Numerical Methods in TEM
Convolution and Deconvolution

Christoph T. Koch

Max Planck Institut für Metallforschung

http://hrem.mpi-stuttgart.mpg.de/koch/Vorlesung

Applications of Convolution in TEM

- Smoothing of data
- Differentiating of data (e.g. divergence, …)
- Simulate the effect of microscope instabilities (e.g. sample vibrations for images, energy fluctuations in spectra)
- Simulate the effect of detector point spread functions (PSF)
- Simulate the effect of microscope aberrations in HAADF-STEM (based on an oversimplifying approximation)

Applications of Deonvolution in TEM

- Inversion of gradient and divergence operations
- Removal of microscope instabilities (e.g. sample vibrations in images, energy instabilities in spectra)
- Removal of microscope aberrations in HAADF-STEM (assuming that the image is the convolution of probe and object function)
- Removal of plural scattering in EELS
- Removal of source energy spread in EELS

Definition of Convolution

1D

$$f \otimes g = \int_{-\infty}^{\infty} f(r') \cdot g(r - r') dr'$$

2D

$$f \otimes g = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x', y') \cdot g(x - x', y - y') dx' dy'$$

In general, the value of F(**r**)=f(**r**)⊗g(**r**) depends on the values of f(**r**) and g(**r**) for all **r**, i.e. across the whole image

Max-Planck Institut für Metallforschung Universität Stuttgart

Computing the gradient by convolution

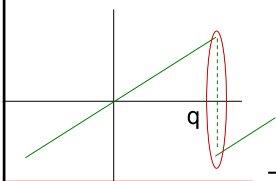
Real space:

$$\frac{df}{dx} = \frac{f_{i+1} - f_i}{\Delta x} = f \otimes D_x^1 \qquad D_x^1 = \begin{pmatrix} 1 & -1 \end{pmatrix}$$

or

$$\frac{df}{dx} = \frac{f_{i+1} - f_{i-1}}{2 \cdot \Delta x} = f \otimes D_x^1 \qquad D_x^1 = \begin{pmatrix} 1 & 0 & -1 \end{pmatrix}$$

Reciprocal space:



Discontinuous edges!

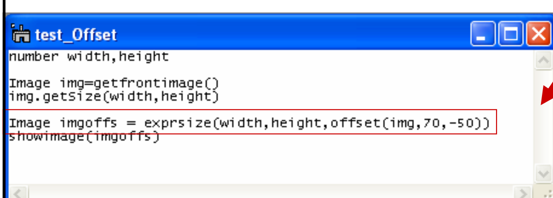
$$\frac{df}{dx} = \frac{d}{dx} \sum F_q \cdot e^{2\pi iqx}$$

$$= \sum F_q \cdot 2\pi iq \cdot e^{2\pi iqx}$$

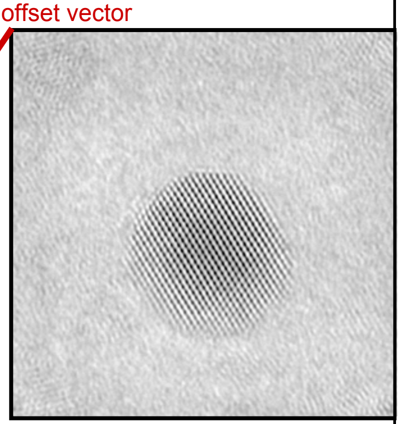
$$\Rightarrow \frac{df}{dx} = FT^{-1}\left\{ 2\pi i \cdot FT\left[f \right] \cdot q \right\}$$

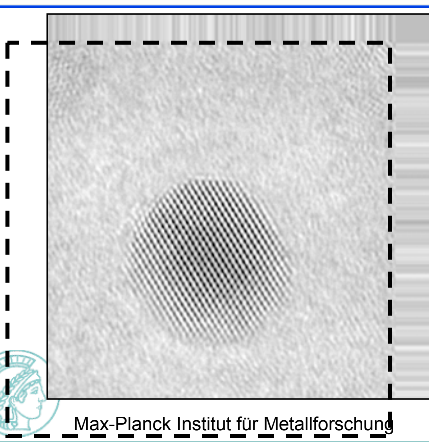
[F_q = Fourier components of f(x)]

The DM function ‘offset(img,dx,dy)’



offset vector

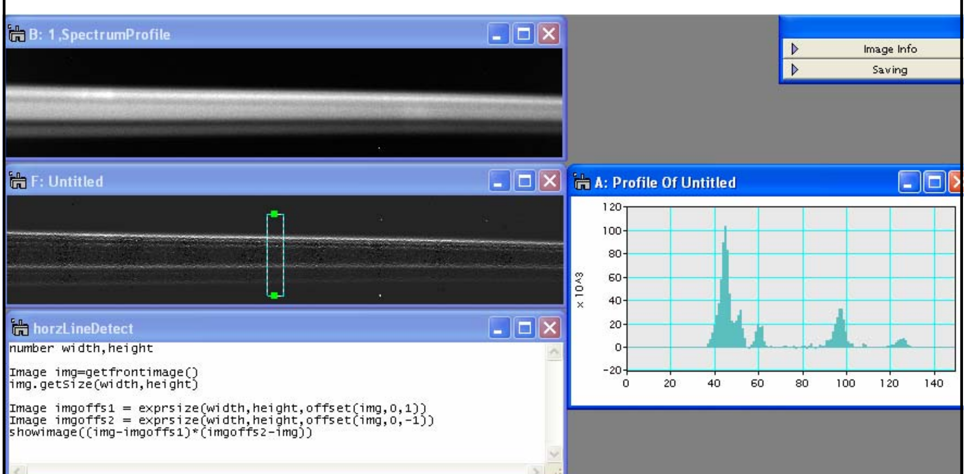




One must provide the command ‘offset’ with the size of the target image bei either:
• create the target image beforehand, or
• use ‘exprsize’

Expand images using ‘offset’

One dimensional derivative: Edge detection



Computing the divergence by convolution

Real space:

$$\Delta f(x,y) = \frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2}$$

$$\Delta f(x,y) = f \otimes D^2_{xy} \qquad \mathbf{D}^2_{xy} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Reciprocal space:

$$\Delta f = \left(\frac{d^2}{dx^2} + \frac{d^2}{dy^2} \right) \sum F_{q_x, q_y} \cdot e^{2\pi i (q_x x + q_y y)}$$

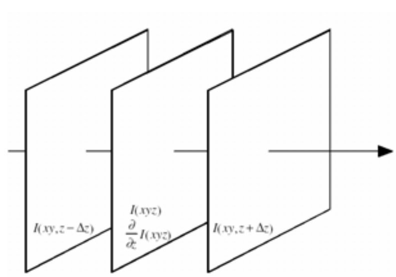
$$= -4\pi^2 \sum \left(q_x^2 + q_y^2 \right) \cdot F_{q_x, q_y} \cdot e^{2\pi i (q_x x + q_y y)}$$

$$\Rightarrow \Delta f = FT^{-1} \left\{ -4\pi^2 \cdot FT[f] \cdot \left(q_x^2 + q_y^2 \right) \right\}$$

Transport of Intensity Equation (TIE)

The wave function satisfies the Schrödinger eqn in free space (Fresnel propagation):

$$\left(2ik \frac{\partial}{\partial z} + \nabla^2_{xy} + 2k^2 \right) \psi(xyz) = 0$$



The phase of the electron wave is then given by:

$$\phi(xyz) = -\frac{2\pi}{\lambda} \nabla^{-2}_{xy} \nabla_{xy} \bullet \left(\frac{1}{I(xyz)} \nabla_{xy} \nabla^{-2}_{xy} \frac{\partial}{\partial z} I(xyz) \right)$$

Inverse Laplace operator

where

$$\psi(xyz) \equiv \sqrt{I(xyz)} \exp\{i\phi(xyz)\} \exp\{i\mathbf{kr}\}$$

Edge detection using the Sobel Filter

(Application: e.g. alignment of EFTEM images with very different contrast)

$$\mathbf{G_x} = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * \mathbf{A} \quad \text{and} \quad \mathbf{G_y} = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \mathbf{A}$$

$$\mathbf{G} = \sqrt{\mathbf{G_x}^2 + \mathbf{G_y}^2}$$

Kernel_x Kernel_y

Intuitive code to compute G_x=Kernel_x*A:

```
for (ix=0;ix<Nx;ix++) {
  for (iy=0;iy<Ny;iy++) {
    for (ix2=-1;ix2<=1;ix2++) {
      for (iy2=-1;iy2<=1;iy2++) {
        SetPixel(Gx,ix,iy,Gx(ix,iy)+Kernel(1+ix2,1+iy2)*A(ix+ix2,iy+iy2));
      }
    }
  }
}
```

computeGx — More efficiently:

```
number width,height
Image img=getfrontimage()
img.getSize(width,height)

Image Gx = CreateFloatImage("Gx",width,height)
Gx = -1*offset(img,1,1) +1*offset(img,-1,1) \
     -2*offset(img,1,0) +2*offset(img,-1,0) \
     -2*offset(img,1,-1)+1*offset(img,-1,-1)

showimage(Gx)
```


The Modulation Transfer Function (MTF)

A sharp image produced by the electron wave on the detector will be smeared by “cross-talk” between the pixels of the detector.

If an electron hits the scintillator (or phosphor screen) above a certain CCD pixel, then the neighboring pixels may also receive a few photons (this is also true for film and imaging plates).

The resulting image is the convolution of the original image with the MTF:

$$I_{\exp}(\vec{r}) = I_{ideal}(\vec{r}) \otimes FT^{-1}\left[MTF(\vec{q})\right]$$
$$= FT^{-1}\left\{FT\left[I_{ideal}(\vec{r})\right] \cdot MTF(\vec{q})\right\}$$

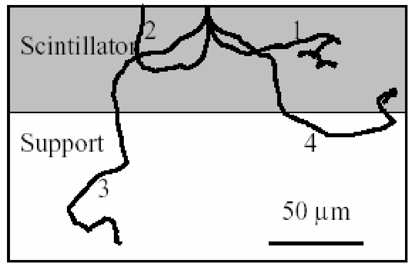
Detector Point Spread Function

1.) Stopped in the scintillator.

2.) Back-scattered from the scintillator.

3.) Stopped in the support.

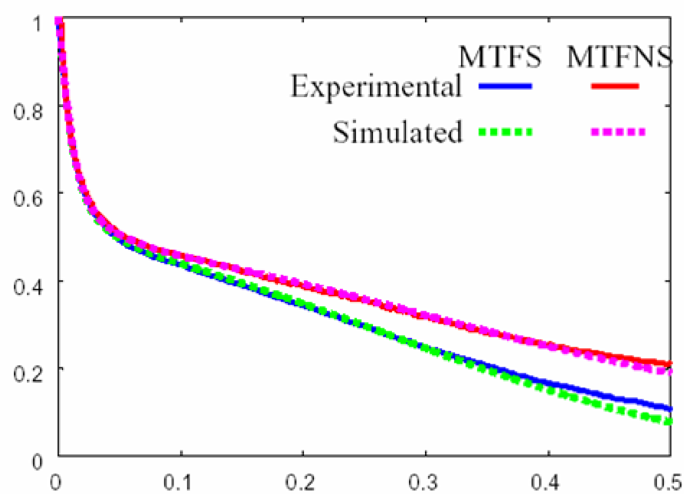
4.) Back-scattered from the support.



Voltage	100 kV	200 kV	300 kV	400 kV
Type 1	82 %	16 %	0.6 %	0 %
Type 2	18 %	20 %	10 %	5 %
Type 3	0 %	48 %	66 %	70 %
Type 4	0 %	16 %	22 %	25 %

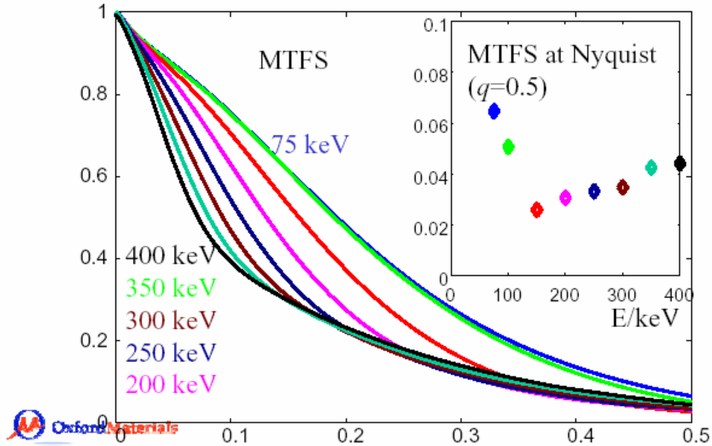
Slide: A. Kirkland

MTFS of a YAG scintillator at 100kV



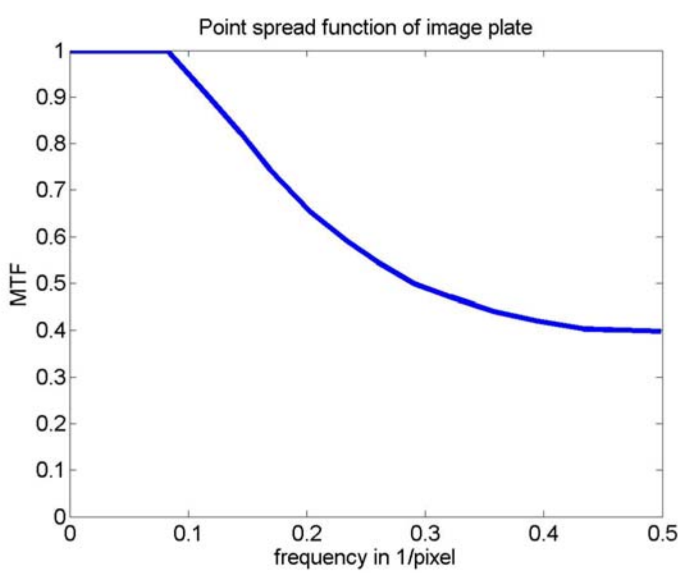
Slide: A. Kirkland

MTFS of a phosphor powder scintillator

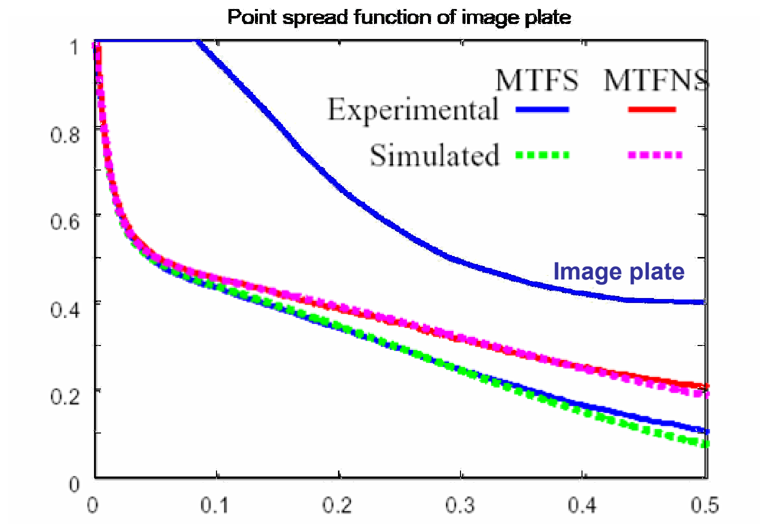


Slide: A. Kirkland

MTF of Image Plate



Comparison Image Plate - CCD



Convolution -> Deconvolution

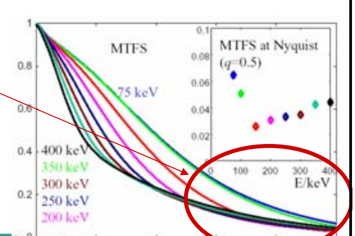
Convolution of an image with the detector MTF (also called point spread function [PSF]):

$$I_{\exp}(\vec{r}) = I_{ideal}(\vec{r}) \otimes FT^{-1}\left[MTF(\vec{q})\right]$$
$$= FT^{-1}\left\{FT\left[I_{ideal}(\vec{r})\right] \cdot MTF(\vec{q})\right\}$$

De-Convolution of an image with the detector MTF:

$$I_{ideal}(\vec{r}) = FT^{-1}\left\{FT\left[I_{\exp}(\vec{r})\right] / MTF(\vec{q})\right\}$$

Problem: At high frequencies the MTF(q) is very small (division by small numbers!) and $I_{exp}(r)$ may be dominated by noise.
=> Noise Amplification!



Avoiding Noise Amplification

Possible solutions to avoid noise amplification are:

1. Impose an upper limit on 1/MTF(q)
2. Lower the upper limit on 1/MTF(q) with increasing q (S/N ratio usually decreases)
3. Let 1/MTF(q) go to zero above a certain resolution q (ideally q should match the resolution present in the image data)
4. Make sure that the deconvolution kernel (e.g. MTF(q)) is smooth. Otherwise this makes errors even worse.
5. Use Richardson-Lucy deconvolution
6. Use Maximum Likelihood deconvolution

1. Upper limit on Inverse of Convolution Kernel

Use the DM command ‘tert’:

```
Image img = getFrontImage()
Image iMTF_lim = tert(1/MTF>thresh, thresh, 1/MTF)
```

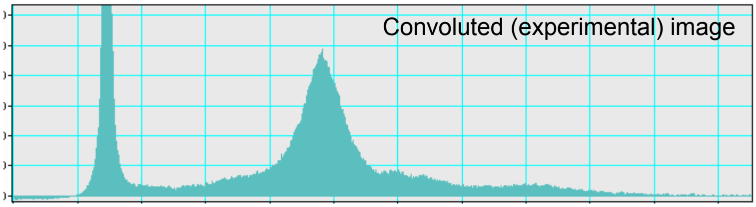
- 1/MTF>thresh → condition
- thresh → condition true
- 1/MTF → condition false

```
Image deconv = realifft(realfft(img)*iMTF_lim)
```

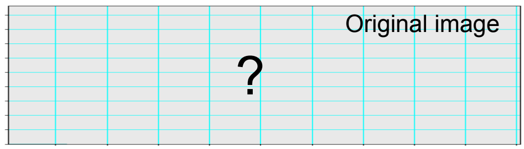
<u>for options 2 & 3</u>: replace thresh with a Gaussian image
(see script “GaussEdgeSmoothingInterp.s”)

Max-Planck Institut für Metallforschung
Universität Stuttgart

Convolution Kernel must be sensible

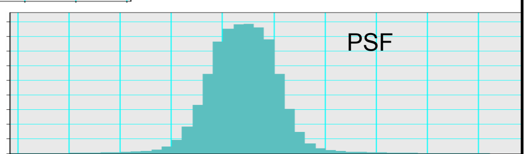


=

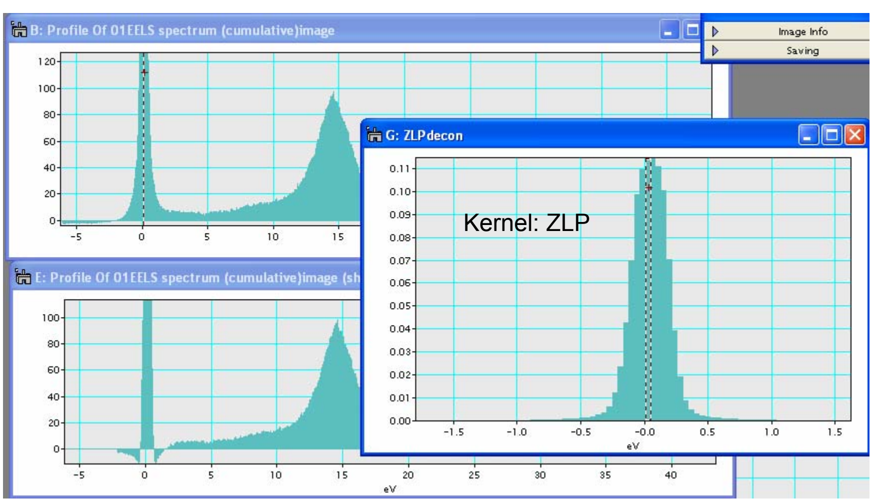


The PSF cannot be broader than the sharpest feature in your image!

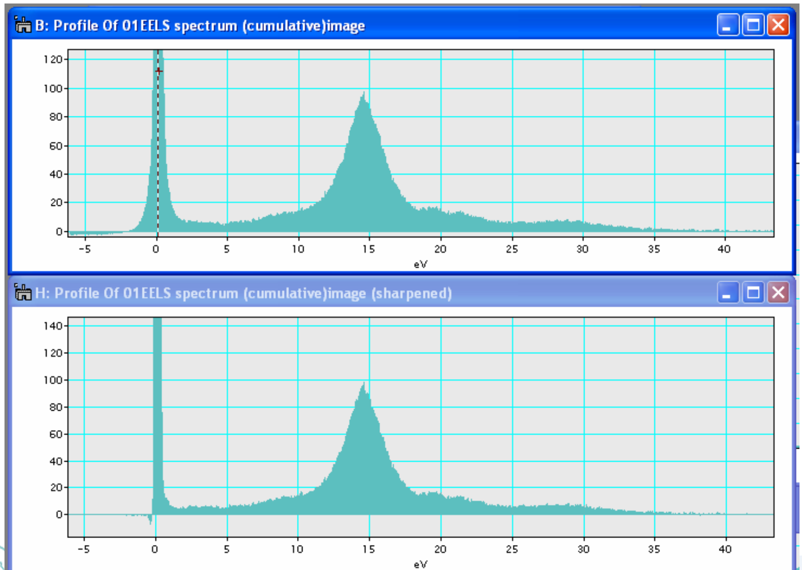
⊗



DM-EELS Function: "Sharpen Spectrum"



‘Sharpen Spectrum’ with ZLP extracted from spectrum

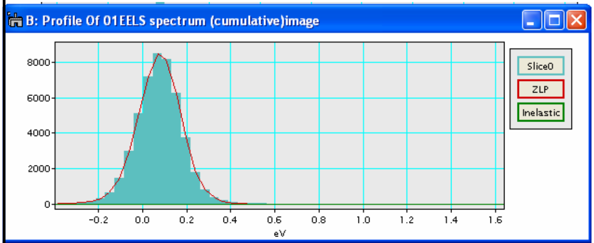


How does ‘Sharpen Spectrum’ work?

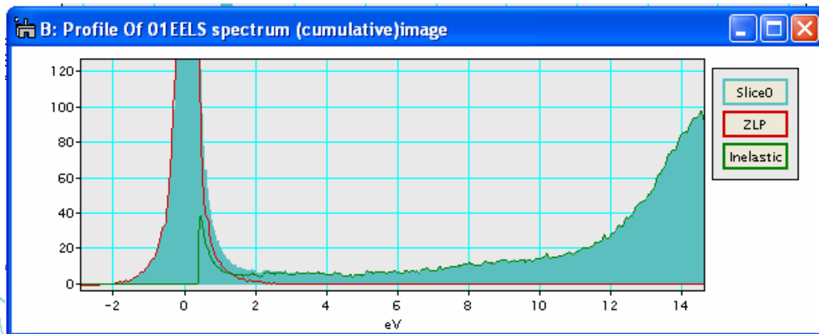
1. Fit a Gaussian to the narrow central portion (within 90% of the maximum, mimimum of 3 channels) of the zero-loss peak [ZLP] (=Convolution kernel)
2. Replace that portion of the ZLP with a δ-function of equal area.
3. Subtract the fitted Gaussian from the original data and set negative pixels of the resulting “Gaussian-subtracted ZLP” to zero.
4. Place the total Intensity difference between the ZLP and the “Gaussian-subtracted ZLP” in a single pixel at the position of the ZLP maximum (=> delta-function).
5. Normalize this modified ZLP to 1 and makesure the delta-function is in pixel(0,0).

Max-Planck Institut für Metallforschung
Universität Stuttgart

How does the ZLP-extraction from the spectrum work



1. A symmetric ZLP is assumed. The right hand side of the ZLP below ¼ of its height is replaced by its left hand side.
2. Negative counts in the "inelastic spectrum" (original – ZLP) are set to zero.



EELS Multiple Scattering Deconvolution

Assuming independent scattering events the intensity in an experimental EELS spectrum can be simulated by the expression

$$I_{\exp}(E) = ZLP(E) \otimes \left[\frac{t}{\lambda} I_{theor}(E) + \frac{1}{2!} \left(\frac{t}{\lambda} I_{theor}(E) \right) \otimes \left(\frac{t}{\lambda} I_{theor}(E) \right) + \ldots \right]$$

$$= ZLP(E) \otimes FT^{-1} \left[\exp\left\{ \frac{t}{\lambda} FT\left[I_{theor}(E) \right] \right\} \right]$$

$$= FT^{-1} \left\{ FT\left[ZLP(E) \right] \cdot \exp\left(\frac{t}{\lambda} FT\left[I_{theor}(E) \right] \right) \right\}$$

This means, in order to extract the true spectrum $I_{theor}(E)$ from an experimental spectrum one must first deconvolute by the ZLP as precisely as possible.

EELS Multiple Scattering Deconvolution (2)

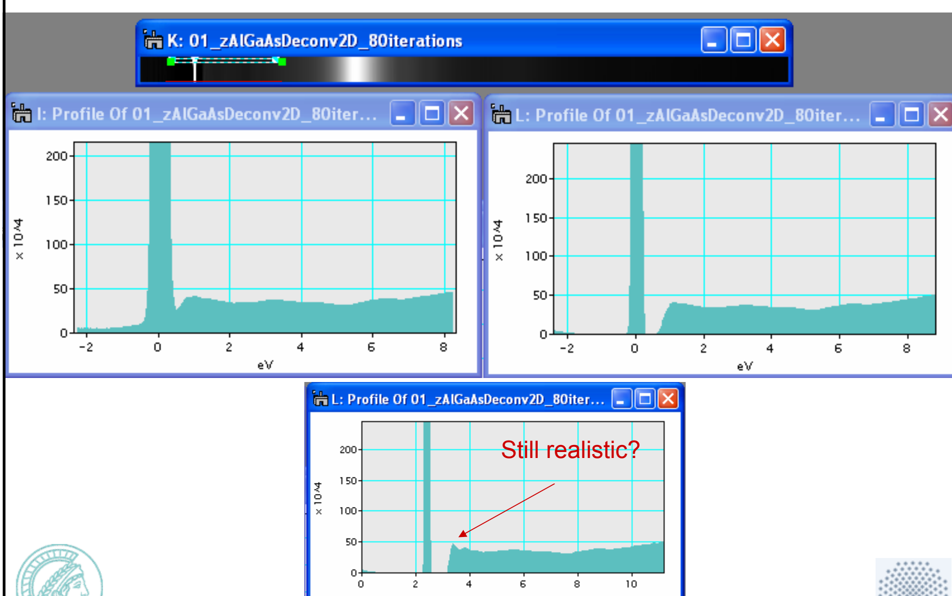
Inverting the previous expression, one can extract the single scattered spectrum from the experimental data according to

$$\frac{t}{\lambda} I_{ss}(E) = FT^{-1}\left\{ \ln\left(\frac{FT\left[I_{theor}(E)\right]}{FT\left[ZLP(E)\right]} \right) \right\}$$

How does an incomplete deconvolution of the ZLP
(or deconvolution by a smoothed ZLP) affect $I_{ss}(E)$?

It mainly affects the resolution of $I_{ss}(E)$. Peak positions and –heights will hardly be affected.

Result of Richardson-Lucy Deconvolution



Maximum Entropy Principle?

2nd law of Thermodynamics:

"The entropy of an isolated system not in equilibrium will tend to increase over time, approaching a maximum value at equilibrium."

Result: In the absence of any constraints, a gas, for example, will distribute evenly.

If constraints are present, then a condition which maximizes entropy and satisfies the constraints will be reached.

Maximum Entropy Image Deconvolution

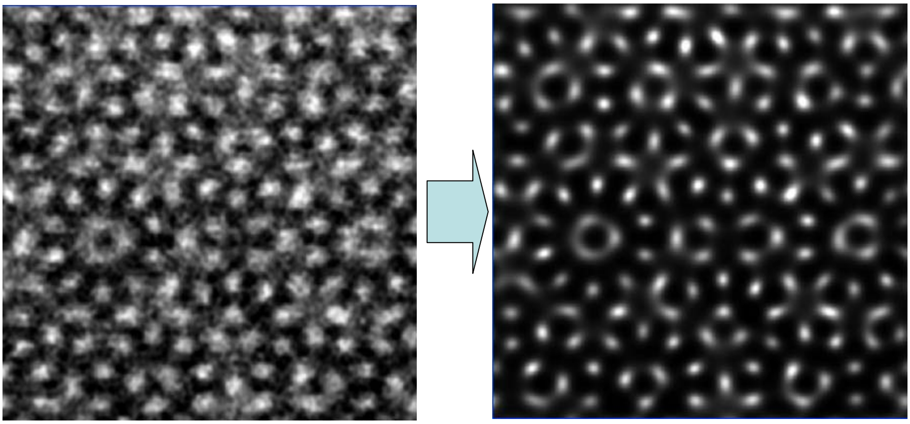
An image that is sought which satisfies the following constraints:

1. Its difference to the original image must be within a certain limit
2. No negative counts occur (unphysical)
3. The noise statistics agree with Gaussian or Poisson statistics
4. The proper MTF / PSF (e.g. in HAADF-STEM) is considered.
5. … other sensible linear constraints …

(Non-linear constraints are also possible, but then the solution is not easily found [see, e.g. Focal Series Reconstruction])

Max-Planck Institut für Metallforschung Universität Stuttgart

Deconvolution HAADF imaging



(from the documentation of the DeConvHAADF plugin by HREMResearch)

Iterative Deconvolution algorithms

Maximum Entropy Algorithm:

$$Q(f^k) = -\sum_i f_i^k \log(f_i^k / f_i^{k-1}) - \lambda \sum_i \frac{(h_i - (g * f^k)_i)^2}{\sigma_i^2}, \textit{ where } f_i^0 = 1$$

Richardson-Lucy [RL] Algorithm:

$$\psi^{k+1}(\xi) = \psi^k(\xi) \int \frac{P(x,\xi)\phi(x)}{\int P(x,\xi)\psi^k(\xi)d\xi} dx \quad \text{using} \quad \psi^0(\xi) \equiv 1$$

$$\phi(x) = \int P(x-\xi)\psi(\xi)d\xi$$

$$\psi(\xi) = \int Q(\xi - x)\phi(x)dx$$

The RL-Algorithm converges to the Maximum Entropy solution

Max-Planck Institut für Metallforschung Universität Stuttgart