - \nabla \mathbf{b}\|^2 + \lambda_3 \frac{\|\nabla \mathbf{l}\|^1}{\|\nabla \mathbf{l}\|^2}, \quad (2.54)$$

and

$$\min_{\mathbf{f}} \|\nabla \mathbf{l} \otimes \mathbf{f} - \nabla \mathbf{b}\|^2 + \lambda_4 \|\mathbf{f}\|^1, \quad (2.55)$$

where λ_3 and λ_4 are two weights. Because $\frac{\lambda_3}{\|\nabla \mathbf{l}\|^2}$ in Eq. (2.54) is, in fact, a weight in each iteration. Its function is similar to λ_1 in Eq. (2.52). Both of the weights decrease during iterations to accommodating more and more details in PSF estimation, which guide blind deconvolution and avoid the trivial delta kernel solution.

2.2.4 Explicit edge prediction for very large PSF estimation

Explicit edge prediction was developed and used in (Monev & Kang 2008, Cho & Lee 2009, Xu & Jia 2010) with shock filter (Osher & Rudin 1990). It directly restores strong edges from intermediate latent image estimates in iterations.

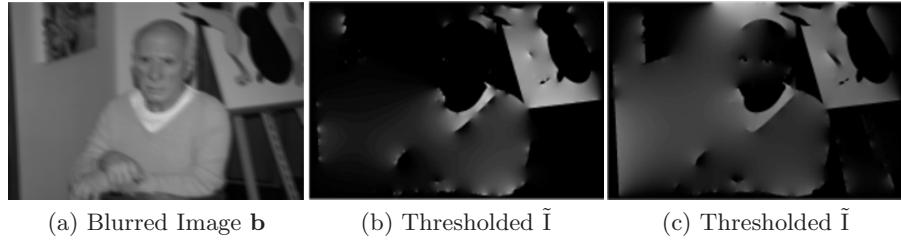


Figure 2.11 Illustration of shock filter. Given the input image (a), optimization is performed in iterations. (b) and (c) are generated in two iterations.

Shock filter performs iteratively. Given an image \mathbf{I} , in pass $t + 1$, the shock filtered result $\tilde{\mathbf{I}}^{t+1}$ is expressed as

$$\tilde{\mathbf{I}}^{t+1} = \tilde{\mathbf{I}}^t - \text{sign}(\Delta \tilde{\mathbf{I}}^t) |\nabla \tilde{\mathbf{I}}^t|, \quad (2.56)$$

where Δ and ∇ are the Laplacian and gradient operators.

Shock filter can be used in iterative blind deconvolution. It produces step-like edges from intermediate latent image estimates produced in each iteration. After removing small-magnitude edges by gradient domain threshold, only a few strongest edges are kept, as illustrated in Figure 2.11(b). Then this thresholded edge map $\tilde{\mathbf{I}}$ substitutes for \mathbf{I} in Eq. (2.51) for PSF estimation in the next iteration.

In early iterations, the thresholded edge map $\tilde{\mathbf{I}}$ is rather coarse and is obviously different from the blur input \mathbf{b} . It thus effectively avoids the trivial solution. In following iterations, more details are added to the edge map, as shown in Figure 2.11(c), to further refine the PSF estimate.

This strategy, however, could suffer from the convergence problem because each shock filtering process might raise the cost resulted from Eq. (2.51) instead of reducing it in each PSF estimation iteration. The shock filtered map also does not guarantee to contain correct edges for large PSF estimation. To address these issues, a general and unified framework was proposed in (Xu et al. 2013) where the edge map is predicted by a family of sparsity functions to approximate L_0 -norm regularization in the new objective. It leads to consistent energy minimization and accordingly fast convergence. The L_0 scheme is mathematically established with high-sparsity-pursuit regularization. It assures only salient change in the image is preserved and made use of.

To simplify mathematical expressions, in what follows, I describe a framework robust in optimization for large-kernel blind deconvolution, which still employs shock filter. It is similar to that of (Xu & Jia 2010). The method starts with construction of an image pyramid with n levels. After processing the coarsest level, its result propagates to the finer one as an initialization. This procedure repeats until all levels are processed. In each level, the method takes a few iterations to select edges and initialize PSF. The final PSF refinement is performed in the highest resolution to improve detail recovery.

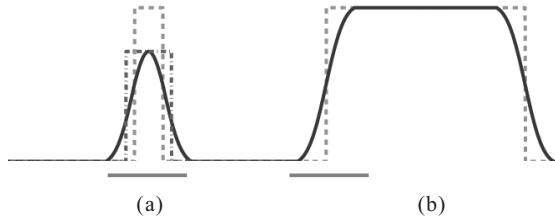


Figure 2.12 Ambiguity of motion deblurring. Two latent signals (dashed lines) in (a) and (b) are Gaussian blurred, shown as the solid curves. In (a), the blurred signal is not total-variation preserving and is shorter than the input. The dot-dash curve with the same height as the blurred signal, however, is an optimal solution during deblurring. The bottom horizontal lines indicate the kernel size.

Edge selection

In this phase, PSF is estimated with salient-gradient map construction and kernel estimation. To make this process fast, coarse image restoration is adopted to quickly obtain the \mathbf{l} estimate.

Initially, the blurred image is Gaussian smoothed and then is shock filtered using Eq. (2.56). Note that the output, i.e., salient-edge map, for many cases, cannot be directly used to guide PSF estimation due to the following fact: *if the scale of an object is smaller than that of the blur kernel, the edge information of the object might adversely affect kernel estimation.*

It is explained with the example shown in Figure 2.12. Two step signals, i.e., dashed curves in (a) and (b), are blurred with a wide Gaussian kernel, yielding signals in solid curves. Due to the small width of the latent signal, its blurred version in (a) reduces height, which mistakes PSF estimation. Specifically, the shorter dash-dot signal, compared to the taller one, has the same total variation as the blurred signal, and thus produce smaller energy in Laplacian regularization. It is more optimal than the ground truth signal when minimizing Eq. (2.44) with $\alpha = 1$. Contrarily, the larger-scale object shown in Figure 2.12(b) has no such ambiguity because it is wider than the kernel, preserving total variation along its edges. This example indicates that if structure saliency is changed by motion blur, corresponding edges produced by shock filter could misguide kernel estimation.

This problem can be tackled by selecting positively informative edges for PSF estimation and eliminating textured regions with fine structures. A metric to measure the usefulness of gradients is

$$r(i) = \frac{|\sum_{j \in N_h(i)} \nabla \mathbf{b}(j)|}{\sum_{j \in N_h(i)} |\nabla \mathbf{b}(j)| + \varepsilon}, \quad (2.57)$$

where \mathbf{b} still denotes the blurred image and $N_h(i)$ is a $h \times h$ window centered at pixel i . ε is to avoid a large r in flat regions. $\nabla \mathbf{b}(j)$ is signed. For a window containing primarily texture patterns, $\nabla \mathbf{b}$ cancel out a lot in measure $|\sum_j \nabla \mathbf{b}(j)|$. In contrast, $\sum_j |\nabla \mathbf{b}(j)|$ is the sum of absolute gradient magnitudes in $N_h(x)$,

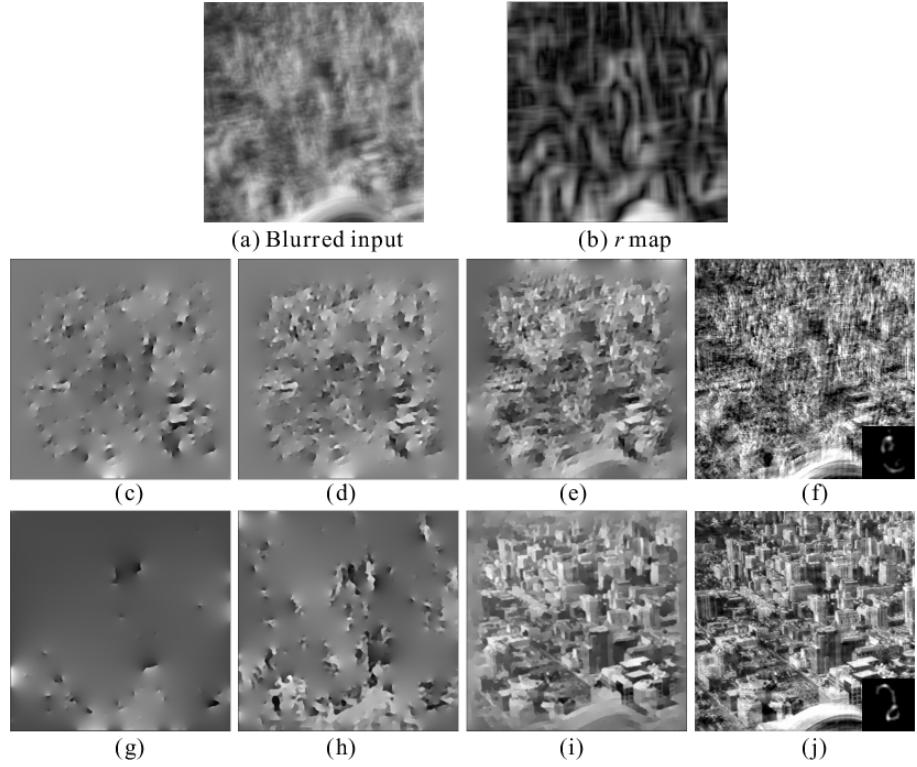


Figure 2.13 Edge selection in kernel estimation. (a) Blurred image. (b) r map (Eq. (2.57)). (c)-(e) I^s in the 1st, 2nd and 7th iterations without considering r . (f) Deblurring result not using the r map. (g)-(i) I^s maps computed according to Eq. (2.59). (j) Our final result. (c)-(e) and (g)-(i) are constructed from ∇I^s by solving Poisson equations.

which estimates how strong image structure is inside the window. Their incorporation in r actually measures whether the window is a texture one or not. A large r implies that local gradients are of similar directions and are not extensively neutralized, while a small r corresponds to either texture or a flat region. Figure 2.13(b) shows the computed r map. More explanations are provided in (Xu, Yan, Xia & Jia 2012).

Pixels belonging to small r -value windows are then removed, encoded in mask

$$M = H(r - \tau_r), \quad (2.58)$$

where $H(\cdot)$ is the Heaviside step function, outputting zeros for negative and zero values and ones otherwise. τ_r is a threshold. Finally, the selected edges are formed by non-zero values in ∇I^s , constructed as

$$\nabla I^s = \nabla \tilde{I} \circ H(M \circ |\nabla \tilde{I}| - \tau_s), \quad (2.59)$$

where \circ denotes element-wise matrix multiplication, \tilde{I} is the shock filtered image

and τ_s is a threshold of gradient magnitudes. Eq. (2.59) excludes part of the gradients, whose values depend jointly on the magnitude $|\nabla \tilde{I}|$ and the prior mask M . This selection process greatly robustifies following kernel estimation.

Figure 2.13(c)-(e) and (g)-(i) illustrate the correspondingly computed I^s maps in different iterations without and with the edge selection operation. The comparison in these two rows manifests that including more edges do not necessarily benefit kernel estimation. They, contrarily, can confuse this process especially in the first a few iterations. So an appropriate image edge selection process is vital. To allow for inferring subtle structures eventually, one can decrease the values of τ_r and τ_s in iterations, to include more and more edges. The maps in (e) and (i) contain similar amount of edges; but the quality significantly differs. The step to produce the results in (f) and (j) is detailed below.

Fast kernel estimation

With the selected edge maps, PSF initialization can be done quickly with a simple quadratic objective function written as

$$E_e(\mathbf{f}) = \|\nabla I^s \otimes \mathbf{f} - \nabla \mathbf{b}\|^2 + \lambda \|\mathbf{f}\|^2, \quad (2.60)$$

Here, \mathbf{f} is constrained in a simple quadratic term thanks to the effective gradient maps ∇I^s . Note that minimizing E_e makes the PSF estimate a bit more noisy compared to that constrained by the Laplacian term in Eq. (2.53). The result will be refined in the following steps.

Based similarly on Parseval theorem and the derivation in Eq. (2.39), computing FFTs on all variables and setting the derivatives w.r.t. \mathbf{f} to zeros yield the closed-form solution

$$\mathbf{f} = \mathcal{F}^{-1} \left(\frac{\overline{\mathcal{F}(\partial_x I^s)} \mathcal{F}(\partial_x \mathbf{b}) + \overline{\mathcal{F}(\partial_y I^s)} \mathcal{F}(\partial_y \mathbf{b})}{\mathcal{F}(\partial_x I^s)^2 + \mathcal{F}(\partial_y I^s)^2 + \lambda} \right), \quad (2.61)$$

where $\mathcal{F}(\cdot)$ and \mathcal{F}^{-1} denote FFT and inverse FFT respectively. $\overline{\mathcal{F}(\cdot)}$ is the complex conjugate operator.

The algorithm for multi-scale kernel initialization is sketched in Algorithm 1. Deconvolution starts from the coarsest level with edge selection in Eq. (2.59), kernel estimation in Eq. (2.61) and fast deconvolution using Eq. (2.8). The estimates then propagate to the next level by spatial upscaling and repeating the above steps to obtain a more accurate result. After all levels are processed, final kernel refinement is performed in the original image resolution as described in what follows.

Sparse kernel refinement

To remove remaining noise from kernel \mathbf{f}^0 output from Algorithm 1, one can apply hard or hysteresis threshold to set small values to zeros. This simple scheme however ignores the blur model, possibly making the truncated kernel less accurate. One example is shown in the top middle of Figure 2.14(b). Only keeping

Algorithm 1 Kernel Initialization

INPUT: Blur image \mathbf{b} and an all-zero kernel with size $h \times h$
Build an image pyramid with levels $\{1, 2, \dots, n\}$.
for $l = 1$ to n **do**
 Compute gradient confidence r for all pixels using Eq.(2.57).
 for $i = 1$ to 5 **do**
 (a) Select edges ∇I^s for kernel estimation (Eq. (2.59)).
 (b) Estimate kernel using Eq. (2.61).
 (c) Estimate the latent image I^l by fast non-blind deconvolution (Eq. (2.8)), and update $\tau_s \leftarrow \tau_s/1.1$, $\tau_r \leftarrow \tau_r/1.1$.
 end for
 Upscale image $I^{l+1} \leftarrow I^l \uparrow$.
end for
OUTPUT: Kernel estimate \mathbf{f}^0 and sharp edge gradient ∇I^s for further refinement

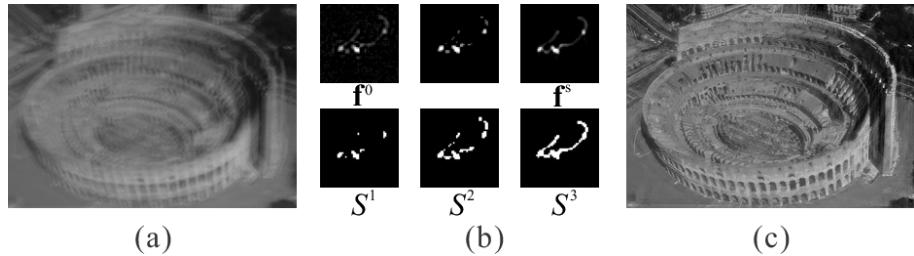


Figure 2.14 Sparse Kernel Refinement. (a) Blurred image. (b) Top row: kernel \mathbf{f}^0 , kernel by simply thresholding to remove small-value elements, and our kernel refinement result \mathbf{f}^s . Bottom row: S^1-S^3 , iteratively detected support regions by the ISD method. (c) Restored image using \mathbf{f}^s .

the large-value elements apparently cannot correctly preserve subtle structure in the motion PSF.

This problem is solved by iterative support detection (ISD) that ensures deblurring quality while removing noise (Wang & Yin 2009, Xu & Jia 2010). The idea is to iteratively secure PSF elements that already have large values by relaxing the regularization penalty, so that these pixels will not be significantly affected by regularization in next-round kernel refinement.

ISD is an iterative method. In each iteration i , after refining the kernel estimate \mathbf{f}^i , a partial support is produced to put large-value elements into set S^i and all others to set \bar{S}^i . This process is denoted as

$$S^i \leftarrow \{j : \mathbf{f}_j^i > \epsilon^s\}, \quad (2.62)$$

where j indexes elements in \mathbf{f}^i and ϵ^s is a positive number evolving in iterations, to form the partial support. ϵ^s can be configured by applying the “first significant

Algorithm 2 Kernel Refinement

INPUT: Initial kernel \mathbf{f}^0 , $\nabla \mathbf{b}$, and ∇I^s (output of Algorithm 1)Initialize the partial support $\overline{S^0}$ with \mathbf{f}^0 (Eq. (2.62)) and set $i = 1$.**repeat** Solve for \mathbf{f}^i by minimizing Eq. (2.63). Produce $\overline{S^i}$ (Eq. (2.62)). $i \leftarrow i + 1$.**until** $\frac{\|\mathbf{f}^{i+1} - \mathbf{f}^i\|}{\|\mathbf{f}^i\|} \leq \epsilon_f$ ($\epsilon_f = 1e^{-3}$ empirically)**OUTPUT:** Kernel estimate \mathbf{f}^s

jump” rule. Briefly, we sort all elements in \mathbf{f}^i in an ascending order w.r.t. their values and compute differences $d_0, d_1 \dots$ between each two nearby elements. Then we exam these differences sequentially starting from the head d_0 and search the first element, d_j for example, that satisfies $d_j > \|\mathbf{f}^i\|_\infty / (2h \cdot i)$, where h is the kernel width and $\|\mathbf{f}^i\|_\infty$ returns the largest value in \mathbf{f}^i . ϵ^s is thus assigned with the value of the j th kernel element. Readers are referred to (Wang & Yin 2009) for extra explanation. Examples of the detected support are shown in the bottom row of Figure 2.14(b). Elements within S are less penalized in optimization, resulting in an adaptive process.

Sparse kernel refinement in each iteration $i + 1$ is achieved by minimizing

$$E(\mathbf{f}) = \frac{1}{2} \|\nabla I^s \otimes \mathbf{f} - \nabla \mathbf{b}\|^2 + \gamma \sum_{j \in \overline{S^i}} |\mathbf{f}_j|. \quad (2.63)$$

Threshold applies softly to kernel refinement in regularization, which automatically maintains element sparsity faithful to the motion PSF. Algorithm 2 depicts the kernel refinement procedure.

To minimize Eq. (2.63) with the partial support, iterative reweighted least squares (IRLS) can be applied. By writing convolution into the matrix multiplication form, the latent image I , kernel \mathbf{f} , and blur input \mathbf{b} are correspondingly expressed as matrix A , vector $\nu(\mathbf{f})$, and vector $\nu(\mathbf{b})$. Eq. (2.63) is then minimized by iteratively solving linear equations w.r.t. $\nu(\mathbf{f})$. In the t -th pass, the corresponding linear equation is expressed as

$$[A^T A + \gamma \text{diag}(\nu(\bar{S})) \text{diag}(\varpi^{-1})] \nu(\mathbf{f})^t = A^T \nu(\mathbf{b}), \quad (2.64)$$

where A^T denotes the transposed A and $\nu(\bar{S})$ is the vector form of \overline{S} . ϖ denotes $\max(\|\nu(\mathbf{f})^{t-1}\|, 1e^{-5})$, which is the weight related to the kernel estimate from the previous iteration. $\text{diag}(\cdot)$ produces a diagonal matrix from the input vector. Eq. (2.64) can be solved by a conjugate gradient method in each pass.

The finally refined kernel \mathbf{f}^s is shown in Figure 2.14(b). It maintains small-value elements, which exist in almost all motion kernels. In the meantime, it is reasonably sparse. Optimization in this phase converges in less than 3 iterations referring to the loop in Algorithm 2.

Finally, given the PSF estimate \mathbf{f} output from this algorithm, high quality

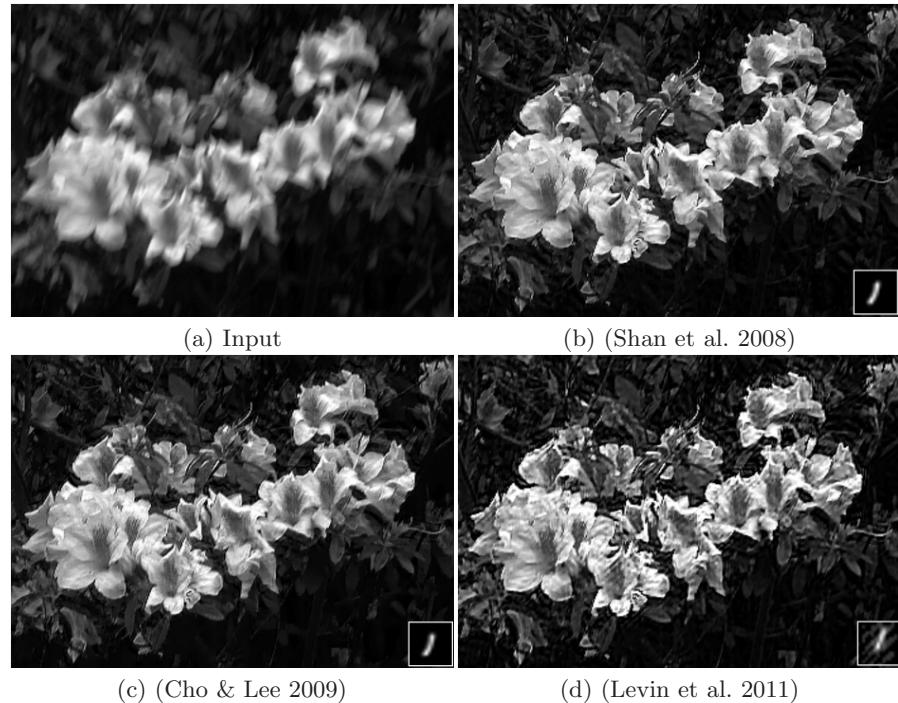


Figure 2.15 Visual comparison of different blind deconvolution results. The input is a camera captured blurred natural image.

(Methods)	(Implementation)	version	255 × 255	800 × 800	1024 × 768
(Fergus et al. 2006)	(Matlab)	v1.2	649.11	4343.24	6916.44
(Shan et al. 2008)	(C++)	v1.0	73.28	417.22	700.23
(Cho & Lee 2009)	(C++)		0.79	5.78	11.60
(Xu & Jia 2010)	(C++)	v3.0	0.80	5.75	13.26
(Levin et al. 2011)	(Matlab)		76.21	1084.19	1737.27
(Krishnan et al. 2011)	(Matlab)		25.60	215.22	273.85
(Xu et al. 2013)	(Matlab)		1.05	5.77	12.23

Table 2.2 Running time (in seconds) of different methods on CPU with respect to three image resolutions. The kernel sizes are respectively 31×31 , 41×41 , and 51×51 for the three sets of data. Note that Matlab implementation can generally be sped up when reprogrammed in C or C++.

latent image reconstruction can be applied by non-blind deconvolution using Eq. (2.31). Figure 2.14(c) shows the restored image that contains correctly reconstructed texture and structure, manifesting the effectiveness of this blind deconvolution framework.

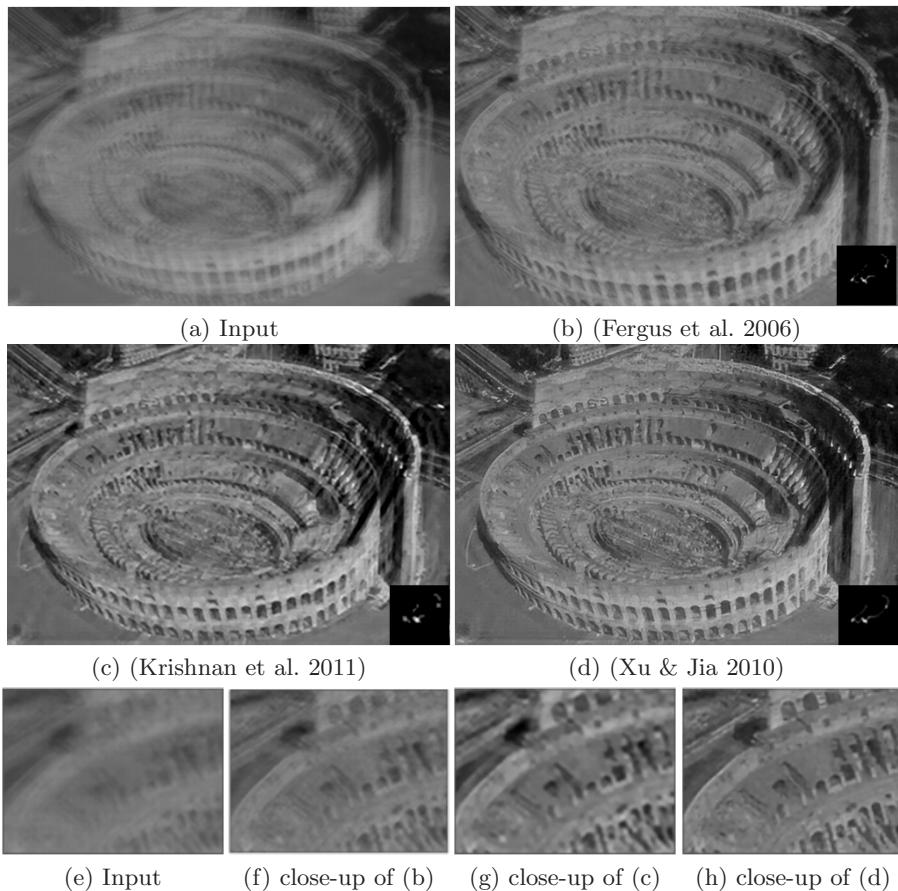


Figure 2.16 Visual comparison of different blind deconvolution results. The input is a camera captured blurred natural image. Close-ups are shown from (e)-(h).

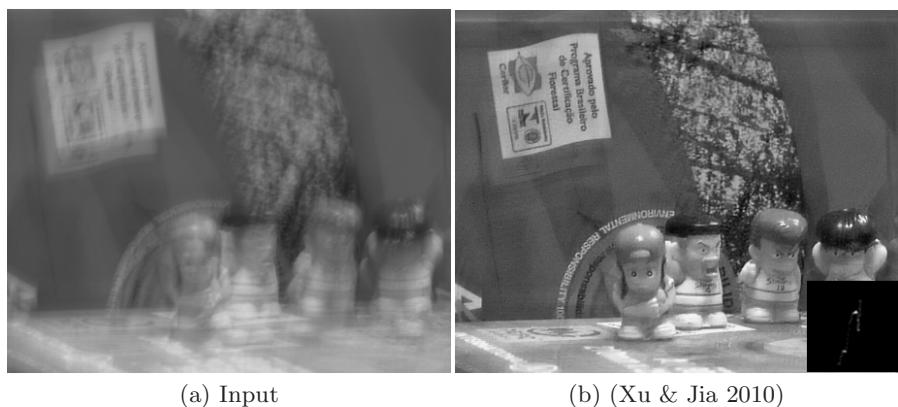


Figure 2.17 Deconvolution of a significantly blurred natural image. The blur kernel is as large as 95×95 .

2.2.5 Results and running time

Two blind deconvolution examples and their results are presented in Figures 2.15 and 2.16. The input blurred natural images are with motion kernels with resolutions 50×50 and 85×85 respectively in a spatial-invariant manner. All methods that are compared in this section can remove part or all of the blur. Difference can be observed by comparing the motion kernel estimates show in the bottom right hand corner of each result and the finally deblurred images, which depend on the quality of kernel estimates and different non-blind deconvolution strategies employed during kernel estimation or after it. Note that these uniform blind deconvolution methods provide basic and vital tools, which avail research in recent years and in future to remove spatially-variant blur from natural images caused by camera rotation and complex object motion.

Figure 2.17 shows a very challenging example where the input image is blurred with a kernel as large as 95×95 pixels. The method of Xu & Jia (2010) can deblur the input image and produce the high quality deconvolution result. The running time of different methods on CPU is given in Table 2.2 tested on a PC with an Intel i7 2.93GHz CPU and 8G memory. There is much room to speed up the methods written in Matlab by programming them in C++. In general, as the alternating energy minimization strategy does not need to estimate marginal probability or sample distributions, it is generally more efficient and flexible.

Finally, code or executable for several representative natural image uniform blind deblurring methods is publicly available. Respective links are listed below.

- (Fergus et al. 2006) (website and code)
<http://cs.nyu.edu/%7efergus/research/deblur.html>
- (Shan et al. 2008) (website and executable)
<http://www.cse.cuhk.edu.hk/leojia/projects/motion%5fdeblurring>
- (Cho & Lee 2009) (website and executable)
<http://cg.postech.ac.kr/research/fast%5fmotion%5fdeblurring/>
- (Xu & Jia 2010) (website and software)
<http://www.cse.cuhk.edu.hk/leojia/projects/robust%5fdeblur>
- (Krishnan et al. 2011) (website and code)
<http://cs.nyu.edu/%7edilip/research/blind-deconvolution/>
- (Levin et al. 2011) (code)
<http://www.wisdom.weizmann.ac.il/%7elevina/papers/LevinEtalCVPR2011Code.zip>
- (Xu et al. 2013) (website and software)
<http://www.cse.cuhk.edu.hk/leojia/projects/l0deblur>

Acknowledgements

Portions of text and figures are reused based on the work that appeared in (Xu & Jia 2010), with kind permission of Springer Science+Business Media.

References

- Andrews, H. & Hunt, B. (1977), 'Digital image restoration', *Prentice-Hall Signal Processing Series, Englewood Cliffs: Prentice-Hall, 1977* 1.
- Ayers, G. & Dainty, J. (1988), 'Iterative blind deconvolution method and its applications', *Optics letters* **13**(7), 547–549.
- Chan, T. & Wong, C. (1998), 'Total variation blind deconvolution', *IEEE Transactions on Image Processing* **7**(3), 370–375.
- Cho, S. & Lee, S. (2009), 'Fast motion deblurring', *ACM Transactions on Graphics (TOG)* **28**(5).
- Fergus, R., Singh, B., Hertzmann, A., Roweis, S. T. & Freeman, W. T. (2006), 'Removing camera shake from a single photograph', *ACM Transactions on Graphics (TOG)* **25**(3), 787–794.
- Fish, D., Brinicombe, A., Pike, E. & Walker, J. (1995), 'Blind deconvolution by means of the richardson-lucy algorithm', *Journal of the Optical Society of America* **12**(1), 58–65.
- Geman, D. & Reynolds, G. (1992), 'Constrained restoration and the recovery of discontinuities', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **14**(3), 367–383.
- Geman, D. & Yang, C. (1995), 'Nonlinear image recovery with half-quadratic regularization', *IEEE Transactions on Image Processing* **4**(7), 932–946.
- Hunt, B. (1973), 'The application of constrained least squares estimation to image restoration by digital computer', *IEEE Transactions on Computers* **100**(9), 805–812.
- Jia, J. (2007), Single image motion deblurring using transparency, in 'CVPR'.
- Jordan, M., Ghahramani, Z., Jaakkola, T. & Saul, L. (1999), 'An introduction to variational methods for graphical models', *Machine learning* **37**(2), 183–233.
- Joshi, N., Szeliski, R. & Kriegman, D. J. (2008), Psf estimation using sharp edge prediction, in 'CVPR'.
- Kim, S.-J., Koh, K., Lustig, M. & Boyd, S. P. (2007), An efficient method for compressed sensing, in 'ICIP', pp. 117–120.
- Krishnan, D. & Fergus, R. (2009), Fast image deconvolution using hyper-laplacian priors, in 'NIPS'.
- Krishnan, D., Tay, T. & Fergus, R. (2011), Blind deconvolution using a normalized sparsity measure, in 'CVPR', pp. 233–240.
- Levin, A., Fergus, R., Durand, F. & Freeman, W. T. (2007), 'Image and depth from a conventional camera with a coded aperture', *ACM Transactions on Graphics (TOG)* **26**(3), 70.
- Levin, A., Weiss, Y., Durand, F. & Freeman, W. T. (2009), Understanding and evaluating blind deconvolution algorithms, in 'CVPR', pp. 1964–1971.
- Levin, A., Weiss, Y., Durand, F. & Freeman, W. T. (2011), Efficient marginal likelihood optimization in blind deconvolution, in 'CVPR', pp. 2657–2664.
- Lucy, L. (1974), 'An iterative technique for the rectification of observed distributions', *Journal of Astronomy* **79**, 745.
- Miller, K. (1970), 'Least squares methods for ill-posed problems with a prescribed bound', *SIAM Journal on Mathematical Analysis* **1**, 52.
- Miskin, J. & MacKay, D. (2000), 'Ensemble learning for blind image separation and deconvolution', *Advances in Independent Component Analysis* pp. 123–141.

- Money, J. & Kang, S. (2008), ‘Total variation minimizing blind deconvolution with shock filter reference’, *Image and Vision Computing* **26**(2), 302–314.
- Osher, S. & Rudin, L. I. (1990), ‘Feature-oriented image enhancement using shock filters’, *SIAM Journal on Numerical Analysis* **27**(4), 919–940.
- Richardson, W. (1972), ‘Bayesian-based iterative method of image restoration’, *Journal of the Optical Society of America* **62**(1), 55–59.
- Shan, Q., Jia, J. & Agarwala, A. (2008), ‘High-quality motion deblurring from a single image’, *ACM Transactions on Graphics (TOG)* **27**(3).
- Tikhonov, A. N. & Arsenin, V. Y. (1977), *Solutions of ill posed problems*, V. H. Winston and Sons.
- Van Cittert, P. (1931), ‘Zum einfluß der spaltbreite auf die intensitätsverteilung in spektrallinien. ii’, *Zeitschrift für Physik A Hadrons and Nuclei* **69**(5), 298–308.
- Wang, Y., Yang, J., Yin, W. & Zhang, Y. (2008), ‘A new alternating minimization algorithm for total variation image reconstruction’, *SIAM Journal on Imaging Sciences* **1**(3), 248–272.
- Wang, Y. & Yin, W. (2009), ‘Compressed Sensing via Iterative Support Detection’, *CAAM Technical Report TR09-30*.
- Wiener, N. (1949), ‘Extrapolation, interpolation, and smoothing of stationary time series: with engineering applications’, *Journal of the American Statistical Association* **47**(258).
- Woods, J. & Ingle, V. (1981), ‘Kalman filtering in two dimensions: Further results’, *IEEE Transactions on Acoustics, Speech and Signal Processing* **29**(2), 188–197.
- Xu, L. & Jia, J. (2010), Two-phase kernel estimation for robust motion deblurring, in ‘ECCV’, pp. 157–170.
- Xu, L., Yan, Q., Xia, Y. & Jia, J. (2012), ‘Structure extraction from texture via relative total variation’, *ACM Transactions on Graphics (TOG)* **31**(6).
- Xu, L., Zheng, S. & Jia, J. (2013), Unnatural l0 sparse representation for natural image deblurring, in ‘CVPR’.
- Yang, J., Zhang, Y. & Yin, W. (2009), ‘An efficient TVL1 algorithm for deblurring multichannel images corrupted by impulsive noise’, *SIAM Journal on Scientific Computing* **31**(4), 2842–2865.
- Yuan, L., Sun, J., Quan, L. & Shum, H.-Y. (2008), ‘Progressive inter-scale and intra-scale non-blind image deconvolution’, *ACM Transactions on Graphics (TOG)* **27**(3).