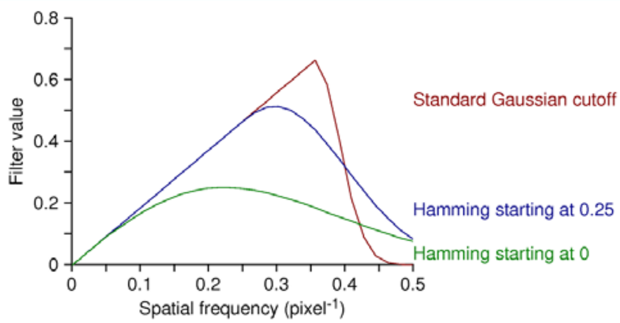
Denoising Cryotomograms with IMOD

Reasons to Denoise Cryotomograms

- Easier segmentation of features
- Presentation
- Particle picking for subvolume averaging
  - The high-resolution information that you hope to bring out with averaging is buried in noise AND makes it hard to see lower-resolution features: “poor contrast”
  - You will often need a separate lower-noise tomogram for particle picking, not suitable for averaging

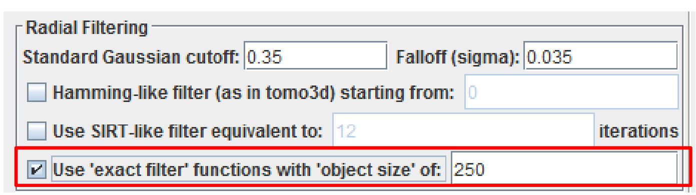
Filters in Tomogram Generation

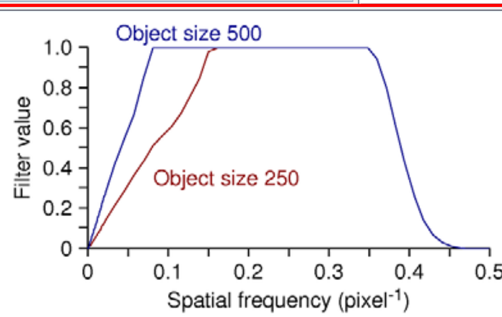
- In addition to SIRT and the SIRT-like filter, there are two other options for increasing contrast by filtering the projection lines before backprojection
- The Hamming-like filter is an alternative to the standard Gaussian filter for attenuating high frequencies



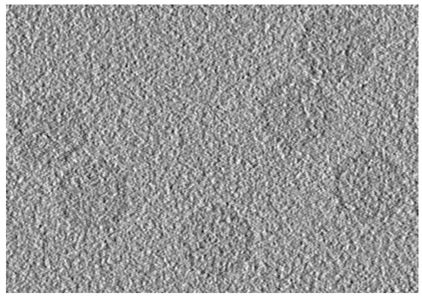
Filters in Tomogram Generation

- The "exact filters" of Harauz and van Heel, like the SIRT-like filter, are an alternative to the linear ramp for accentuating low frequencies
  - Visible in Advanced mode
  - The bigger the "object size", the greater the weighting of low frequencies



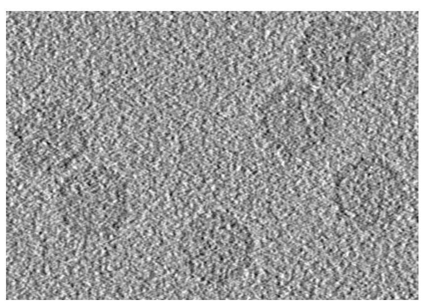


Standard Gaussian Cutoff



Default radius and sigma

Hamming-like Filter

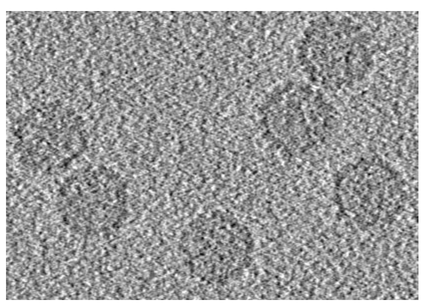


Starting at zero

Exact Filters

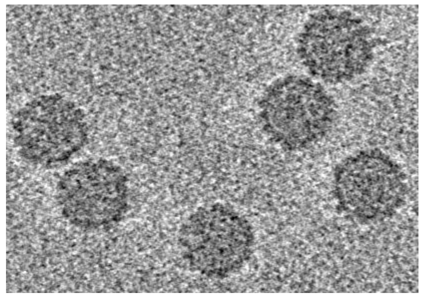
Object size 250

Exact Filters



Object size 500

SIRT-like Filter



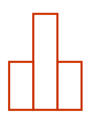
Equivalent to 12 iterations

Smoothing Kernel Filters

- Kernel filtering involves replacing every pixel with a weighted sum of a block of pixels
- The standard smoothing filter has a simple 3x3 kernel

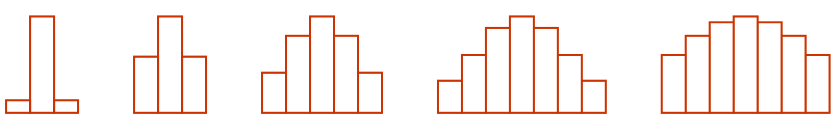
1	2	1
2	4	2
1	2	1

Top view
(all divided by 16)



Side view through
central pixel

- For more flexibility, weights can be set from a real-space Gaussian with a chosen sigma. The standard filter corresponds to σ = 0.85.
- The kernel is 3x3 pixels for σ ≤ 1, 5x5 for 1 < σ ≤ 2, or 7x7 for σ > 2

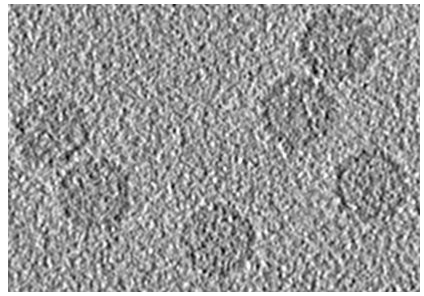


σ = 0.5 σ = 1.0 σ = 1.5 σ = 2.0 σ = 3.0

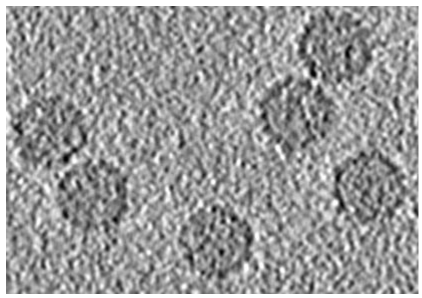
- For even more filtering than σ = 3, you would have to iterate – but σ = 3 already smooths a lot!

Kernel Smoothing Filter

Default



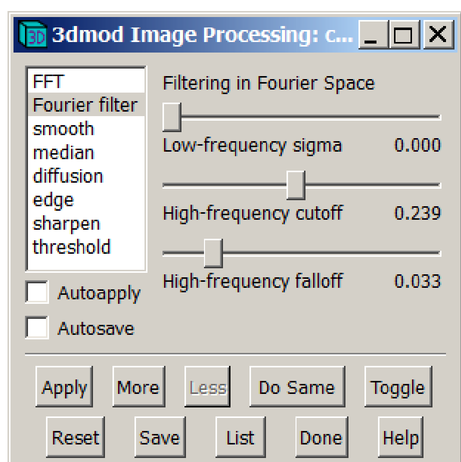
Sigma 1.7



Sigma 3.0

How to Apply Simple Filters

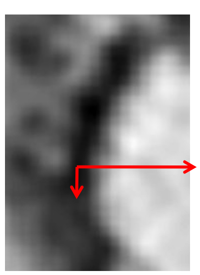
- For simple filtering, the procedure is to test a filter in the 3dmod image processing window then run it on the whole volume in another program
  - Smoothing: clip smooth –l sigma –n iterations in_file out_file
- 3dmod now runs the command for you, with the menu entry File-Process File



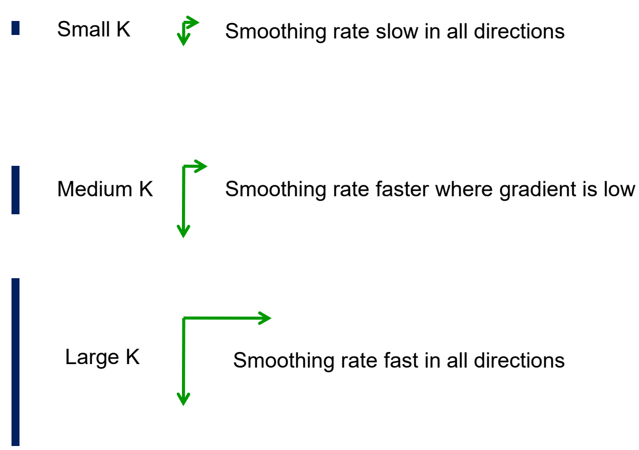
Nonlinear Anisotropic Diffusion (NAD)

- What does that mean?
  - Diffusion: iterative kernel smoothing that diffuses densities between neighboring pixels
  - Anisotropic: not the same in all directions, but less in directions with bigger gradients in density
  - Nonlinear: amount of diffusion in a direction is controlled in nonlinear way by gradient relative to a threshold
- Is supposed to preserve/enhance edges by smoothing along edges and not across them
- Requires two parameters to be selected
  - Number of iterations is intuitive: the more, the smoother
  - K value sets the threshold for blocking diffusion: nonintuitive, has to be found by trying different values and picking the one that gives desired result

The Role of the K Parameter in NAD



Gradients at one point in image



Operational Points on NAD

- NAD should be run through Etomo
  - Don't be confused by low-quality NAD in 3dmod and clip: the real NAD program is nad_eed_3d from Frangakis and Hegerl
  - It requires 36x as much memory as voxels, so it is not practical to run it on a whole tomogram at once
  - It is very time-consuming, so parameter settings need to be worked out on a small test volume
  - The interface in Etomo helps you excise a test volume, test different K values and different numbers of iterations, and run the process on the whole volume in chunks

Running NAD

- In Etomo, select it from the Front Page, or select File – New Nonlinear Anisotropic Diffusion
- A test volume of 200 x 200 x 24 pixels will run reasonably quickly; it can be thinner (down to 16 pixels) if you need to see larger area
- Start with wide range of K values
  - K is relative to the gradients (thus intensities) in the file; this means integer data will need higher K values than byte data for same effect
  - For byte data, try 0.4,1.6,6.4,26,102 or 0.4,1.0,2.5,6.4,16,40,100
  - K tests are run in parallel to the extent possible
- Pick a K range that gives the kind of filtering desired and rerun at finer intervals in the range
  - You will see K values where intensities "plateau" over many pixels
  - Lower K values preserve edges while reducing this effect, but may not allow much smoothing
  - Higher K values give essentially isotropic smoothing
- Vary the iterations to pick the amount of smoothing
- When running on whole volume, memory = 36 x voxels in chunk so default chunk is 14 M voxels.