An Overview of the Maximum Entropy Method of Image Deconvolution

Michael Gary Grotenhuis

A University of Minnesota – Twin Cities “Plan B” Master’s paper

Introduction

Long before entering the consumer home, digital imagery transformed observational astronomy. It provided quantified data to observations, such as the apparent brightness of a star, along with all the other advantages that can be associated with computerized information. And while astronomy was the first science to take advantage of the digital picture, scores of other studies have followed suit. Biologists automatically track the movements of microscopic organisms; chemists discover the composition of distant objects; meteorologists track the weather from satellites.

While the Charge Coupled Device (CCD), the electronic backbone of the digital picture, continues to develop, some have devoted themselves to improving the images with software alone. The impetus is two-fold: first, that there is always the desire to get the most out of data, and second, that there are many situations where implementing the latest technology is prohibitively expensive. For example, consider the Hubble Space Telescope, which initially had a mirror that caused image distortion. Eventually astronauts replaced the mirror, but for years there was the need to utilize the faulty, yet expensive telescope. To improve the Hubble's images, a software algorithm was created to somewhat remove the distortion.

In fact, the Hubble images were improved by performing a computational deconvolution method, the same type of method that is the subject of this paper. In a perfectly focused, noiseless image there is still a warping caused by a point-spread function (PSF). The PSF is a result of atmospheric effects, the instrument optics, and anything else that lies between the scene being captured and the CCD array. It is well-named, as it is mathematically the same as the image taken of a perfect point of light. Consider a cloudless night and a canopy of stars. While every star is at a sufficient distance to be regarded as a point, it does not appear so, mostly because the atmosphere causes a blurring of the light. This is the essence of the point spread function.

Mathematically, the point spread function affects the image in a manner called convolution, a particular type of integral. Deconvolution, then, is the attempt to separate the scene (or "truth") from the point spread function and digital image. The first step in this process is to identify the point spread function, which in many cases can only be done with careful modeling. However, nature has provided astronomy with the perfect means for an empirical detection – the exposure of a star. This method is not perfect, as we must still consider instrument noise and systematic errors, but superior to mathematical models that cannot completely capture the necessary details. The next step is the actual deconvolution – the attempt to find "truth" given the measured image and point-spread-function. There are many deconvolution methods, but the subject of this paper is the one called the Maximum Entropy Method (MEM).

We will begin by learning how the CCD array works, followed by what convolution and the point-spread function are as well as their role in distorting images. Then we will focus on partially removing the distortion using the MEM. Should the reader need more clarification or more information regarding the MEM, I would suggest visiting the websites listed in the reference section, which I found to be very useful.

The CCD Array

In a grayscale digital image, each pixel represents brightness. While “brightness” may initially sound like a somewhat abstract and relative term, the way in which it is measured is not. A digital camera uses a Charge-Coupled Device (CCD) to capture an image. The CCD is an array – a rectangular chip broken into tiny cells, and within each cell, a semiconductor. When the camera is instructed to take a picture, the shutter is opened and the light from the scene is focused onto the array. When an increment of light, or photon, moves from the scene through the optics and into a particular cell, it interacts with an electron in the semiconductor. The electron is imparted with enough energy to move from the semiconductor to a conduction region within the cell. After the camera closes the shutter, the charge in each cell is moved systematically through the array to a register which reads in the amount of charge that originated from each cell. This signal is directly proportional to the number of photons that reached the cell during the exposure. So, in a digital image, the brightness measurement for each pixel is literally a photon count. And incredibly, the device is sensitive enough that more than half the photons (within the desired wavelength range) that are focused onto a particular cell in the CCD array are eventually counted. The 2009 Nobel Prize in Physics was partially awarded to the inventors of the CCD array, Willard S. Boyle and George E. Smith.

There are a number of sources of error involved with CCDs, but usually they can be well-characterized. Color images can be taken by filtering out the unwanted regions of the spectrum, though the details vary significantly. Ultimately we are left with a quantified measurement of light, opening the door for techniques that improve our utilization of imagery.

Convolution

Convolution is a type of mathematical integral that appears often in image processing and signal processing. It involves two functions, and is represented by

$h(t) = f(t) * g(t)$ or $h(t) = f(t) \otimes g(t)$.

Convolution is defined as

$$h(t) = f(t) * g(t) = \int_{-\infty}^{\infty} f(\tau) g(t-\tau)\, d\tau$$

and is commutative, so

$$h(t) = f(t) * g(t) = g(t) * f(t) .$$

Convolution is best understood visually and with an example. So, as an example, we will convolve the step function, defined as

$$f(t) = \begin{cases} 1 & 0 \leq t \leq 1 \\ 0 & otherwise \end{cases}$$

with itself, so $g(t) = f(t)$. The first step is to reflect $g(t)$ over the y-axis and then shift it all the way to $-\infty$. Note that if $g(t)$ was not symmetric about its center as it is with this example, its shape would be reflected as well. The function $f(t)$, now called $f(\tau)$, is not changed in any way.

We will begin the convolution by moving the altered $g(t)$ function from $-\infty$ to ∞. In the integral, $g(t)$ is now changed to $g(t-\tau)$, where the τ variable represents its reflected structure and t represents how much it has been shifted from the y-axis. The situation can be seen visually in Figure 1.

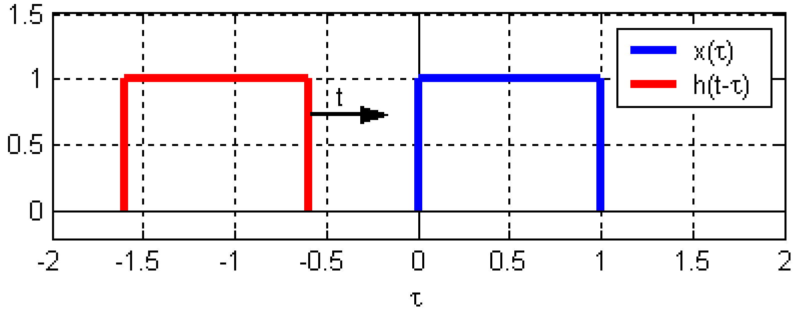

Figure 1: Example convolution image for $t < 0$. All convolution images from Gjendemsjø (2007)

Note that the convolution integral is itself a function, so we expect to get a function of t when we are finished. Recall that the t variable now indicates the position of $g(t-\tau)$. To carry out the integral, we multiply $g(t-\tau)$ for a given value of t with the function $f(\tau)$, and take the integral over τ. In our example, in the region from $t = -\infty$ to 0, either $g(t-\tau)$ or $f(\tau)$ is zero for all τ (except for when τ is exactly zero) , so the result of the convolution is zero in this region. However, once t moves from negative to positive, there is overlap of $g(t-\tau)$ and $f(\tau)$. This is represented in Figure 2.

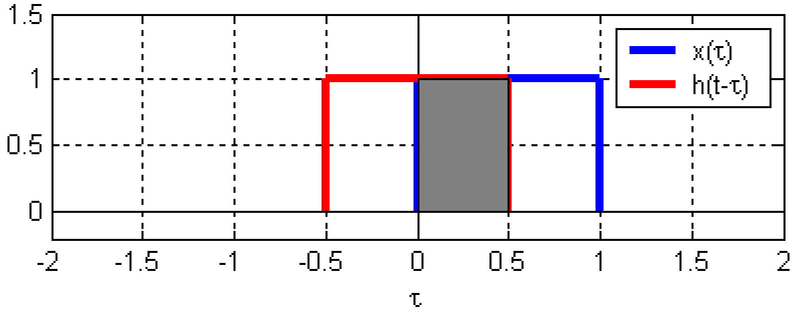

Figure 2: Example convolution image for $0 \leq t \leq 1$

In the region from $t = 0$ to $t = 2$, we have overlap, and the result of the convolution for a given t is given by multiplying the functions together and taking the integral. The integral will be at a maximum at $t = 1$, where the non-zero regions of the two functions completely overlap each other. After that, the convolution integral will decrease toward zero, as the multiplied area decreases as shown in Figure 3.

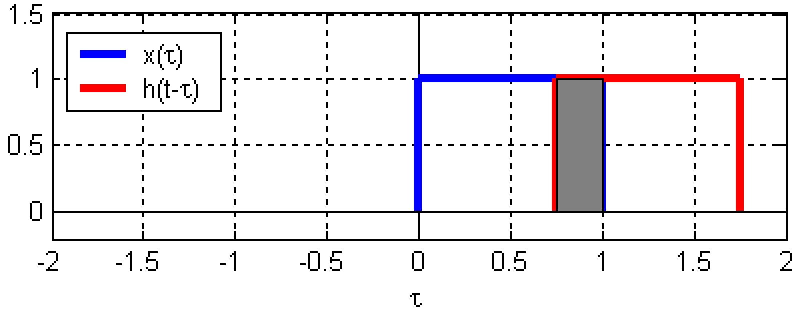

Figure 3: Example convolution image for $1 \leq t \leq 2$

The integral reaches zero at $t = 2$, and is zero in the entire region from $t = 2$ to $t = \infty$, since there is no longer any overlap, as shown in Figure 4.

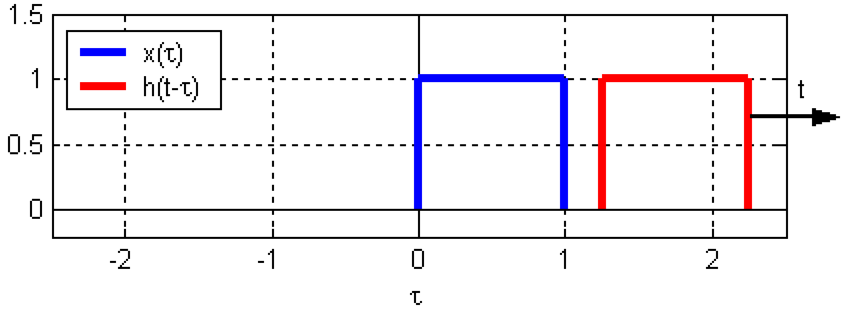

Figure 4: Example convolution image for $t > 2$

The entire result of the convolution is shown in Figure 5. Note the peak at $t = 1$ where the two shapes completely overlapped. If the student wishes to compute a convolution integral, he or she might try duplicating the result in this simple example. Note that the integral will need to be defined separately in each differing region, and that the limits of integration will depend on t. In this example, the different regions are from $t = -\infty$ to 0, from $t = 0$ to $t = 1$, from $t = 1$ to $t = 2$, and from $t = 2$ to $t = \infty$.

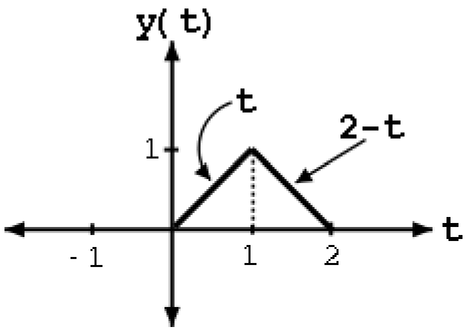

Figure 5: Example convolution result

The Point-Spread Function

You might think that the CCD array is a limiting factor in resolution, in that it integrates all the photons within each cell so there is no way to pinpoint exactly where each photon came from. That might be true were it not for the point-spread function (PSF). The point spread function is well-named, as it represents the spreading of light through the optics and onto the CCD array from a perfect point (infinitely small) of light. A Gaussian is sometimes used as an approximation for a PSF, as is shown in Figure 6.

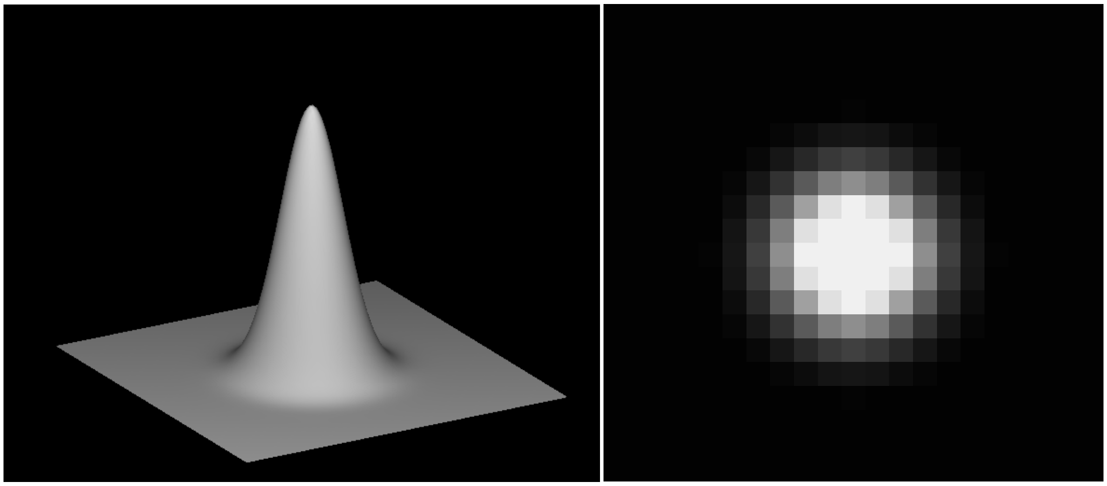

Figure 6: Visual representation of a Gaussian point-spread function, left: as a 3-d surface, right: as an image

In practice, every optical system has its own unique point-spread function. Certainly, our knowledge of quantum mechanics forces us to replace the idea that light is spread evenly through the PSF with a more statistical understanding. In that sense, the PSF is somewhat like a wave-function.

Usually an imaging system is created in such a way that the point-spread function is over-sampled, meaning its influence is spread over many pixels. It is the PSF, then, that is the limiting factor in resolution. In astronomy, this fact allows the measuring of an optical system's PSF via a long exposure of a star – in practice, a perfect point of light.

Certainly, real images are not merely a single point of light. The effect of the point-spread function with a real image is to take every originating point of light from the scene and spread it into neighboring regions. In an image, the spreading of each original point of light combines with the spreading from other points. A simple example of this is shown in Figure 7.

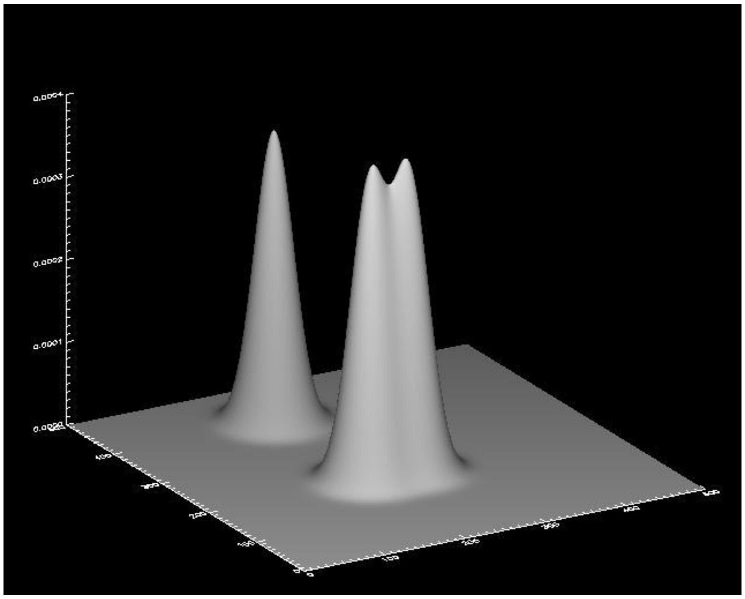

Figure 7: Visual representation of an image where the influence from two points of light overlap due to the point-spread function

Figure 7 represents an incredibly simple measured image of three perfect points of light. In a real image, there is a multitude of such points. I will call this originating light, the scene that exists before the PSF has taken effect, "truth". Mathematically, our measured image, without considering noise, is represented as

$$Measured\ Image = \mathrm{truth}\ *\ \mathrm{PSF}\ .$$

That is to say that our measured image is the convolution of truth with the point-spread function. In practice, we deal with a discrete image (the measurement from our CCD), and we represent the above equation as

$$I_k = \sum_{i=0}^{N-1} O_i PSF_{k,i} \qquad \text{Equation 1}$$

where I_k is the measured image, O_i is "truth", $PSF_{k,i}$ is the PSF, and N is the dimension of the image. This is a one-dimensional equation but the extension to two dimensions is obvious, and in many cases the equation can be separated into the two directional components. I will use the one dimensional version for the sake of simplicity.

Clearly it would be ideal to get "truth" using our measured image and PSF. Doing so would increase resolution – our ability to see detail in the data. This practice is called deconvolution, and one particular method, the Maximum Entropy Method (MEM), is the subject of this paper.

An Aside – The Fourier Relationship Of Convolution

The following is an aside for those who are familiar with the Fourier Transform:

Convolution and the Fourier Transform share a special – and very simple – relationship. If we denote the convolution of "truth" and PSF as

$$i(x) = o(x) \, * \, psf(x)$$

then, in Fourier space,

$$\bar{I}(\bar{\xi}) \propto \bar{O}(\bar{\xi}) \cdot \overline{PSF}(\bar{\xi})$$

where I have used a proportionality because there is a constant that depends upon convention. Given this knowledge, deconvolution should be simple: find the Fourier transform of the image and PSF, divide to get

$$\frac{\bar{I}(\bar{\xi})}{\overline{PSF}(\bar{\xi})} \propto \bar{O}(\bar{\xi})$$

and then take the inverse transform (of course, the image and PSF must have a valid Fourier transform, which is another discussion entirely). In an ideal world, where one can take a perfect measurement without any added noise, then yes, this solution would be ideal. The problem is that the PSF contains small high-frequency components, and the division operation with the PSF-transform in the denominator amplifies the high-frequency noise. The high frequencies correspond to high detail in the actual image, so the Fourier method of dividing obscures the image detail that we are hoping to recover (O'Haver 2008).

Some researchers do choose to perform deconvolution using the Fourier method combined with filters. This does not yield a perfect solution, but neither does any other deconvolution method. And to be fair, it should be mentioned that the Maximum Entropy Method produces the "smoothest" (smallest detail) image possible.

Bayes' Theorem and Image Deconvolution

Bayes' Theorem is an important facet of probability theory that concerns conditional probabilities. Consider the following equation:

$$P(B|A)\,P(A) = P(A|B)\,P(B)$$

Equation 2

where $P(A)$ is the "prior probability" of A; $P(B)$ the "prior probability" of B; $P(B|A)$ the "conditional probability" of B, given A; and $P(A|B)$ the "conditional probability" of A, given B. The "prior probabilities" are essentially the same as a standard probability, so $P(A)$ would be the probability of event A occurring without any knowledge about B. The "conditional probabilities" refer to the probability of an event occurring knowing that a related event has occurred. So $P(B|A)$ is the probability of event B occurring knowing that event A has occurred.

The concept of conditional probabilities, like convolution, is best understood with an example. So, for instance, if $P(A)$ is the probability of encountering a celebrity in a given day and $P(B)$ is the probability that you are in Hollywood, California in a given day; then $P(A|B)$ would be the likelihood of encountering a celebrity given that you are in Hollywood and $P(B|A)$ would be the probability that you are in Hollywood given that you encountered a celebrity. Convince yourself that Equation 2 is true.

Bayes' Theorem is a simple rearrangement of Equation 2:

$$P(A|B) = \frac{P(A)\,P(B|A)}{P(B)}$$

In our situation, we are trying to maximize the probability that the answer we get from our deconvolution is the same as "Truth", given the data – the blurred image. Our method, then, is a

Bayesian one, as it is based upon maximizing a probability in Bayes' Theorem. The specific equation is (McLean 2008)

$$P(O|I,M) = \frac{P(I|O,M)\,P(O|M)}{P(I|M)}$$

where 'O' is "truth", 'I' is the blurred image – the given data, and 'M' is the model of our deconvolution – the answer we get. Clearly, we want to maximize $P(O|I,M)$, the probability that we have obtained "truth" from our model and the data. Interestingly, there is a term, $P(O|M)$, that does not depend on the data. Using the MEM, this term is a measure of the entropy, a quantity that will be defined later.

The Maximum Entropy Method

Armed with a basic understanding of the digital image, the PSF, and convolution, we can now discuss deconvolution. Typically, we are trying to deconvolve "truth" from our knowledge of the image and PSF. The complexity of the problem lies in the fact that any one pixel of our image contains only a partial signal from the corresponding location in the scene and also contains partial signals from all the other sources of light in the scene, though in practice we need only concern ourselves with the neighboring sources. Of course, there is also noise.

The Maximum Entropy Method, as introduced by Alhassid et al. (1977), is a means for deconvolving "truth" from an image and PSF. We can actually motivate the MEM using a thermodynamic or an information theory definition of entropy, and it is satisfying to know that the two different schools of thought lead us to the same method.

Motivating the MEM Using Information Theory: Kangaroos

To understand the information theory perspective of the problem, we use an example (Gull et al. 1984): the kangaroo problem.

We have been studying the kangaroo during a short Australian visit and have arrived at two conclusions: one third of kangaroos have blue eyes, and one third are left-handed. We are asked to estimate the percentage of kangaroos that both have blue eyes and are left-handed. Basically, we are being asked to fill in Table 1.

	Left-handed: True	Left-handed: False
Blue eyes: True	P_1	P_2
Blue eyes: False	P_3	P_4

Table 1: Overall statistics for kangaroo problem

And we have the following constraints:

$P_1+P_2 = P_1+P_3 =1/3$

$P_1+P_2 + P_3+P_4 =1$

Clearly, we have incomplete information. Table 2 provides a few examples of how we could answer.

	L-H:T	L-H: F
Blue: True	1/9	2/9
Blue: False	2/9	4/9

No correlation

	L-H:T	L-H: F
Blue: True	1/3	0
Blue: False	0	2/3

Positive correlation

	L-H: T	L-H: F
Blue: True	1/9	2/9
Blue: False	2/9	4/9

Negative correlation

Table 2: Three possible conclusions for kangaroo problem

Due to our limited information, none of these options is considered to be more likely. However, if we must answer the question, information theory insists that the “best” answer is that in which there is no correlation between having blue eyes and left-handedness, since the data do not suggest one. In this simple example, of course, we could answer by showing the different possibilities. In a digital picture, however, we have too many options, and supplying the “best” response becomes a near necessity.

Now, we undertake the mathematical problem of generalizing this idea. We look for a function of the form

$$R(p) = \sum_{i=0}^{3} r(p_i)$$

Equation 3

that we maximize to find our solution of least correlation, where the p_i are the proportions and r is the function acting on each proportion. It is possible to derive the form of $r(p_i)$, but we will use an empirical formulation. The fact that we seek an additive form of $R(p)$ should not be surprising, since other forms would suggest a correlation between the proportions.

Using the constraints, we can reconstruct the kangaroo table as shown here:

	Left-handed: True	Left-handed: False
Blue eyes: True	P_0	1/3 – P_0
Blue eyes: False	1/3 – P_0	1/3 + P_0

Table 3: Re-write of Table 1 with constraints added

and, of course, we seek a maximum of $R(p)$ using the first derivative:

$$\frac{dR(p_0)}{dp_0} = 0$$

Table 4 shows the results of the maximization on p_0.

$r(p_i)$	Result for Kangaroos: p_0
$-p_i \ln p_i$	1/9 = 0.11111
$-{p_i}^2$	1/12 = .08333
$\ln\ (p_i)$	.13013
${p_i}^{1/2}$	.12176

Table 4: Results of maximization of Equation 3 for various functional forms (Gull et al. 1984)

The form of $r(p_i)$ which satisfies our desire for an uncorrelated answer is $-p_i \ln p_i$. In the next section where we seek a more physical description of entropy, we will informally derive this functional form. From an information theory standpoint, though, entropy is thought of as the amount of disorder, or lack of correlation, in a set of data. In our example, we define entropy quantitatively as

$$R = -\sum_{i=0}^{N-1} p_i \ln p_i \qquad \text{Equation 4}$$

Though we dealt with a somewhat abstract kangaroo problem, the extension to digital images is simple. In an image, our $p_i's$ are now the proportion of the total image brightness that belongs in a particular pixel (what we would measure with no PSF effect), and our constraints become

$$I_k = \sum_{i=0}^{N-1} p_i PSF_{k,i} \qquad \text{Equation 5}$$

and

$$\sum_{i=0}^{N-1} p_i = 1 \qquad \text{Equation 6}$$

Usually these constraints do not provide a unique answer in themselves, and we must use the principle of maximum entropy to obtain our desired image.

Motivating the MEM Using Statistical Mechanics: Monkees

For the student who owns the text "Introduction to Quantum Mechanics" by David J. Grifitths (1995) there is an excellent treatment on the basics of quantum statistical mechanics that complements this and the next section very well. The website by Steinbach (2010) is also helpful.

Statistical mechanics relates measurements at a macroscopic level to the mechanics of the microscopic. Macroscopically, we consider variables like temperature and volume (macrostates), whereas microscopically we are concerned with single particles being in particular states (microstates). Usually we surrender the idea that we can keep track of all the particles and which states they are in, though we can easily measure the total energy. Imagine, though, that we could count all the *possible* microstates that give the same total energy. If all the microstates are equally likely, then we define a statistical mechanics measure of entropy:

$$S = k_B \ln \Omega \qquad \text{Equation 7}$$

where S is entopy, k_B is Boltzmann's constant, and Ω is the number of microstates that give rise to our measured macrostate.

In our deconvolution problem, we have many different possible macrostates – many different possible "truths" that fit the constraints of our problem defined in Equation 5 and Equation 6. With the MEM, we choose the macrostate that has the highest entropy – the greatest number of ways that the photons can be arranged to give the same result. Just like statistical mechanics assumes that all microstates are equally likely, we assume that the processes (outside of the constraints) which create our image are random, and therefore all arrangements of photons which create the image are equally likely.

To represent a random process, researchers in many studies sometimes use the analogy of "monkeys at typewriters". The idea is that each monkey hits one key at a time at random, and after enough time has passed, the monkey produces the complete works of Shakespeare or some other key progression that does not appear random in any way (the time involved is usually stated in terms of the age of the universe, so I wouldn't bring a typewriter to the zoo hoping to get a Pulitzer). In the spirit of "monkees at typewriters" we consider "monkeys tossing balls", often used to describe the statistics of the MEM.

Each pixel in our CCD array is a bin large enough to contain an enormous number of balls. While we turn our heads, a monkey tosses identical balls (photons), one at a time, into a bin at random (please do not get confused with the "one at a time" statement – certainly in a digital image there are many photons striking the CCD in a given amount of time. I merely need a way to label the photons: the "first", "second", and so on.)

Let's consider two example images that both meet the constraints defined in Equation 5 and Equation 6. Each image has a total of four photons (balls). In image A, all the photons are in one pixel (bin). In image B, there are two photons in one pixel and two photons in another pixel. We are asked to find the most likely image of the two. With image A, there is only one possible configuration: photons one through four were deposited into one pixel. With image B, we have more options, three of which are shown in Figure 8.

	Photon 1	Photon 2	Photon 3	Photon 4
Pixel 1	X	X		
Pixel 2			X	X

	Photon 1	Photon 2	Photon 3	Photon 4
Pixel 1	X		X	
Pixel 2		X		X

	Photon 1	Photon 2	Photon 3	Photon 4
Pixel 1	X			X
Pixel 2		X	X	

Figure 8: Three possible microstates for image B macrostate

Since we did not see how the photons struck the CCD (we did not observe the monkey tossing the balls), we have to assume that each possible arrangement is equally likely. Therefore, Image B is the more-likely macrostate, allowing many more possibilities of incoming photons (many more ways the monkey could have thrown the balls). This is a fundamental difference from the information theory (kangaroo) derivation, because we *do* assert that the image we choose – the one with the greatest entropy in the statistical mechanics sense - is the most likely.

We still need to get a general equation for entropy the same as Equation 4. We start with the first pixel (bin). We wish to know how many ways we can choose, from all the photons (balls) incorporated in the image, the photons in the first pixel (calling this pixel the "first" really has no meaning except that we wish to consider each pixel once and separately from the rest of the pixels). Out of N total photons in the image, we have N ways to pick the first photon, N-1 to pick the second, and so on; in general there are

$$\frac{N!}{(N-N_1)!}$$

ways out of N_1total photons in the first pixel. However, choosing in this manner counts the permutations within the pixel (bin). Since we do not care about the order in which we have chosen the N_1 photons, we divide by $N_1!$, resulting in the binomial coefficient:

$$\frac{N!}{N_1!\,(N-N_1)!}$$

Now we continue the process for the second pixel, for which $N - N_1$ pixels remain in the image, so we have:

$$\frac{(N-N_1)!}{N_2!\,(N-N_1-N_2)!}$$

The product of all the pixels gives the total number of possible microstates:

$$\frac{N!}{N_1!\,(N-N_1)!}\;\frac{(N-N_1)!}{N_2!\,(N-N_1-N_2)!}\;\frac{(N-N_1-N_2)!}{N_2!\,(N-N_1-N_2-N_3)!}\cdots = N!\prod_{n=1}^{M}\frac{1}{N_n!}$$

where M is the total number of pixels that have photons.

Now we convert the N_n using the notation from Equation 4

$$N_n = p_i\,N$$

and insert it to get

$$\Omega = N!\prod_{n=1}^{M}\frac{1}{(N\,p_i)!}$$

Next we use this equation for the total number of microstates in the entropy equation, Equation 7.

$$S = k_B\,\ln\,(N!\prod_{n=1}^{M}\frac{1}{(N\,p_i)!})$$

Maximizing the function does not depend upon the constant in front (except that it is positive) and I will drop it from now on. Using the laws of products within logarithms, we can expand the above equation

$$S = \ln N! - \sum_{i=0}^{M}\ln(Np_i)!$$

We can then use the Stirling approximation on the second term since the total number of photons and the number of photons in a pixel is usually large ($\ln L! \sim L \ln L - L$):

$$S = N \ln N - N - \sum_{i=0}^{M} N\, p_i \ln(N\ \ p_i) + \sum_{i=0}^{M} N p_i$$

And after again using the law of products within logarithms, we have

$$S = \mathrm{N} \ln N - N - N \sum_{i=0}^{M} \mathrm{p_i} \ln\ (N\ ) - N \sum_{i=0}^{M} p_i \ln p_i + N$$

$$= -\mathrm{N} \sum_{i=0}^{M} p_i \ln p_i$$

This is not quite the same form as Equation 4, but in finding the maximum entropy the constant N in front is irrelevant (just like the k_B constant).

Continuing on: An Expression for the Probability Function

We now seek an expression for the probability function, given the constraints of Equation 5 and Equation 6 , and the understanding that we wish to maximize Equation 4. Maximizing a multi-variable function subject to constraints can be achieved using Lagrange multipliers. In general, a function $F(x_1, x_2, x_3, \ldots)$; with constraints $f_1(x_1, x_2, x_3, \ldots) = 0$, $f_2(x_1, x_2, x_3, \ldots) = 0$, …; is maximized by introducing a new function $G\left(x_{1,}x_2, x_3, \ldots, \lambda_1, \lambda_2, \lambda_3 \ldots\right) = F + \lambda_1 f_1 + \lambda_2 f_2 + \lambda_3 f_3$ where the λ's are the Lagrange multipliers. We find the maximum for F by setting the derivatives of G to zero, i.e.

$$\frac{\partial G}{\partial x_i} = 0$$

$$\frac{\partial G}{\partial \lambda_i} = 0$$

for all i.

In our case (Hollis et al. 1992),

$$G = -\sum_{i=0}^{M} p_i \ln p_i + \lambda_0 (1 - \sum_{i=0}^{M} p_i) + \sum_{k=0}^{K} \lambda_k (I_k - \sum_{i=0}^{M} p_i\, PSF_{i,k})$$

From the first derivative we get

$$\frac{\partial G}{\partial p_i} = 0 = -ln\, p_i - (1 + \lambda_0) - \sum_k \lambda_k PSF_{i,k}$$

We will absorb the 1 inside the parenthesis into the λ_0 constant. This is allowed because, after taking the other two derivatives, λ_0 is not in the remaining equations.

We can rearrange the above equation to get:

$$p_i = \exp\ (-\lambda_0 - \sum_k \lambda_k\, PSF_{k,i}) \qquad \textbf{Equation 8}$$

The next derivative we take merely gives us one of our constraint equations, Equation 6:

$$\frac{\partial G}{\partial \lambda_0} = 0 = 1 - \sum_i p_i$$

We can combine Equation 6 and Equation 8 like so:

$$1 = \sum_i \exp\ (-\lambda_0 - \sum_k \lambda_k\, PSF_{k,i})$$

Therefore,

$$\exp\ (\lambda_0) = \sum_i \exp(-\sum_k \lambda_k PSF_{k,i})$$

The remaining derivative gives us our other constraint:

$$\frac{\partial G}{\partial \lambda_k} = 0 = \ I_k - \sum_i p_i\, PSF_{k,i}$$

Inserting p_i from Equation 8 gives

$$0 = I_k - \sum_i (\exp\ (-\lambda_0 - \sum_s \lambda_s PSF_{s,i}))\ PSF_{k,i}$$

Changing the summation index inside the exponent is important. We cannot use i in both situations because the indexes arose from separate equations. Finally, we multiply by exp (λ_0) to arrive at

$$0 = \sum_i (\exp\ (-\sum_s \lambda_s PSF_{s,i}))((I_k) - \sum_i \exp\ (-\lambda_s PSF_{s,i}))\ PSF_{k,i})$$

We cannot solve for the Lagrange multipliers ($\lambda's$) analytically. In the next section I will discuss a computational solution.

A Computational Method For Maximum Entropy Deconvolution

The method presented here follows both Hollis et al. (1992) and Alhassid et al. (1977).

We will start by introducing a trial set of Lagrange multipliers under the constraint

$$\exp(\lambda_0^T) = \sum_i \exp(\sum_k \lambda_k^T PSF_{k,i}) \qquad \textbf{Equation 9}$$

which is formed from Equation 6 and Equation 8.

The following is known as Gibb's inequality, for which I have provided a proof in Appendix B:

$$\sum_i p_i \ln(\frac{p_i}{p_i^T}) \geq 0$$

And therefore:

$$-\sum_i p_i \ln(p_i) \leq -\sum_i p_i \ln(p_i^T) \qquad \textbf{Equation 10}$$

Our goal will be to minimize the RHS of this inequality. While it may seem strange that we are minimizing in order to maximize entropy, consider that entropy was maximized through the Lagrange multiplier method in the section above. Our initial trial set of λ's (Lagrange multipliers) does not take the actual image into account, so the overall computation will find the set of Lagrange multipliers that minimizes the RHS of Equation 10, thereby meeting the constraints from Equation 5 and Equation 6.

We can rewrite the RHS of Equation 10 using

$$p_i^T = \exp\ (-\lambda_0^T - \sum_k \lambda_k^T\ PSF_{k,i})$$

And Equation 5 and Equation 6 to get

$$F = \lambda_0^T + \sum_k \lambda_k^T I_k$$

which is a new expression for the RHS of Equation 10.

We can find the minimum of F via

$$0 = \frac{\partial F}{\partial \lambda_k^T} = \frac{\partial \lambda_0^T}{\partial \lambda_k^T} + I_k$$

By implicit differentiation of Equation 9, combined with Equation 6, we get a new equation for the minimum of F

$$G = \frac{\partial F}{\partial \lambda_k^T} = -\sum_i p_i^T PSF_{k,i} + I_k$$

If we choose our trial solutions perfectly, these equations will all equal zero. To get to that point, we use a recursive relation

$$\lambda_k^T(new) = \ \lambda_k^T\,(old) - \ G_k$$ Equation 11

Consider one loop to see how this brings us consecutively closer to a solution: if our p_i^T is too big, then G will be negative. The recursive relation will make λ_k^T bigger, making the next p_i^T for the next loop smaller.

Finally, we have arrived at an overall maximum entropy solution.

Using The Maximum Entropy Method

Before judging the effectiveness of the MEM, we must create a methodology by which we can compare a deconvolved image to “truth”. I present the “truth” image.

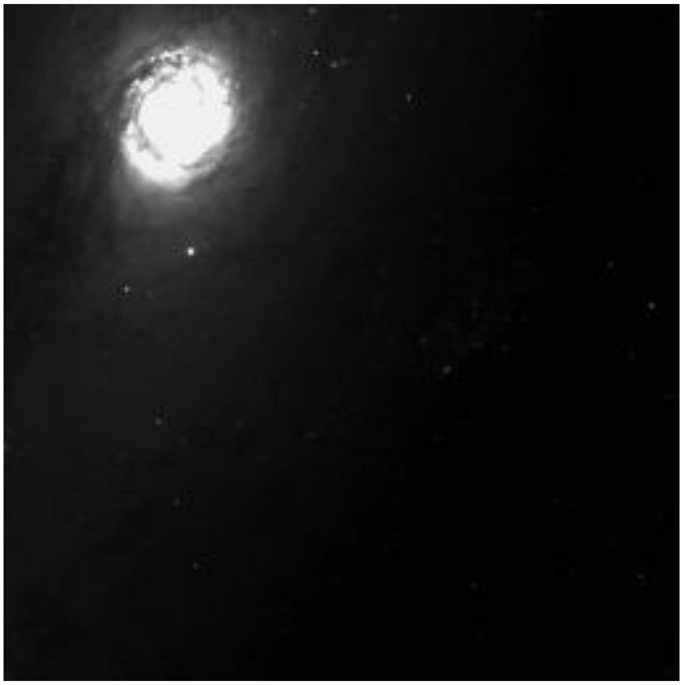

Figure 9: “Truth” image (image obtained from Space Telescope Science Institute website)

In fact, the “truth” image is an imperfect image in its own right. It was taken by a telescope where the optics has a corresponding Point-Spread Function that blurred truth (*real* truth). If we had a measurement or model of the PSF for the telescope we could attempt to deconvolve truth from this image. I will not attempt to do so for two reasons: First, we have no idea what truth really looks like, so we would not be able to compare the deconvolved result to what it should really look like – again, real truth. Second, noise in the image would affect the deconvolution, and while I do want to show how image noise affects the MEM, I also want to show how well the MEM works without noise. We should consider that situation to be an unobtainable ideal case from which we can judge the best possible performance of the MEM.

With that understanding, we convolve the “Truth” image (Figure 9) with a PSF (Figure 10) to create the blurred image (Figure 11). We will hope to get “Truth” (Figure 9) back from this image. At this point, we introduce no image noise.

The PSF is displayed as a surface in three dimensions. The higher the surface, the greater the PSF value, or intensity. It can also be displayed as an image. Both representations are shown in Figure 10.

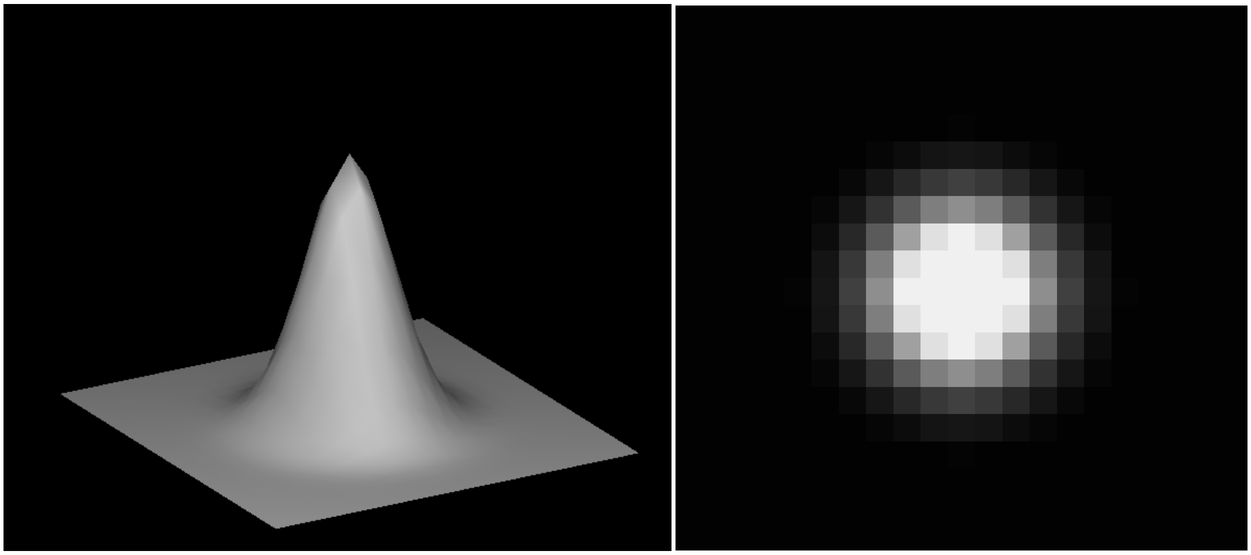

Figure 10: PSF used for example image represented as a 3-d surface (left) and an image (right)

Figure 11: Example image obtained by convolving “truth” (Figure 9) with the PSF (Figure 10)

The MEM can be computationally implemented in many ways, but we will use an algorithm derived upon the analysis performed in the previous section. I have provided the program used in Appendix A (Varosi et al. 1997). This algorithm does provide the option of using a faster recursive relation than that of Equation 11, but otherwise the code should be recognizable from the equations in the previous section.

The algorithm is iterative, with each iteration making an improvement upon the deconvolved image. The end result of using the MEM is smoothest image possible, equivalent to the greatest entropy. Figure 12 shows the output from the 1st, 2nd, 10th, 100th, 500th, and 1500th iteration. Each iteration took approximately one second on a 2.1 GHz machine, and the image is 256 by 256 pixels. I did use the faster recursive relation option. I did not incorrectly paste the image after the 1st iteration. It is there to show how the MEM begins with a completely smooth image (blank) and then fills in detail. You can see that the image improvement plateaus, especially considering the little perceivable difference between the 500th and 1500th iterations. Also note that, even without any noise, the MEM is not perfect, which you can easily see when comparing the result of the 1500th iteration in Figure 12 to Figure 9, or “truth”.

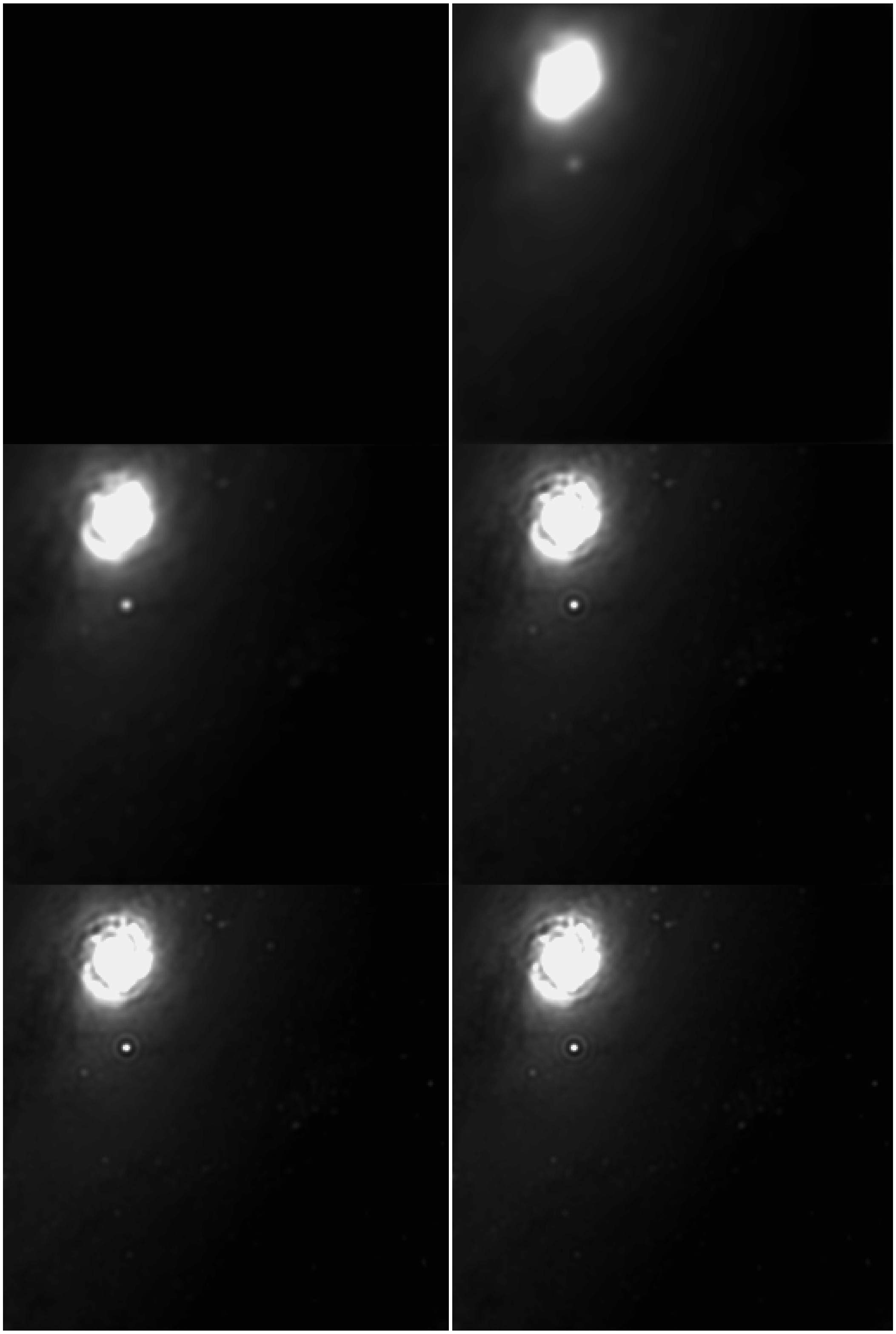

Figure 12: Deconvolution result after using MEM algorithm after 1 iteration (top left), 2 iterations (top right), 10 iterations (middle left), 100 iterations (middle right), 500 iterations (bottom left), and 1500 iterations (bottom right)

With digital images, there are two types of noise that affect measurements, and we will consider both here. We begin with “shot noise”. Looking at our “truth” image, consider that there is a physical process behind every pixel that is producing some number of photons in a given amount of time. However, there is some variablilty in the rate at which the photons are produced. Statistically, when we are measuring a process that outputs something that we count in an amount of time, it follows the Poisson distribution. This distribution is a special type of Gaussian where the standard deviation equals the square root of the mean number of counts. An important fact to consider is that, as the mean increases, the percent of the image that is noise decreases. In that sense, measurements of bright scenes (or longer exposures) are less noisy than dark ones, as the signal-to-noise ratio increases with brightness. If you have a dark digital image on your computer, use a simple image manipulation program and increase the brightness. Because this will multiply both the signal and noise by the same number (it does not do the same thing that a longer exposure would), the signal-to-noise ratio does not change, and you will see a noisey image. In fact, the image will probably look a lot like Figure 13. Here, I have taken our “truth” image and used a random number generator with each pixel to vary the brightness via the Poisson distribution. Figure 14 shows our new blurred image. I have made the assumption that the point-spread-function is mostly accounted for near the telescope, which is why I first added noise to the “truth” image and then convolved it with the PSF. The 1500th iteration of the MEM is shown in Figure 15, and you can clearly see a noisier result than the corresponding iteration with no noise in Figure 12.

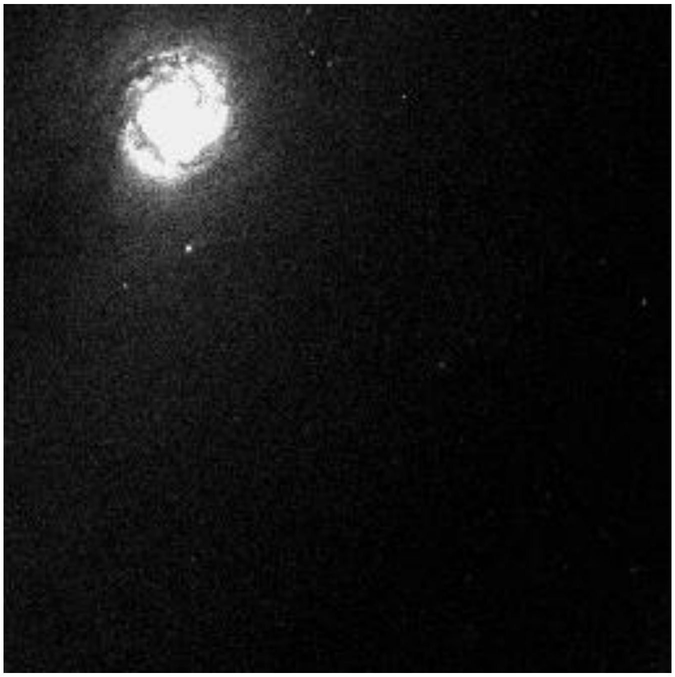

Figure 13: “Truth” image with Poisson noise added

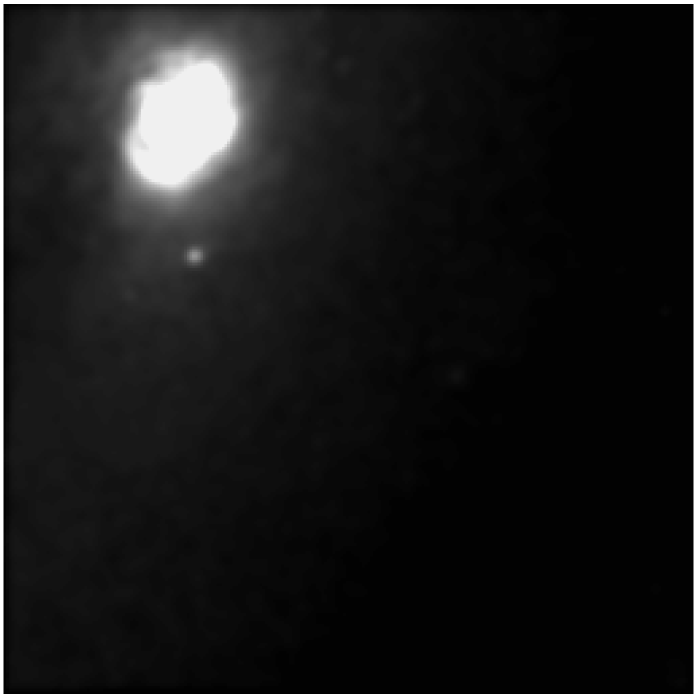

Figure 14: Result of “truth” image with poisson noise added after convolution with PSF

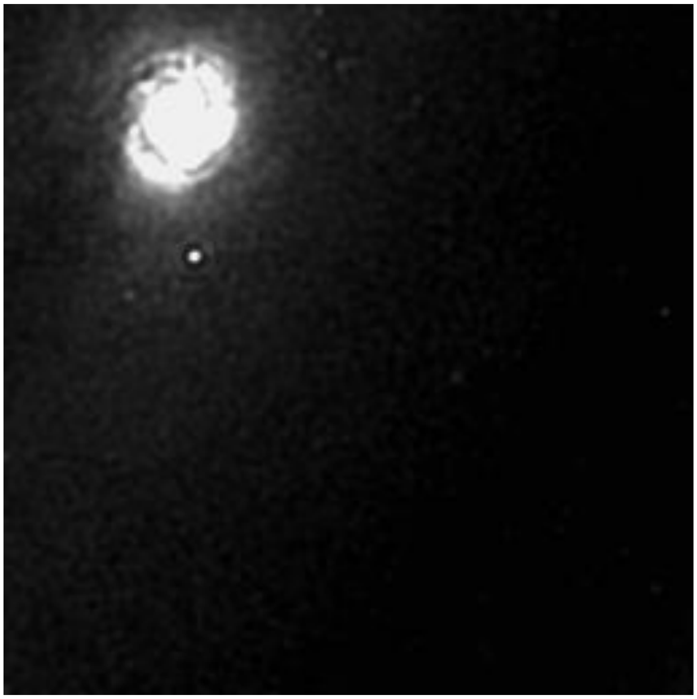

Figure 15: Deconvolution result after 1500 iterations using Poisson-noised “truth”

The other types of noise that affect an image are mostly due to thermal energy or amplifier noise from the electronics. These sources are usually represented by Gaussian distributions with a mean and standard deviation that depend heavily on the CCD and related electronics. Figure 16 shows our “truth” image after I have added Gaussian noise with a mean of zero and a standard deviation of one. Certainly it is fallacious to assume that the noise can be aptly characterized by one Gaussian distribution, but it is the best we can do with no information about the device used for the measurement. In this case, it is appropriate to add the noise after the creation of the blurred image since the noise is heavily signal-independent.

It is interesting to see that, while it is difficult to see any difference in our Gaussian-noised blurred image from the original blurred image in Figure 11, it has a drastic result of the MEM deconvolution, shown in Figure 17. This is perhaps because the noised image manifests itself with greater entropy than is real. It is also worth mentioning that the simple MEM algorithm we used did not take noise into account, where a more robust method certainly would.

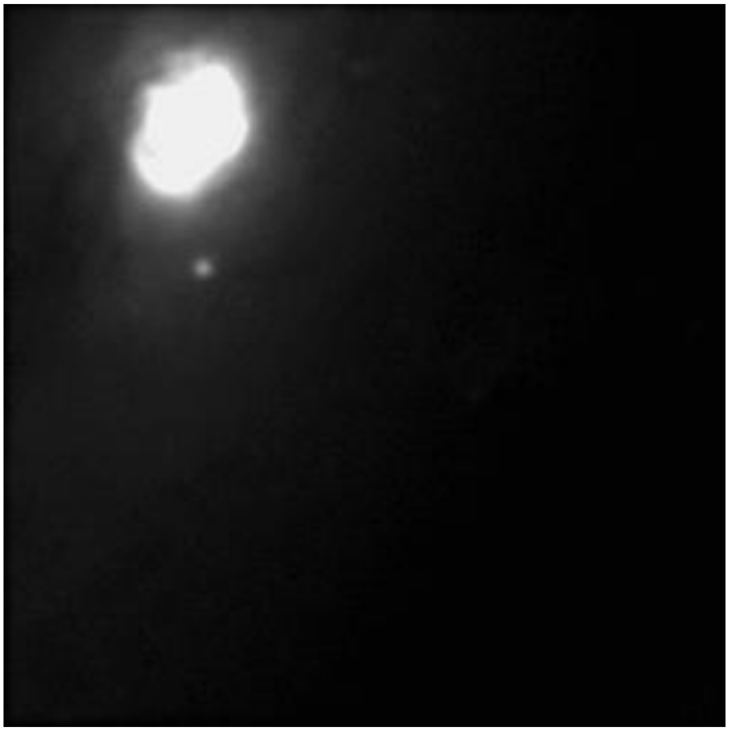

Figure 16: “Truth” image after convolution with PSF followed by adding Gaussian noise

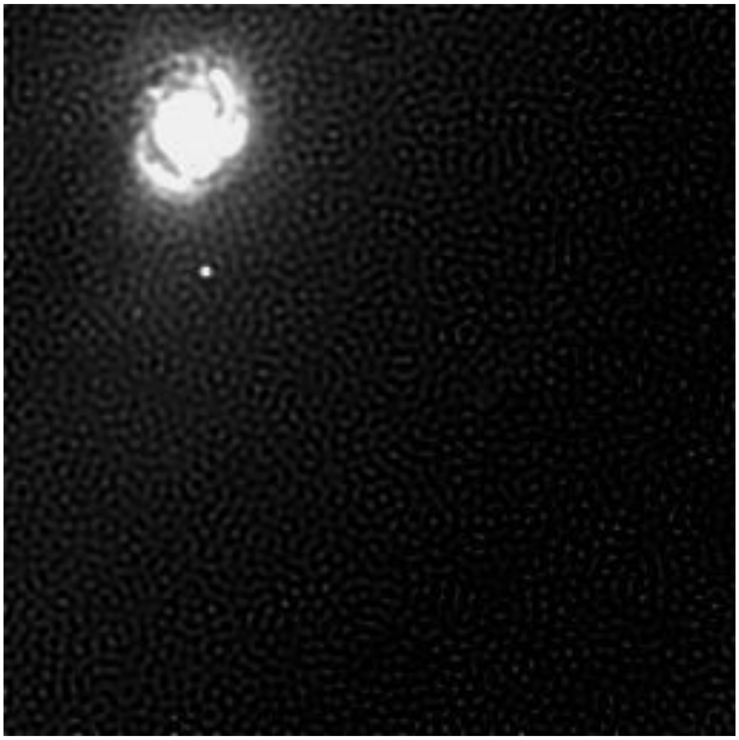

Figure 17: Deconvolution result after 1500 iterations using Gaussian-noised “truth”

While using the MEM for standard deconvolution is a powerful process in its own right, it has other capabilites that I will not demonstrate in this paper. One of these is the ability to somewhat deconvolve an image *with no model or measurement of the point-spread function*. This is called blind deconvolution. In another capacity, the MEM can be used to recover parts of images where all the pixel information has been lost.

With the rapid increase in computing speed combined with better technology, deconvolution methods like the Maximum Entopy Method are likely to be continuously improved, literally sharpening our understanding of our universe.

Appendix A: The MEM program used

This is the actual Maximum Entropy Program used for this paper. This was obtained from The IDL Astronomy User's Library from NASA-Goddard at http://idlastro.gsfc.nasa.gov/.   This program was originally written by Frank Varosi in 1992 and was converted into IDL by W. Landsman in 1997. It does not take noise into account.

```
;+
; NAME:
;       MAX_ENTROPY
;
; PURPOSE:
;       Deconvolution of data by Maximum Entropy analysis, given the PSF
; EXPLANATION:
;       Deconvolution of data by Maximum Entropy analysis, given the
;       instrument point spread response function (spatially invariant psf).
;       Data can be an observed image or spectrum, result is always positive.
;       Default is convolutions using FFT (faster when image size = power of 2).
;
; CALLING SEQUENCE:
;       for i=1,Niter do begin
;       Max_Entropy, image_data, psf, image_deconv, multipliers, FT_PSF=psf_ft
;
; INPUTS:
;       data = observed image or spectrum, should be mostly positive,
;                                       with mean sky (background) near zero.
;       psf = Point Spread Function of instrument (response to point source,
;                                                       must sum to unity).
;       deconv = result of previous call to Max_Entropy,
;       multipliers = the Lagrange multipliers of max.entropy theory
;               (on first call, set = 0, giving flat first result).
;
; OUTPUTS:
;       deconv = deconvolution result of one more iteration by Max_Entropy.
;       multipliers = the Lagrange multipliers saved for next iteration.
;
; OPTIONAL INPUT KEYWORDS:
;       FT_PSF = passes (out/in) the Fourier transform of the PSF,
;               so that it can be reused for the next time procedure is called,
;   /NO_FT overrides the use of FFT, using the IDL function convol() instead.
;   /LINEAR switches to Linear convergence mode, much slower than the
;               default Logarithmic convergence mode.
;       LOGMIN = minimum value constraint for taking Logarithms (default=1.e-9).
; EXTERNAL CALLS:
;       function convolve( image, psf ) for convolutions using FFT or otherwise.
; METHOD:
```

```
;            Iteration with PSF to maximize entropy of solution image with
;            constraint that the solution convolved with PSF fits data image.
;            Based on paper by Hollis, Dorband, Yusef-Zadeh, Ap.J. Feb.1992,
;            which refers to Agmon, Alhassid, Levine, J.Comp.Phys. 1979.
;
;       A more elaborate image deconvolution program using maximum entropy is
;       available at
;       http://sohowww.nascom.nasa.gov/solarsoft/gen/idl/image/image_deconvolve.pro
; HISTORY:
;            written by Frank Varosi at NASA/GSFC, 1992.
;            Converted to IDL V5.0   W. Landsman   September 1997
;-

pro max_entropy, data, psf, deconv, multipliers, FT_PSF=psf_ft, NO_FT=noft, $
                                  LINEAR=Linear, LOGMIN=Logmin, RE_CONVOL_IMAGE=Re_conv

        if N_elements( multipliers ) LE 1 then begin
                multipliers = data
                multipliers[*] = 0
          endif

        deconv = exp( convolve( multipliers, psf, FT_PSF=psf_ft, $
                                                        /CORREL, NO_FT=noft ) )
        totd = total( data )
        deconv = deconv * ( totd/total( deconv ) )

        Re_conv = convolve( deconv, psf, FT_PSF=psf_ft, NO_FT=noft )
        scale = total( Re_conv )/totd

        if keyword_set( Linear ) then begin

                multipliers = multipliers + (data * scale - Re_conv)

          endif else begin

                if N_elements( Logmin ) NE 1 then Logmin=1.e-9
                multipliers = multipliers + $
                        aLog( ( ( data * scale )>Logmin ) / (Re_conv>Logmin) )
          endelse
end
```

Appendix B: Proof of Gibb's Inequality

We start with an inequality valid for all $x > 0$. To convince yourself, you may want to graph both sides of the inequality.

$$\ln x \ \leq x - 1$$

We continue, considering only those i for which p_i and p_i^T are non-zero, a set we call I:

$$-\sum_{i \in I} p_i ln \frac{p_i^T}{p_i} \geq -\sum_{i \in I} p_i (\frac{p_i^T}{p_i} - 1)$$

$$-\sum_{i \in I} p_i ln \frac{p_i^T}{p_i} \geq -\sum_{i \in I} p_i^T + \sum_{i \in I} p_i$$

$$-\sum_{i \in I} p_i ln \frac{p_i^T}{p_i} \geq 0$$

$$-\sum_{i \in I} p_i \ln p_i^T + \sum_{i \in I} p_i \ln p_i \geq 0$$

We still need to include those p_i and p_i^T that are zero. The following are true:

$$0 \ln\left(\frac{0}{p_i^T}\right) = 0 \quad \text{and} \quad p_i \ln\left(\frac{p_i}{0}\right) = \infty$$

Therefore,

$$\sum_i p_i \ln(\frac{p_i}{p_i^T}) \ \geq 0$$

References

Alhassid, Y., Agmon, N., Levine, R.D. 1978, Chem. Phys. Let., 53, 22

Gjendemsjø, A., Baraniuk, R. 2007, Convolution – Complete Example, http://cnx.org/content/m11541/latest/

Griffiths, D. J. 1995, in Introduction to Quantum Mechanics (Upper Saddle River, Prentice Hall)

Gull, S.F., Skilling, J. 1984, Proc. IEE, 131, 646

Hollis, J.M., Dorband, J.E., Yusef-Zadeh, F. 1992, ApJ, 386, 293

McLean, I.S. 2008, in Electronic Imaging in Astronomy (New York, Springer)

O'Haver, T.O. 2008, http://terpconnect.umd.edu/~toh/spectrum/Deconvolution.html

Steinbach, P.J. 2010, http://cmm.cit.nih.gov/maxent/letsgo.html

Varosi, F., Landsman, W. 1997, http://idlastro.gsfc.nasa.gov/ftp/pro/image/max_entropy.pro