

Deconvolution using natural image priors

Anat Levin Rob Fergus Frédo Durand William T. Freeman

Massachusetts Institute of Technology, Computer Science and Artificial Intelligence Laboratory

1 Introduction

This document provides a detailed description of the deconvolution algorithms used in the paper: A. Levin, R. Fergus, F. Durand and W. T. Freeman, *Image and Depth from a Conventional Camera with a Coded Aperture*, SIGGRAPH 2007. An implementation of the methods described below is available online at: <http://groups.csail.mit.edu/graphics/CodedAperture>

Given an observed blurry image y and a filter f the deblurring problem can be defined as finding a sharp image x such that:

$$y = f * x \quad (1)$$

A variety of different methods exist for solving for x , such as the Richardson-Lucy deconvolution algorithm. However, in this note we propose alternative strategies that make use of priors on natural images.

We start by noting that convolution is a linear operator and thus eq 1 can be written as

$$y = C_f x \quad (2)$$

where C_f is an $N \times N$ convolution matrix, and the images x, y are written as $N \times 1$ element vectors. It will also be useful to express the convolution in the frequency domain, where $F(v, \omega)$, the Fourier transform of C_f is a diagonal matrix:

$$Y(v, \omega) = F(v, \omega)X(v, \omega) \quad (3)$$

Here X, Y are the frequency representation of x, y and v, ω are coordinates in the frequency domain.

If the matrix C_f is a full rank matrix, and no noise is involved in the imaging process, the simplest approach to deconvolve y is to invert C_f and define $x = C_f^{-1}y$. Or in the frequency domain, $X(v, \omega) = Y(v, \omega)/F(v, \omega)$. This, however, is very rarely stable enough. For example, the inverse is not defined in frequencies (v, ω) for which $F(v, \omega) = 0$. Even in case $|F(v, \omega)|$ is not exactly 0 but small, the inverse is very sensitive to noise. That is, if the Y we are observing includes some noise $Y(v, \omega) = F(v, \omega)X(v, \omega) + n$, then the inverse will produce $Y(v, \omega)/F(v, \omega) = X(v, \omega) + n/F(v, \omega)$, so, when $|F(v, \omega)|$ is small the noise contribution is increased.

To overcome these difficulties it is helpful to introduce prior into the deconvolution process, and search for an x which will more or less explain the observed image y , but will also be a good image according to a prior model of natural images. That is, we would like to find the maximum a-posteriori explanation for x given y :

$$x = \operatorname{argmax}_x P(x|y) \propto P(y|x)P(x) \quad (4)$$

Assuming the noise in the measurement process is an i.i.d Gaussian noise with variance η we can express the likelihood as:

$$P(y|x) \propto e^{-\frac{1}{2\eta^2} \|x - C_f y\|^2} \quad (5)$$

For the prior on x we use a set of filters g_k and prefer that the image response for this filter set be small:

$$P(x) = e^{-\alpha \sum_{i,k} \rho(g_{i,k} * x)} \quad (6)$$

where i sums over image pixels and $g_{i,k}$ denotes the k th filter centered at pixel i , for some function ρ . The simplest choice of filters g_k is the horizontal and vertical derivative filters: $g_x = [1 \ -1]$ and $g_y = [1 \ -1]^T$. It can also be useful to include second order derivatives, or the more sophisticated filters learned by [Roth and Black 2005; Weiss and Freeman 2007]. The function ρ is selected to be a sparse, or heavy tailed function, and in the provided implementation we used $\rho(z) = |z|^{0.8}$ (other popular parameterizations of a sparse prior are student-t distributions [Roth and Black 2005] and scale mixtures of Gaussians [Portilla et al. 2003; Weiss and Freeman 2007; Fergus et al. 2006]). However, for computational reasons, we will start by considering the case of Gaussian priors, that is, setting $\rho(z) = |z|^2$. Intuitively, the above priors suggest that images are often smooth and their derivatives are close to zero.

Taking the log of equations 4,5,6, the maximum a-posteriori explanation for x reduces to minimizing:

$$\|y - C_f x\|^2 + w \sum_{i,k} \rho(g_{i,k} * x) \quad (7)$$

where $w = \alpha \eta^2$. By minimizing eq 7 we search for the x minimizing the reconstruction error $\|C_f x - y\|^2$, with the prior preferring x to be as smooth as possible. The following sections describe several approaches for minimizing eq 7.

2 Deconvolution using a Gaussian Prior

We start with the simpler case, in which the image prior is Gaussian and $\rho(z) = |z|^2$.

By differentiating eq 7 with respect to x and setting the derivative to zero, we can conclude that the optimal solution can be found by solving a sparse set of linear equations: $Ax = b$ for

$$A = C_f^T C_f + w \sum_k C_{g_k}^T C_{g_k} \quad b = C_f^T y \quad (8)$$

2.1 Deconvolution in the frequency domain

The simplest way to solve eq 8 is in the frequency domain. Since A is a sum of convolution matrices, its diagonal in the frequency domain and we get

$$(|F(v, \omega)|^2 + w \sum_k |G_k(v, \omega)|^2) X(v, \omega) = F(v, \omega)^* Y(v, \omega) \quad (9)$$

and as a result

$$X(v, \omega) = \frac{F(v, \omega)^* Y(v, \omega)}{|F(v, \omega)|^2 + w \sum_k |G_k(v, \omega)|^2} \quad (10)$$

If no prior is used ($w = 0$) eq 10 will reduce to dividing Y by F . Including priors, however, will pull the solution toward zero, especially in frequencies in which $|F(v, \omega)|$ is small. It should also be noted that the response of a derivative filter is larger in the higher frequencies and small in the lower frequencies. Thus, the prior effect will be stronger in the higher frequencies. This is a reasonable

thing to do since in those frequencies, the image content is expected to be small and the noise contribution is stronger.

In the frequency domain, the deconvolution problem can be solved extremely efficiently (a matter of seconds even for 2M pixel images) the most expensive part being the Fourier transform. The price to pay, however, is that expressing convolution in the frequency domain assumes the convolution operation is fully cyclic and this assumption is wrong along image boundaries. Therefore, applying deconvolution in the frequency domain usually result in artifacts at the image edges. For the application described in [Levin et al. 2007], having to clip a narrow strip along image boundaries out of a 2M pixel image is not a huge price to pay.

In the accompanying code, a frequency domain deconvolution is implemented in the function `deconvL2_frequency.m`.

2.2 Deconvolution in the spatial domain

To avoid the cyclic convolution assumption we would like to include in the convolution matrices C_f, C_{gk} only valid pixels (or in other words- to omit rows corresponding to boundary pixels) and thus the convolution matrices will have less rows than columns. The penalty for doing this is that the convolution operation will no longer be diagonal in the frequency domain. We can still solve eq 8 explicitly in the spatial domain. However, if the image is large, or if the support of the filters is large, explicitly inverting the matrix A in eq 8 will be impractical. In the accompanying code, we solved eq 8 numerically using the Conjugate Gradient algorithm (e.g. [Barrett et al. 1994]). The bottleneck in each iteration of this algorithm is the multiplication of each residual vector by the matrix A . Luckily the form of A (Eq 8) enables this to be performed relatively efficiently as a concatenation of convolution operations.

It should be noted that the usage of a numerical solver means that while each iteration reduces the error score, the global optimum will not necessarily be achieved, or at least, won't be achieved within a small number of iterations. The default iteration number in the accompanying implementation was adjusted to the convolution filter of our coded aperture. Other filters might result in a completely different numerical behavior.

An implementation of Conjugate Gradient deconvolution in the spatial domain is provided in the function `deconvL2.m`.

3 Deconvolution using a Sparse Prior

The advantage of a Gaussian image prior is that the resulting optimization problem is convex and the optimal solution can be derived in closed form as in eq 8. However, the distribution of derivative filters in natural images is sparse and not Gaussian. The advantage of a sparse prior for several image processing applications has been demonstrated by numerous recent papers [Portilla et al. 2003; Weiss and Freeman 2007; Fergus et al. 2006; Roth and Black 2005; Levin and Weiss To appear]. While a Gaussian prior prefers to distribute derivatives equally over the image, a sparse prior opts to concentrate derivatives at a small number of pixels, leaving the majority of image pixels constant. This produces sharper edges, reduces noise and helps to remove unwanted image artifacts such as ringing. The drawback of a sparse prior is that the optimization problem is no longer convex, and cannot be minimized in closed form. To optimize this, we use an iterative re-weighted least squares process (IRLS) (e.g. [Meer 2004; Levin and Weiss To appear]).

The IRLS approach minimizes costs of the form:

$$\sum_j \rho(A_{j \rightarrow} x - b_j) \quad (11)$$

(The deblurring problem can be written in this form, by setting the row vectors $A_{j \rightarrow}$ as the rows of the convolution matrices C_{gk}, C_f). The IRLS approach poses the problem as a sequence of standard least square problems, each least square problem re-weighted by solution at the previous step. The minimization of each least square problem is equivalent to solving a sparse set of linear equations.

The IRLS algorithm proceeds as follows:

- Initialization: set $\psi_j^0 = 1$
- Repeat till convergence:
 - Let $\bar{A} = \sum_j A_{j \rightarrow}^T \psi_j^{t-1} A_{j \rightarrow}$ and $\bar{b} = \sum_j A_{j \rightarrow}^T \psi_j^{t-1} b_j$. x^t is the solution for $\bar{A}x = \bar{b}$.
 - Set $u_j = A_{j \rightarrow} x^t - b_j$ and

$$\psi_j^t(u_j) = \frac{1}{u_j} \frac{d\rho(u_j)}{du}$$

In the provided deblurring implementation we used a prior of the form $\rho(u_j) = |u_j|^{0.8}$. Thus, the re-weighting term reduces to

$$\psi(u_j) = \max(|u_j|, \epsilon)^{0.8-2}$$

where $|u_j|$ was replaced with $\max(|u_j|, \epsilon)$ to avoid division by zero. We note that initializing the IRLS using $\psi_j^0 = 1$ is equivalent to initializing with the Gaussian prior solution. The re-weighting weights $\psi(u_j)$ will come out higher for image locations in which the derivative of the current guess is too small, and lower when the current guess is too high. Thus, the effect of the re-weighting process is to pull toward zero derivatives in image locations in which the current guess is smooth, and to pay the derivative penalty along the edges in the current guess.

Since in this process the derivative in each pixel is re-weighted in a different way, the re-weighting means that eq 8 is not a convolution and cannot be solved in the frequency domain (regardless of the boundaries modeling), so we are forced to work in the spatial domain using the Conjugate Gradient algorithm. Our observation was that the re-weighting step usually makes the system less homogeneous, and slows down the convergence of the numerical solver. For our coded aperture kernel, solving the re-weighted least square system required more Conjugate Gradient than solving the non-reweighted system of section 2.2. Experimentally, we found that the quality of the results obtained was dependent on the number of conjugate gradient iterations used.

An implementation of sparse deconvolution using the IRLS approach is provided in the function `deconvSps.m`.

4 Comparison

Figure 1 demonstrates the difference between the reconstructions obtained with a Gaussian prior and a sparse prior. While the sparse prior produces a sharper image, both approaches produce better results than the classical Richardson-Lucy deconvolution scheme.

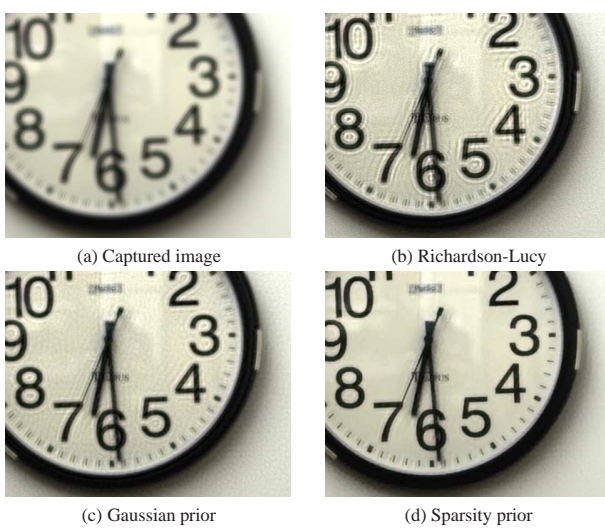


Figure 1: Comparison of deblurring algorithms applied to an image captured using our coded aperture. Note the ringing artifacts in the Richardson-Lucy output. The sparsity prior output shows less noise than the other two approaches.

5 Implementation

The accompanying code provides an implementation of the 3 deconvolution strategies described above. To get started, the user should check the file `demo.m`. Deconvolution subroutines are implemented in the following files:

1. `deconvL2_frequency.m`: Deconvolution in the frequency domain, assuming a Gaussian derivative prior.
2. `deconvL2.m`: Deconvolution assuming a Gaussian derivative prior in the spatial domain, using the Conjugate Gradient algorithm.
3. `deconvSps.m`: Deconvolution assuming a sparse derivative prior in the spatial domain, using the iterative reweighted least squares algorithm.

The user should note that all of the above functions assume the convolution filter has an odd number of pixels in both dimensions. Finally, a small tip from other users- if you aren't happy with the deconvolution output, you might need to flip your filter: `filt=flipplr(flipud(filt))`.

References

- BARRETT, R., BERRY, M., CHAN, T. F., DEMMEL, J., DONATO, J., DONGARRA, J., EIJKHOUT, V., POZO, R., ROMINE, C., AND DER VORST, H. V. 1994. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, 2nd Edition*. SIAM, Philadelphia, PA.
- FERGUS, R., SINGH, B., HERTZMANN, A., ROWEIS, S. T., AND FREEMAN, W. 2006. Removing camera shake from a single photograph. *ACM Transactions on Graphics, SIGGRAPH 2006 Conference Proceedings, Boston, MA 25*, 787–794.
- LEVIN, A., AND WEISS, Y. To appear. User assisted separation of reflections from a single image using a sparsity prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

LEVIN, A., FERGUS, R., DURAND, F., AND FREEMAN, W. T. 2007. Image and depth from a conventional camera with a coded aperture. *ACM Transactions on Graphics, SIGGRAPH*.

MEER, P. 2004. Robust techniques for computer vision. *Emerging Topics in Computer Vision*.

PORTILLA, J., STRELA, V., WAINWRIGHT, M., AND SIMONCELLI, E. P. 2003. Image denoising using scale mixtures of gaussians in the wavelet domain. *IEEE Transactions on Image Processing*.

ROTH, S., AND BLACK, M. J. 2005. Fields of experts: A framework for learning image priors. In *CVPR*.

WEISS, Y., AND FREEMAN, W. T. 2007. What makes a good model of natural images? In *CVPR*.