Deconvolution algorithms for fluorescence and electron microscopy

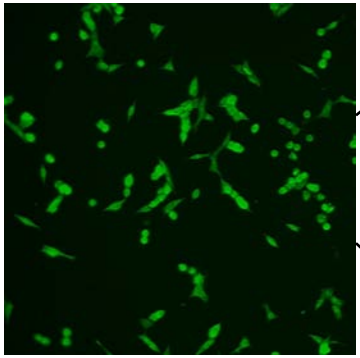
Siddharth Shah

Thesis Defense,
August 10, 2006,
Department of Biomedical Engineering

Outline

1. Motivation and Problem Statement
2. 2-D and 3-D Blind Deconvolution using Fourier Decoupling
3. 2-D Blind Deconvolution using the QUILL image model
4. Automatic Contrast Transfer Function estimation
5. Edge-preserving deconvolution of Electron Microscopy images
6. Conclusions
7. Future Work

A Real World Problem



Acquired light microscopic image with a wide field microscope after extensive preparation of Sample. Image of lower quality than needed.

Buy confocal microscope, Re-prepare slides and reaquire image

Hardware solution

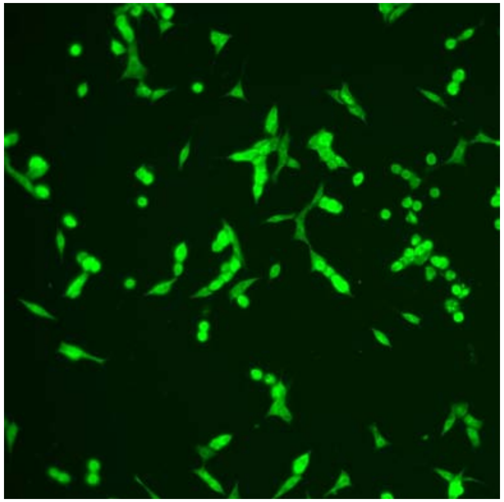
- Changing to confocal might not improve Image quality (esp thick samples.
- Preparation of slides might be very time consuming and expensive.
- Confocal microscopes can be very expensive.

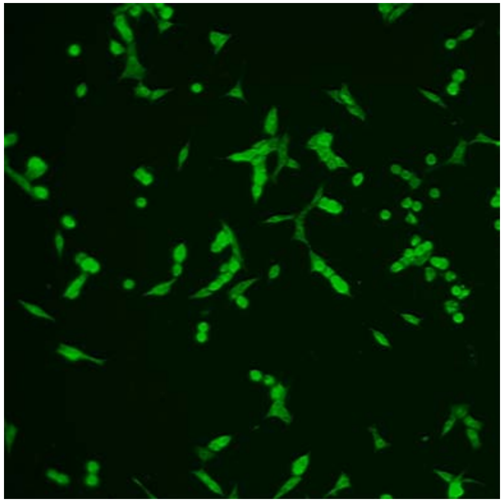
Acquire point Spread function (PSF) Plug into one of the many Deconvolution algorithms

Software solution

- PSF determination can be very time consuming.
- Deconvolution algorithms can be very slow and may require a lot of memory. Running them on large images may be intractable.
- Many deconvolution problems only work for certain types of images
- Many algorithms (esp. linear) can cause artifacts. This can be a big cause for concern for the biological researcher (Don't want to "discover" new structures)

The answer





A method that can restore images that is

- Cheap
- Fast
- Simple to use
- Robust i.e. is not sensitive to type or image or noise
- Does not require PSF estimation (hopefully)

The panacea?

Blind Deconvolution

“It’s not only impossible, it’s hopelessly impossible”

Julian C Christou

European Southern Observatory

Deconvolution: Problem Definition Compact Support Case

DATA PSF OBJECT NOISE

$$y(n_1,n_2) = h(n_1,n_2)^{**}\, u(n_1,n_2) + n(n_1,n_2)$$

where $u(n_1,n_2) = 0$ unless $0 \le n_1,n_2 \le M-1$

$h(n_1,n_2) = 0$ unless $0 \le n_1,n_2 \le L-1$

$y(n_1,n_2) = 0$ unless $0 \le n_1,n_2 \le N-1$ $\quad N=L+M-1$

$n(n_1,n_2)$ is zero mean 2-D white Gaussian noise

This is the linear deconvolution model. It is only valid for incoherent imaging systems.

PROBLEM

Given only data $y(n_1,n_2)$ reconstruct the object $u(n_1,n_2)$ and the PSF $h(n_1,n_2)$

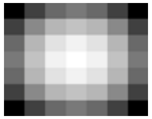
Compact Support

Support: Smallest rectangle encompassing area of interest.

Compact support: Support of the image is finite and within boundaries of convolved image.

MxM

$**$



LxL

$=$

M+L-1 x M+L-1

In other words, for the compact support deconvolution problem the entire object of interest is assumed to be contained in the image.

Partial Data Problem

No assurance that complete object is contained in the given image

DATA PSF OBJECT NOISE

$$y(n_1,n_2) = h(n_1,n_2) ** \, u(n_1,n_2) + n(n_1,n_2)$$

where $u(n_1,n_2) = 0$ unless $0 \leq n_1,n_2 \leq M-1$

$h(n_1,n_2) = 0$ unless $0 \leq n_1,n_2 \leq L-1$

$y(n_1,n_2) = 0$ unless $L-1 \leq n_1,n_2 \leq M-1$

PROBLEM

Reconstruct MxM portion of the image from (M-L+1)x (M-L+1) portion of data. (blurred image is smaller than the object)

Partial data problem is much harder than the compact support problem. As there is no unique solution even if PSF is known.

Blind Deconvolution of even point spread functions from compact support images

What we are going to show…

An algorithm that performs blind deconvolution that is

- Fast
- Parallelizable
- A linear algebraic formulation
- Non iterative (at least for the Least Squares solution)

Problem Formulation

DATA PSF OBJECT NOISE

$$y(n_1,n_2) = h(n_1,n_2)^{**}\, u(n_1,n_2) + n(n_1,n_2)$$

where $u(n_1,n_2) = 0$ unless $0 \le n_1,n_2 \le M\text{-}1$

$h(n_1,n_2) = 0$ unless $0 \le n_1,n_2 \le L\text{-}1$

$y(n_1,n_2) = 0$ unless $0 \le n_1,n_2 \le N\text{-}1$ $N=L+M-1$

n(n1,n2) is zero mean 2-D white Gaussian noise

CRITICAL ASSUMPTION

$$h(n_1,n_2) = h(L-n_1,L-n_2) \text{ (even PSF)}$$

PROBLEM

Given only data $y(n_1,n_2)$ reconstruct the object $u(n_1,n_2)$ and the PSF $h(n_1,n_2)$

Validity of Assumption

Many optical PSFs are symmetrical

EXAMPLE The theoretical microscopic PSFs are always symmetric. Hence, potential applications of this algorithm are in

- 2-D microscopy
- 3-D microscopy

Problem Ambiguities

Ambiguity	Solution
1. Scale Factor: If $\{h(i_1,i_2,i_3), u(i_1,i_2,i_3)\}$ is a solution, then $\{ch(i_1,i_2,i_3),(1/c)u(i_1,i_2,i_3)\}$ is also a solution (c is any real constant)	We consider the problem solved when the image is determined to a scale factor.
2. Translation: If $\{h(i_1,i_2,i_3), u(i_1,i_2,i_3)\}$ is a solution then so is $\{h(i_1+d_1,i_2+d_2,i_3+d_3), u(i_1-d_1,i_2-d_2,i_3-d_3)\}$	We specify supports for PSF and object.
3. Exchange: We need to distinguish $h(i_1,i_2,i_3)$ from $u(i_1,i_2,i_3)$.	We assume is not an even function unlike the PSF which is even. Or Assume $L \neq M$ (usually $L << M$)

Solution in 1D

$$y(n) = h(n)*u(n),\ h(n) \text{ is even}$$

Taking z transform

$$Y(z) = H(z)U(z) = z^{L} H(\frac{1}{z})U(z)$$

$$\boxed{Y(z)z^{M}U(\frac{1}{z}) = z^{N+1}Y(\frac{1}{z})U(z)}$$

Equating coefficients, we get the following matrix

$$\begin{bmatrix} y(0) & 0 & 0 & y^*(N-1) & 0 & 0 \\ y(1) & \ddots & \vdots & y^*(N-2) & \ddots & \vdots \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \vdots & \ddots & y(N-2) & \vdots & \ddots & y^*(1) \\ \vdots & 0 & y(N-1) & 0 & 0 & y^*(0) \end{bmatrix} \begin{bmatrix} u(M-1) \\ \vdots \\ u(0) \\ -u(0) \\ \vdots \\ -u(M-1) \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ 0 \end{bmatrix}$$

Toeplitz Structure

1D Example

Solve: {24,57,33}={h(0),h(0)}*{u(0),u(1)}

$$\begin{bmatrix} 33 & 0 & 24 & 0 \\ 57 & 33 & 57 & 24 \\ 24 & 57 & 33 & 57 \\ 0 & 24 & 0 & 33 \end{bmatrix} \begin{bmatrix} u(0) \\ u(1) \\ -u(1) \\ -u(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

u(0)=8, u(1)=11
Correct up to a scale factor

2D and 3D Solution

2D Solution

$$\boxed{Y(z_1, z_2)(z_1 z_2)^M U(\frac{1}{z_1}, \frac{1}{z_2}) = (z_1 z_2)^N Y(\frac{1}{z_1}, \frac{1}{z_2}) U(z_1, z_2)}$$

3D Solution

$$\boxed{Y(z_1, z_2, z_3)(z_1 z_2 z_3)^M U(\frac{1}{z_1}, \frac{1}{z_2}, \frac{1}{z_3}) = (z_1 z_2 z_3)^N Y(\frac{1}{z_1}, \frac{1}{z_2}, \frac{1}{z_3}) U(z_1, z_2, z_3)}$$

3D Solution

$$Y(z_1, z_2, z_3)(z_1 z_2 z_3)^M U\left(\frac{1}{z_1}, \frac{1}{z_2}, \frac{1}{z_3}\right) = (z_1 z_2 z_3)^N Y\left(\frac{1}{z_1}, \frac{1}{z_2}, \frac{1}{z_3}\right) U(z_1, z_2, z_3)$$

Equating coefficients we would get a doubly nested Toeplitz matrix

Matrix size: $(2M + L-2)^3$ X $(2M^3)$

Q: So we have solved the 3D problem ?
A: Not quite !! If M=17 and L=8 then the matrix size is 4913 X1024

It will be intractable to use this method “as is” in 3D !

Fourier Decoupling

Consider the 2-D case

$$\boxed{Y(z_1,z_2)(z_1z_2)^M U(\frac{1}{z_1},\frac{1}{z_2}) = (z_1z_2)^N Y(\frac{1}{z_1},\frac{1}{z_2})U(z_1,z_2)}$$

Setting $z_2 = y_k = e^{\frac{j2\pi k}{M}}$ and $z_1 = x_k = e^{\frac{j2\pi k}{M}}$ we get

$$Y(z_1,y_k)y_k{}^M z_1{}^M U*(\frac{1}{z_1^*},y_k) = y_k{}^N z_1{}^N Y^*(\frac{1}{z_1^*},y_k)U(z_1,y_k)$$

and

$$Y(x_{k,}z_2)x_k{}^M z_2{}^M U*(x_k,\frac{1}{z_2^*}) = x_k{}^N z_2{}^N Y^*(x_k,\frac{1}{z_2^*})U(x_k,z_2)$$

The point ?

The last equation two equations are decoupled into a set of M 1D problems !

How about 3-D ?

Let $x_k = e^{j2\pi k/M}, y_k = e^{j2\pi k/M}, z_k = e^{j2\pi k/M}$

Then

$$Y(z_1, y_k, z_3)(z_1 y_k z_3)^M U(\frac{1}{z_1}, \frac{1}{y_k}, \frac{1}{z_3}) = U(z_1, y_k, z_3)(z_1 y_k z_3)^N Y(\frac{1}{z_1}, \frac{1}{y_k}, \frac{1}{z_3})$$

Using conjugate symmetry $U(\frac{1}{z_1}, \frac{1}{y_k}, \frac{1}{z_3}) = U*(\frac{1}{z_1 *}, \frac{1}{y_k *}, \frac{1}{z_3 *}) = U*(\frac{1}{z_1 *}, y_k, \frac{1}{z_3 *})$

$$Y(z_1, y_k, z_3)(z_1 y_k z_3)^M U*(\frac{1}{z_1 *}, y_k, \frac{1}{z_3 *}) = U(z_1, y_k, z_3)(z_1 y_k z_3)^N Y^*(\frac{1}{z_1 *}, y_k, \frac{1}{z_3 *})$$

The last equation is decoupled into a set of M 2-D problems. We can decouple each of these 2-D problems to M 1-D problems.

Simplifying life 1-D at a time

So we broke down a huge 3D problem to M simpler 2D problems

What next ?

Substitute for x_k in each 2D problem and you would get M 1D problems in z

$$\boxed{Y(x_k, y_k, z_3)(x_k y_k z_3)^M U*(x_k, y_k, \frac{1}{z_3*}) = U(x_k, y_k, z_3)(x_k y_k z_3)^N Y^*(x_k, y_k, \frac{1}{z_3*})}$$

To summarize

We broke up a large 3D problem into M^2 1D problems !

But what about the scale factors ?

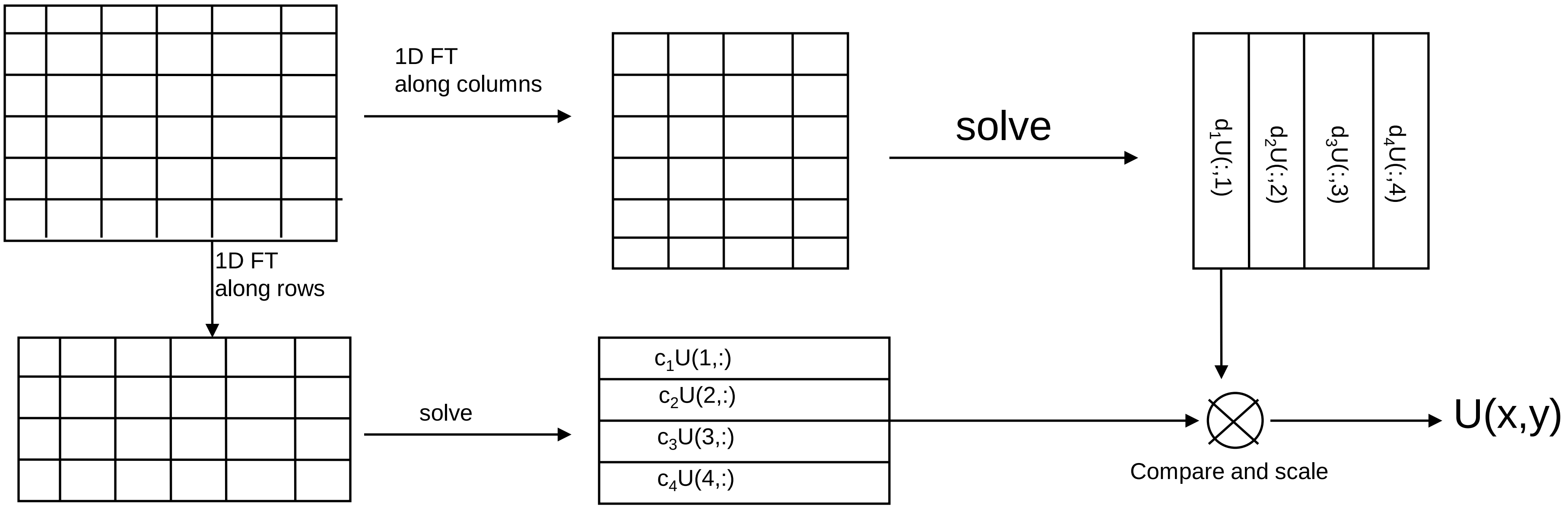
Weighing the scales… (2D)

Note that each 1D problem will be correct upto a scale factor.

Decoupling 2D to 1D → Each row is solved to a scaled factor.

How do we get the whole 2D solution correctly ?

Solve along columns and compare coefficients



1D FT along columns

solve

$d_1U(:,1)$

$d_2U(:,2)$

$d_3U(:,3)$

$d_4U(:,4)$

1D FT along rows

solve

$c_1U(1,:)$

$c_2U(2,:)$

$c_3U(3,:)$

$c_4U(4,:)$

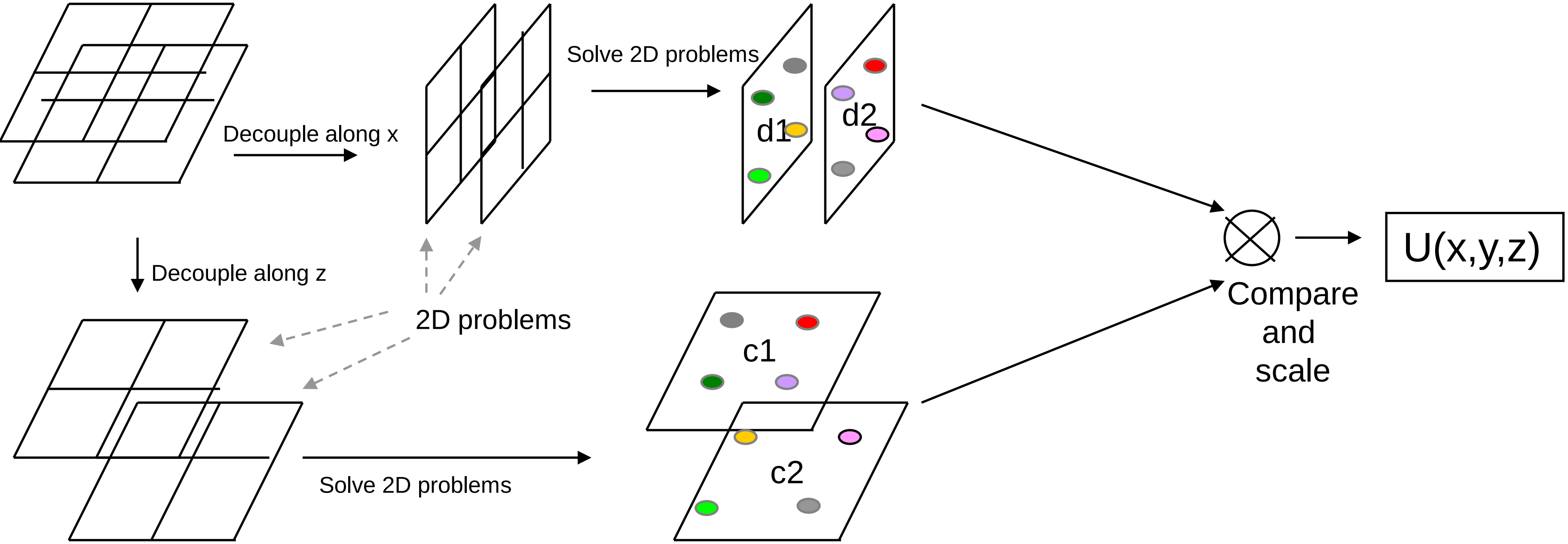
Compare and scale

U(x,y)

Weighing the scales… (3D)

We just learned how a 2D problem could be correctly scaled

Solve 3D → Decouple 3D to 2D → Decouple 2D to 1D → Solve 1D → Scale 1D Sol^{ns} → 2D solutions → Scale 2D Sol^{ns} → Solve 3D



Fourier Decoupling Algorithm

Algorithm (3-D)

1. Take FFT along either rows, columns or depth
2. Now we have multiple problems in 2D (M problems in 2D)
3. For each 2D problem we take FFT along Rows or columns
4. For each 2D problem we get multiple 1D problem. (M^2 problems in 1-D)
5. So solve 1D Problem and then scale the 2-D problem.
6. Scale the 3-D problem.

What about noise ?

In presence of noise the nullspace of the toeplitz structure no longer exists.

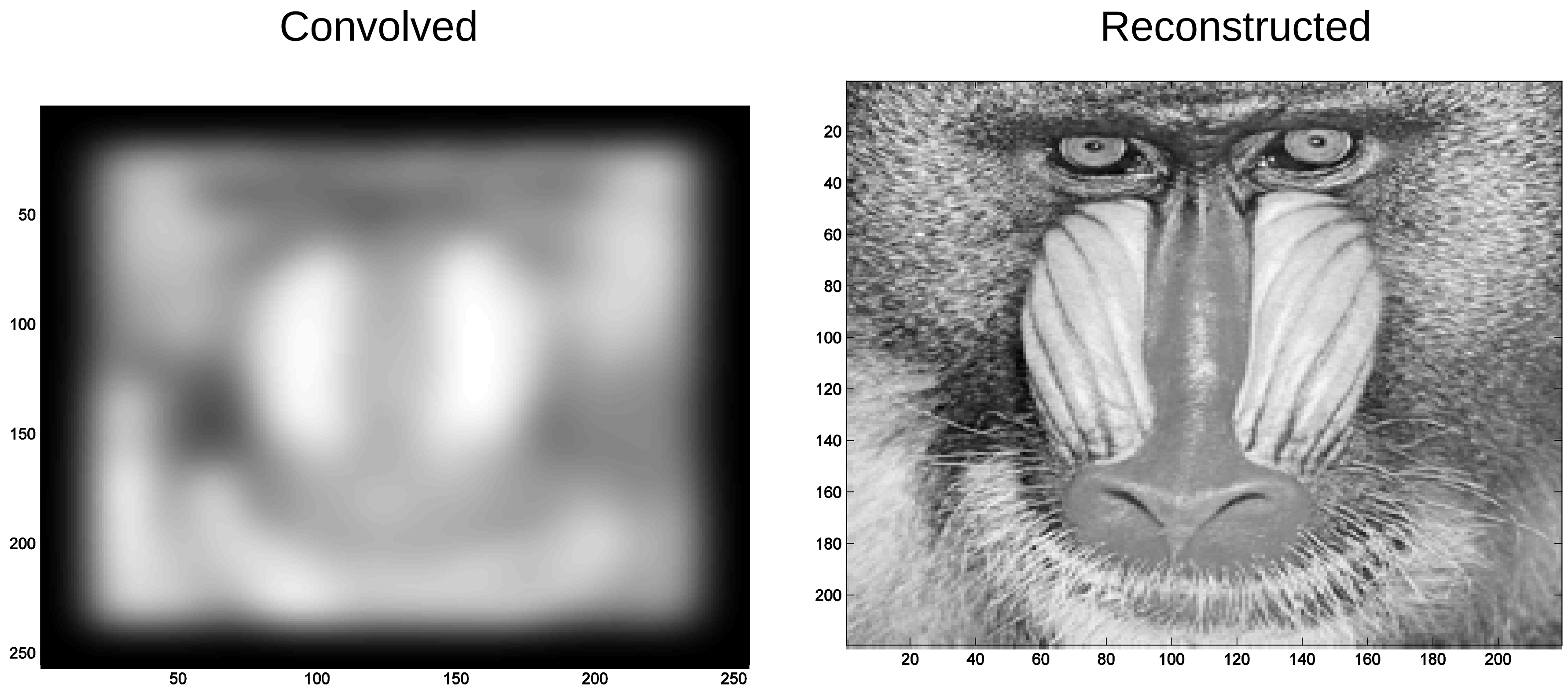
We can find “nearest” nullspace using Least Squares (LS) (fast)

Can use structure of matrix to solve by structured least squares (STLS)
(slow but more accurate)

We can show that such norm minimization will give us the Maximum Likelihood Estimate of the object $u(n_1,n_2)$ for white Gaussian noise random field.

Simulations

2-D: No Noise



Convolved

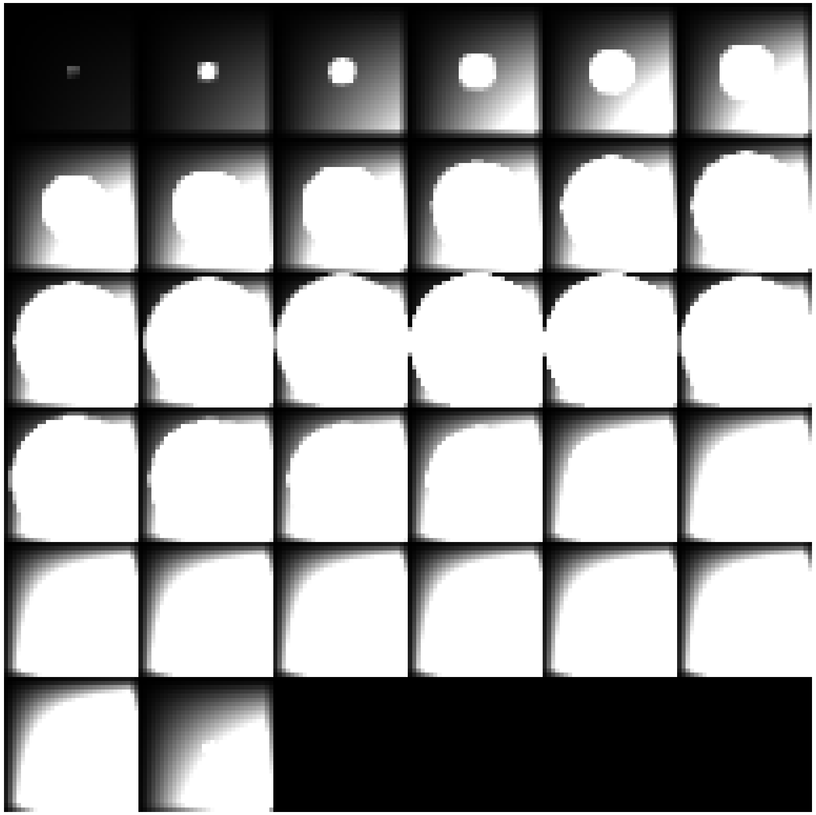
Reconstructed

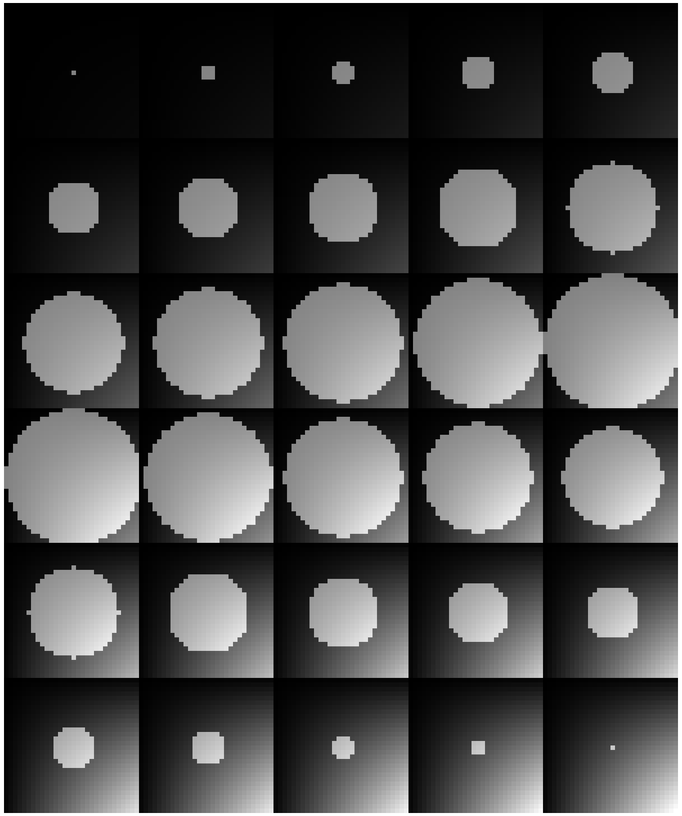
Image 221X221. PSF 35X35

Simulations

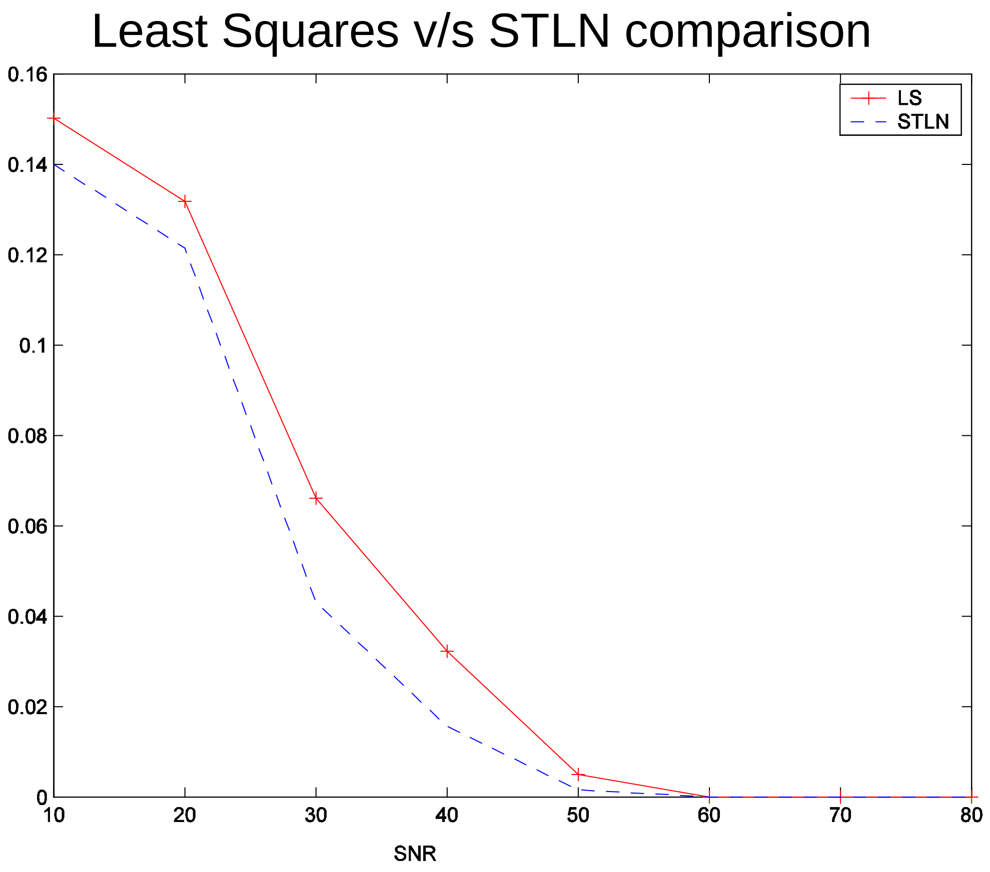
3-D: No Noise

Synthetic bead image (3X3X3), (3X3X3) PSF, no noise





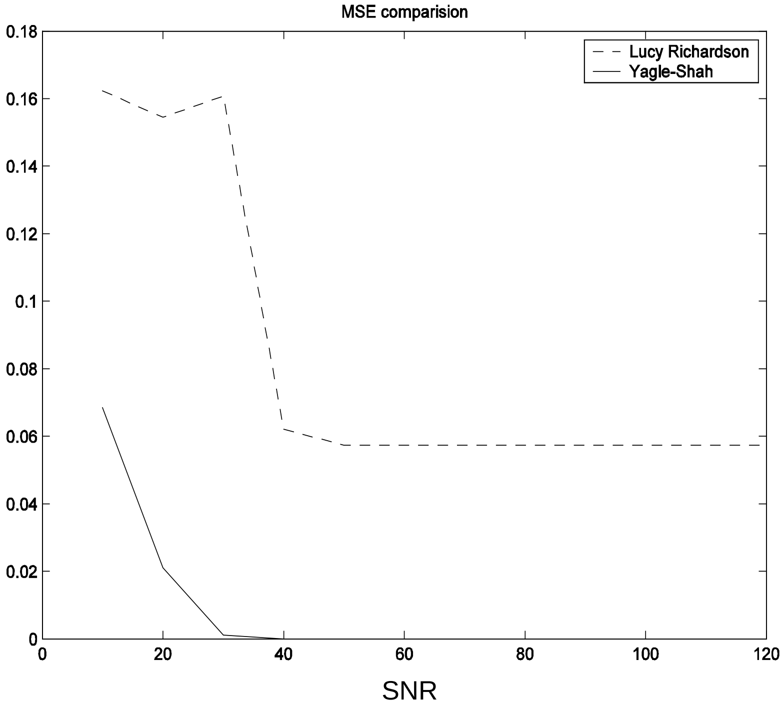
Simulations: With noise



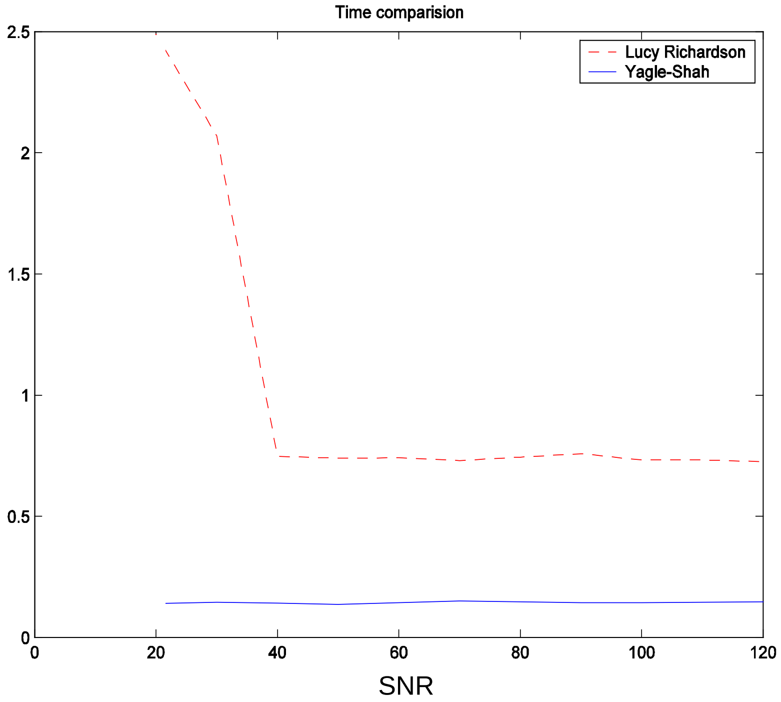
7X7X7 image. 3X3X3 PSF.
50 iterations per SNR

Least squares does well at high SNRs but at low and medium SNRS STLN is better.

Comparison with Lucy Richardson



MSE



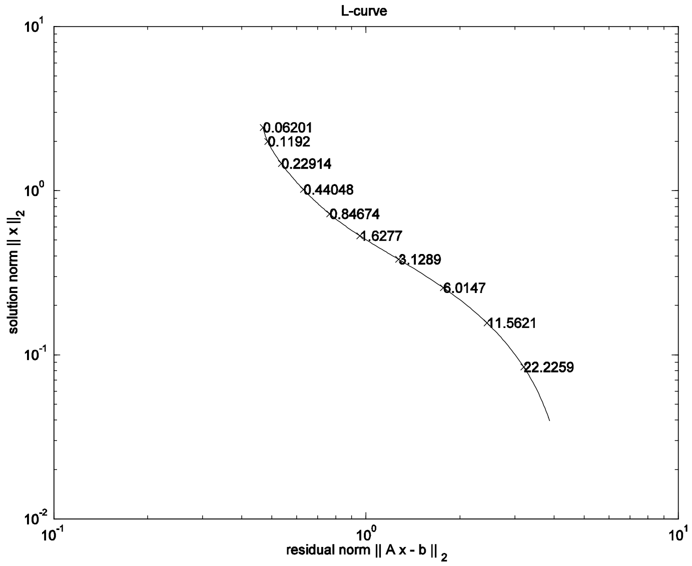
Time

Our algorithm gave a lower MSE. In LR Final accuracy even in absence of noise depends on initial PSF estimate.

Our algorithm need only a fixed amount of time to solve independent of the SNR. LR needs more time and time to solve depends on the SNR.

Attempts at Regularization

- With real data algorithm is rather sensitive to noise.
- For 3-D data with lots of black spaces, the algorithm found multiple null-spaces.
- Attempts at regularization were marginally successful (Tikhonov)



Finding Lambda using L-Curve method

L-curve of one of the Fourier decomposed Problems is shown here

Note that the L-Curve is hardly a L here !

Regularization is hard as there is no optimal point (at least using the least squares approach)

Where we stand

- Algorithm successfully deconvolves 2-D and 3-D simulated data in both the presence and absence of noise.
- Performance better than the Lucy-Richardson algorithm
- Problems with “planes of zeros” for 3-D data and “lines of zeros” for 2-D data. Presently overcome with adding low amounts of white Gaussian noise

Publications

- Fast 3-D Blind Deconvolution of even point spread functions, Yagle AE and Shah S, *presented Photonics West 2004, San Jose, CA*

QUILL Model based deconvolution

The Idea

By representing an image as a subsampled version itself convolved with a "basis" kernel we can reformulate the problem as a single input multiple output type of problem. This problem is easily solvable using Bezout's Lemma.

1D Example

Consider the following sequence

$$x(n)=\{x(0),0,x(2),0,x(4),0,\ldots\}*\{1/2,1,1/2\}=\{x(0),\frac{x(0)+x(2)}{2},x(2),\frac{x(2)+x(4)}{2},\ldots\}$$

Linear spline

Downsampled data

Note: We can change the spline to any other one. However, the linear spline gave us The best empirical results.

QUILL Model: 1D

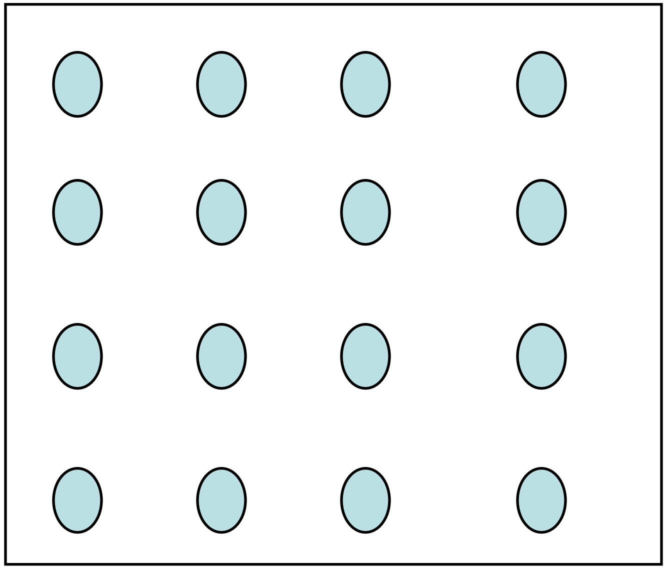
If x(n) is a 1D signal then

$$x(n) = \Sigma x(2i)\phi(n-2i) = \tilde{x}(n) * \phi(n)$$

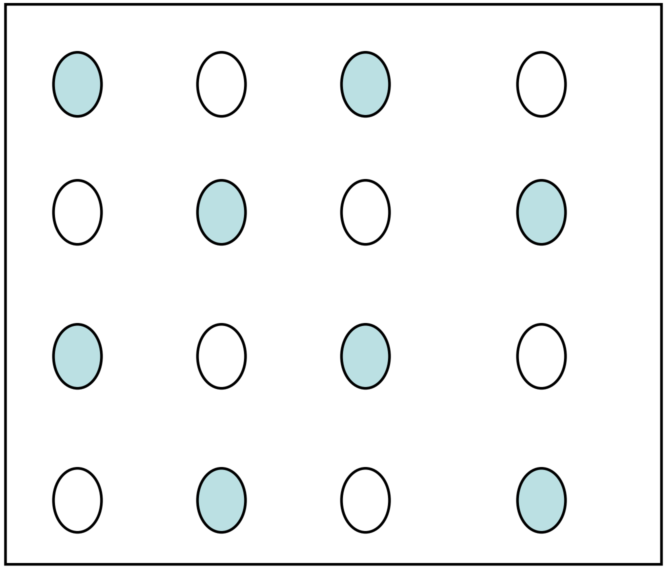
Where
Φ(n) is a basis of some kind and
$\tilde{x}(n)$ Is obtained by setting x(n)=0 for odd n.

Quincunx Sampling

$$\tilde{x}(n_1,n_2)=\left\{\begin{matrix} x(n_1,n_2) \\ 0 \end{matrix}\right\}\begin{matrix} \text{if } \mathrm{n}_1+\mathrm{n}_2 \text{ is even} \\ \text{otherwise} \end{matrix}$$



16 Pixel Image



Quincunx Version

QUILL model: 2D

If $x(n_1,n_2)$ is a 2D image, we do the following:

1. Take its Quincunx sampled version. Call this $\tilde{x}(n_1,n_2)$

2. 2X2 Upsample this image along lines inclined at 45 and 135 degrees. We call this $\tilde{x}2(n_1,n_2)$

3. We then convolve the upsampled version of the image with 2D linear spline basis function so that $\tilde{x}2(n_1,n_2)$ is now Interpolated Linearly

QUILL = Quincunx Upsampled Interpolated Linearly

$$\boxed{x_{QUILL}(n_1,n_2) = \tilde{x}2(n_1,n_2) ** \phi 1(n_1,n_2) ** \phi 2(n_1,n_2)}$$

Quincunx

Upsampler

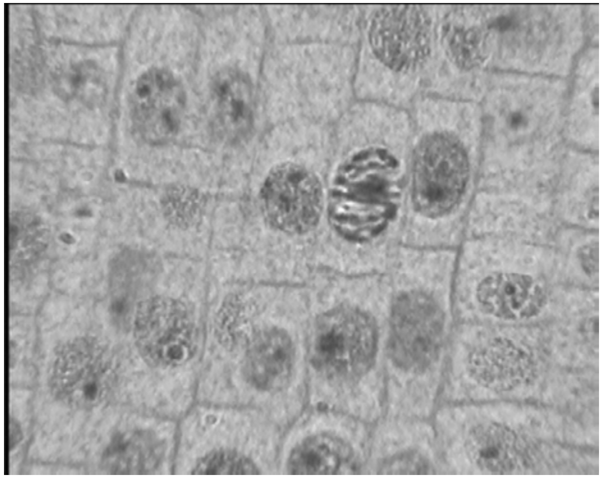
Linear Interpolator

QUILL in Action
The Good

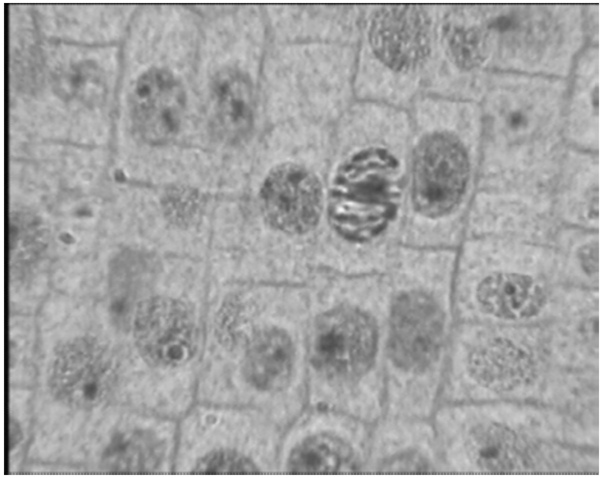
Before

After

Excellent model fit.
It is a highly sampled Image (1024 X 1024) so not much is lost during modelling.



Light microscopy image of onion cell. Very good fit as there is very little high frequency info.



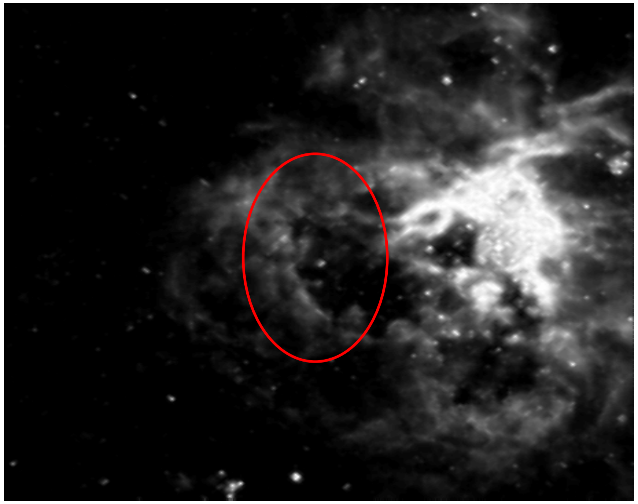
QUILL in Action
The Bad

Before

$x(n_1, n_2)$

Poor representation of Point sized objects. This Is because point sized Objects are high Frequency information.

After

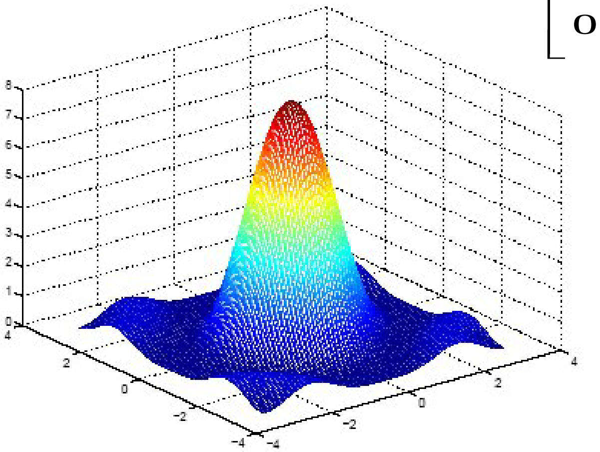


1. QUILL is a good algorithm of choice for large highly sampled images there is little high frequency information. (2D and 3D microscopy. Some MRI intensity images)
2. QUILL is a bad algorithm for images that are small and/or have point sized objects. This means that it would be bad for astronomical and ultrasound applications (speckle)

Implicit assumptions in QUILL (Maybe not so implicit!!)

$$\boxed{x_{QUILL}(n_1,n_2) = \tilde{x}2(n_1,n_2) ** \phi 1(n_1,n_2) ** \phi 2(n_1,n_2)}$$

$$\phi(n_1,n_2) = \begin{bmatrix} 0 & 0 & \frac{1}{4} & 0 & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ \frac{1}{4} & 0 & 1 & 0 & \frac{1}{4} \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{4} & 0 & 0 \end{bmatrix} ** \begin{bmatrix} 0 & \frac{1}{4} & 0 \\ \frac{1}{4} & 1 & \frac{1}{4} \\ 0 & \frac{1}{4} & 0 \end{bmatrix}$$



The Fourier Spectrum of Φ

Φ is fixed. So if we know x2 we know the image

A Link: QUILL as SIMO

$$
\begin{aligned}
y(n_1,n_2) &= h(n_1,n_2) * * x(n_1,n_2) \\
&= h(n_1,n_2) * * (\phi(n_1,n_2) * * \tilde{x}2(n_1,n_2)) \\
&= \tilde{h}(n_1,n_2) * * \tilde{x}2(n_1,n_2)
\end{aligned}
$$

If we now use the following notation

$$
\begin{aligned}
y^1(n_1,n_2) &= y(2n_1,2n_2) \\
y^2(n_1,n_2) &= y(2n_1+1,2n_2) \\
y^3(n_1,n_2) &= y(2n_1,2n_2+1) \\
y^4(n_1,n_2) &= y(2n_1+1,2n_2+1) \\
\tilde{h}^1(n_1,n_2) &= \tilde{h}(2n_1,2n_2) \\
\tilde{h}^2(n_1,n_2) &= \tilde{h}(2n_1+1,2n_2) \\
\tilde{h}^3(n_1,n_2) &= \tilde{h}(2n_1,2n_2+1) \\
\tilde{h}^4(n_1,n_2) &= \tilde{h}(2n_1+1,2n_2+1)
\end{aligned}
$$

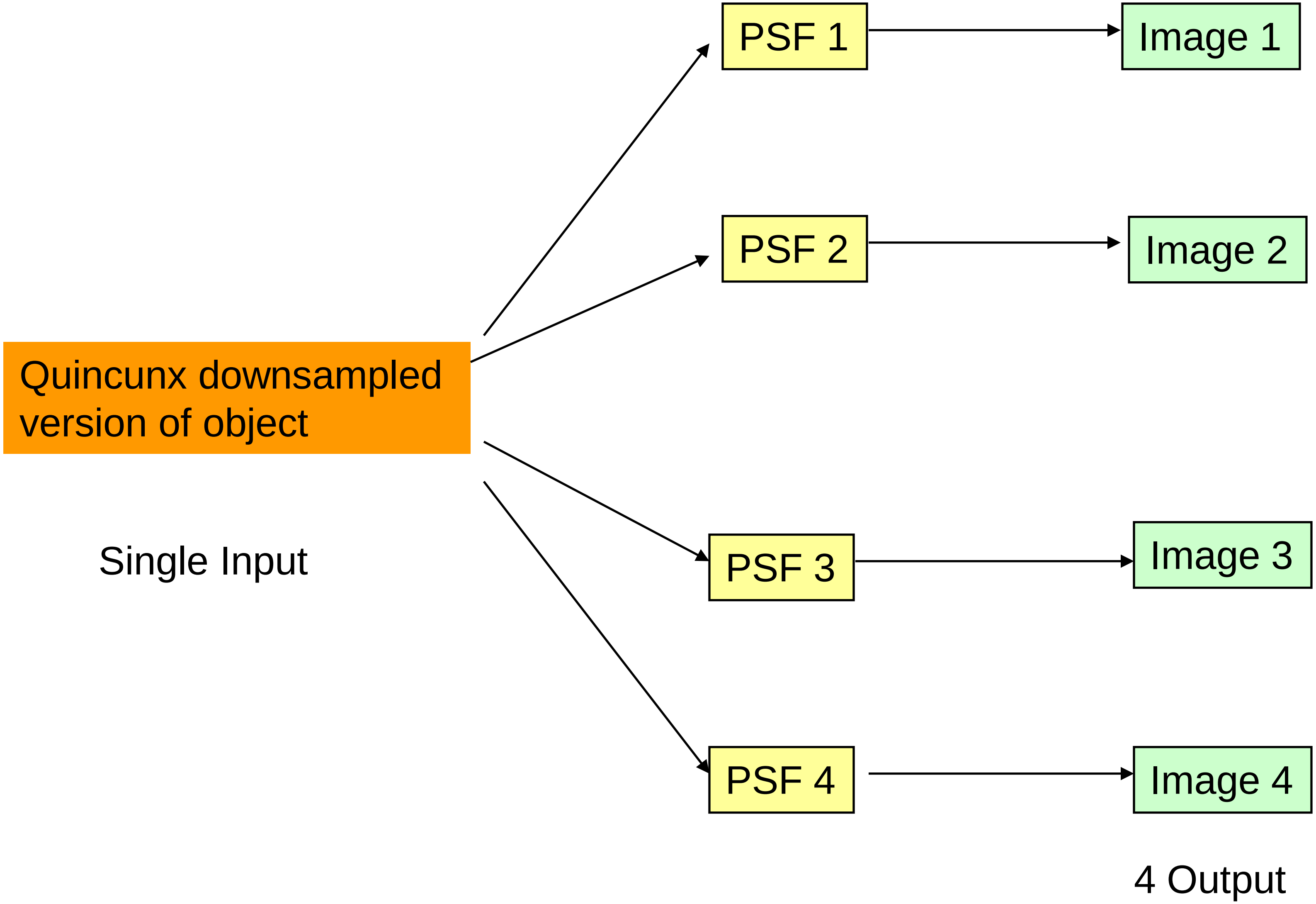
then we have the following relationships

Downsampled versions
Of PSF

$$
\begin{aligned}
y^1(n_1,n_2) &= \tilde{h}^1(n_1,n_2) * * \tilde{x}(n_1,n_2) \\
y^2(n_1,n_2) &= \tilde{h}^2(n_1,n_2) * * \tilde{x}(n_1,n_2) \\
y^3(n_1,n_2) &= \tilde{h}^3(n_1,n_2) * * \tilde{x}(n_1,n_2) \\
y^4(n_1,n_2) &= \tilde{h}^4(n_1,n_2) * * \tilde{x}(n_1,n_2)
\end{aligned}
$$

Quincunx
sampled

A Link: QUILL as SIMO



Quincunx downsampled version of object

PSF 1

PSF 2

PSF 3

PSF 4

Image 1

Image 2

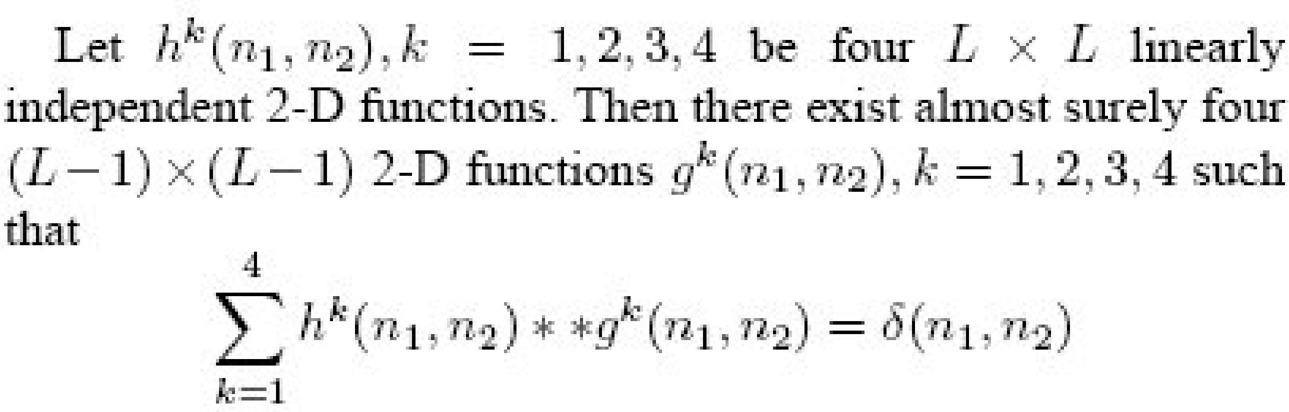
Image 3

Image 4

Single Input

4 Output

Critical Result from SIMO
The 2D Bezout's Lemma



Let $h^k(n_1, n_2), k = 1, 2, 3, 4$ be four $L \times L$ linearly independent 2-D functions. Then there exist almost surely four $(L-1) \times (L-1)$ 2-D functions $g^k(n_1, n_2), k = 1, 2, 3, 4$ such that

$$\sum_{k=1}^{4} h^k(n_1, n_2) * *g^k(n_1, n_2) = \delta(n_1, n_2)$$

This is NOT akin to finding an inverse filter

Solving the QUILL deconvolution problem

We can show easily that

$$\sum_{k=1}^{4} y^k(n_1, n_2) **g^k(n_1, n_2) = \tilde{x}(n_1, n_2)$$

Example: Consider the following problem

$$\begin{bmatrix} 1 & 2 & -2 & -4 \\ -1 & -1 & 5 & 1 \\ 4 & -2 & -2 & 6 \\ 3 & -1 & -8 & 0 \end{bmatrix} ** \begin{bmatrix} 1 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 & 0 & 5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 6 & 0 & 0 & 0 & 7 & 0 & 0 & 0 & 8 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 2 & -2 & -4 & 2 & 4 & -4 & -8 & 3 & 6 & -6 & -12 \\ -1 & -1 & 5 & 1 & -2 & -2 & 10 & 2 & -3 & -3 & 15 & 3 \\ 4 & -2 & 2 & 14 & 0 & -20 & 1 & 22 & 2 & -26 & -6 & 18 \\ 3 & -1 & -12 & -4 & 26 & 2 & -21 & -5 & 34 & 2 & -24 & 0 \\ 6 & 12 & 4 & -32 & -1 & 38 & 6 & -38 & -2 & 46 & -16 & -32 \\ -6 & -6 & 42 & 2 & -39 & -7 & 50 & 2 & -48 & -8 & 40 & 8 \\ 24 & -12 & -12 & 36 & 28 & -14 & -14 & 42 & 32 & -16 & -16 & 48 \\ 18 & -6 & -48 & 0 & 21 & -7 & -56 & 0 & 24 & -8 & -64 & 0 \end{bmatrix}$$

QUILL Deconvolution at work

Solution Steps

1. We know size of the deconvolver (L-1) That helps us set the y matrix.
2. We know where the zeros are in the QUILL version of the image.
3. So we can extract those rows and form a matrix with the y values on the LHS and the zeros on the RHS.
4. Null space of this extracted matrix gives us the deconvolver (g).
5. Once you have g , you can find the QUILL image

$$\sum_{k=1}^{4} y^k(n_1,n_2) ** g^k(n_1,n_2) = \tilde{x}(n_1,n_2)$$

$$\begin{bmatrix} 1 & 2 & -1 & -1 \\ 4 & -2 & 3 & -1 \\ 6 & 12 & -6 & -6 \\ 24 & -12 & 18 & -6 \\ -2 & -4 & 5 & 1 \\ 2 & 14 & -12 & -4 \\ 4 & -32 & 42 & 2 \\ \boxed{-12} & \boxed{36} & \boxed{-48} & \boxed{0} \\ 2 & 4 & -2 & -2 \\ 0 & -20 & 26 & 2 \\ -1 & 38 & -39 & -7 \\ 28 & -14 & 21 & -7 \\ -4 & -8 & 10 & 2 \\ 1 & 22 & -21 & -5 \\ 6 & -38 & 50 & 2 \\ -14 & 42 & -56 & 0 \\ 3 & 6 & -3 & -3 \\ 2 & -26 & 34 & 2 \\ -2 & 46 & -48 & -8 \\ 32 & -16 & 24 & -8 \\ -6 & -12 & 15 & 3 \\ -6 & 18 & -24 & 0 \\ -16 & -32 & 40 & 8 \\ -16 & 48 & -64 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 6 \\ 0 \\ 0 \\ 4 \\ 0 \\ \boxed{0} \\ 2 \\ 0 \\ 7 \\ 0 \\ 0 \\ 5 \\ 0 \\ 0 \\ 3 \\ 0 \\ 8 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\boxed{x_{QUILL}(n_1,n_2) = \tilde{x}2(n_1,n_2) ** \phi 1(n_1,n_2) ** \phi 2(n_1,n_2)}$$

Why is QUILL so fast ?

1. The degee of computational complexity for the nullspace problem depends on the width of the matrix columns. Here it is 4*(L-1)^2. It only depends on L. Typical PSFs are small (<14X14)

2. The complexity “does not care” if the image is large or small. For a larger image there will be more multiplication operations in the end to find the unknown non zero points. Multiplication operations are very fast

3. Because of the QUILL model we are only finding the values of a fraction of the points of the original image (12.5% of the points)

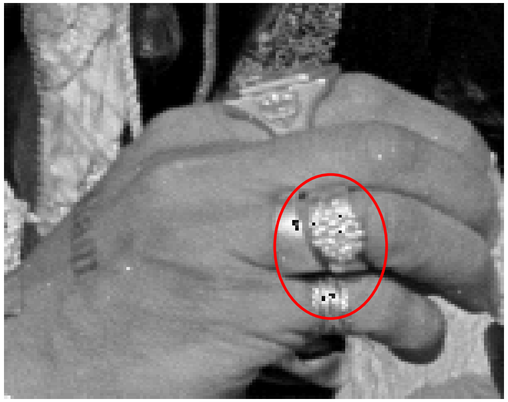
4. Memory requirements are not very high. A fraction of y values only need to be arranged in the Toeplitz-Block-Toeplitz form and the nullspace is easily found.

5. Many fast nullspace finding algorithms are available.

6. Last and most important, the algorithm is non-iterative.

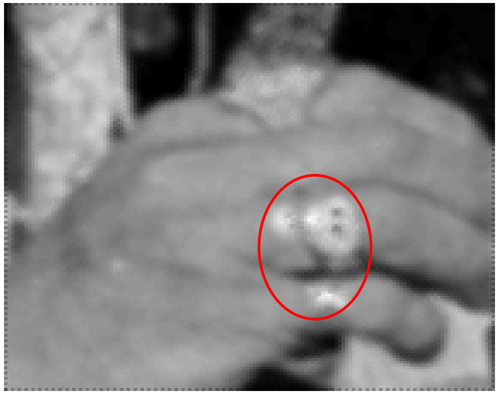
Simulations

No noise deconvolution



Original

Convolved with Gaussian shaped PSF



Restored

- Deconvolution while successful did not give us the original image → Modelling error

- Restored large areas well but point features were not properly deconvolved.

Regularization: The fidelity noise tradeoff

Consider the least squares solution to the problem: Ax=b

$$x = V \mathrm{diag}\left[\frac{1}{s_i}\right] U^T b \qquad \text{where} \qquad A = USV^T \qquad \text{(SVD)}$$

- Some matrices have a very small s_i
- Hence the small s_i*'s dominate the solution.*

This is bad news when there is noise in the image, since small singular values are typically dominated by noise.

What can be done ?

You regularize the solution

Regularization: The fidelity noise tradeoff

What can be done ?

Get rid of the small Singular values

$$x = Vdiag\left[\frac{1}{s_i}\right]U^T b$$

Only some s_i's are chosen

Truncated Singular Value Decomposition (TSVD)

Formulate a fidelity to solution to noise tradeoff equation and solve it

$$\min\left\|\begin{bmatrix}A\\ \alpha I\end{bmatrix}x-\begin{bmatrix}b\\ o\end{bmatrix}\right\|_2^2$$

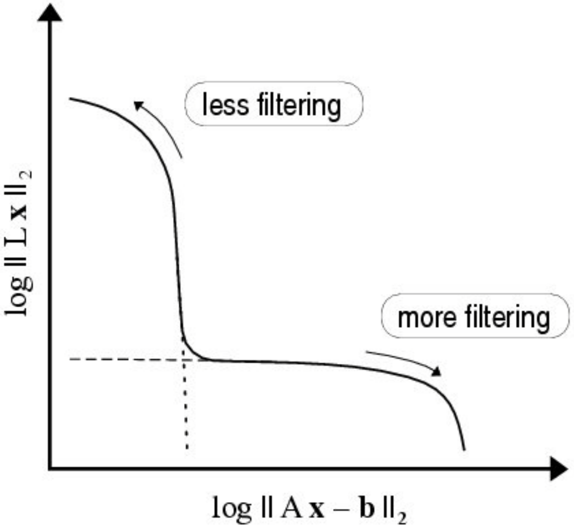
Filter factors

$$x = Vdiag\left[\frac{s_i^2}{(s_i^2+\alpha^2)s_i}\right]U^T b$$

When s_i's are small compared to α they are suppressed. When s_i's are large, the s_i terms dominate.

The elusive Mr α

The most obvious question for Tikhonov regularization : How do you find α ?



- If you solve Tikhonov for different α and plot Log ||Lx|| v/s log ||Ax-b|| you get the adjacent curve

- We want the alpha that gives us the corner point: The lowest product of Log ||Lx|| and log ||Ax-b||

Problems

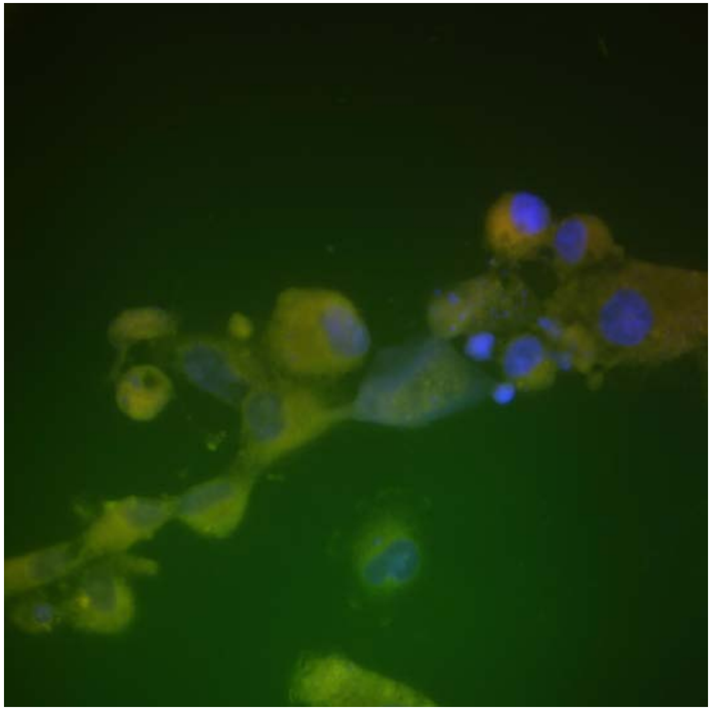
1. Sometimes fails in practice. Finding α is still a black art needing luck and patience.
2. Tikhonov over regularizes many times. Thus, while smoothing the image the method overall over blurs the edges. What we need is something that leaves the edges sharp yet smooths the noisy gradient areas.

Anisotropic Diffusion

Edge Preserving regularization

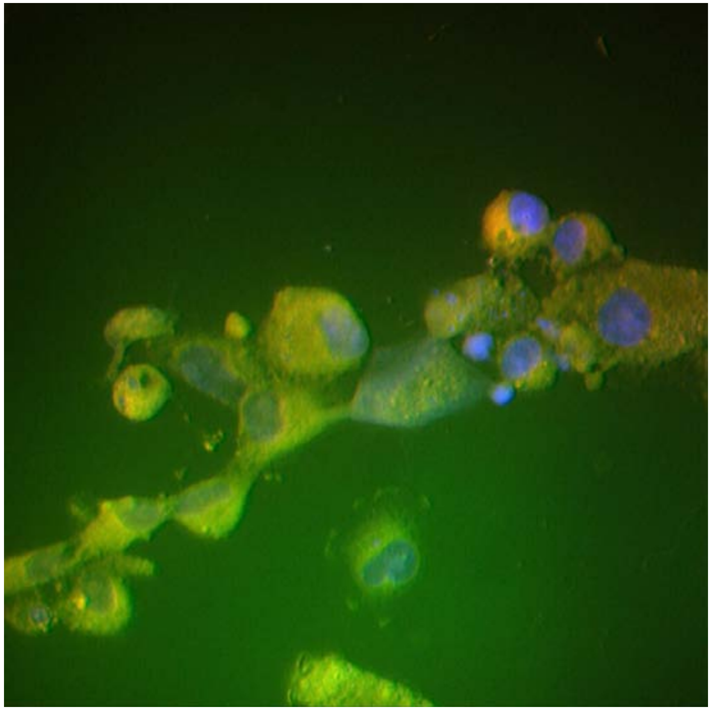
Regularizing QUILL: Results with real 2-D microscopy data

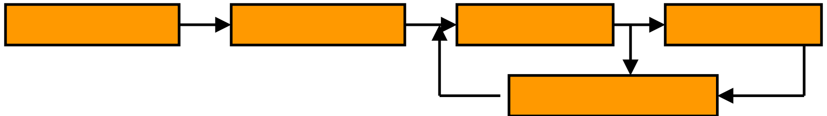
Test: Microscopic images downloaded online (truly blind application)

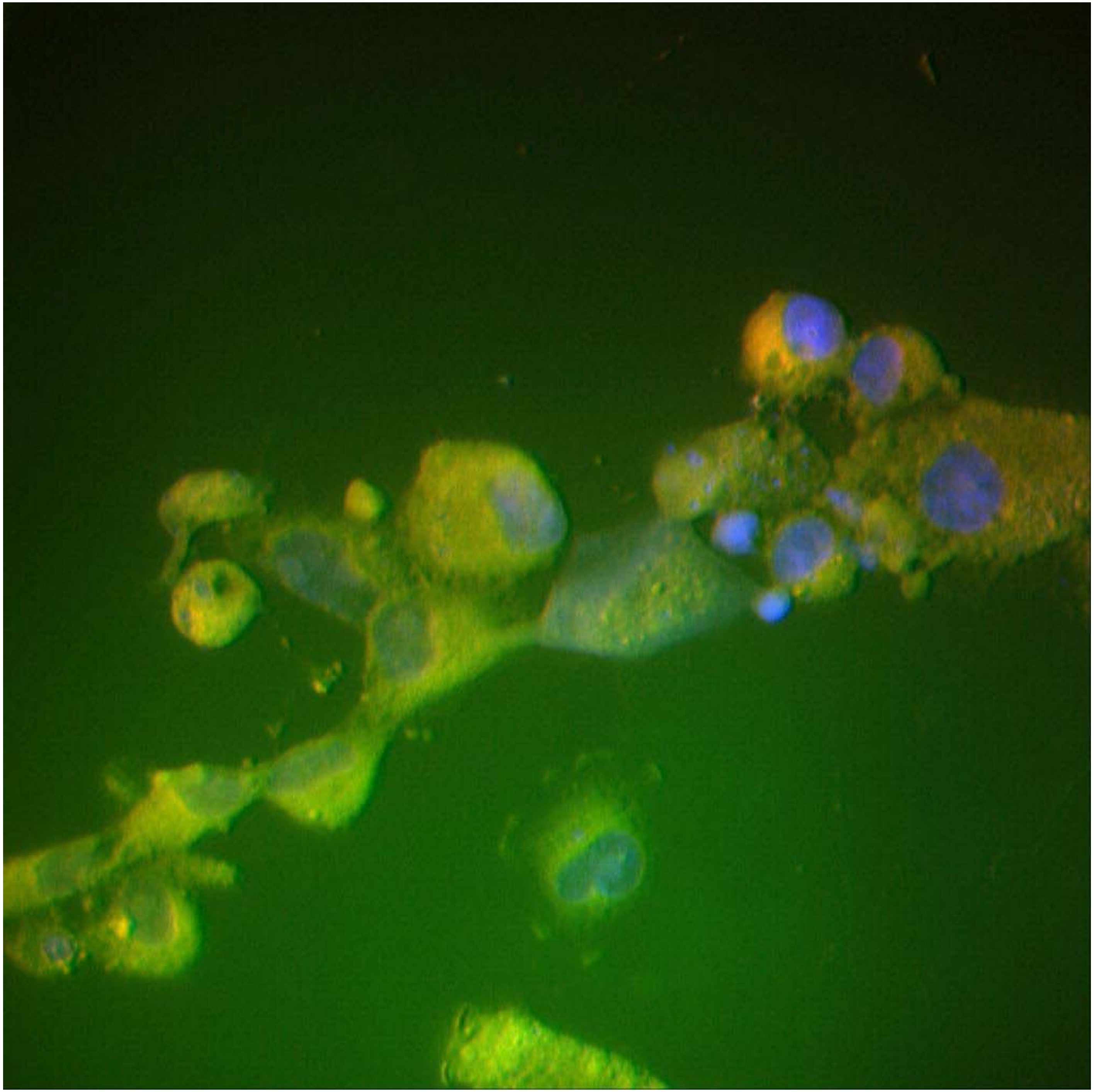


Assume

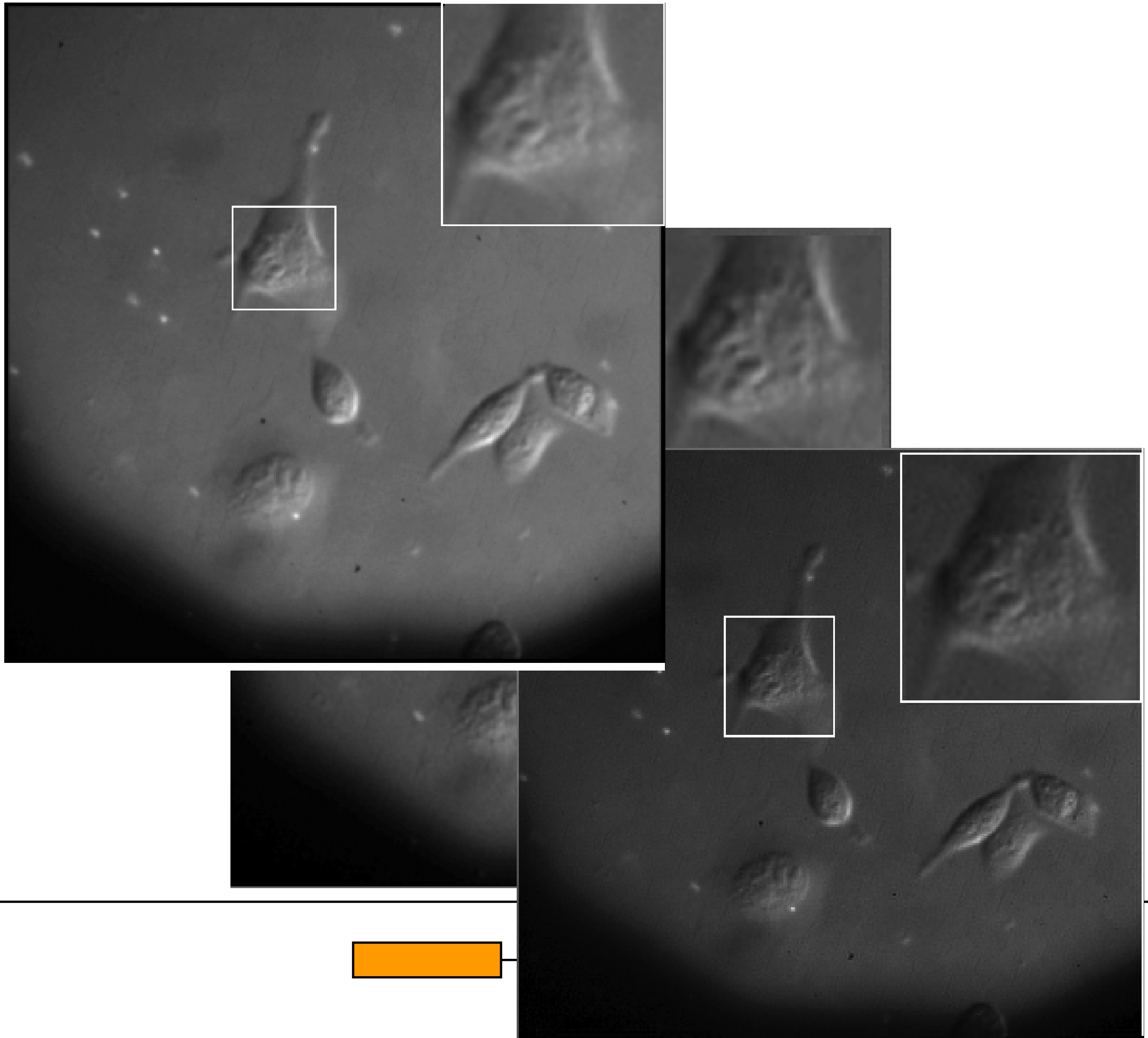
PSF:4X4
TSVD







Comparison with Lucy-Richardson

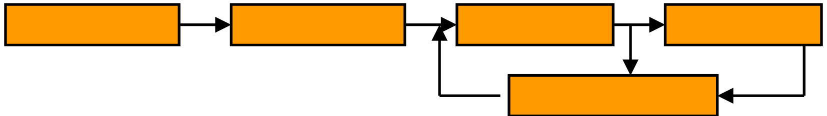


I wish to thank Dr Mary-Ann-Mycek for the image.

Comparison with Lucy-Richardson

- The QUILL algorithm improved visual image quality noticeably.
- It did not take a long time for this 480X480 image as it is a 1 pass algorithm

- The Lucy-Richardson algorithm did not improve the image quality (visually). The starting PSF was assumed to be Gaussian. This is a reasonable approximation.
- It took many iterations to come to a reasonably stable solution and even then there are ringing artifacts in the solution.



Publications

- 2-D Blind Deconvolution from partial data using Bezout’s lemma and the QUILL model, Yagle AE and Shah S, *Accepted, IEEE Trans Image Processing*

Deconvolution in Electron Microscopy

EM Imaging equation

For thin biological specimen, one can use the weak phase weak amplitude approximation to derive the image formation equation.

In the frequency domain:

$$\boxed{I(q,\phi) = CTF(q,\phi)O(q,\phi)E(q,\phi) + S(q,\phi) + N}$$

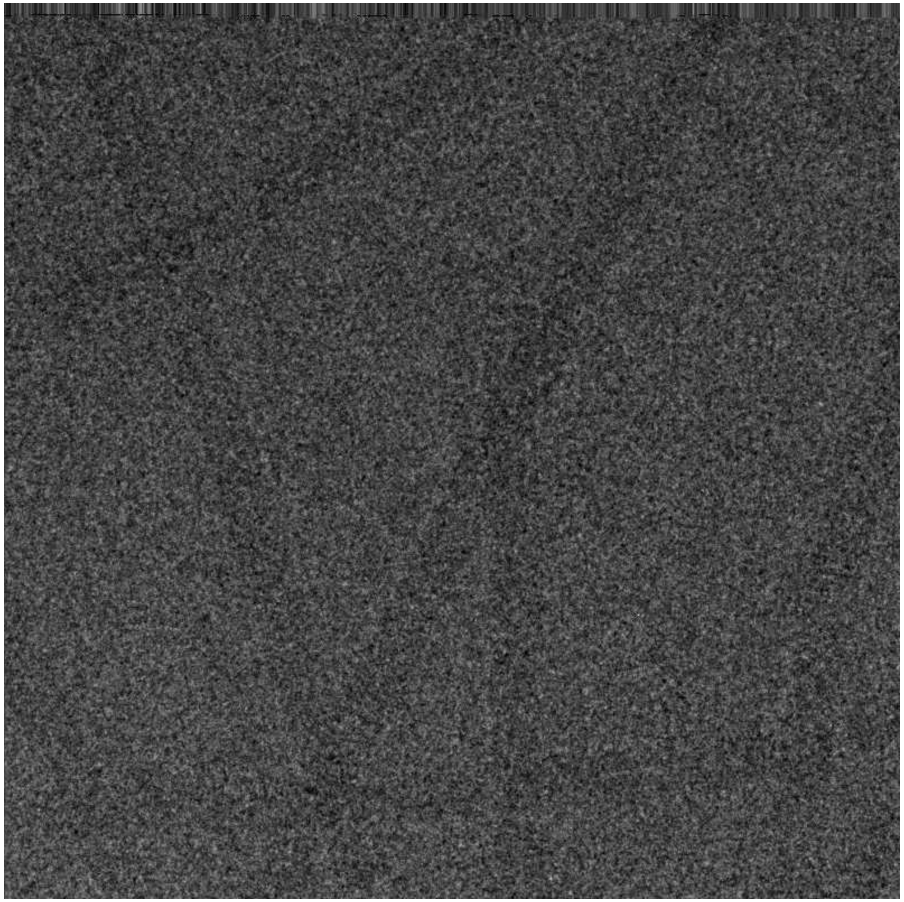
Image CTF Object Background Noise

Goal: Recover $O(q,\phi)$

The other terms

$CTF(q,\phi)$	The Contrast Transfer Function. It is the Fourier transform of the PSF. Theoretically well defined.
$S(q,\phi)$	Background signal due to inelastic scatter. Although it contains no object information, it is technically NOT noise.
N	Noise due to photon counting and quantization. Mixture of Poisson And Gaussian noise.
$E(q,\phi)$	Envelope Function. Due to the incoherence of the electron beam the system response decays with increasing frequencies. This is Mathematically summarized in the envelope function. Its not a big factor at low resolutions.

Why is EM deconvolution difficult ?



Cryo-EM micrograph of Tobacco Mosaic Virus (TMV). Note the poor contrast

High background signal and high noise levels are common in most cryo-EM images. This makes:

1. CTF estimation difficult.
2. Implementation of blind deconvolution algorithms difficult.

Strategy

We adopt a two step approach to solve this problem

1. Develop a successful CTF estimation algorithm that can estimate the CTF Accurately using little or no manual input.

2. Deconvolve individual 2-D micrographs of EM images with the acquired CTF estimate using an edge preserving penalized least squares approach.

The deconvolved 2-D micrographs are then used to reconstruct a 3-D model.

CTF Estimation

CTF Formula

$$CTF(q,\phi)=\sqrt{1-\alpha^2}\,\sin(\chi(q,\phi))-\alpha\cos(\chi(q,\phi))$$

Amplitude contrast

Wavelength of e$^-$ beam

$$\chi(q,\phi)=\frac{\pi}{2}\left[C_s\lambda^3q^4-\lambda\,q^2(2\Delta f_{mean}+\Delta f_{diff}\,\cos(2\phi-2\phi_a))\right]$$

Spherical aberration of e$^-$ lens

Mean defocus

Differential defocus

Astigmatism angle

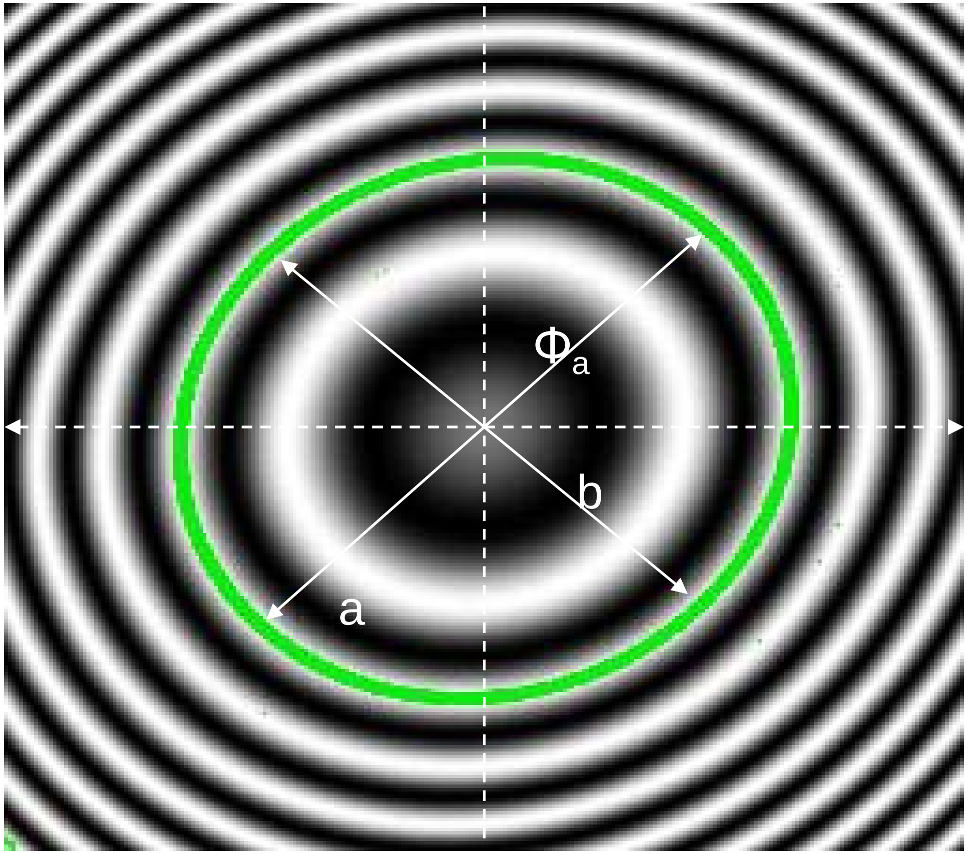
🟦 Known parameter

🟪 Unknown parameter

CTF Parameters
A Geometrical Perspective

$$\Delta f_{mean} = \frac{a+b}{2}$$

$$\Delta f_{diff} = b - a$$

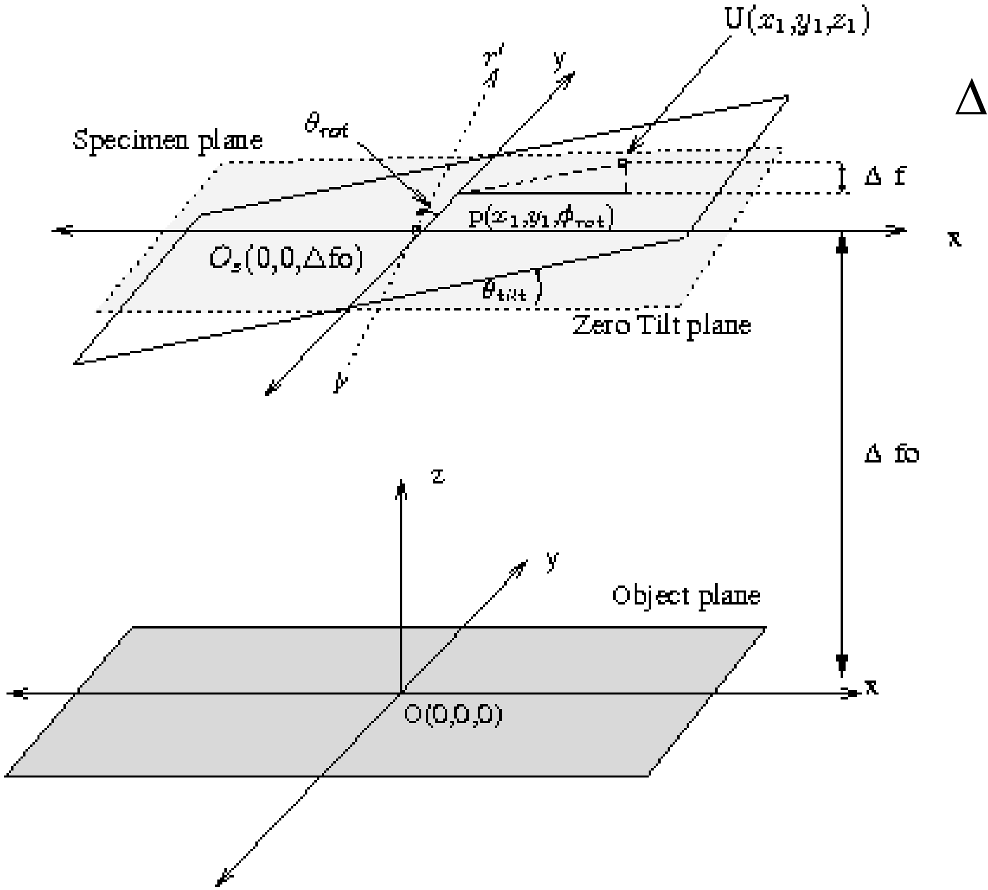


Differential defocus and astigmatism angle are astigmatism parameters

α serves as a phase delay for the Sinusoidal wave

Many algorithms attempt to solve this estimation by radially averaging the CTF first to increase SNR. Obviously this will NOT capture the astigmatism.

Mean defocus: Another view
Tilted case



Mean defocus at point U

$$\Delta f_{mean} = (x\cos\theta_{rot} + y\sin\theta_{rot})\tan\phi_{tilt} + \Delta f_0$$

Where

θ_{rot} Angle made by rotation axis

ϕ_{tilt} Tilt angle of specimen plane

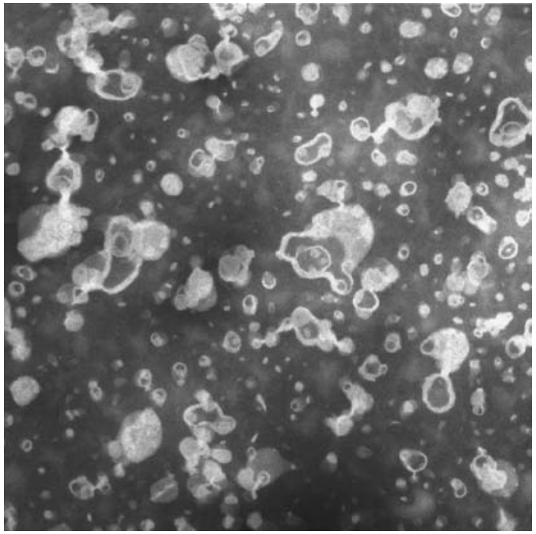
Δf_0 Mean defocus at rotation axis

For a tilted specimen mean defocus is not constant across the specimen plane.

Types of EM samples

Negative Staining

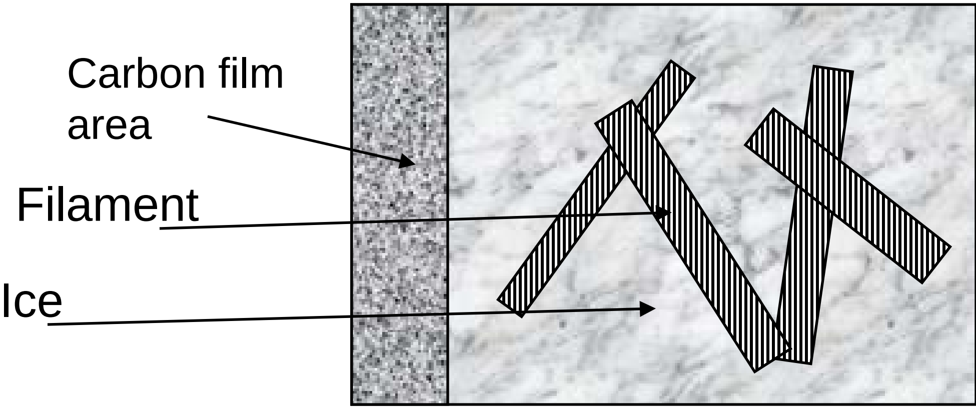
- Biological sample is treated with heavy salts.
- Sample is dehydrated and may undergo significant transformation.
- Images have high contrast (SNR).
- Resolution is lower as it is limited to grain size of the staining salt.



Negative stain TEM of lipid droplets isolated from macrophage foam cells.*

Cryo

- Biological sample is flash frozen in ice.
- Sample is preserved in native state.
- Images have low SNR
- Many times there is an underlying carbon support film.

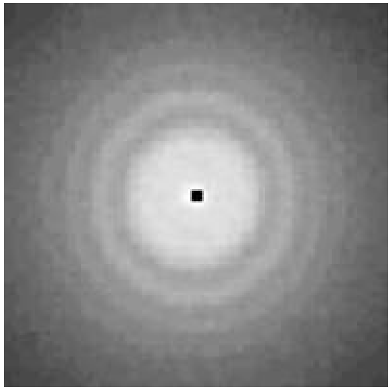


Schematic Diagram of a cryo sample

*Source: Image by Jay Jerome, Ginny Kellner-Weibel, George Rothblat Vanderbilt Website

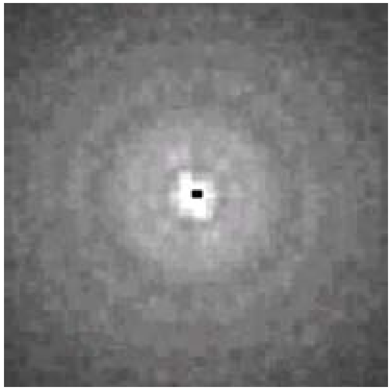
‘Types’ of CTFs

Easy: Negative Staining



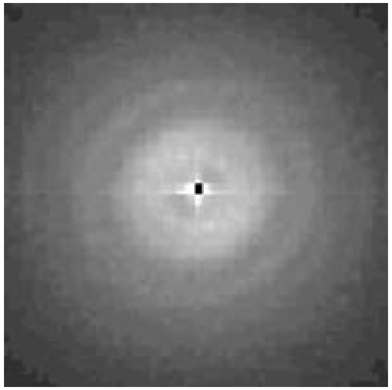
3 orders of CTF rings are seen

Harder: Tilted Cryo with carbon backing



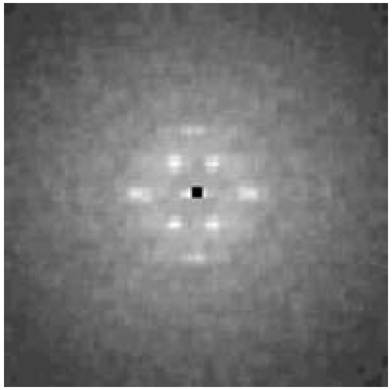
Due to sampling Within a stripe, SNR is lower than in the planar case

Hard: Cryo with carbon backing



Fewer orders of CTF rings are seen but Contrast is quite good.

Hardest: Cryo with specimen in ice



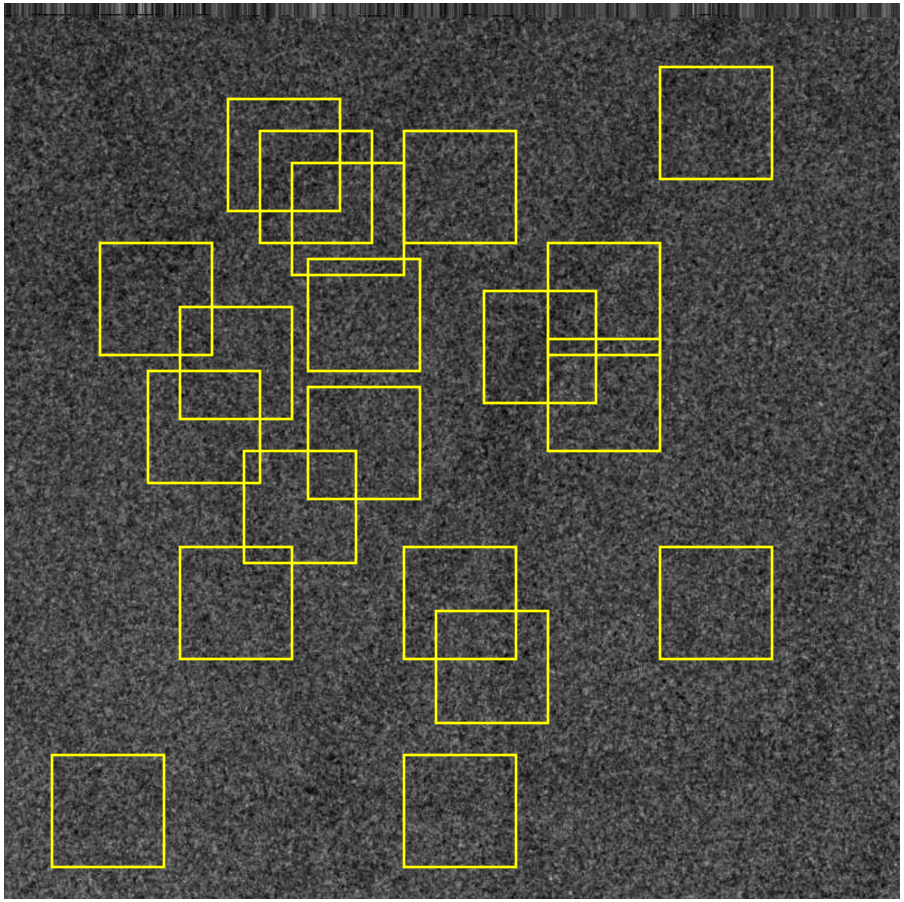
Barely any rings Seen. The power Spectrum of the flagella filament is clearly seen in the CTF power spectrum.

CTF Parameter Estimation

Step 1: Acquiring CTF power spectrum estimate

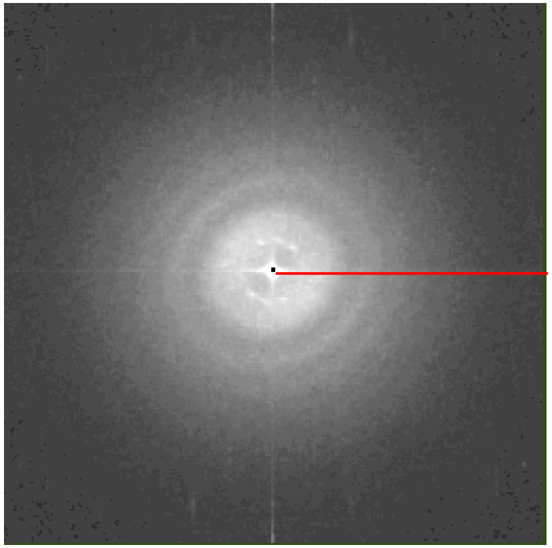
1. Calculate the power spectrum of 50-100 sample windows whose locations are randomly chosen.

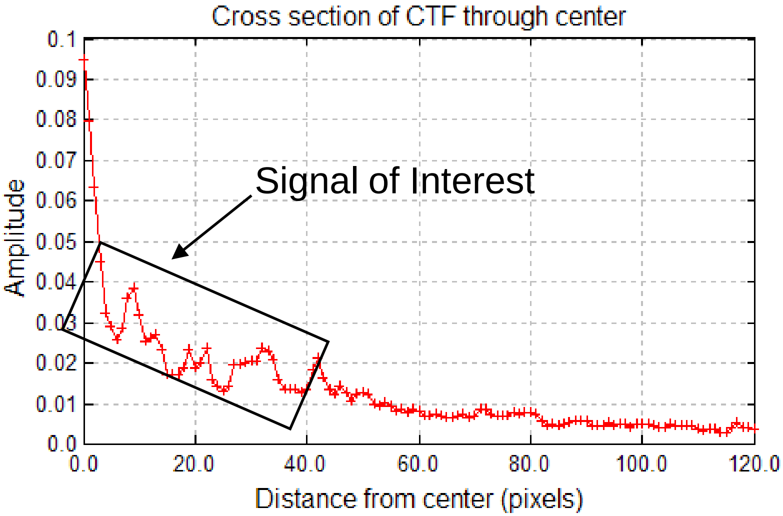
2. Average the power spectrum to get (hopefully) a good CTF power spectrum estimate.



Random sampling strategy was chosen over overlapped periodogram averaging to reduce influence of object power spectrum.

Step 2: Estimating CTF power spectrum background





The background S(q,Φ) is quite significant and much larger than the signal of Interest. So background must be estimated and subtracted.

Present strategy: Radially average CTF and estimate background by fitting a 1-D mixed exponential signal.

Problem: No one said the background was circular !

Background must be estimated in 2D

Step 2: Estimating CTF power spectrum background

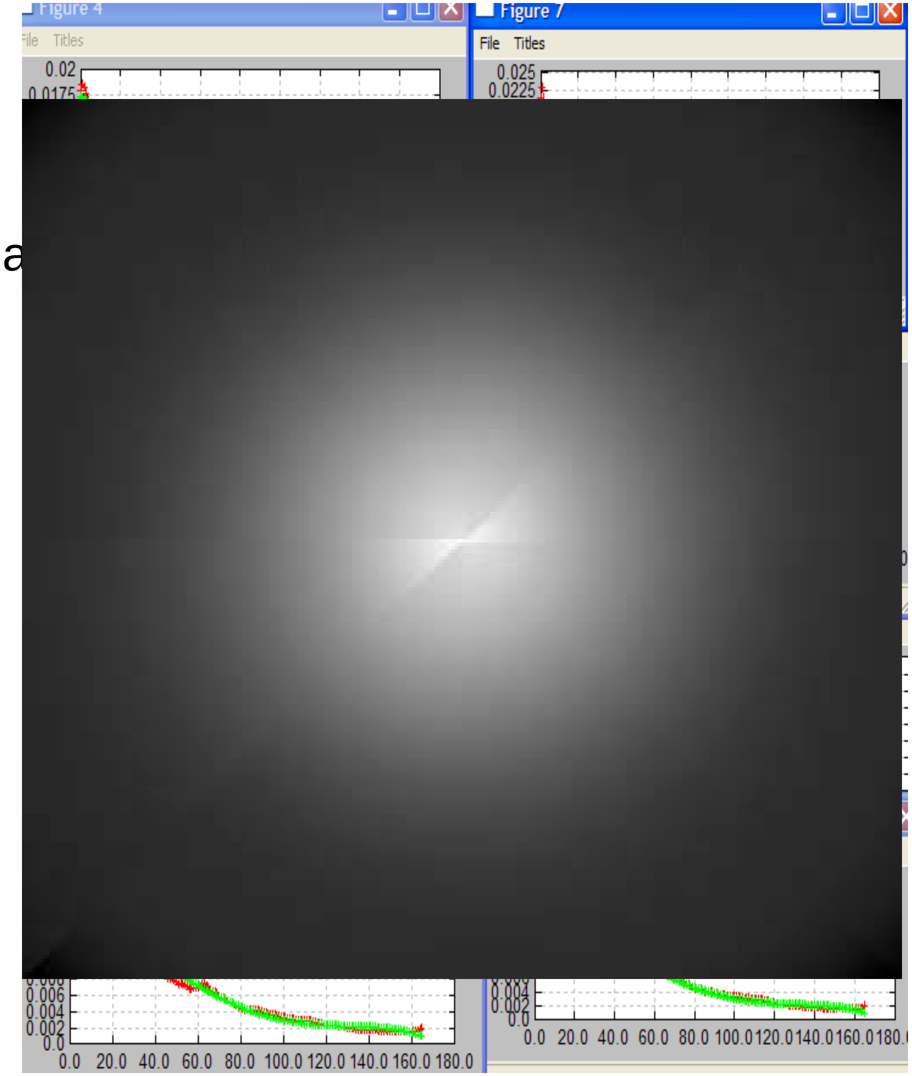
Problem with 2D approach

1. Noise. Averaging smooths out the signa
   for a good fit.
2. What equation to use for a 2D fit ?

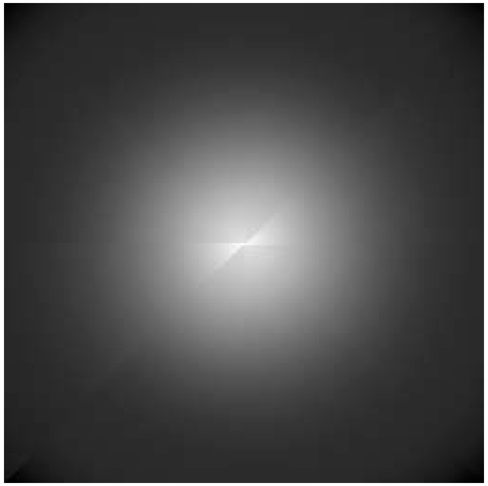
A conservative compromise

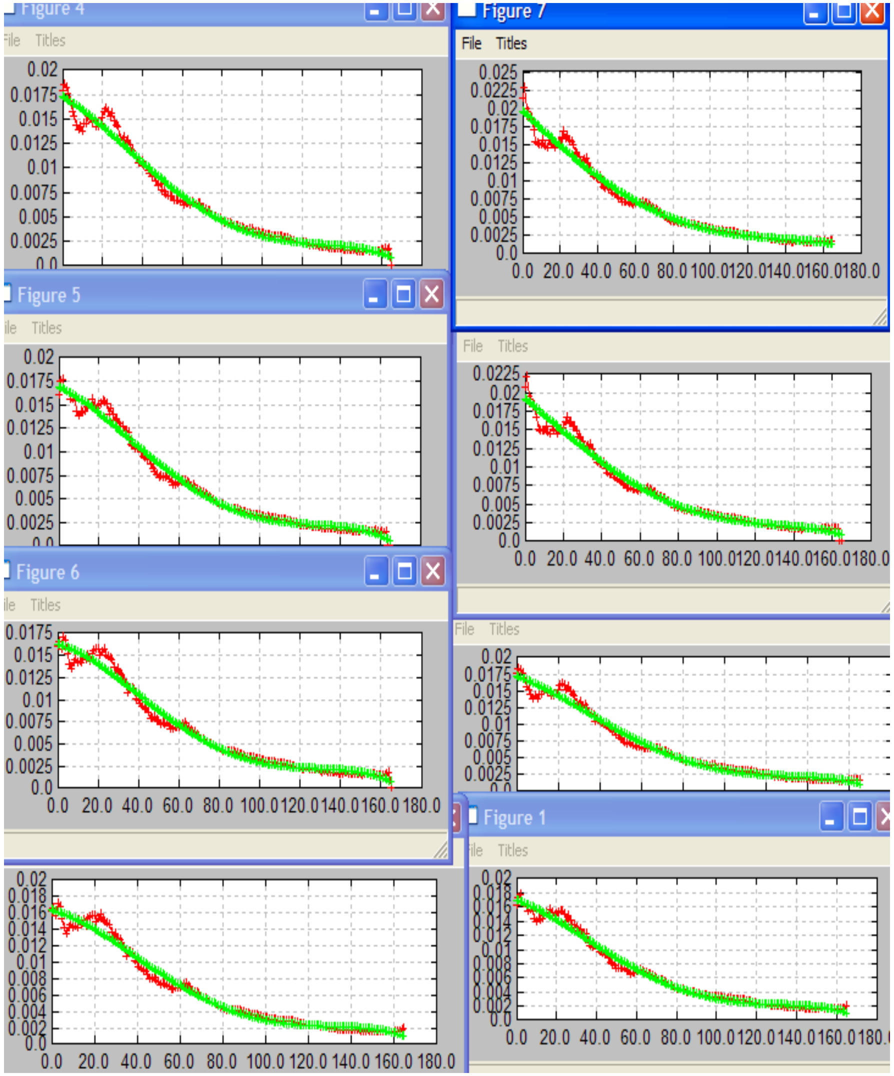
1. Divide up the image into 8 sectors.
2. Calculate fits using a quartic polynomial
   for radially averaged CTF in each sector.

Accounts for astigmatism somewhat….



Step 2: Estimating CTF power spectrum background



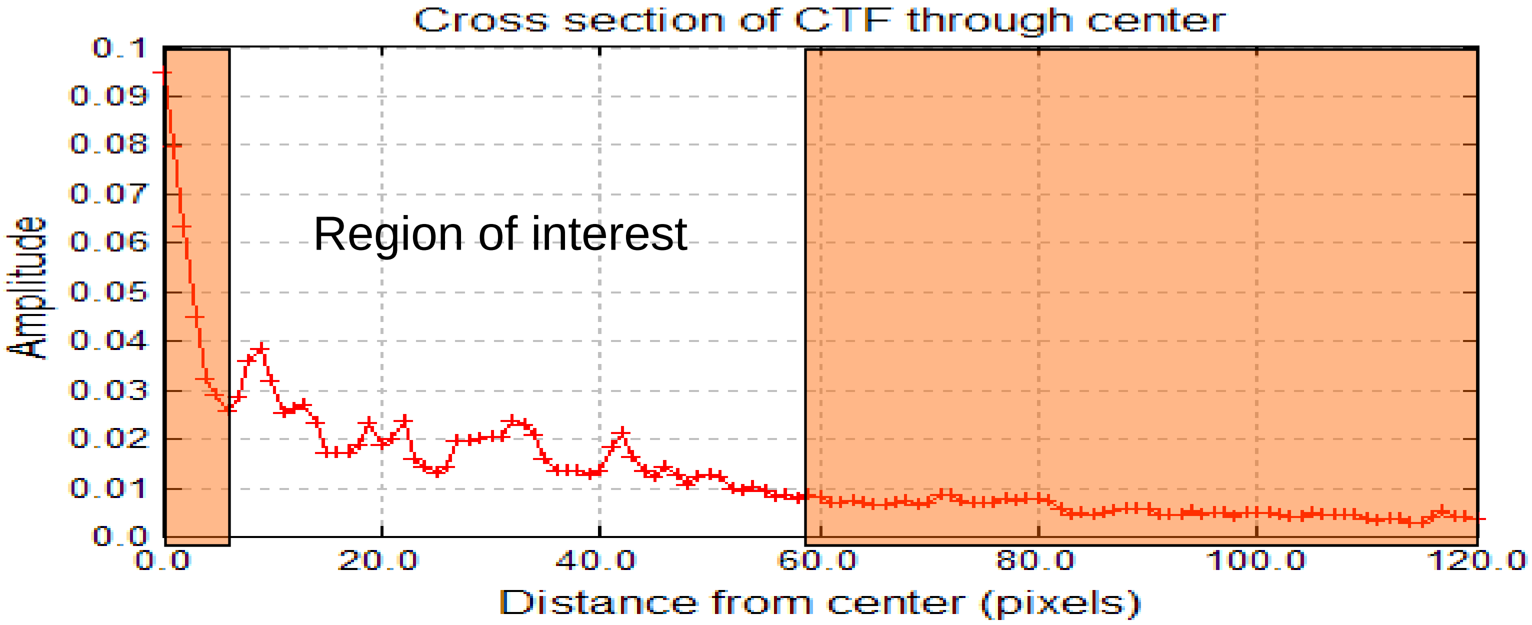


How about a mixed exponential fit ?

I found the polynomial fit does better, although it does gives rise to ripples at the higher frequencies.

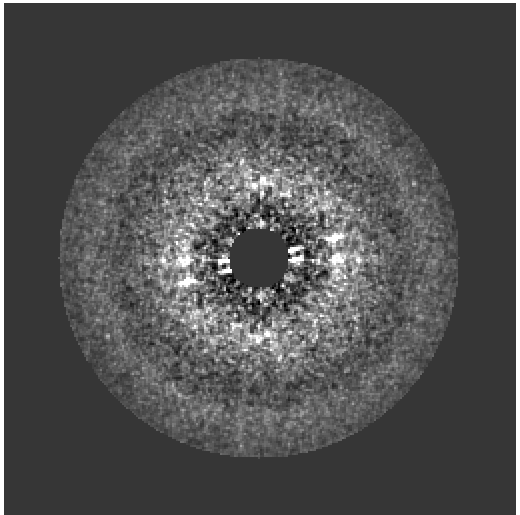
After subtraction we get a residual CTF.

Step 3: Masking



Region of interest

The masked residual CTF



Step 3: Masking

Inner Mask

1. Make a very conservative guess and do a 8 point grid search with CTF parameters to get initial guess Parameters for CCG.
2. Use guess parameters to calculate approximate location of first zero of CTF.
3. Set inner mask radius using guess parameters.

Outer Mask

1. Radius can be set automatically by choosing radius at which 95-99% of the residual CTF signal is contained.

Step 4: Parameter Estimation

Use Constrained Conjugate Gradients to minimize a cross-correlation cost function.

Goal :Minimize

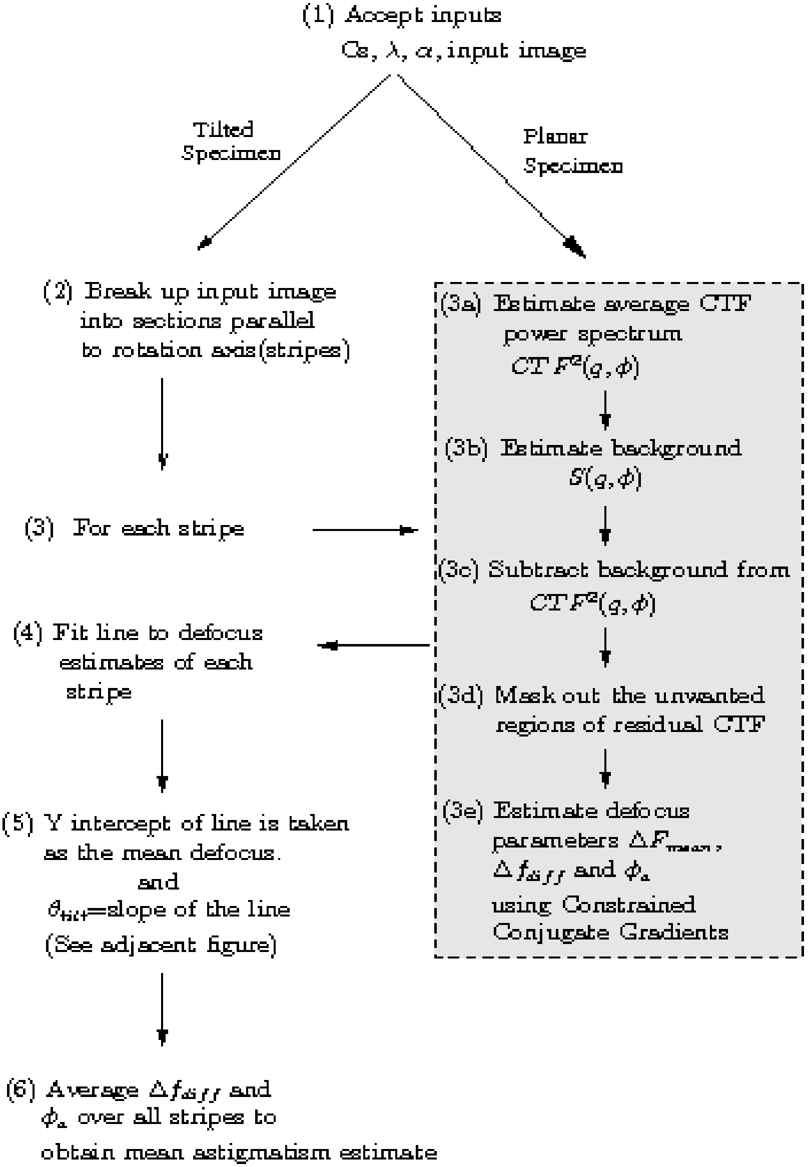
$$\psi = 1 - \frac{\Sigma CTF_{res}^{2} CTF^{2}}{\sqrt{\Sigma CTF^{4} \Sigma CTF_{res}^{4}}}$$

Parameter constraints

Parameter	Min	Max
Mean defocus	0.05 µm	10 µm
Differential defocus	0 µm	1 µm
Astigmatism tilt (deg)	0	180

Amplitude contrast is very difficult to estimate accurately and is NOT estimated.

Flowchart of planar and tilted CTF estimation algorithms



(1) Accept inputs
Cs, λ, α, input image

Tilted Specimen

Planar Specimen

(2) Break up input image into sections parallel to rotation axis (stripes)

(3) For each stripe

(3a) Estimate average CTF power spectrum $CTF^2(q,\phi)$

(3b) Estimate background $S(q,\phi)$

(3c) Subtract background from $CTF^2(q,\phi)$

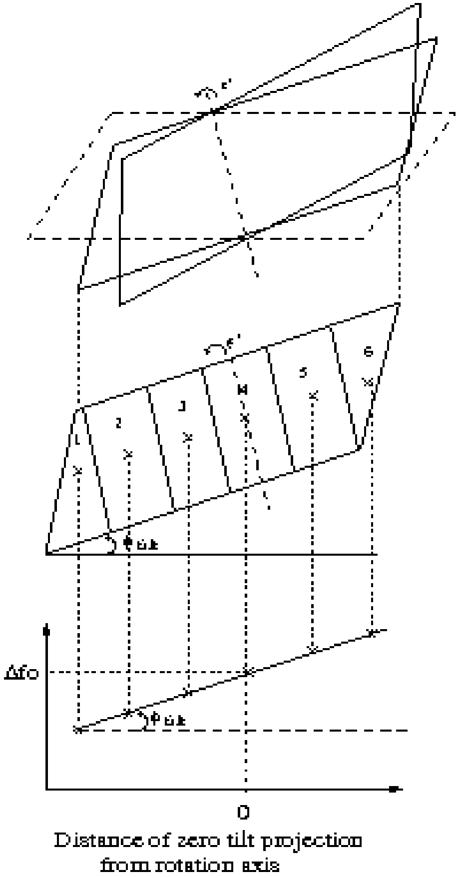
(3d) Mask out the unwanted regions of residual CTF

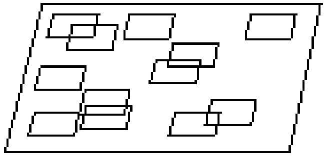
(3e) Estimate defocus parameters ΔF_{mean}, Δf_{diff} and ϕ_a using Constrained Conjugate Gradients

(4) Fit line to defocus estimates of each stripe

(5) Y intercept of line is taken as the mean defocus.
and
θ_{tilt}=slope of the line
(See adjacent figure)

(6) Average Δf_{diff} and ϕ_a over all stripes to obtain mean astigmatism estimate





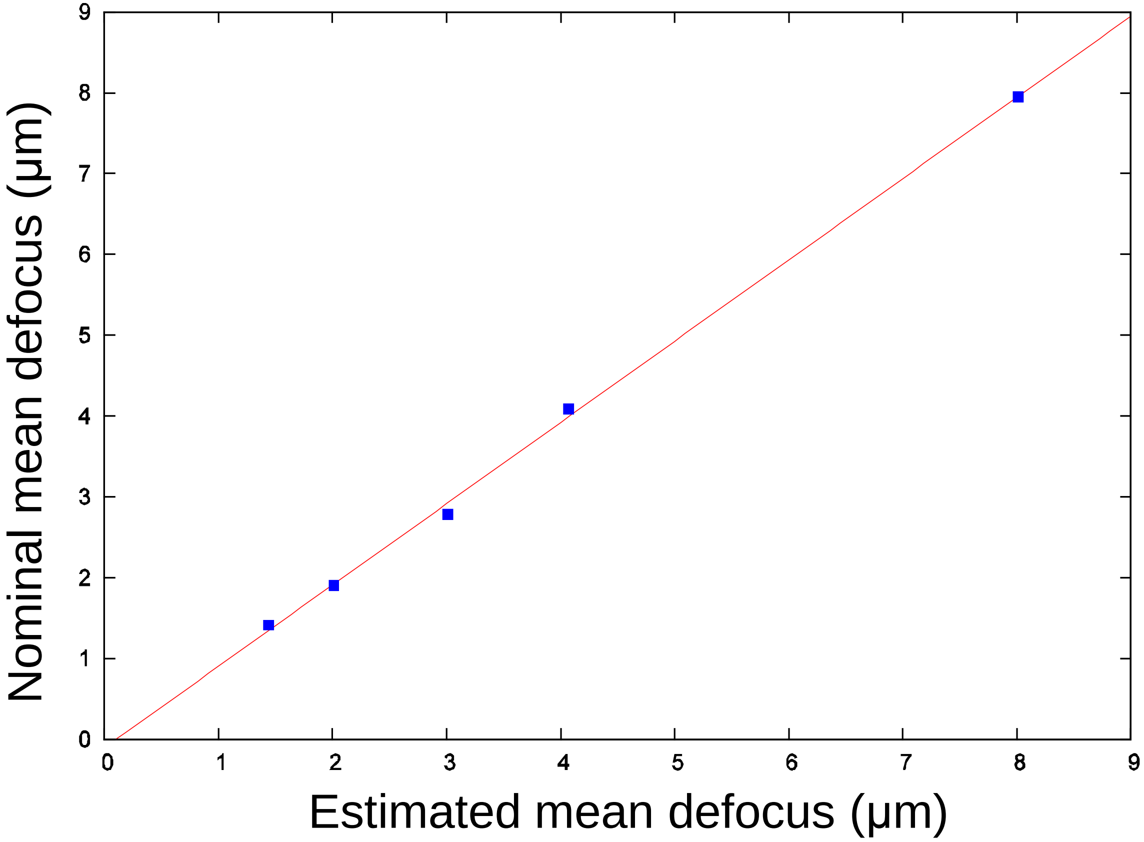
Random sampling of CTF power spectrum at 50 points

Results

Defocus Series experiment: Cryo sample backed with carbon support film

Mean defocus read from instrument has a zero error and is a not a reliable measurement.

The difference between two measurements is accurate as zero error is cancelled out.



Zero defocus offset: 0.099 µm
Slope of fit line: 1.005
Correlation of fit: 0.999

Results

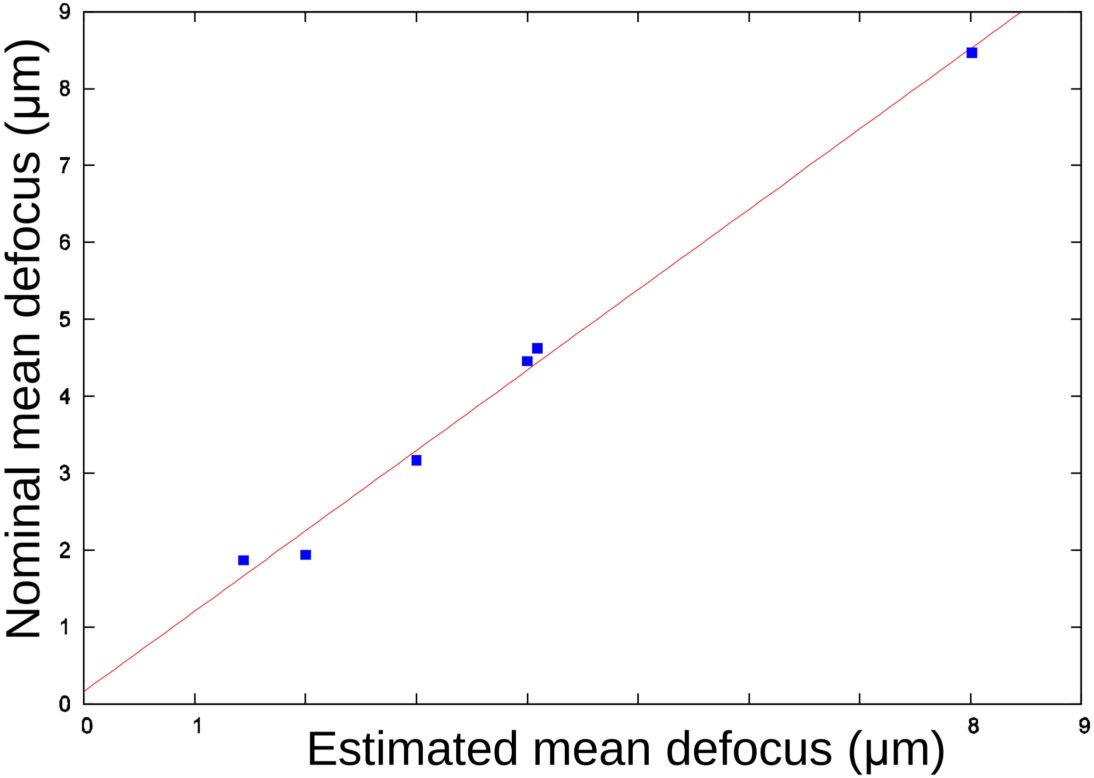
Defocus Series experiment: Sample backed with carbon support film tilted to 45°

The cryo sampled was tilted to 45 degrees. Images were recorded at different defocii.

Defocus parameter estimates Were obtained along stripes parallel to the rotation axis.

Mean defocus at rotation axis Was recorded.

For the line fit, the tilt angle measured from the instrument Was used.



Zero defocus offset: -0.156 µm
Slope of fit line: 1.045
Correlation of fit: 0.997

Results

Validation

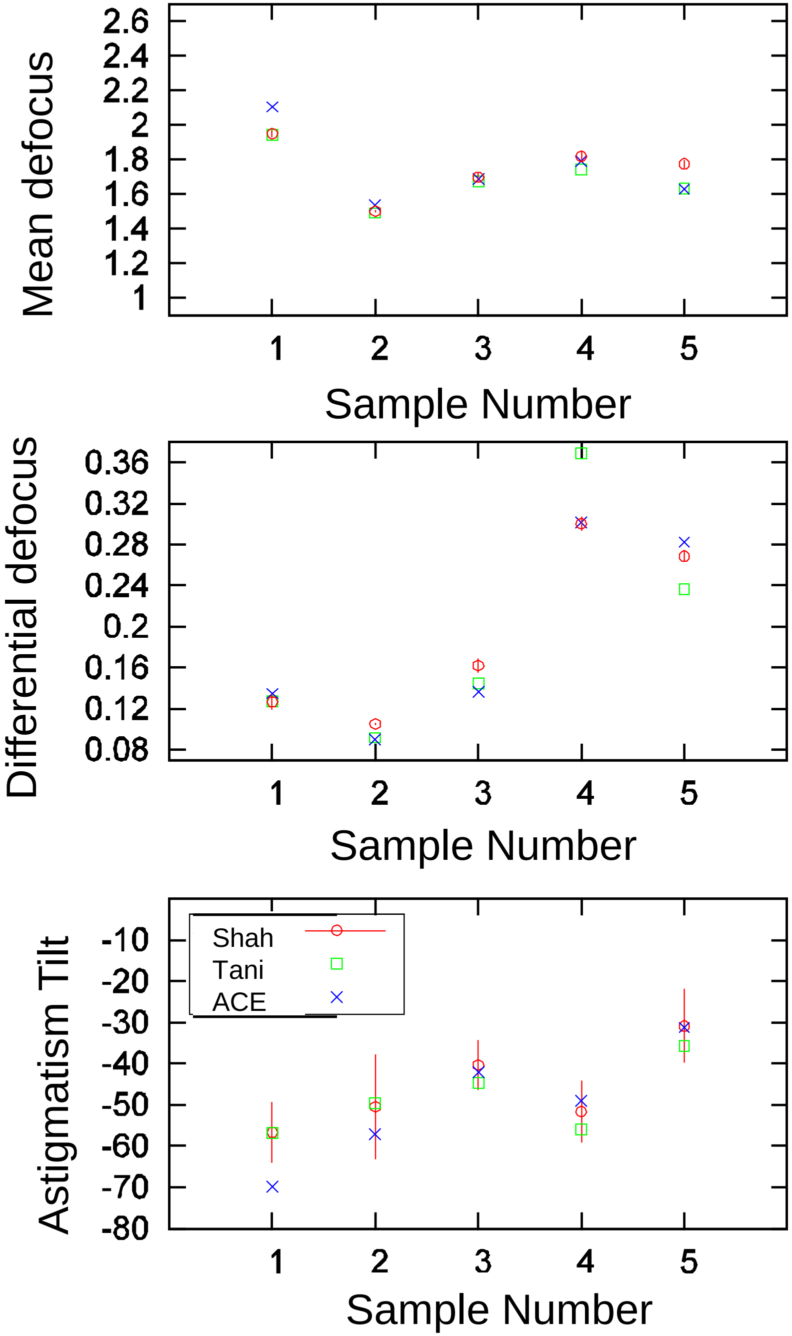
10 Carbon film estimates were compared to estimates using

Automatic CTF Estimation (ACE)
Uses edge detection for estimating astigmatism and then performs elliptical averaging to estimate CTF in 1-D

PLTCTFX
Semi automatic CTF estimation algorithm that performs 1-D CTF estimation in 5 sectors.

Good agreement among all algorithms.

Comparison of CTF parameters



Results

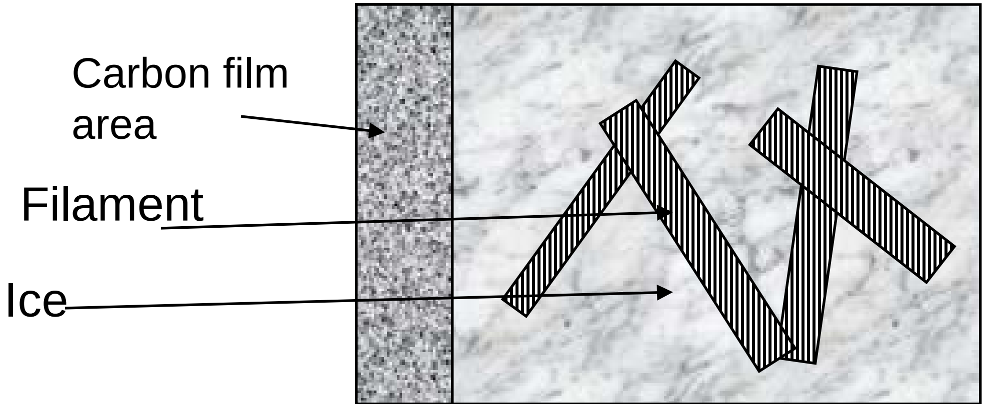
Protein Embedded in Ice case

By far the hardest case

a. There is no carbon film support.
b. The power spectrum of the protein interferes with the CTF
c. Noise levels are very high
d. Specimen are often very thin. So one cant sample over a large area.
e. Fewer if any CTF rings are seen. More rings are vital to getting astigmatism information.

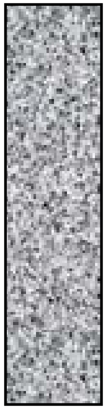
Until now, no algorithm has completely succeeded in estimating defocus parameters for this case.

Experiment: Estimated defocus parameters from a micrograph of bacteria flagella Filaments and compared it with carbon film estimates from a neighbouring areas

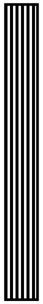


Results

Protein Embedded in Ice case



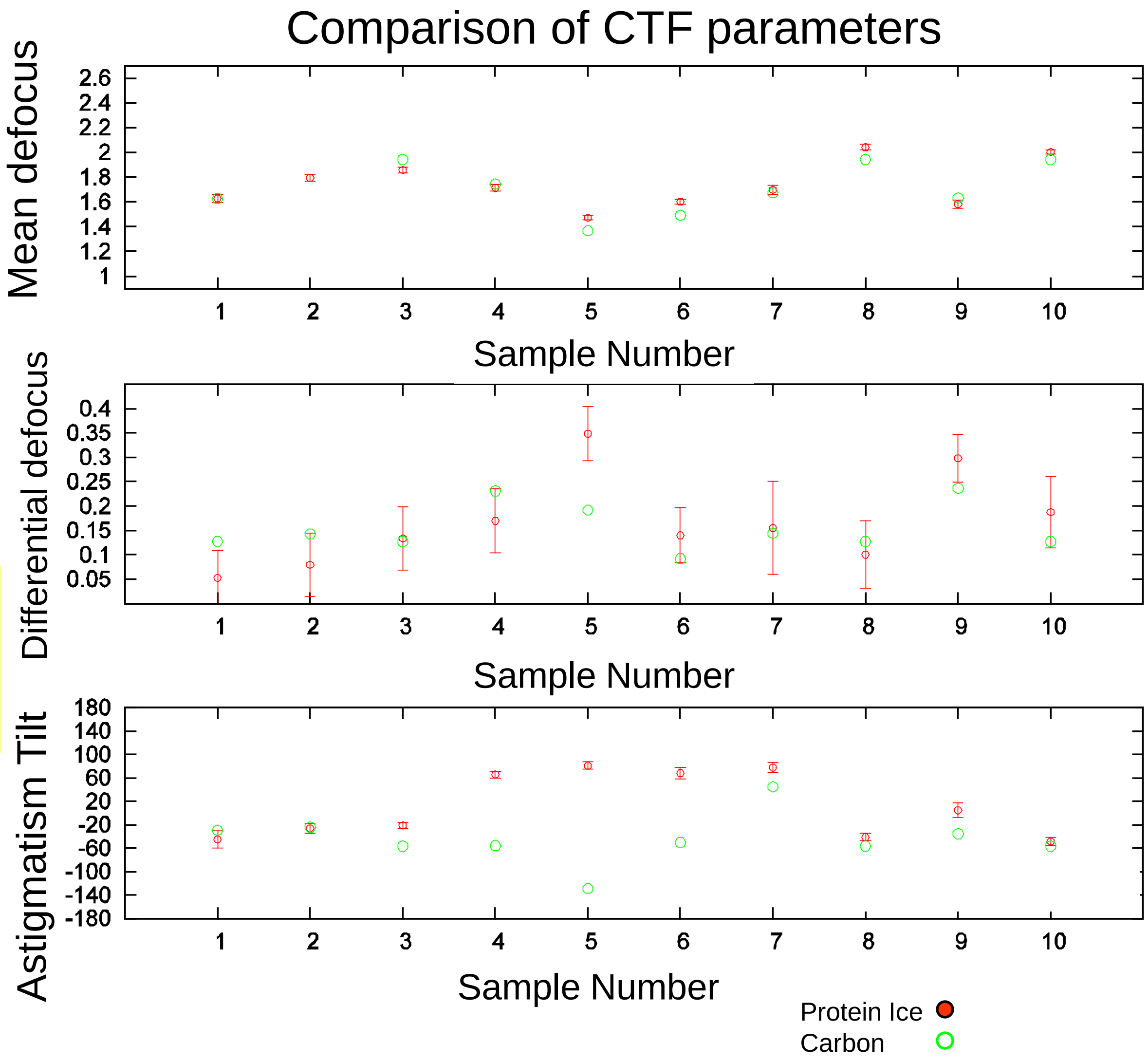
Estimate #1



Estimate #2

Mean defocus estimates appear quite accurate but astigmatism parameters are off.

Until now, no algorithm has completely succeeded in estimating defocus parameters for this case.



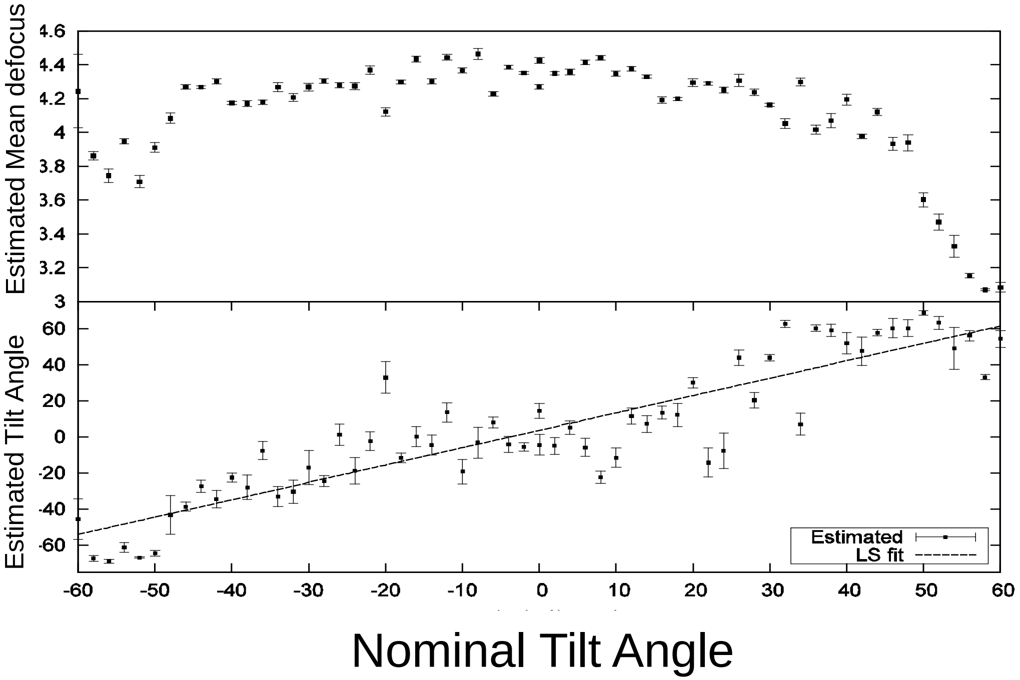
Comparison of CTF parameters

Results

Tomographic Tilt Series

Set of low dose images acquired from -60 to +60 degrees. For each image, defocus parameters at rotation axis were estimated estimated.

Angle estimates are not very accurate But the general trend is good.



Mean defocus at axis, exhibits inverse U shaped profile. This is because sample under observation is not exactly at rotation axis but slightly above it. As a result when it is rotated it becomes closer to the observation plane. This is called **Eucenricity**.

GUI Demo

Publications and other developments

Automatic Contrast Transfer Function estimation for cryo and cryo-tomographic EM images, Siddharth Shah, Erik Hom, Koji Yonekura, John Sedat, David Agard, *in review*, Journal of Structural Biology

EM Deconvolution

The imaging equation

$$\boxed{I(q,\phi) = CTF(q,\phi)O(q,\phi)E(q,\phi) + S(q,\phi) + N}$$

To simplify the equation, we ignore the background. It shall be treated as ‘noise’.

For the time being, we will only aim to deconvolve out the CTF and not the envelope function. In other words we recover the envelope function filtered version of the object.

Simplified Equation

$$\boxed{I(q,\phi) = CTF(q,\phi)O(q,\phi) + N}$$

Look familiar ?

Previous

$$Y(q,\phi) = H(q,\phi)O(q,\phi) + N$$

Current

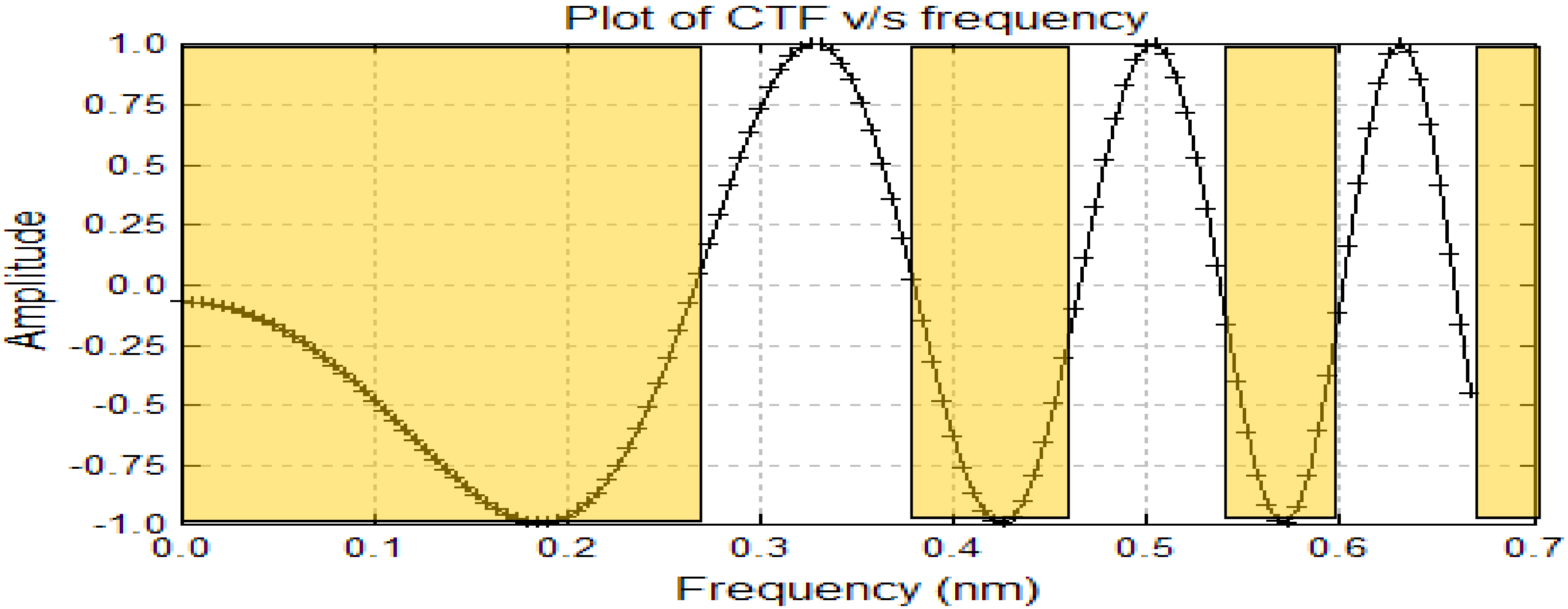
$$I(q,\phi) = CTF(q,\phi)O(q,\phi) + N$$

It is the same problem after the approximations.
Standard deconvolution methods can be applied

Using the previous algorithm the CTF can be estimated easily. The question remains as to what algorithm should be used.

Present ‘State of the Art’: Phase Flipping

Phase Flipping: This is the most commonly used CTF correction method. Only Corrects for the phase distortions of the CTF. Does not correct for amplitude effects.



Multiply frequencies corresponding to orange areas by -1.
This is only correct if the CTF were a rectangular wave !

Present 'State of the Art': Wiener Filtering

Wiener Filtering: Also very common. Attempts to solve the deconvolution problem in a regularized linear least squares framework.

What is the framework ?

Consider Linear least squares

$$\hat{O} = \arg\min_{O} \left\| I - OH \right\|_2$$

This approach will never give a good solution due to the ill-conditioned and ill-posed Nature of the problem. Therefore all deconvolution algorithms use a regularizing term

$$\hat{O} = \arg\min_{O} \left\| I - OH \right\|_2 + \lambda_R R$$

The key here is the choice of R

Wiener Filter

When $R = \|O\|^2$ the solution to the previous equation is a simple inverse filter

$\hat{O} = IW$,where $$W = \frac{CTF^*}{CTF + \lambda_R}$$

This is the Wiener filter. The chief advantage is that it is fast and easy to implement.

Disadvantage

- From a statistical perspective this choice of R discourages large values of the object O. Most images are non-zero, so this is not a good choice.

Visually,

- When λ is small, the high frequency information appears sharp but there is a lot of noise.
- When λ is large, the noise is diminished but the object edges appear blurred.

A more sensible regularizer would prevent sudden discontinuities, i.e. preserve smooth Surfaces while discouraging ‘jaggedness’.

Quadratic Gradient Regularizer

When $R = \|\nabla O\|^2$ we minimize large gradients in the deconvolved image.

The solution for this choice of R is also very simple and easy to implement.

Disadvantage
The same noise-resolution problem as Wiener filtering. No choice of λ appears to preserve noise and edge information simultaneously.

Both choice of R are not optimal because they are linear regularizers.

We need a regularizer that
- Strongly penalizes small gradients that are due to noise
- Weakly penalizes large gradients that are due to edges and perhaps some noise.

Such an edge-preserving behavior is exhibited by Huber functionals.

Edge preserving regularizer

Consider

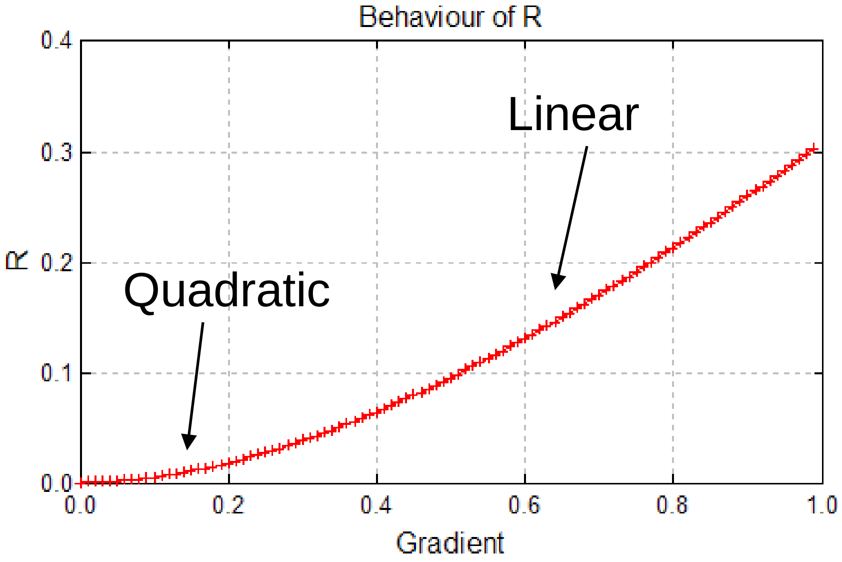
$$R = \nabla O - \ln(1 + \nabla O)$$

For small R

$$R = \frac{\nabla O^2}{2}$$

For large R

$$R = \nabla O$$



For this choice of R, no analytical solution exists. So we use an optimization Algorithm such as Constrained Conjugate Gradients.

The experiment

Experimental Design

•5 samples of the bacterial flagella filament were deconvolved. This was chosen because the bacterial flagella has been resolved to atomic resolution. So we have a good reference.

•They were also deconvolved using other conventional algorithms such as phase flipping, wiener filtering, phase flipping with amplitude correction (two stage deconvolution using phase flipping and wiener filtering.

• The filaments were reconstructed in 3-D using the programs of Mimori et al.

Analysis

• Visual inspection.

• Numerically, we compared the error in phase of the Fourier transform between the deconvolution reconstruction and the reference atomic resolution dataset at various resolutions.

Flowchart

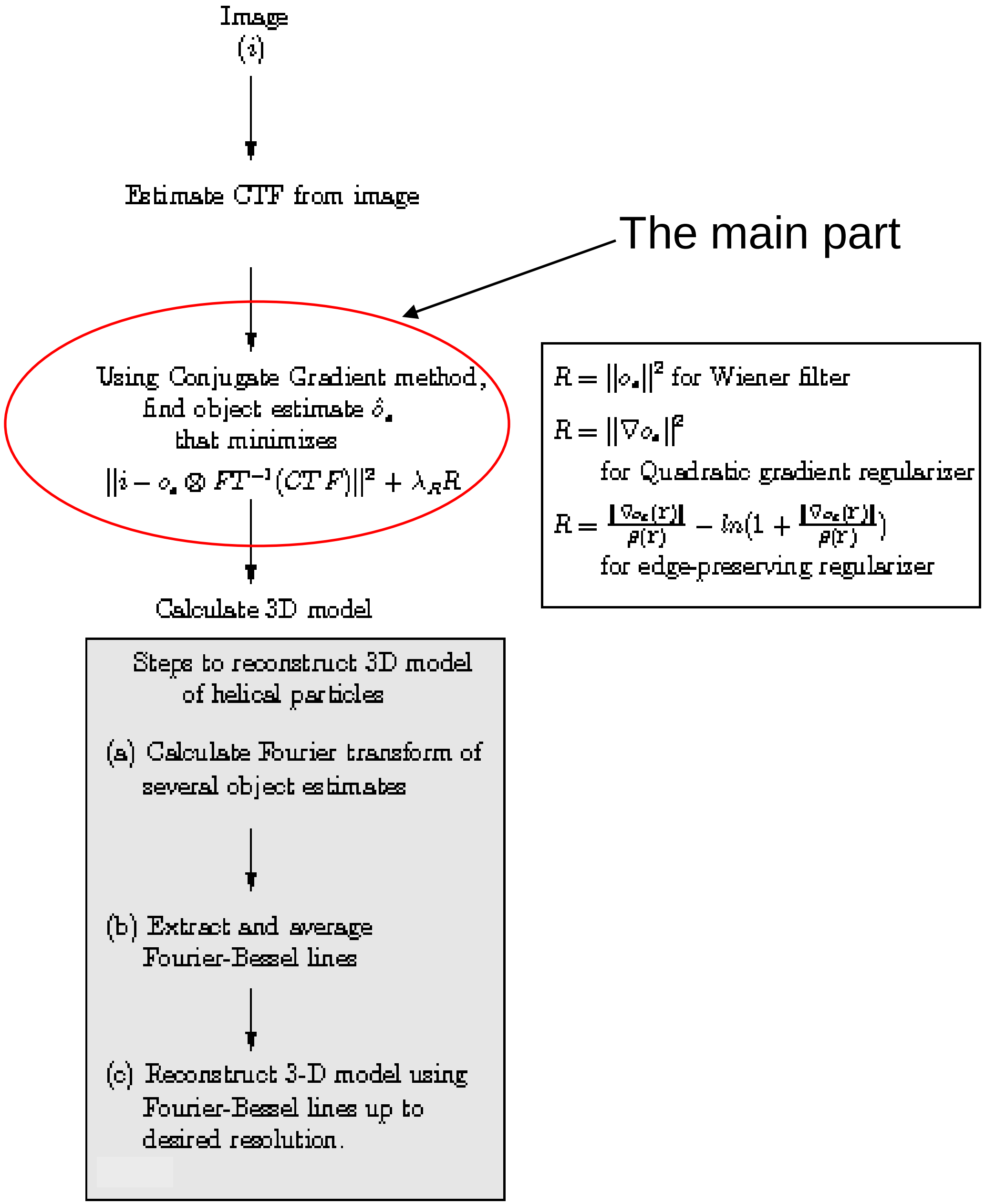


Image
(i)

Estimate CTF from image

The main part

Using Conjugate Gradient method,
find object estimate $\hat{o}_s$
that minimizes

$$||i - o_s \otimes FT^{-1}(CTF)||^2 + \lambda_R R$$

$R = ||o_s||^2$ for Wiener filter

$R = ||\nabla o_s||^2$
for Quadratic gradient regularizer

$R = \frac{|\nabla o_s(\mathbf{r})|}{\beta(\mathbf{r})} - ln(1 + \frac{|\nabla o_s(\mathbf{r})|}{\beta(\mathbf{r})})$
for edge-preserving regularizer

Calculate 3D model

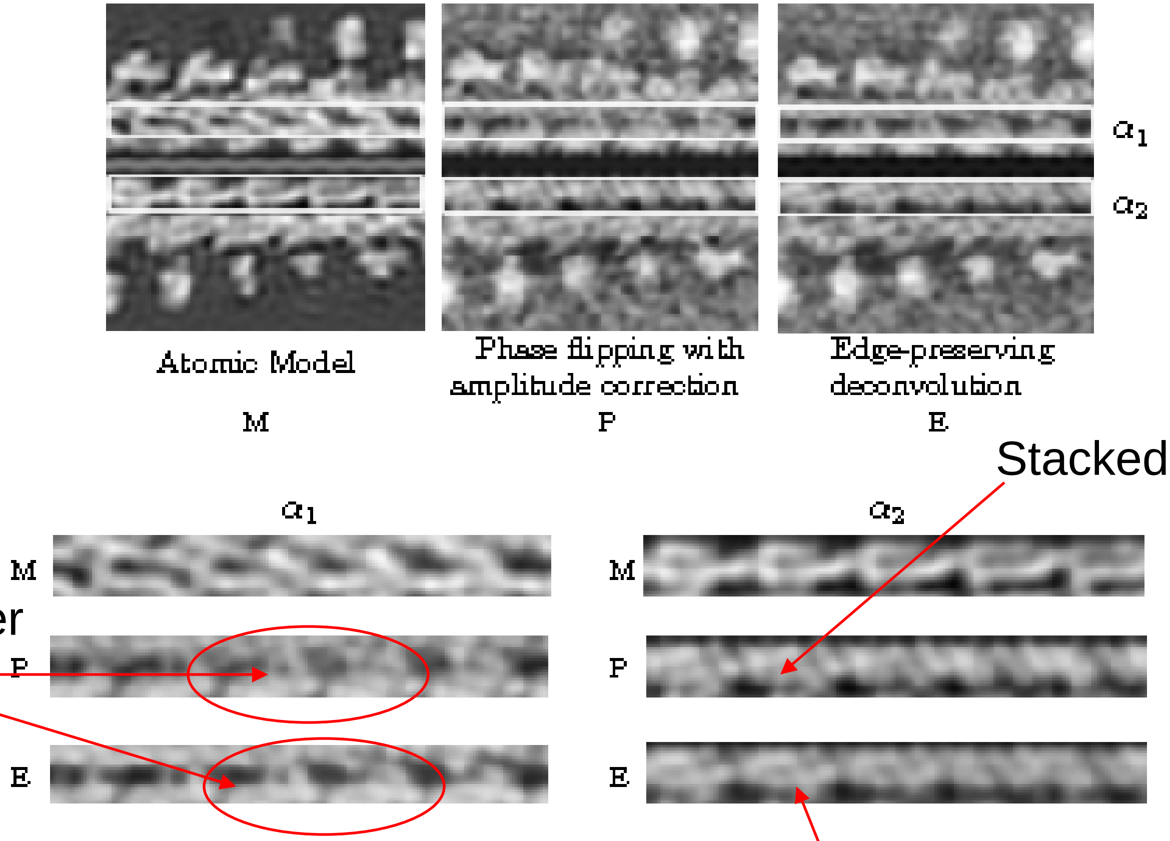
Steps to reconstruct 3D model of helical particles

(a) Calculate Fourier transform of several object estimates

(b) Extract and average Fourier-Bessel lines

(c) Reconstruct 3-D model using Fourier-Bessel lines up to desired resolution.

Results



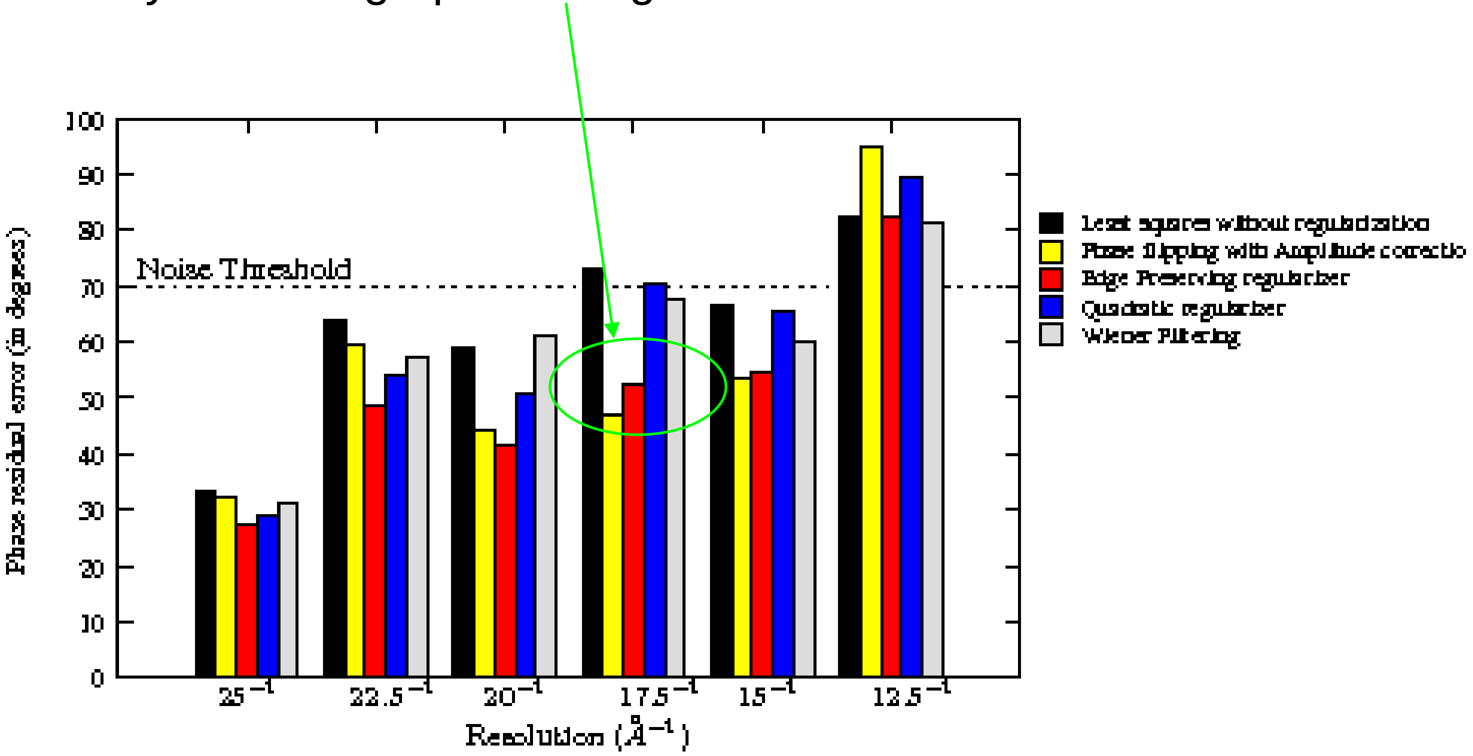
Stacked disk like shape

α helices are better seperated

Shape is better preserved

Results

Why is the edge-preserving method not the best at this resolution ?



- Across all but one resolution, the edge preserving method gave superior phase residuals.

Special Acknoledgement

I would like to thank professors David Agard and John Sedat for letting me work in their labs for a year and for financial support too.

My EM work would not have been possible without them.

Conclusions

The 2-D and 3-D blind even PSF algorithm

Salient Features

1. Simulation results indicate that the algorithm is able to restore images with finite support that have been convolved with even PSFs in 2-D and 3-D.
2. The algorithm is very fast due to the Fourier decoupling steps that breaks up a Single large problem into many small tractable 1-D problems.

Problems

1. Will only work well for compact support images.
2. PSF is often not symmetric, so algorithm will not work well for those images.

Conclusions

The 2-D blind deconvolution algorithm of QUILL images

Salient Features

1. Algorithm performs blind deconvolution of images that are modeled well by the QUILL approximation.
2. As algorithm is non-iterative, it is fast.

Problems

1. Algorithm will not work for objects that are poorly approximated by the model.

Conclusions

Automatic CTF estimation algorithm

Salient Features

1. Only one of two algorithms to date that is fully automatic
2. Only algorithm that can estimate defocus parameters of tilted images.
3. User friendly implementation using a GUI in open source, Numerical Python platform.

Problems

1. Background is fit using a polynomial least squares fit. It should be fit in a constrained least squares approach.
2. Background is fit in 1-D radial averages of sectors. A full 2-D fit would be better.
3 Envelope function is not estimated.

Conclusions

Edge-preserving deconvolution of EM images

Salient Features

1. The first edge-preserving deconvolution algorithm for EM images.
2. Preliminary results indicate performance better than other state-of-the-art algorithms at the lower resolutions.

Problems

1. Results are still preliminary. More testing is needed.
2. Why did we get poor results at one of the resolutions ?
3. Envelope function needs to be included as part of the problem formulation.

Future Work

Blind Deconvolution Algorithms

1. Both even PSF and QUILL deconvolution algorithms need to be tested more on biological datasets.

2. The QUILL algorithm in particular is suited to high speed deconvolution of low resolution images, which is common in optical microscopy.

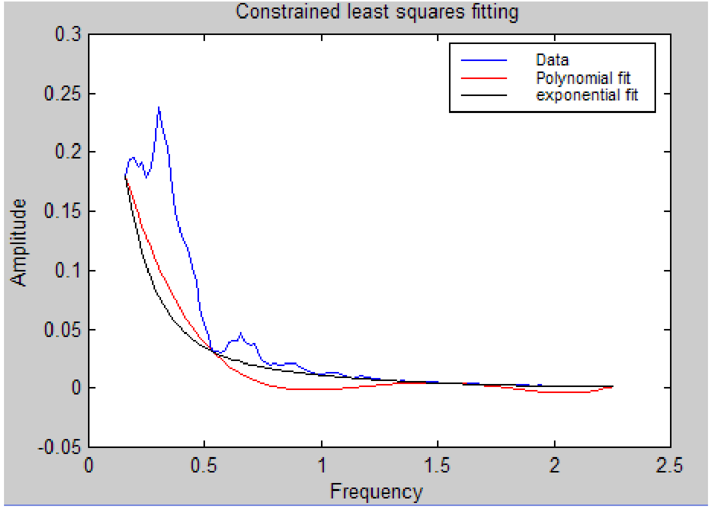
3. The QUILL algorithm needs to be tested with other regularizers. The use of edge-preserving regularizers with the QUILL algorithm appears to be promising.

Future Work

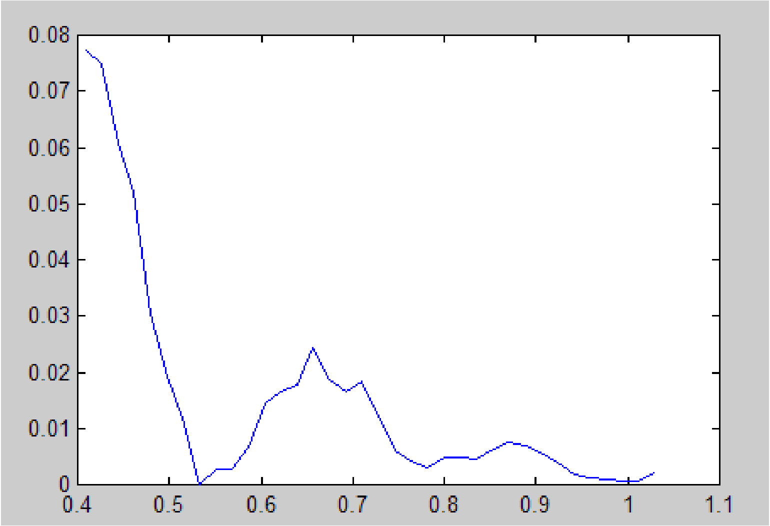
CTF Estimation Algorithm

1. Perform background estimation in 2D using a constrained least squares approach.
2. Constrained least squares can also be used to perform envelope function estimation.

Demo



Estimate background
Using constrained LS

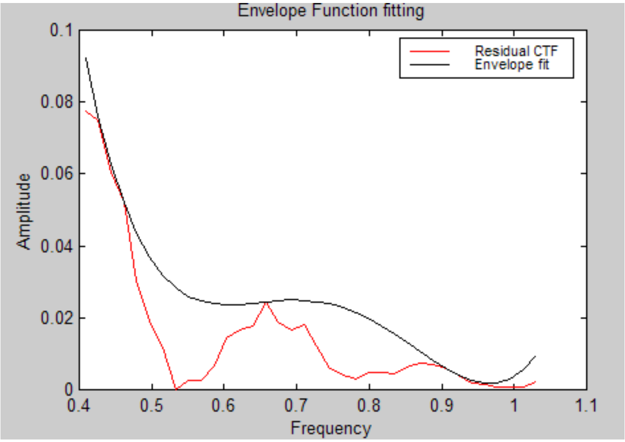


Residual CTF

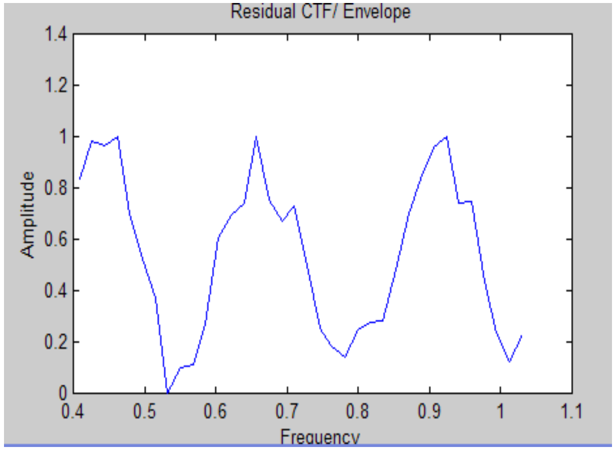
Future Work

CTF Estimation Algorithm

1. Perform background estimation in 2D using a constrained least squares approach.
2. Constrained least squares can also be used to perform envelope function estimation.



Fit Envelope to residual CTF



Residual CTF/ Envelope Fit

Future Work

EM Deconvolution Algorithm

Immediate

1. Algorithm needs to be tested for a number of EM samples, not just helical particles.
2. We need to understand why the algorithm did not perform well at 17.5 A resolution
3. Envelope function needs to incorporated into the problem formulation.

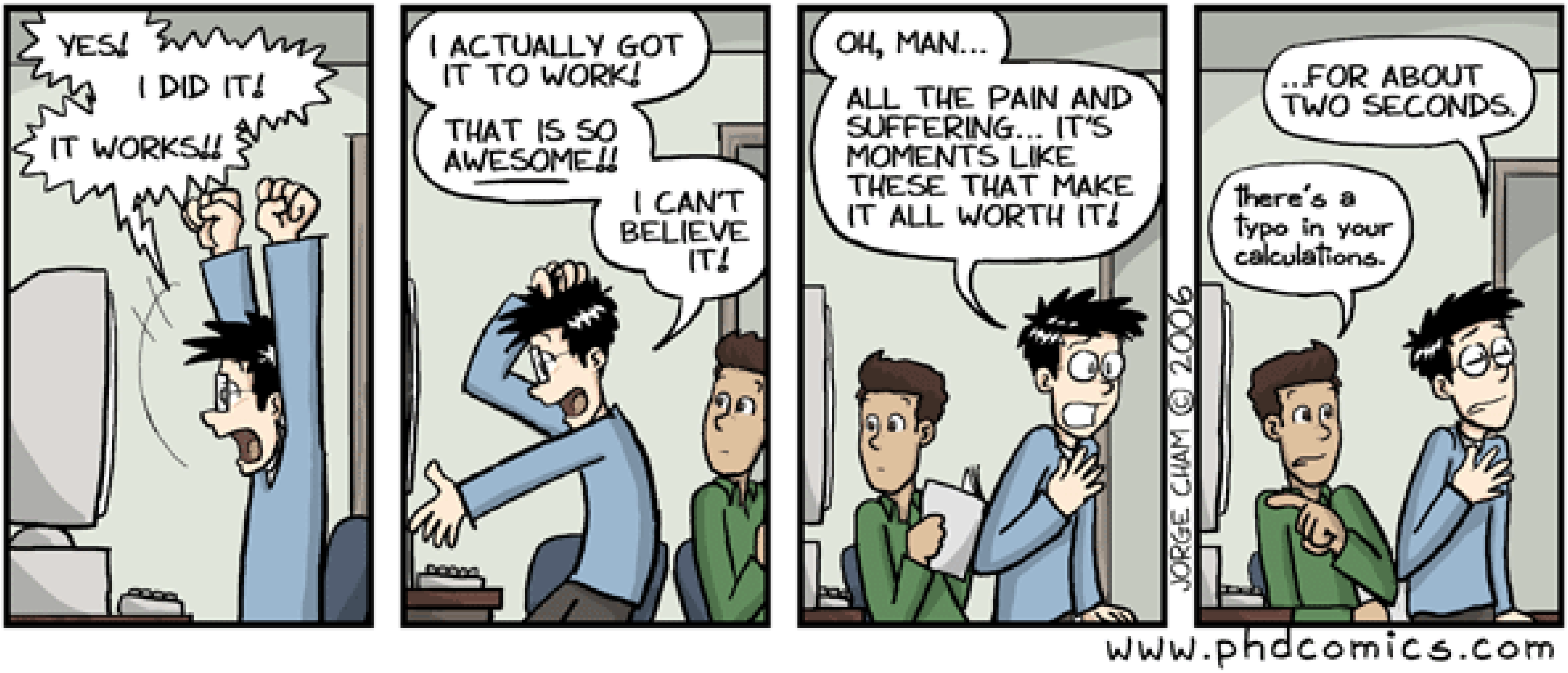
Long term

1. Adapt algorithm for tomographic data. Two possible approaches

   a. Break up image into strips where CTF is assumed constant.
   b. Formulate it as a single 2-D problem with spatially varying PSF. More difficult but elegant.

2. Myopic deconvolution approach of AIDA/MISTRAL.

The End ?


YES! I DID IT! IT WORKS!!
I ACTUALLY GOT IT TO WORK!
THAT IS SO AWESOME!!
I CAN'T BELIEVE IT!
OH, MAN...
ALL THE PAIN AND SUFFERING... IT'S MOMENTS LIKE THESE THAT MAKE IT ALL WORTH IT!
JORGE CHAM © 2006
there's a typo in your calculations.
...FOR ABOUT TWO SECONDS.
www.phdcomics.com