Image Deconvolution

Xiang Hao

Scientific Computing and Imaging Institute, University of Utah, UT,
hao@cs.utah.edu

Abstract. This is a assignment report of Mathematics of Imaging course. The topic is image deconvolution.

Keywords: Imaging System, Image Deconvolution, Inverse Problem

1 Introduction

"Imaging is the representation or reproduction of an object's outward form; especially a visual representation " [1]. Imaging system is all around our life, such as a digital camera and a scanner.

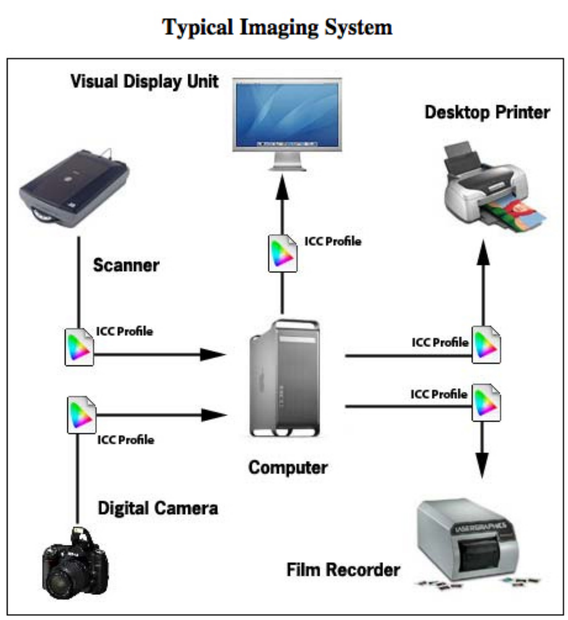

Fig. 1. Typical Imaging System. Image form [3].

Usually, linear Imaging System can be represented as

$$< \phi_i, x >= \int \phi_i(v|u)x(u)du = y_i, i = 1, 2, ...M, \tag{1}$$

where $<, >$ is the dot product between ϕ_i and x, u denotes object space, and v denotes the detector space, x is the original image (object), ϕ_i is the point spread function (PSF), which describe how the information x transfer to the detector, and the detected data is y_i. M is the number of the detected data. Our goal is usually reconstruct the x given y_i and ϕ_i for all i.

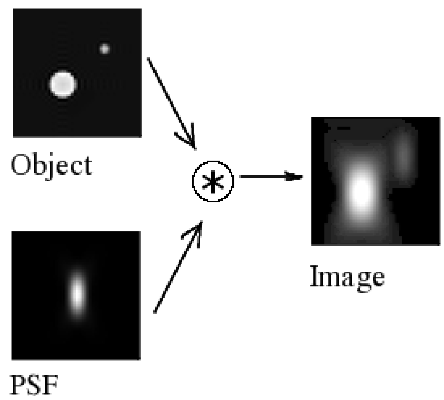

Fig. 2. Point spread function. Image is from Wikipedia [2].

If ϕ_i is shift invariant, the dot product becomes the convolution operator and the reconstruction becomes a deconvolution process. For example, if x is just a one dimensional signal, ϕ_i is a Gaussian function with mean i, then the detected data y_i is just x weighted by ϕ_i. In other words, y_i is the convolution of x and ϕ_i. In discrete setting, the above equation is equivalent to the linear system $Ax = y$, where x is a N by 1 vector, y is a M by 1 vector containing y_i, A is a M by N matrix, each row of A is the discretized ϕ_i. For example, it may look like [0, 0.1, 0.2, 0.4, 0.2,0.1,0,....], and we can solve x by solving the linear system. If M is not equal to N, depending on the value of N and M, the linear system can be either underdetermined or overdetermined. Usually, in deconvolution problem, M is equal to N, therefore there is a unique solution if A has full rank.

Usually, there are some uncertainty in the detected data y_i, which means y_i has noise. So in this case we can not directly solve the linear system $Ax = y$, instead, we want to maximum the liklihood of the detected data based on the noise distribution. The noise distribution depends on the imaging system. Here we will discuss two common noise distribution, namely, Gaussian distribution and Poisson distribution.

2 Method and Implementation

2.1 Gaussian Model

The Gaussian distribution with mean μ and variance σ^2 is

$$p(t) = \frac{1}{\sqrt{2\pi}\sigma} \exp^{-\frac{(t-\mu)^2}{2\sigma^2}} .$$

In Gaussian model, we assume y_i is Gaussian distributed with mean $< \phi_i, x >$ and variance σ^2, so

$$p(y_i) = \frac{1}{\sqrt{2\pi}\sigma} \exp^{-\frac{(y_i - <\phi_i, x>)^2}{2\sigma^2}},$$

then the joint probability of y is

$$p(y) = \prod_{i=1}^{M} p(y_i) = \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^M \exp^{-\frac{\sum_i^M (y_i - <\phi_i, x>)^2}{2\sigma^2}} = \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^M \exp^{-\frac{\|Ax-y\|_2^2}{2\sigma^2}} .$$

Since σ is constant, though we do not know it, it is easy to see that we can maximize $p(y)$ by minimizing $\|Ax - b\|$, where A is a M by N matrix. Now the problem is a least squares problem, and we know the solution to this problem is $x = (A^T A)^{-1} A^{-1} y$. If M is equal to N and A is a full rank matrix, minimizing $\|Ax-b\|$ is equivalent to solve the linear system $Ax = b$, which assumes no noise. This leads to a interesting conclusion: assuming Gaussian noise is "equivalent" to assume no noise in this case.

2.2 Implementation of Gaussian Model

$$\|Ax - b\|_2^2 = \int (y - \phi \otimes x)^2 dv$$

By Parseval's equality,

$$\int (y - \phi \otimes x)^2 dv = \int F((y - \phi \otimes x)^2) d\omega = \int (Y(\omega) - \Phi(\omega) X(\omega))^2 d\omega, \quad (2)$$

where F denotes the Fourier transform.

Use calculus of variation, we can compute the gradient of (2) as $Y\Phi - X\Phi^2$, so finally,

$$X(\omega) = \frac{Y(\omega)\Phi(\omega)}{\Phi(\omega)^2}, \text{and } x(u) = F^{-1}(X(\omega)), \quad (3)$$

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

If we use white noise prior, $\|x\|_2^2$, we just need to add μ to the denominator when we compute $X(\omega)$ [4], thus equation (3) becomes

$$X(\omega) = \frac{Y(\omega)\Phi(\omega)}{\Phi(\omega)^2 + \mu}, \text{and } x(u) = F^{-1}(X(\omega)),$$

Tips When computing $\Phi(\omega)$, we need to make sure the domain of $\phi(v|u)$ is the same as the domain of $x(u)$.

2.3 Poisson Model

The Poisson distribution is

$$p(k,\lambda) = \frac{\lambda^k \exp^{-\lambda}}{k!},$$

where k is the expected number of occurrences, and λ is the number of occurrences.

In Poisson model, we assume y_i is Poisson distributed with expected value $< \phi_i, x >$, so

$$p(y_i) = \frac{< \phi_i, x >^k \exp^{-<\phi_i,x>}}{y_i!},$$

then the joist probability of y is

$$p(y) = \prod_{i=1}^{M} p(y_i) = \prod_{i=1}^{M} \frac{< \phi_i, x >^{y_i} \exp^{-<\phi_i,x>}}{y_i!},$$

and the log of this likelihood is

$$\log(p(y)) = \sum_{i=1}^{M} y_i \log(< \phi_i, x >) - < \phi_i, x > - \log(y_i!) \tag{4}$$

When we maximize (4), we can ignore $\log(y_i!)$ since it is known. So we can just maximize

$$\sum_{i=1}^{M} y_i \log(< \phi_i, x >) - < \phi_i, x >$$

By using the Expectation Maximization algorithm [5], the above equation can be maximized by the following formulas

$$\begin{aligned} &\text{Expectation: } z^{k+1}(u) = x^k(u) \int \frac{\phi(v|u)y(v)}{\int x(u)\phi(v|u)du} dv \\ &\text{Maximization: } x^{k+1}(u) = z^{k+1}(u) \end{aligned} \tag{5}$$

2.4 Implementation of Poisson Model

The implementation of the Poisson Model is straightforward once we obtain the above formulas (5). We just to compute summaries instead of computing integrals.

Let $f(v)$ is

$$\int \frac{y(v)}{\int x(u)\phi(v|u)du} dv.$$

One tricky thing is that when I compute $\int \phi(v|u)f(v)$, since $\phi(v|u)$ is a Gaussian blurring function, which means $\phi(v|u) = \phi(u|v)$, so I actually compute

$\int \phi(u|v)f(v) = \int \phi(u-v)f(v)$, which is a convolution and easy to be implemented.

When we add a prior, we just need to change the maximization step. For example, if we add the white noise prior, $\|x\|_2^2$, the maximization step would be $x^{k+1}(u) = x^k(u) - \epsilon\left(1 - \frac{z^{k+1}(u)}{x^k(u)} - \lambda x^k(u)\right)$. If we add the H_1 prior, $\|\nabla x\|_2^2$, the maximization step would be $x^{k+1}(u) = x^k(u) - \epsilon\left(1 - \frac{z^{k+1}(u)}{x^k(u)} - \lambda \Delta x^k(u)\right)$.

3 Results

We test 6 different algorithms in this section:

Algorithm1: Gaussian model

Algorithm2: Gaussian model with Prior. We use white noise prior here.

Algorithm3: Poisson model

Algorithm4: Poisson model with Prior. We use either whiter noise or H_1 prior.

Algorithm5: deconvlucy algorithm in Matlab using two parameters: deconvlucy(blurred Image, PSF);

Algorithm6: deconvlucy algorithm in Matlab using three parameters: deconvlucy(blurred Image, PSF,20,sqrt(V));

3.1 Effect of the width of the Gaussian

In Figure 3, we blur the clear image with different width and there are no noise in the blurred image, then we apply different algorithms, namely Gaussian model, Gaussian model with prior, Poisson model, Poisson model with prior, and deconvlucy algorithms in Matlab, to the blurred images. The prior (regularization) we used here is the white noise prior.

For the top image, the Gaussian model gets the best result, and Poisson model's result is alos good and the de-noising regularization does not improve the performance of both Poisson and Gaussian model. This makes sense since the white noise prior is saying there are some noise in the image, but the truth is that there are no noise in the image, so the prior information is not right, which leads the prior models to get worse resuls.

For the bottom image, the Gaussian model fails when the width of the Gaussian blurring increases and the de-noising regularization help the Gaussian model obtain reasonable result. I guess for the Gaussian midel the effect of increasing the width is similar to adding noise, that is why the white noise prior helps the Gaussian model get reasonable result. For the Poisson model, the results are not good.

I test the deconvlucy algorithms in Matlab with two different parameters, for the bottom image, one of them gets better results comparing to the other algorithms.

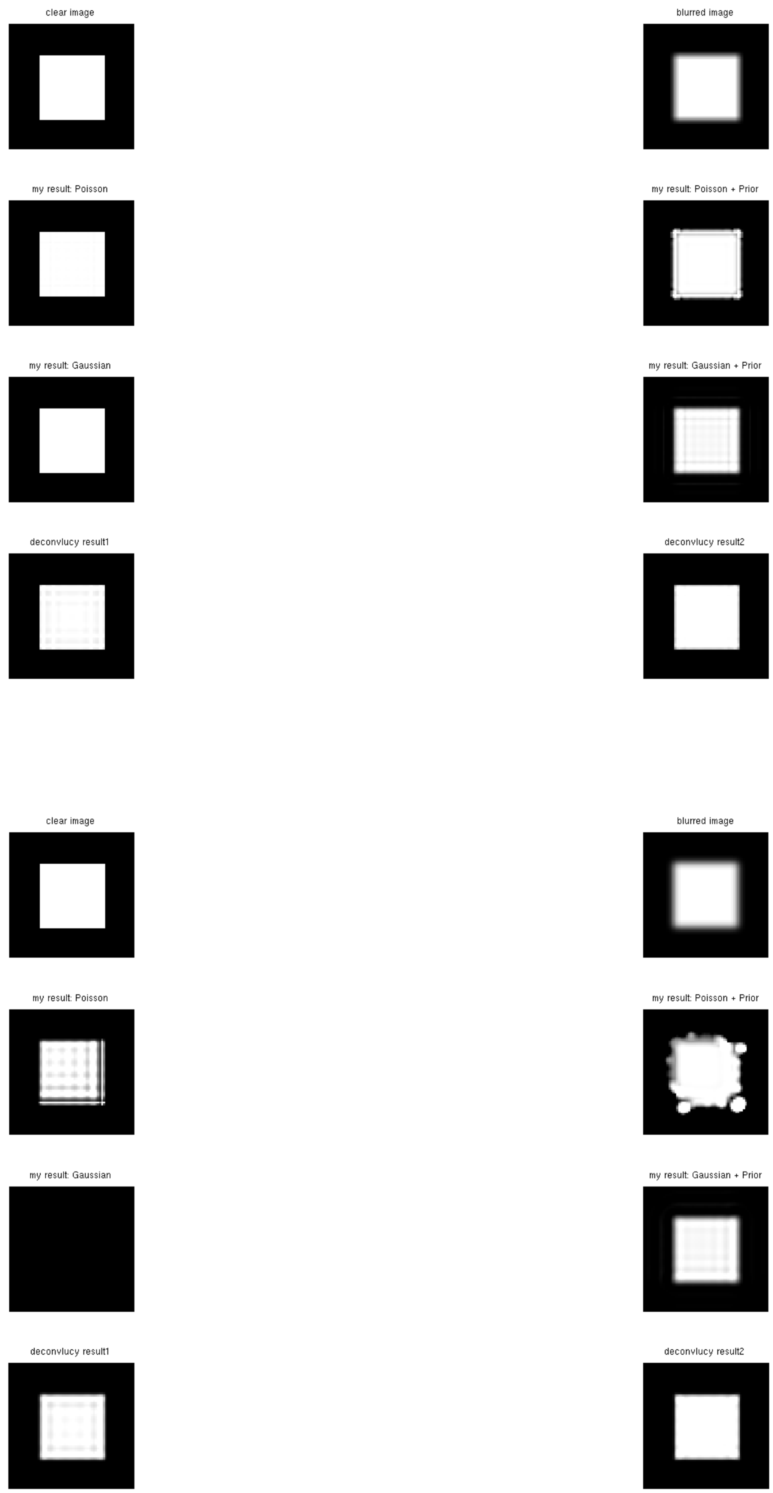

Fig. 3. Image deconvolution with different width of Gaussian. The top image has width 5 and the bottom width has width 8.

3.2 Different noises

In Figure 4, the top and bottom images have difference noise levels, The bottom one has more noise. The top one has Gaussian noise with mean 0 and variance 0.001, and the bottom one has Gaussian noise with mean 0 and variance 0.005. We use H_1 prior for the Poisson model and use white noise model for the Gaussian model.

The Gaussian model fails in both cases. This is caused by the noises: $Y(\omega)$ is not zero while $\Phi(\omega)$ is zero, thus $X(\omega)$ would be very large in this case. The algorithms generally obtain worse results in the bottom image comparing to the top one, but there are no significant differences between the results of the Gaussian model with prior algorithm. Over the six algorithms, the Gaussian model with prior algorithm gets the best performance since the blurred image has noises and the prior information, white noise prior, is correct. In addition, the deconvlucy algorithms in Matlab do not get better results in this case.

In Figure 5, we test our algorithms on a blurred image with Poisson noise. The Poisson model is better than the Gaussian model. If we look carefully, we can find more artifacts in the result of the Gaussian model comparing to the result of the Poisson model. The prior models does not help and make the results even worse. This should be cause by the prior information is not right: the noise is Poisson distributed rather than Gaussian distributed. One of the deconvlucy also gets good result.

4 Discussion

Generally, the Poisson model is more stable than the Gaussian model when noises appear or the width of the Gaussian blurring increases. The regularization (prior information) does not help when there are no noise and the width of th Gaussian blurring is small. When the width of the Gaussian blurring is large or the noise appears, the regularization might help, but it depends on what prior we use. In a word, if we use the correct prior information, the information should help, otherwise, the results would get worse.

References

1. Imaging. In *http://en.wikipedia.org/wiki/Imaging.*
2. Point spread function. In *http://en.wikipedia.org/wiki/Point_spread_function.*
3. Typical imaging system. In *http://www.gigaart.net/vis201/imaging_system.jpg.*
4. M. Bertero and P. Boccacci. Image deconvolution. In Valtere Evangelista, Laura Barsanti, Vincenzo Passarelli, and Paolo Gualtieri, editors, *From Cells to Proteins: Imaging Nature across Dimensions*, volume 3 of *NATO Security through Science Series*, pages 349–370. Springer Netherlands, 2005.
5. Sean Borman. The expectation maximization algorithm: A short tutorial. unpublished paper available at http://www.seanborman.com/publications. Technical report, 2004.

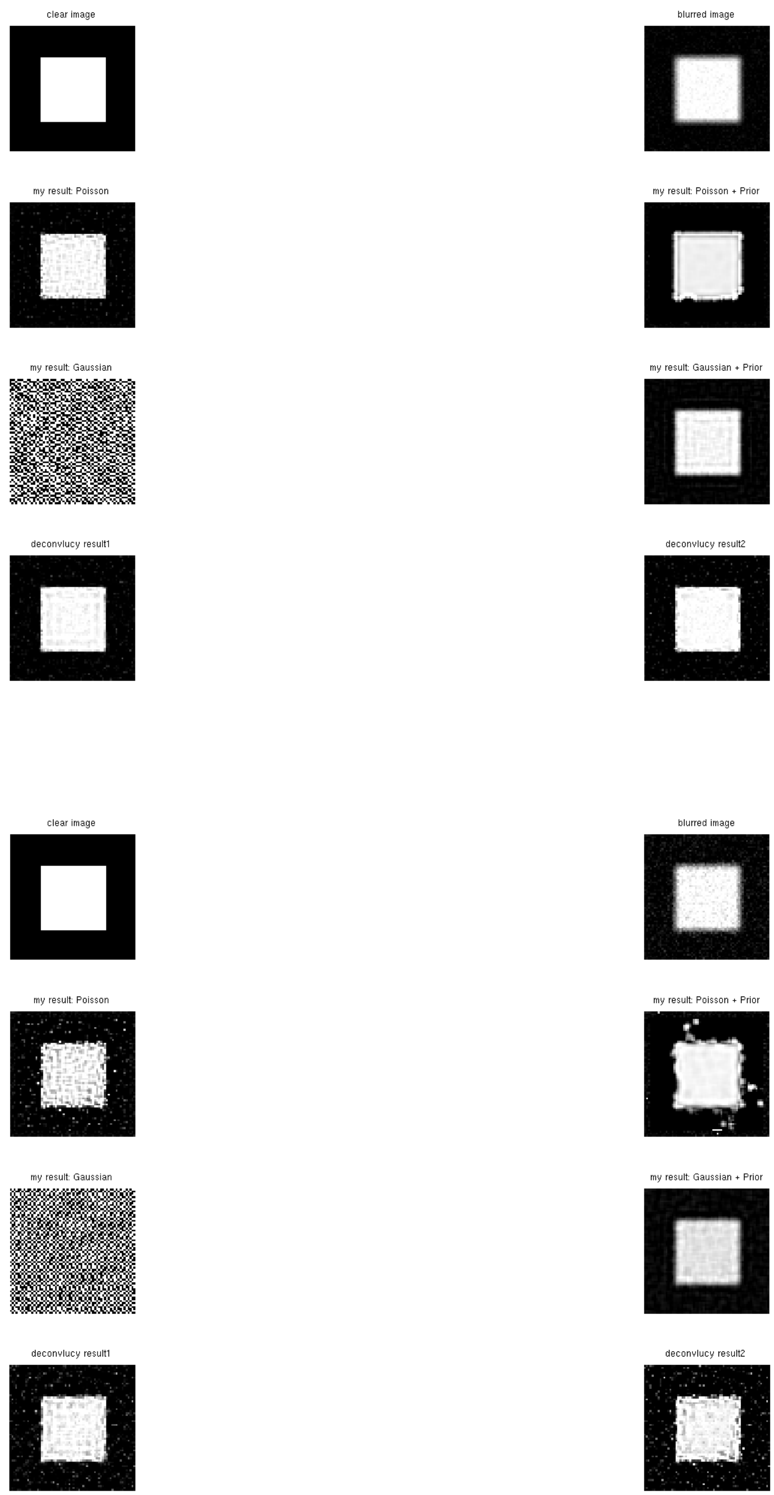

Fig. 4. Image deconvolution with different Gaussian noises. The top image has Gaussian noise (0, 0.001) and the bottom one width (0, 0.005).

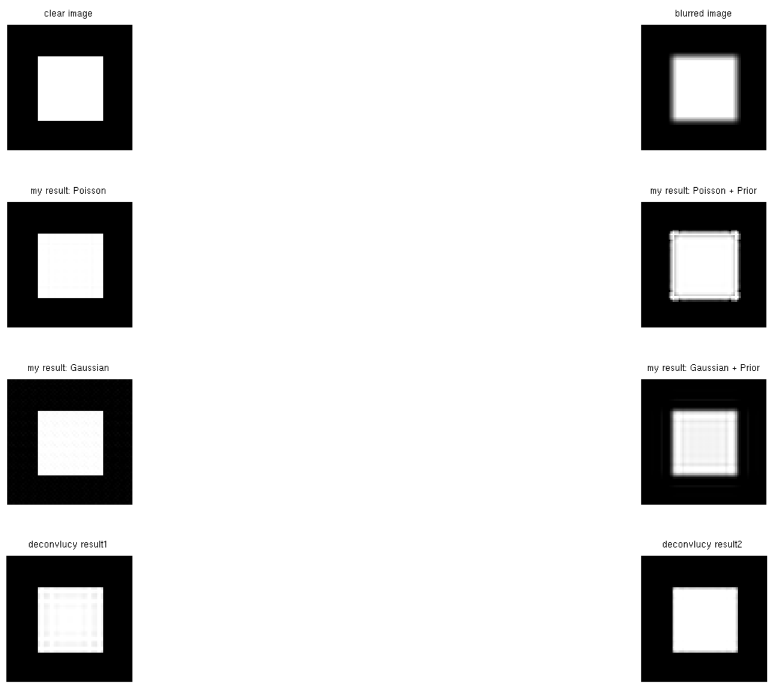

Fig. 5. Image deconvolution with Poisson noise