# Blind Deconvolution in Flourescent Microscopy

Tal Kenig
Professor Tziki Kam
Professor Arie Feuer

# Problem formulation

The problem at hand can be formulated as

$$g = n(h * f)$$

where:

- *f* denotes the imaged object
- *h* denotes the imaging system point spread function (psf), which is assumed to be linear and space invariant
- *g* denotes the image acquired by the imaging system*
- *n* denotes a pixel - wise noise function
- * denotes the 3D convolution operator.

The goal is to estimate *f* and possibly also *h* from *g* and any additional prior knowledge available.

# Prior knowledge available

Prior knowledge assumed to be available:

1. The psf is symmetric: *h(x,y,z)=h(-x,-y,-z)*
2. The psf is radial - symmetric in the x-y plane:

$$h(x,y,z) = h_{rad}(\sqrt{x^2+y^2}, z) \equiv h_{rad}(r,z)$$

3. The object is non-negative: $f(x,y,z) \geq 0, \forall x,y,z$
4. The noise is a Poisson process and therefore can be characterized by Poisson statistics with an expectation *f*h*:

$$p(g(x,y,z)\,|\,f,h) = \frac{(f*h)(x,y,z)^{g(x,y,z)}\, e^{-(f*h)(x,y,z)}}{g(x,y,z)!}$$

# Simulation framework

## psf simulation

The psf has been simulated according to the following formula (Born and Wolf, Principles of Optics):

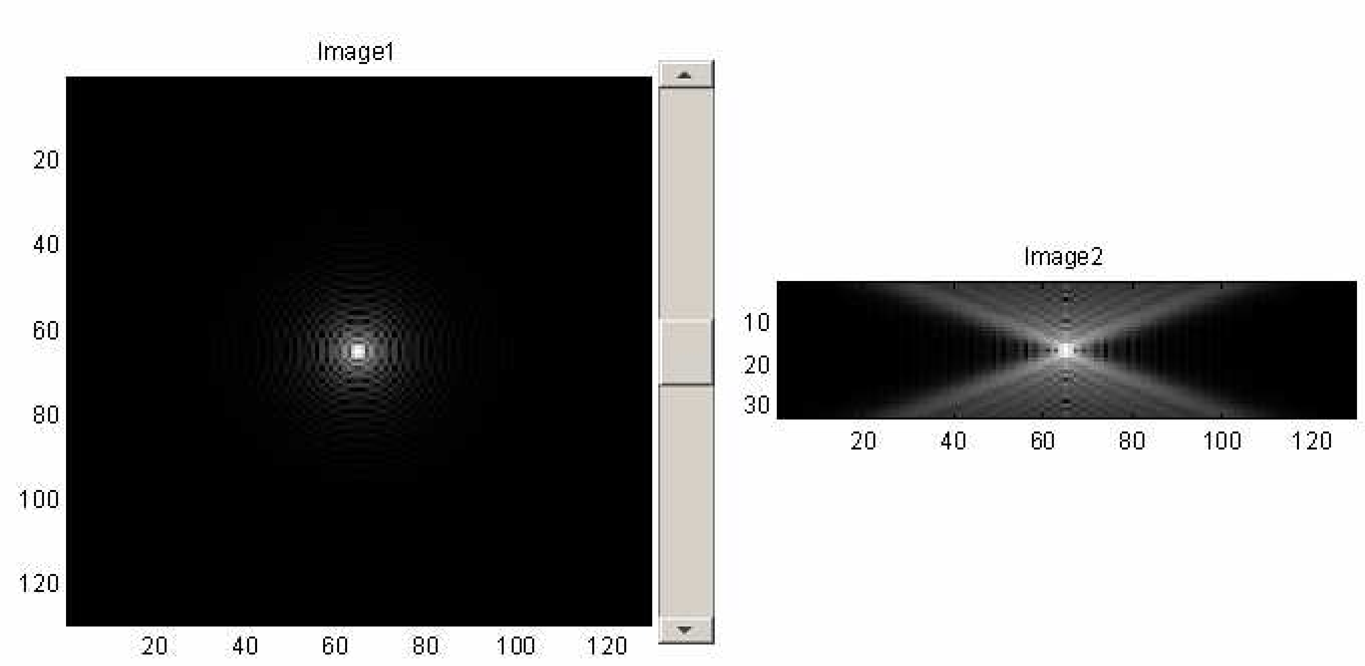
$$\boxed{h_{rad}(r,z) = I_0 \left| \int_0^1 J_0\left(\frac{k\cdot NA}{n_0\cdot m} r\rho\right)\cdot e^{-\frac{1}{2}i\rho^2\left(\frac{k\cdot NA}{n_0\cdot m}\right)^2 z} \rho d\rho \right|^2}$$

- $J_0$ is a Bessel function of the first kind
- $k$ is the wave frequency in the medium $\quad k = \dfrac{2\pi n_0}{\lambda}$
- $\lambda$ is the light wavelength
- NA is the numerical aperture
- $n_0$ is the refractive index in the immersion medium
- $m$ is the CCD magnification.
- $I_0$ is a scaling factor and was chosen so that $h$ integrate to unity.

Typical values used for our psf simulation:

$$n_0 = 1.518, \quad \lambda = 0.5\mu m, \quad NA = 1.45, \quad m = 1$$

Simulated psf (displayed with a logarithmic colormap).

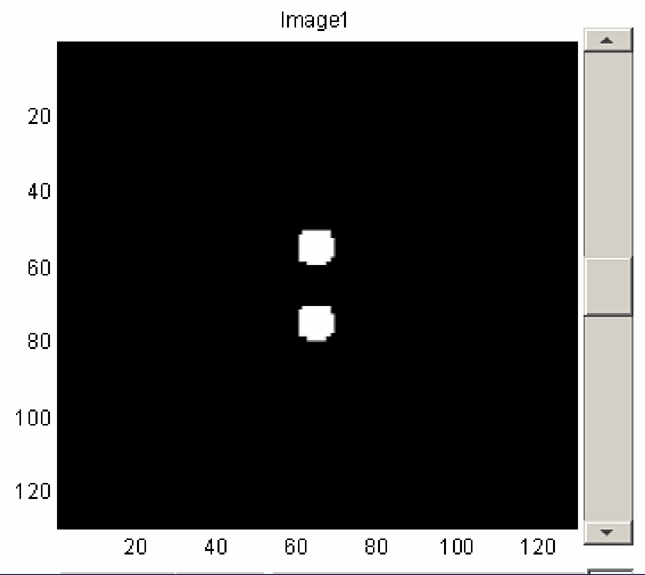


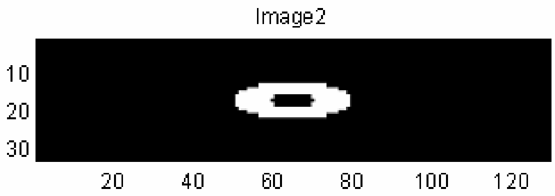
An x-y plane section at z=0

A y-z plane section at x=0

# Object simulation

In order to simulate an imaged object, an image of a torus was generated





An x-y plane section at z=0.

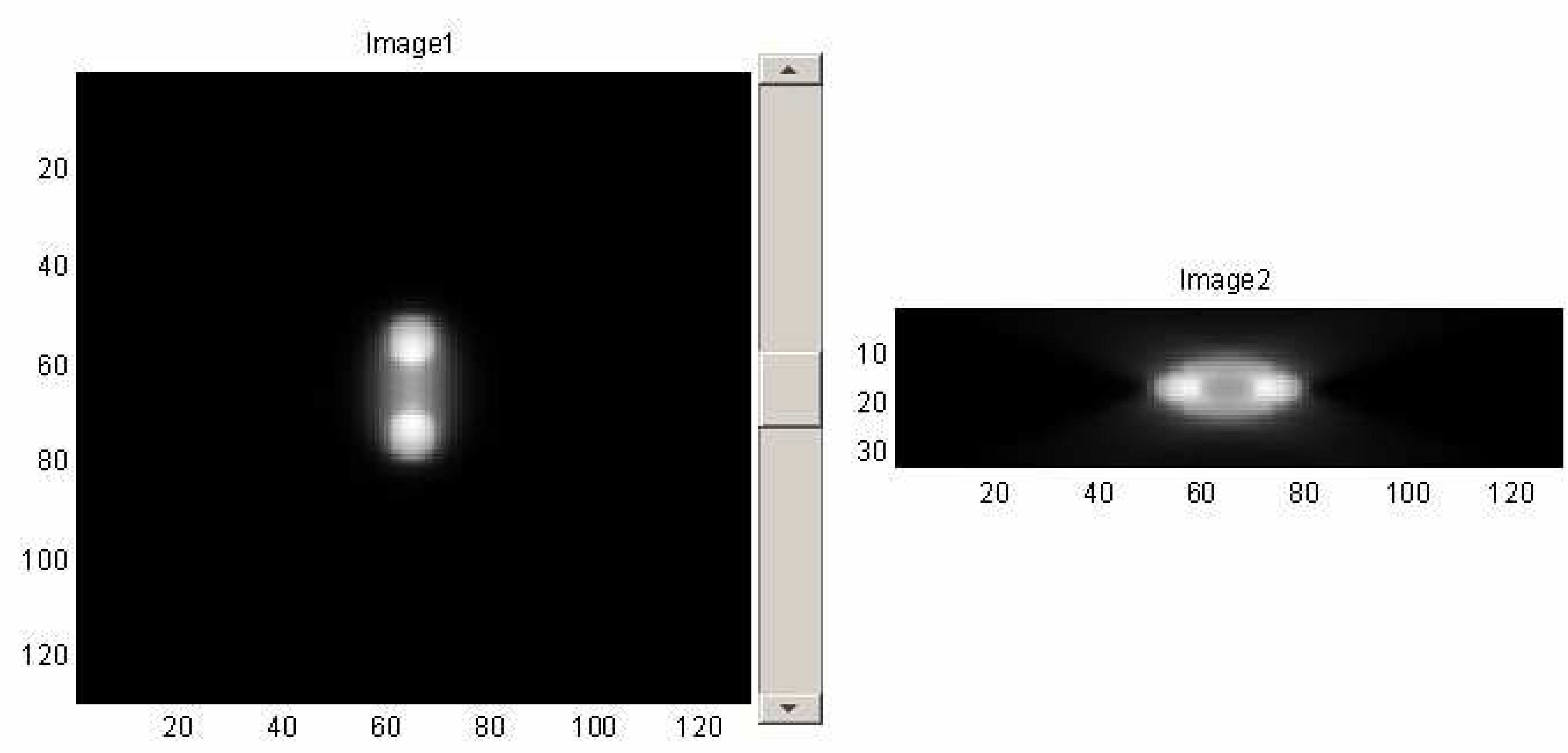
A y-z plane section at x=0

# Data modeling

- The blurred image was simulated by convolving the object with the psf.
- Poisson noise was generated by first linearly scaling the blurred image to a maximal pixel value of 1000, which is typical for a wide - field fluorescence image acquired under good conditions.

  Then, for each pixel, a Poisson distributed random number was drawn out of a Poisson distribution with a mean (and variance) equal to the pixel value.

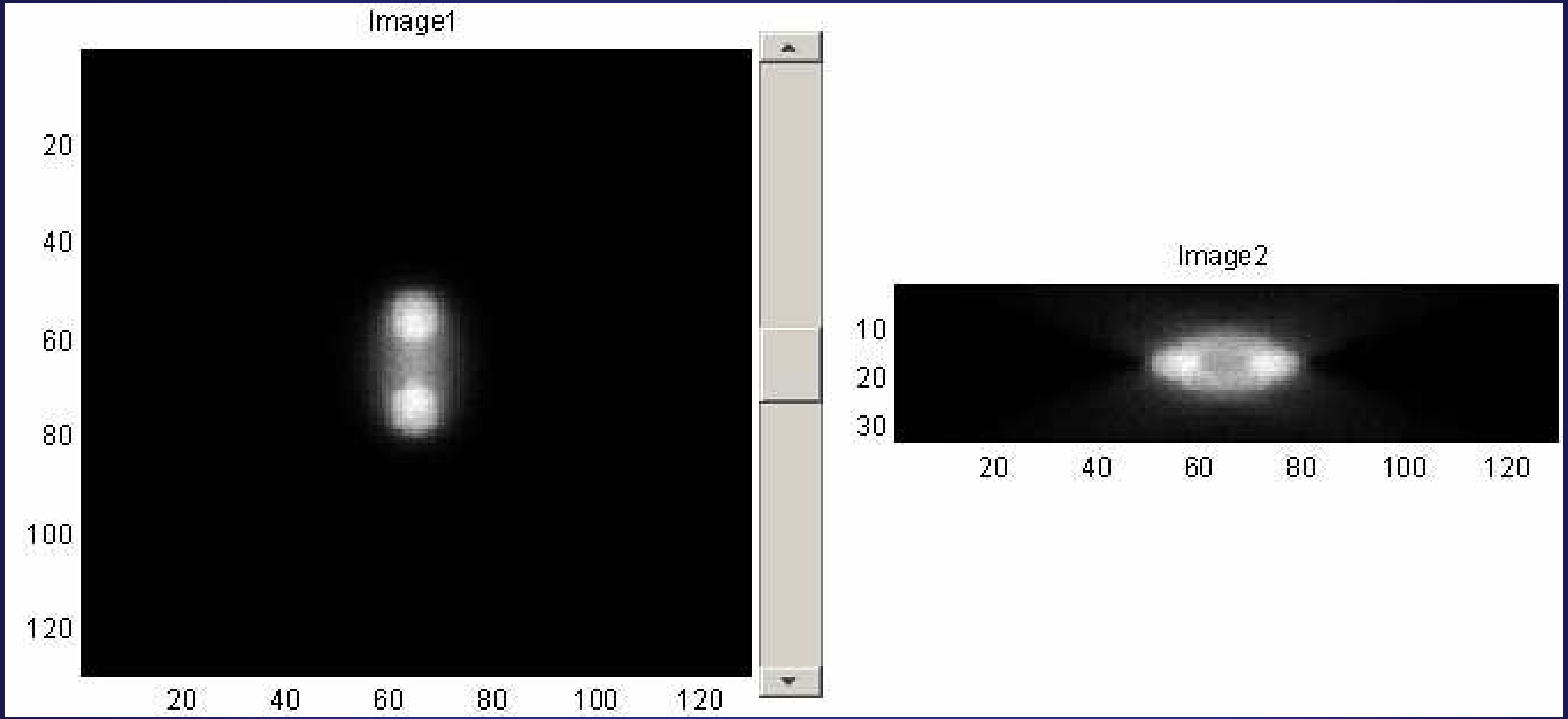
# Blurred object.



An x-y plane section at z=0.

A y-z plane section at x=0

# Blurred and noisy object.



An x-y plane section at z=0.

A y-z plane section at x=0

Visually inspecting previous figures, it is notable that the noise is not overwhelming.

Wide - field microscope is a sensitive imaging device, i.e. there is little photon loss.

Therefore, the acquired image has a typical signal intensity of approximately 200 to 1000 photons per detector pixel, depending on the imaging conditions.

For the simulated signal, the maximal pixel value was 1000 = noise standard deviation (for Poisson noise), namely,

$$\boldsymbol{STD_{noise}} = \sqrt{1000} = 31.6 \Rightarrow 3.16\% \quad \text{of signal}$$

For a weaker signal of approximately 200 photon counts per pixel,

$$\boldsymbol{STD_{noise}} = \sqrt{200} = 14.14 \Rightarrow 7.07\% \quad \text{of signal}$$

Although the noise may seem to be visually insignificant, we will later observe that it might have an adverse effect on the blind deconvolution process, unless properly accounted for.

# Maximum Likelihood Expectation Maximization (MLEM)

The MLEM algorithm seeks to find

$$\hat{f} = \arg\max_{f} \{p(g \mid f)\}$$

Recall

$$p(g \mid f) = \prod_{x,y,z} \frac{(f * h)(x,y,z)^{g(x,y,z)} e^{-(f*h)(x,y,z)}}{g(x,y,z)!}$$

For simplicity, we can equivalently find the extreme points
of any monotonic increasing function of *p(g|f)*, and specifically of *log(p(g|f))*.

Simplifying the expression, we get:

$$log(p(g \mid f)) = \sum_{x,y,z} [g \cdot log(f * h) - f * h - log(g!)]$$

Differentiating w.r.t *f(i,j,k)* and equating to zero, yields:

$$\sum_{x,y,z}\left[h(x-i,y-j,z-k)\cdot\left(\frac{g(x,y,z)}{(f*h)(x,y,z)}-1\right)\right]=0$$

This is true since

$$(f*h)(x,y,z)=\sum_{i,j,k}f(i,j,k)h(x-i,y-j,z-k)$$

Namely

$$\frac{\partial}{\partial f(i,j,k)}(f*h)=h(x-i,y-j,z-k)$$

So that

$$h(-x,-y-z)*\left(\frac{g(x,y,z)}{(f*h)(x,y,z)}-1\right)=0$$

Assuming that *h* sums to unity, we get

$$h(-x,-y-z)*\left(\frac{g(x,y,z)}{(f*h)(x,y,z)}\right)=1$$

Multiplying both sides by *f* yields

$$f\cdot\left(\frac{g(x,y,z)}{(f*h)(x,y,z)}\right)*h(-x,-y-z)=f$$

Which leads to the basic MLEM iteration

$$\boxed{\hat{f}_{n+1}=\hat{f}_n\cdot\left(\frac{g}{\hat{f}_n*h}\right)*h^s}$$

Where $\hat{f}_n$ denotes an estimate for *f* at iteration *n*.

The superscript *s* denotes symmetrical reflection:

$$\xi^s(x,y,z)=\xi(-x,-y,-z)$$

# Blind MLEM

In blind deconvolution algorithms, *h* is unknown.

In order to derive an algorithm that will overcome this by simultaneously estimating both *f* and *h*, we differentiate the log likelihood expression w.r.t. *h(i,j,k)* and equate to zero

$$f(-x,-y-z)*\left(\frac{g(x,y,z)}{(f*h)(x,y,z)}-1\right)=0 \quad \Rightarrow$$

$$f(-x,-y-z)*\left(\frac{g(x,y,z)}{(f*h)(x,y,z)}\right)=\sum_{x,y,z} f(x,y,z)$$

Multiplying both sides by *h*  yields:

$$\frac{h}{\sum\limits_{x,y,z} f(x,y,z)}\cdot\left(\frac{g(x,y,z)}{(f*h)(x,y,z)}\right)*f(-x,-y-z)=h$$

Which leads to the MLEM iteration for *h*:

$$\hat{h}_{n+1} = \frac{\hat{h}_n \cdot \left( \dfrac{g}{f * \hat{h}_n} \right) * f^{\,s}}{\sum\limits_{x,y,z} f(x,y,z)}$$

As $\sum_{x,y,z} f(x,y,z)$ is constant value and $h$ sums to unity, we can drop the denominator term and get the blind MLEM equations:

$$\boxed{\begin{aligned} \hat{f}_{n+1} &= \hat{f}_n \cdot \left( \frac{g}{\hat{f}_n * \hat{h}_n} \right) * \hat{h}_n^s \\ \hat{h}_{n+1} &= \hat{h}_n \cdot \left( \frac{g}{\hat{f}_n * \hat{h}_n} \right) * \hat{f}_n^s \end{aligned}}$$

Where $\hat{f}_n, \hat{h}_n$ denote estimates for *f* and *g*, at iteration *n*, respectively.

# Prior knowledge enforcement

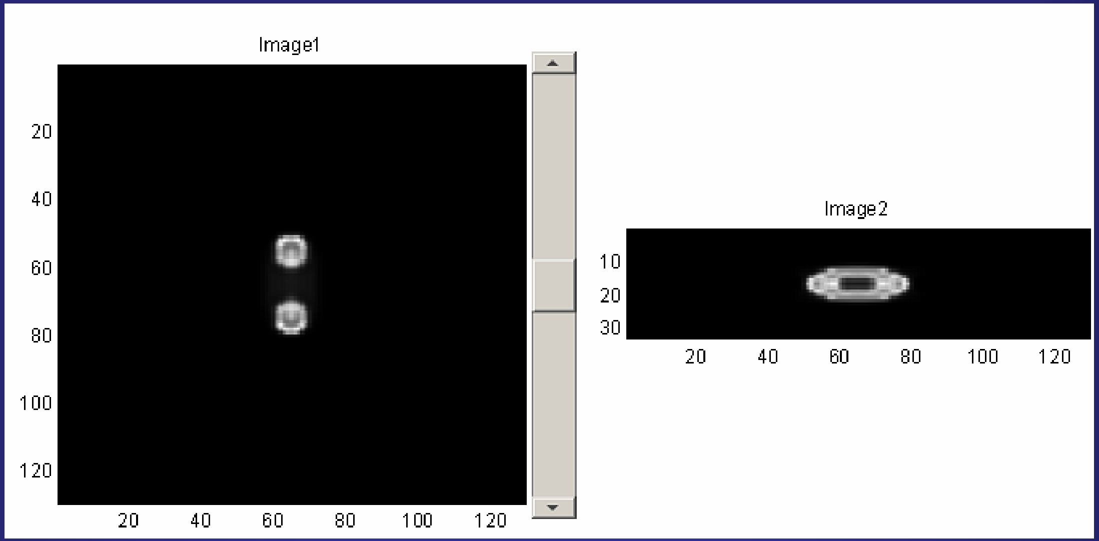
Each MLEM iteration is followed by a projection stage, where prior knowledge is enforced.

## psf symmetry

$$\mathrm{P}\,(h)(r,z)=\frac{1}{2\cdot\#(R=r)}\sum_{R=r}\sum_{Z=z,-z}h(R,Z)$$

Where $\mathrm{P}$ denotes the projection operator.

# Deblurred object (noiseless) using the psf symmetry prior.



An x-y plane section at z=0.  A y-z plane section at x=0

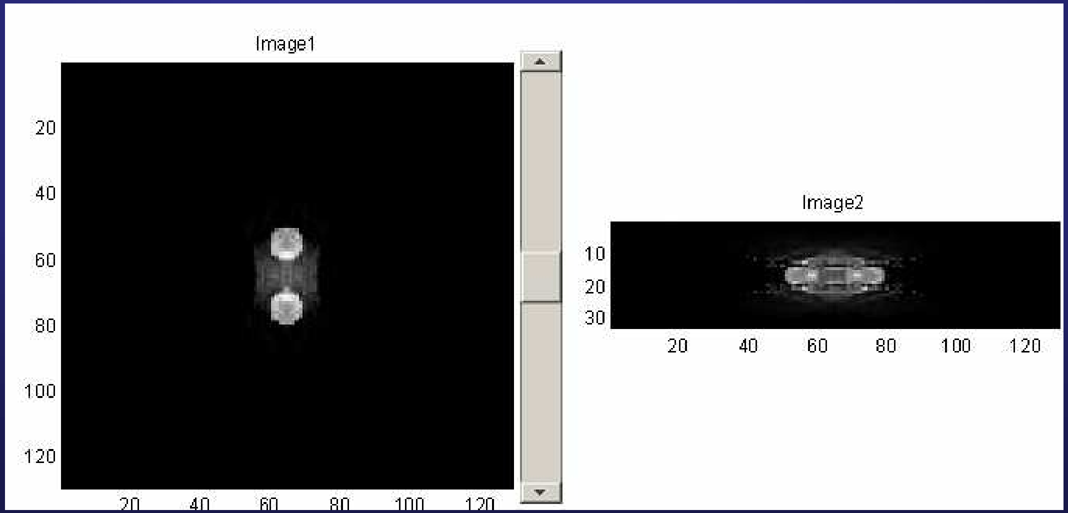
# Phase prior

$\boldsymbol{h}$ symmetric $\Rightarrow$ **H** real

$\Rightarrow$ under noiseless conditions, the phase of $\boldsymbol{F}$ equals the phase of $\boldsymbol{G}$.

$$\Rightarrow \; : \mathrm{P}\,(f)(x,y,z) = \mathrm{F}^{-1}\left\{ |\,F(u,v,w)\,| \cdot e^{i\varphi(G(u,v,w))} \right\}(x,y,z)$$

Deblurred object (noiseless), using phase prior and psf symmetry prior.



It can be seen that the incorporation of the phase prior introduces false noise into the reconstructed image.

The source is unreliable phase values in areas where **H** vanishes, or almost vanishes, due to numerical errors.

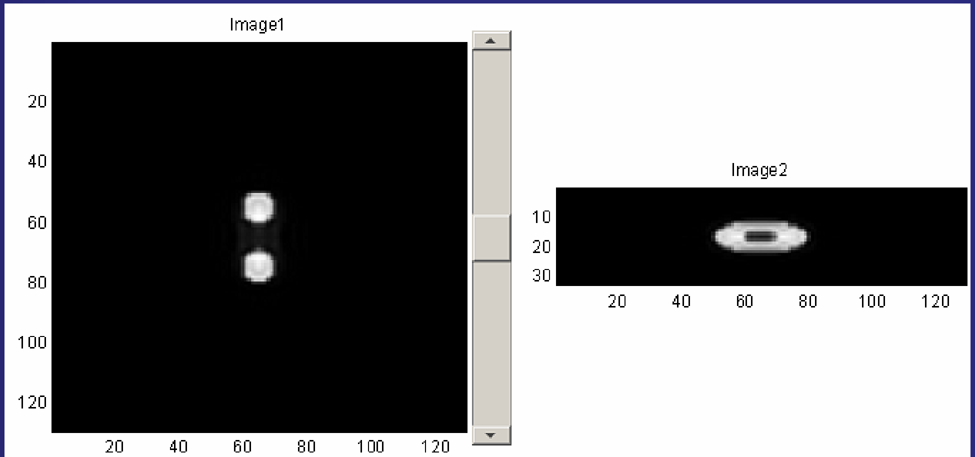
In order to verify this assumption, additional prior information was introduced to the algorithm, in the form of the psf frequency domain support.

$$\mathcal{P}(f)(x,y,z)=\mathcal{F}^{-1}\left\{\begin{array}{cc}|F(u,v,w)|\cdot e^{i\varphi(G(u,v,w))}, & |H(u,v,w)|>c\\ F(u,v,w), & else\end{array}\right\}(x,y,z)$$

Where *c* is a positive constant.

Namely, the phase prior is enforced only at frequency domain locations where *H* does not 'vanish'.

Deblurred object (noiseless), using phase, psf frequency support and psf symmetry priors.



We note that the psf frequency support prior significantly improves the reconstruction results.

However, we thought that this information is usually not available.

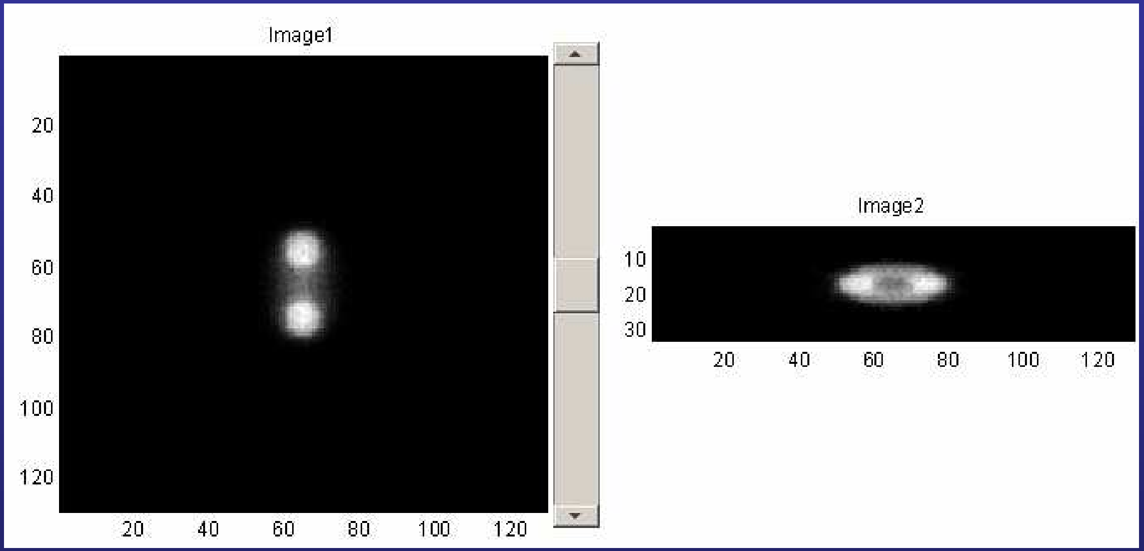
Attempts were made to estimate the psf frequency domain support out of the corrupted image *g*, as well as out of the psf estimates $\hat{h}_k$, yielding poor results (similar to those without the support prior).

The above leads to the conclusions:

- The phase prior is a way to introduce valuable information regarding the imaged object into the deconvolution process.
- However, in order to effectively use the phase prior, one must possess an accurate estimate of the psf frequency support.

# Simulations including Poisson noise

An attempt was made to apply the MLEM algorithm to a noisy blurred image.



Deblurred object (noisy), using only psf symmetry prior.

As can be seen, the presence of Poisson noise prevents the algorithm from converging, yielding a result which is still noisy and blurred.

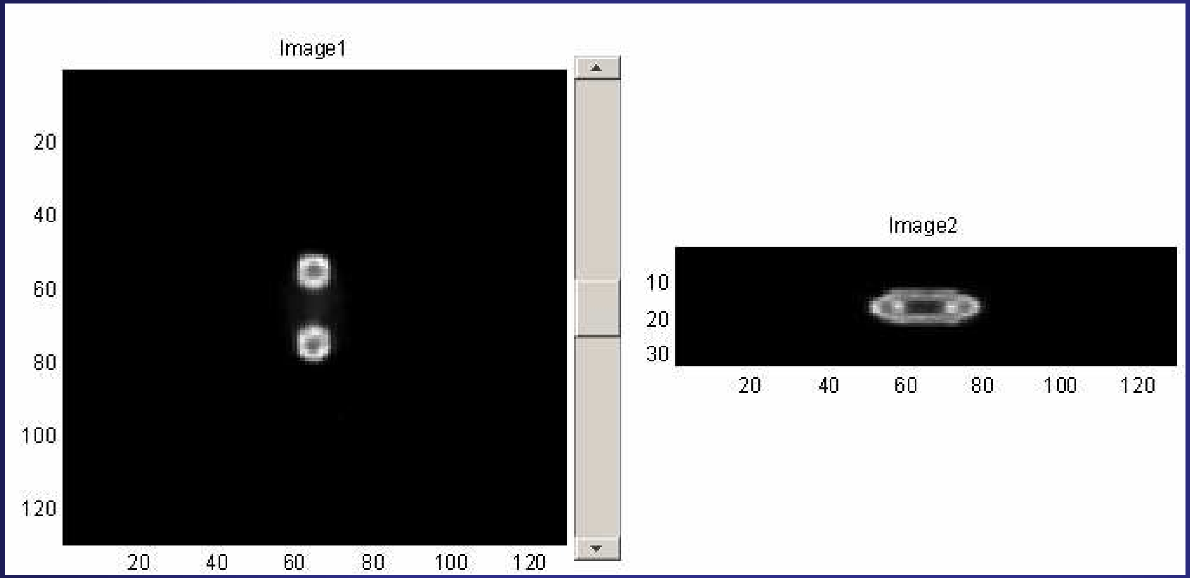
# Overcoming the noise

T overcome the noise, a general scheme of incorporating denoising within the MLEM framework (Starck, Pantin and Murtagh) was used.

$$
\begin{aligned}
R_n &= g - \hat{f}_n * \hat{h}_n \\
\bar{R}_n &= Denoise(R_n) \\
\hat{f}_{n+1} &= \hat{f}_n \cdot \left( \frac{\hat{f}_n * \hat{h}_n + \bar{R}_n}{\hat{f}_n * \hat{h}_n} \right) * \hat{h}_n^s \\
\hat{h}_{n+1} &= \hat{h}_n \cdot \left( \frac{\hat{f}_n * \hat{h}_n + \bar{R}_n}{\hat{f}_n * \hat{h}_n} \right) * \hat{f}_n^s
\end{aligned}
$$

This is based on the observation that $\hat{f}_n * \hat{h}_n$ is likely to be smooth hence $\boldsymbol{R_n}$ will contain the noise.

As an initial experiment, the RD-MLEM was implemented using a 3x3x3 median filter for the *Denoise()* function yielding promising results.



# Penalized MLEM

In our latest result we note that even in the absence of noise, or when the noise is handled well, there are oscillatory artifacts in the vicinity of the edges in the reconstructed image.

In an attempt to overcome this phenomenon we decided to introduce into the algorithm some additional prior knowledge of *f*.

The penalized MLEM (P.J. Green, 1990) attempts to do just that by looking for:

$$\hat{f} = \arg\max_f \{p(f \mid g)\}$$

Using Bayes law, the log of this expression is

$$\log\left(p(f \mid g)\right) = \underbrace{log\left(p(g \mid f)\right)}_{\text{standard MLEM}} + \underbrace{log\left(p(f)\right)}_{\text{prior for f}} - log\left(p(g)\right)$$

A general prior probability function can have the form:

$$p(f) = \mu \cdot \exp(-P(f))$$

Where $\boldsymbol{P(f)}$ can be any function which returns low values for inputs that agree with prior knowledge of f and vice versa (a penalty function).

Then

$$log(p(f \mid g)) = \sum_{x,y,z} \left[ g \cdot log(f * h) - f * h - log(g!) \right] - \beta \cdot P(f) + \text{constant}$$

Differentiating with respect to f and some further development leads to:

$$\hat{f}_{n+1} = \frac{\hat{f}_n \cdot \left( \frac{g}{\hat{f}_n * h} \right) * h^s}{1 + \beta \cdot \frac{\partial}{\partial f} P(f) \mathrm{j}_{\hat{f}_n}}$$

We note that $\frac{\partial}{\partial f}P(f)$ is evaluated at the previous iteration estimate, in order to keep calculations manageable.

This is often referred to in literature as the "One Step Late" (OSL) regularization.

Since the expression here differs from the original one only by terms that are independent of h, the iteration for h does not change.

Hence, the blind P-MLEM iteration is:

$$\hat{f}_{n+1} = \frac{\hat{f}_n \cdot \left( \frac{g}{\hat{f}_n * \hat{h}_n} \right) * \hat{h}_n^s}{1 + \boxed{\beta \cdot \frac{\partial}{\partial f} P(f)\mathrm{j}_{f_n}}}$$

The additional term

$$\hat{h}_{n+1} = \hat{h}_n \cdot \left( \frac{g}{\hat{f}_n * \hat{h}_n} \right) * \hat{f}_n^s$$

# Total Variation penalty

One option for the function *P(f)* is to take it as the total variation (TV) of *f*:

$$P(f) = \int_x \int_y \int_z \left\| '' \; f \right\| dxdydz = \int_x \int_y \int_z \sqrt{f_x^2 + f_y^2 + f_z^2}\, dxdydz$$

Where

$$f_x \equiv \frac{\partial f}{\partial x},\ f_y \equiv \frac{\partial f}{\partial y},\ f_z \equiv \frac{\partial f}{\partial z}$$

The logic behind this choice is that minimization of the TV leads to suppression of small oscillations while maintaining crisp, sharp edges (Rudin, Osher and Fatemi, 1992).

$\frac{\partial}{\partial f}P(f)$ cannot be calculated directly.

But, writing $P = \int\limits_x \int\limits_y \int\limits_z L(x, y, z, f, f_x, f_y, f_z)dxdydz$ and using

the Euler - Lagrange equation

$$\frac{\partial}{\partial f}P = \frac{\partial L}{\partial f} - \frac{\partial}{\partial x}\frac{\partial L}{\partial f_x} - \frac{\partial}{\partial y}\frac{\partial L}{\partial f_y} - \frac{\partial}{\partial z}\frac{\partial L}{\partial f_z}$$

We get in the TV case

$$\frac{\partial}{\partial f}P = -\frac{\partial}{\partial x}\frac{f_x}{\sqrt{f_x^2 + f_y^2 + f_z^2}} - \frac{\partial}{\partial y}\frac{f_y}{\sqrt{f_x^2 + f_y^2 + f_z^2}} - \frac{\partial}{\partial z}\frac{f_z}{\sqrt{f_x^2 + f_y^2 + f_z^2}} = -div\left(\frac{\nabla f}{\|\nabla f\|}\right)$$

which can be calculated in each iteration.

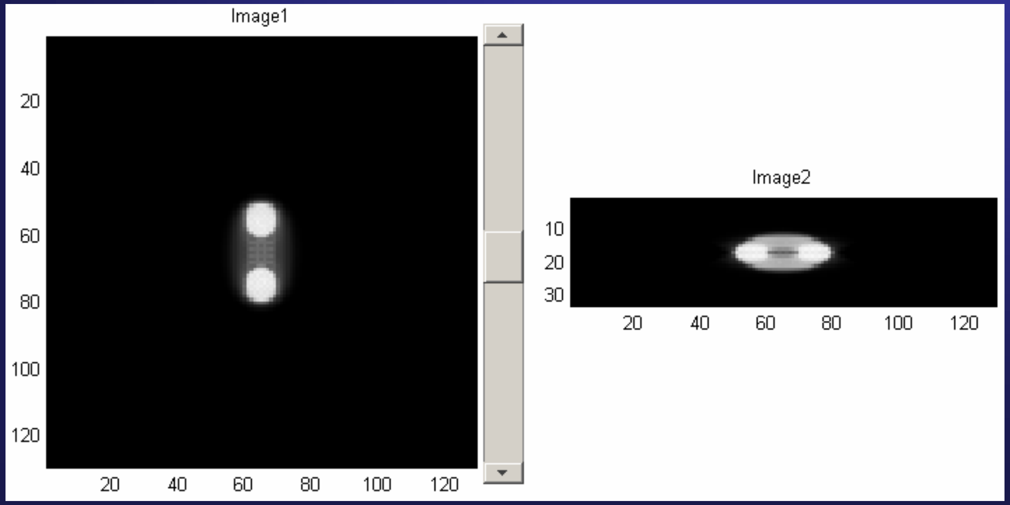
We note that the term $div\left(\frac{\nabla f}{\|\nabla f\|}\right)$ has a physical interpretation and can be recognized as the mean curvature of *f*.

Hence, the TV P-MLEM iteration becomes

$$\hat{f}_{n+1} = \frac{\hat{f}_n \cdot \left(\frac{g}{\hat{f}_n * \hat{h}_n}\right) * \hat{h}_n^s}{1 - \beta \cdot div\left(\frac{\nabla \hat{f}_n}{\|\nabla \hat{f}_n\|}\right)}$$

Our first experiment with the TV P-MLEM was deblurring of a noiseless image.

The first choice was $\beta = \mathbf{0.002}$ as suggested by Dey et. al. (Inria group).

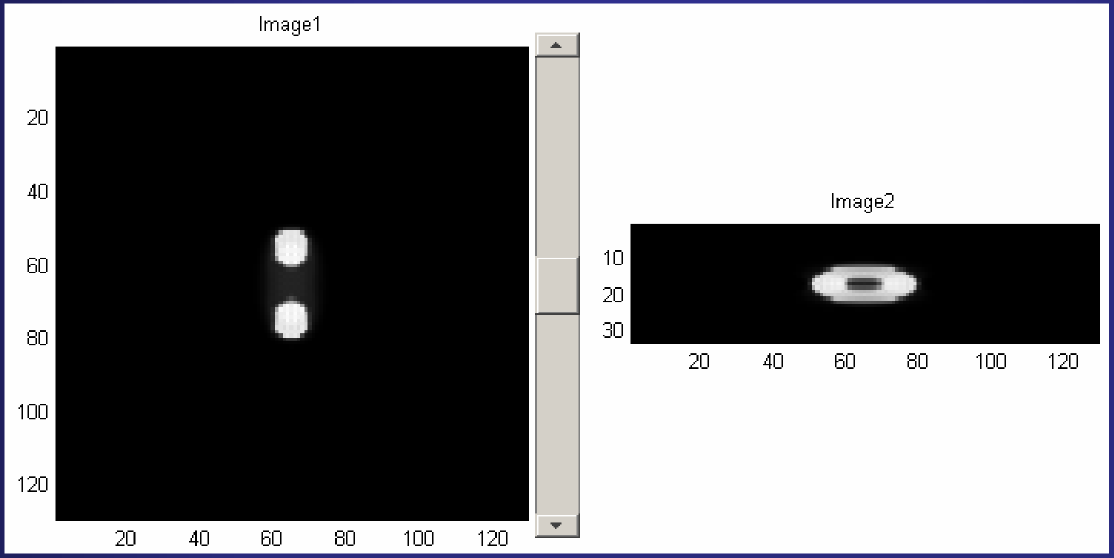


Deblurred object, using the TV P-MLEM and psf symmetry prior, $\beta = 0.002$.

Clearly, the oscilatory artifacts around the edges were eliminated, but the image  is still somewhat blurry.

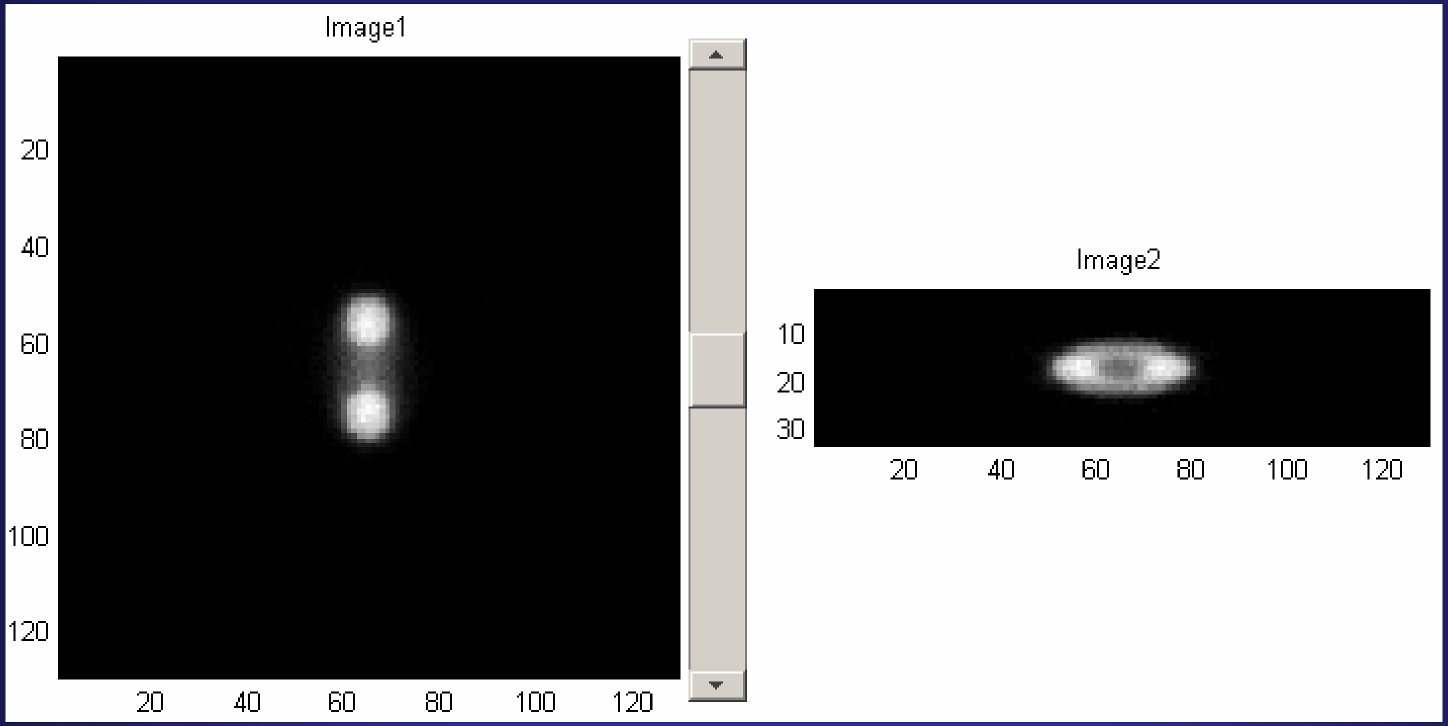
It seems like that the penalty term prevented the solution from converging.

To further investigate this effect we reduced the value of $\beta$



Deblurred object, using the TV P-MLEM and psf symmetry prior, $\beta = 0.000125$.

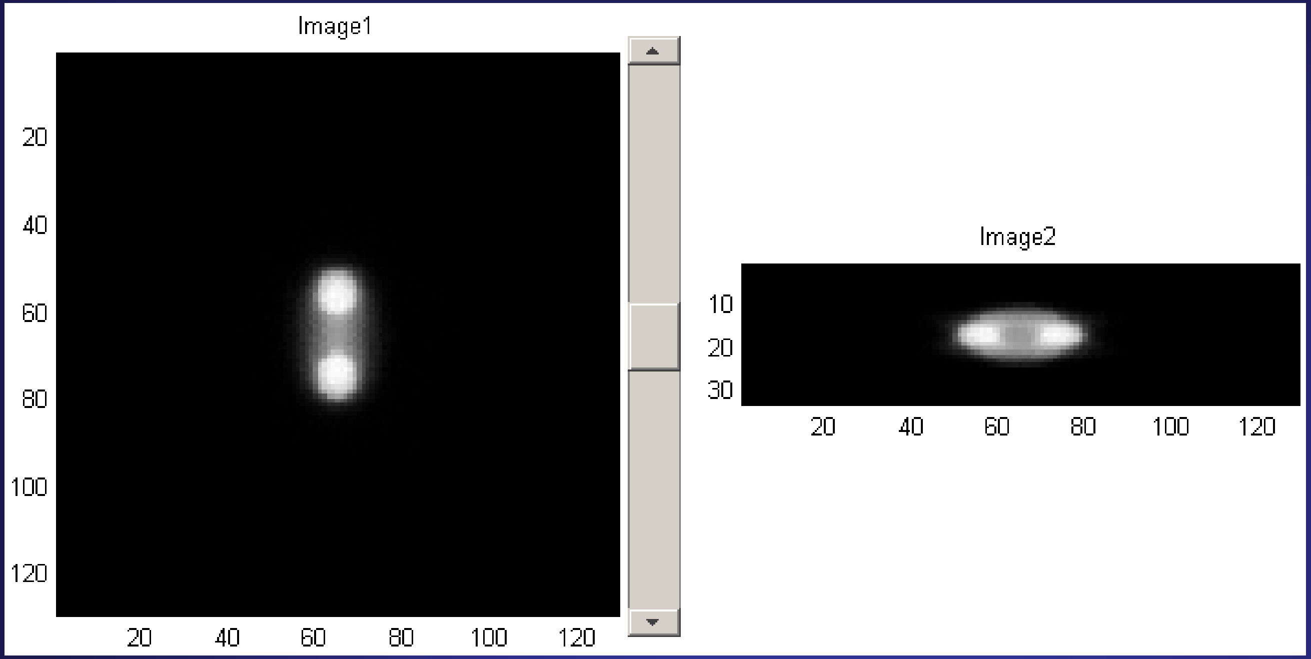
Encouraged by this result, the next step was trying the algorithm with a noisy image:



Deblurred noisy object, using the TV P-MLEM and psf symmetry prior, $\beta =$ 0.000125.

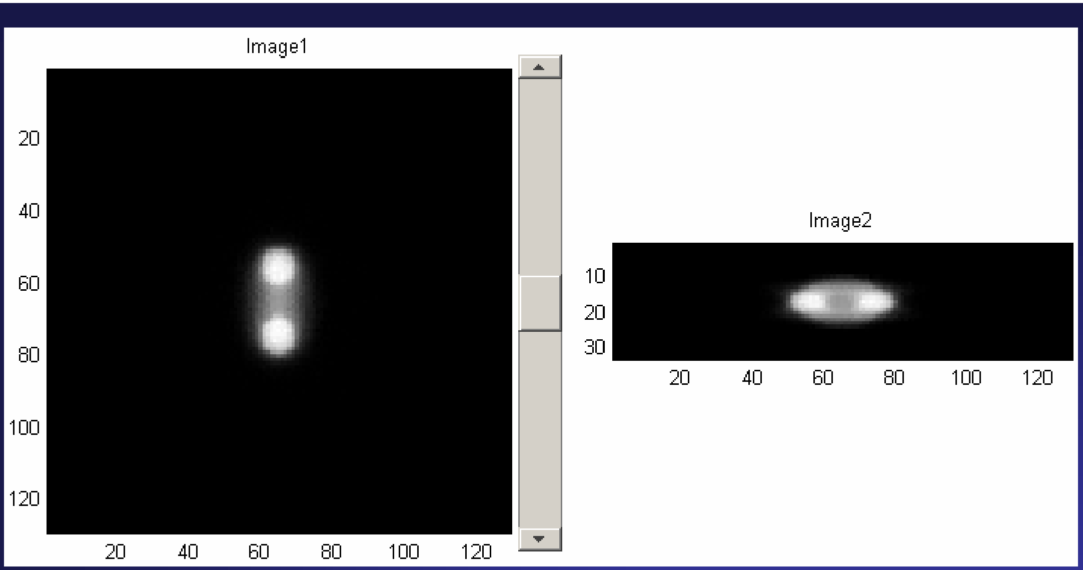
It seems that the penalty term is not effective in suppressing the Poisson noise.

Therefore, higher values of $\beta$ were tried, again.

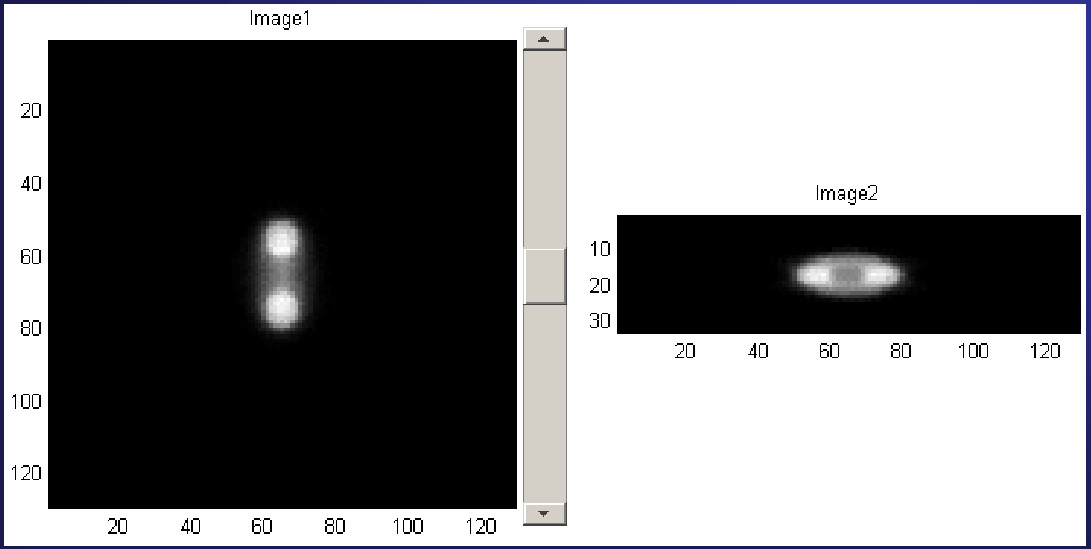


Deblurred noisy object, using the TV P-MLEM and psf symmetry prior, $\beta = 0.002$.

It seems that high $\beta$ values suppress the noise, but also prevent the reconstruction of sharp edges.



Deblurred noisy object, using the TV P-MLEM and psf symmetry prior, $\beta$ = 0.001.



Deblurred noisy object, using the TV P-MLEM and psf symmetry prior, $\beta$ = 5.625e-4.

It seems that high $\beta$ values suppress the noise, but also prevent the reconstruction of sharp edges.

If the value of $\beta$ is lower, the noise prevents the image estimate from converging.

This leads to:

# The Residual Denoised Penalized MLEM

Combine the TV-P-MLEM with the RD-MLEM to give us the residual denoised penalized MLEM (RDP-MLEM).

The overall iteration will then be:

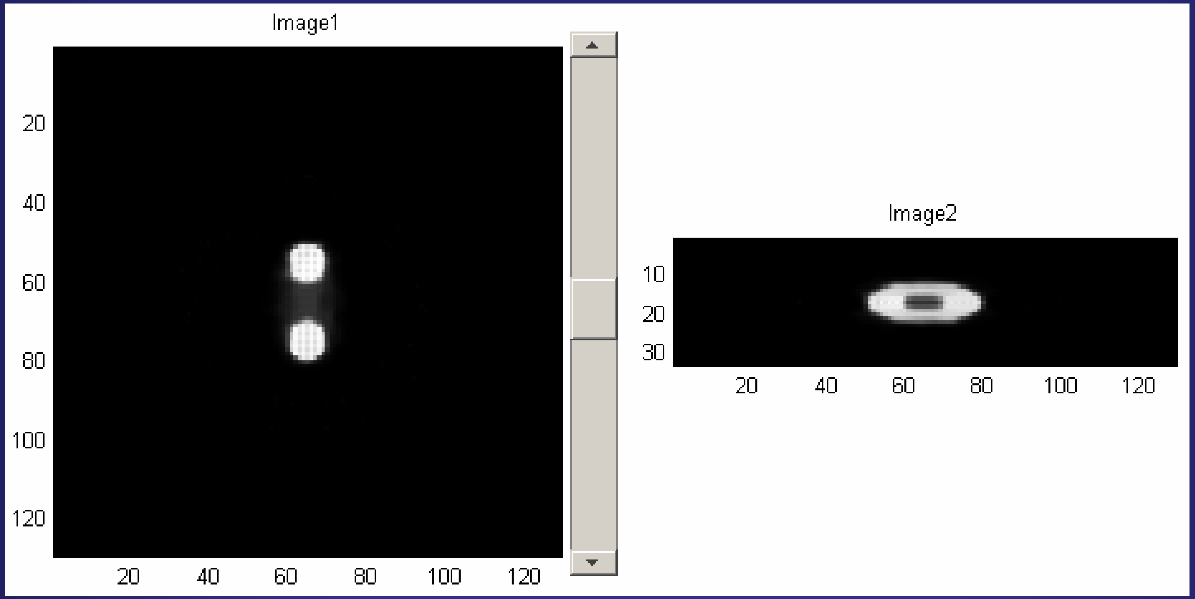
$$R_n = g - \hat{f}_n * \hat{h}_n$$

$$\bar{R}_n = Denoise(R_n)$$

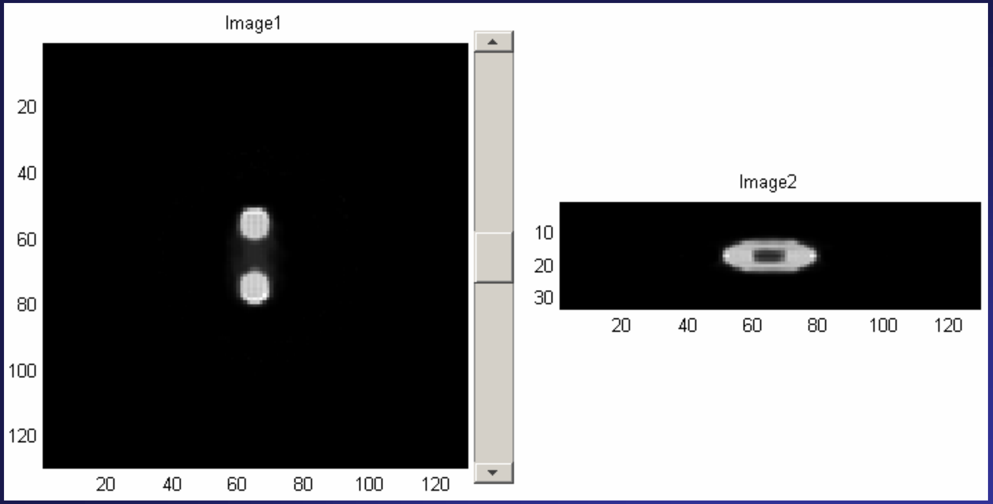
$$\hat{f}_{n+1} = \frac{\hat{f}_n \cdot \left( \frac{\hat{f}_n * \hat{h}_n + \bar{R}_n}{\hat{f}_n * \hat{h}_n} \right) * \hat{h}_n^s}{1 - \beta \cdot div \left( \frac{\text{"}\ \hat{f}_n}{\left\| \text{"}\ \hat{f}_n \right\|} \right)}$$

$$\hat{h}_{n+1} = \hat{h}_n \cdot \left( \frac{\hat{f}_n * \hat{h}_n + \bar{R}_n}{\hat{f}_n * \hat{h}_n} \right) * \hat{f}_n^s$$

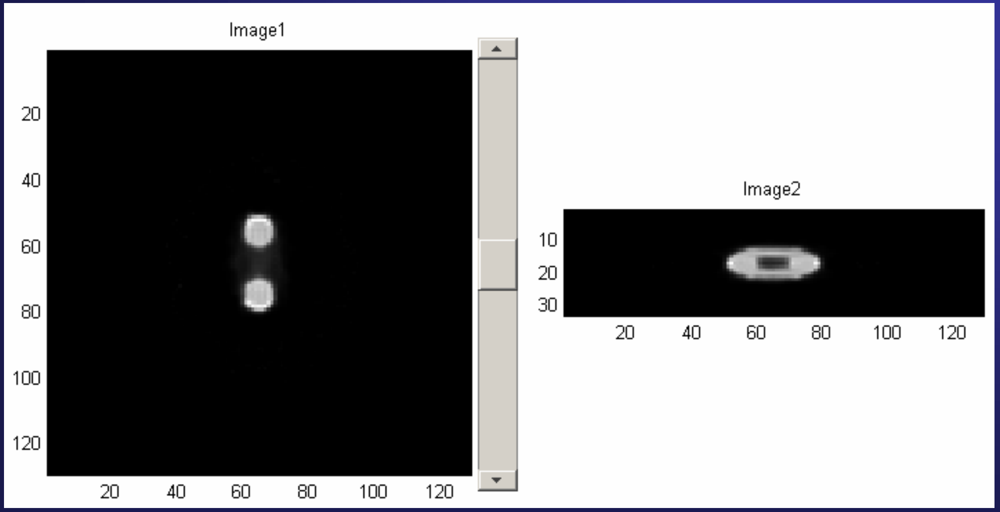
The RDP-MLEM algorithm has been tested with the 3x3x3 median filter as the *Denoise()* function and a few values for *β*. Results are brought in the following figures:



Deblurred noisy object, using the TV RDP-MLEM and psf symmetry prior, *β* = 0.000125.



Deblurred noisy object, using the TV RDP-MLEM and psf symmetry prior, $\beta$ = 6.25e-5.



Deblurred noisy object, using the TV RDP-MLEM and psf symmetry prior, $\beta$ = 4.6875e-5.

# Our next planned steps

- Turns out that the supports of both the psf and its Fourier transform depend on two parameters only: The numerical aperture (NA) and the refractive index of immersion oil.
  As both these parameters are available we can further experiment with the phase prior with hope of achieving improved performance.

- A more realistic model of the pdf is not symmetric in the Z-axis. We intend to modify our model to accommodate for this fact and repeat all the experiments we have seen.

- As a final test of the resulting blind deconvolution algorithm we intend to test in on real images from fluroescent microscopes.