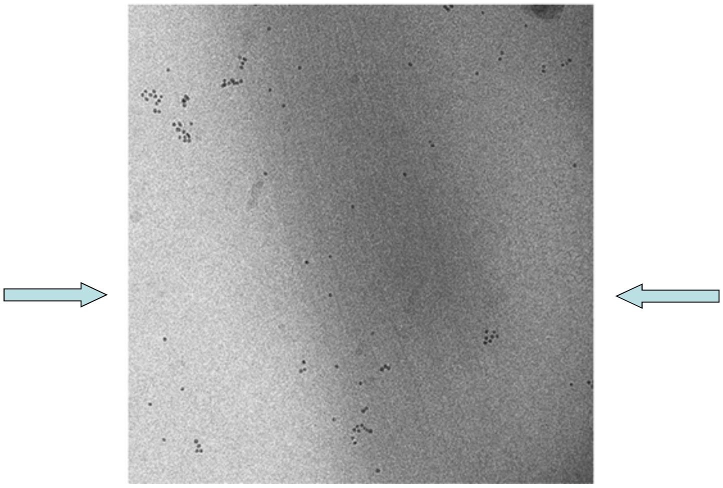
	June 26 (Mon)
8:00	Registration/Security
	Visitor's Center
9:00	Welcome
	Principles and General Aspects of Electron Tomography
	David
10:00	
	Coffee Break
11:00	Alignment and Other Challenges in Reconstructing Cryotomograms with IMOD
	David
12:00	Lunch
1:00	
	Walkthrough Building a Cryo Tomogram With and Without Fiducials
	Cindi
2:00	
3:00	
	Coffee Break
	Batch Processing of Tomograms with IMOD
4:00	David
	Reconstruct a Cryo Tomogram with BatchRunTomo
5:00	
6:00	Mixer/Dinner
	Wah Chiu Keynote Address
7:00	
8:00	
9:00	

Principles and General Aspects of Electron Tomography

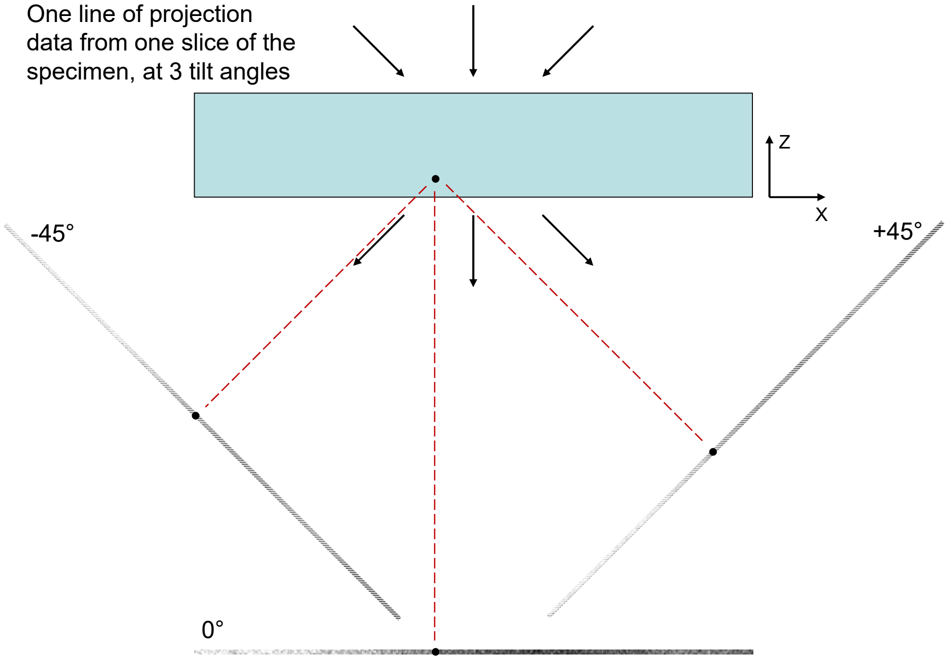
Aligned Cryo-Tilt Series of Flagella of a Giardia Cell

±60º at 2º increments



Horizontal line of projection data

One line of projection data from one slice of the specimen, at 3 tilt angles

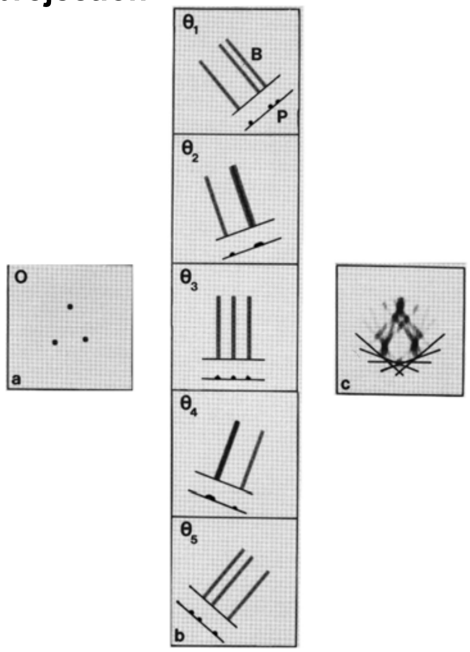


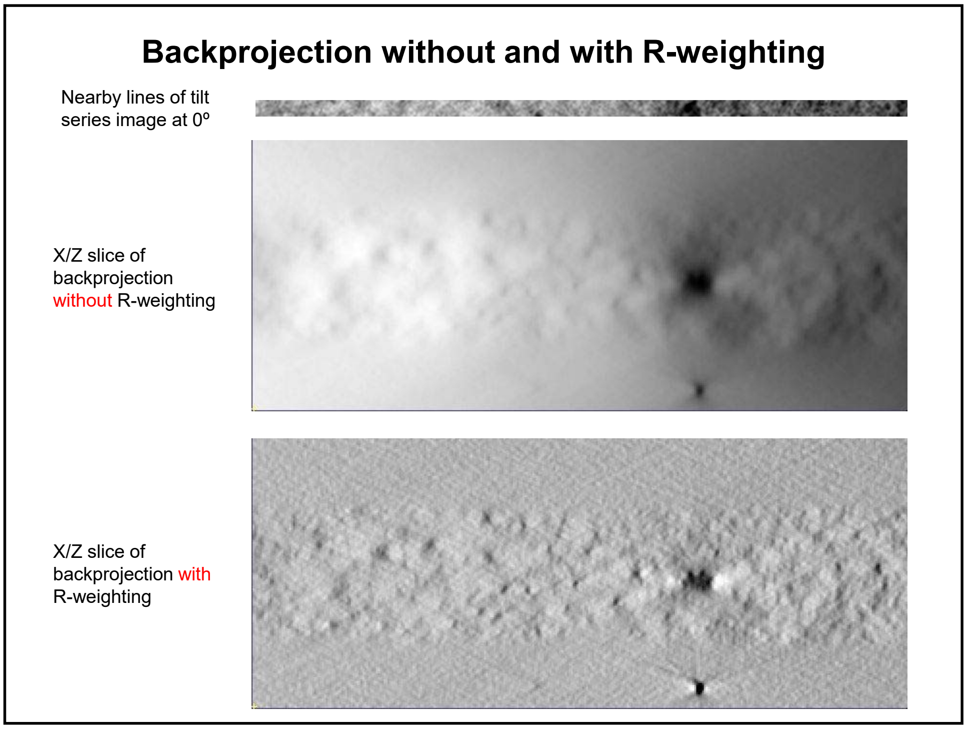
Backprojection

P - Projection of 3 points onto a line at various tilt angles

B – Backprojection of projection data into reconstruction volume

(From Radermacher, 1992)



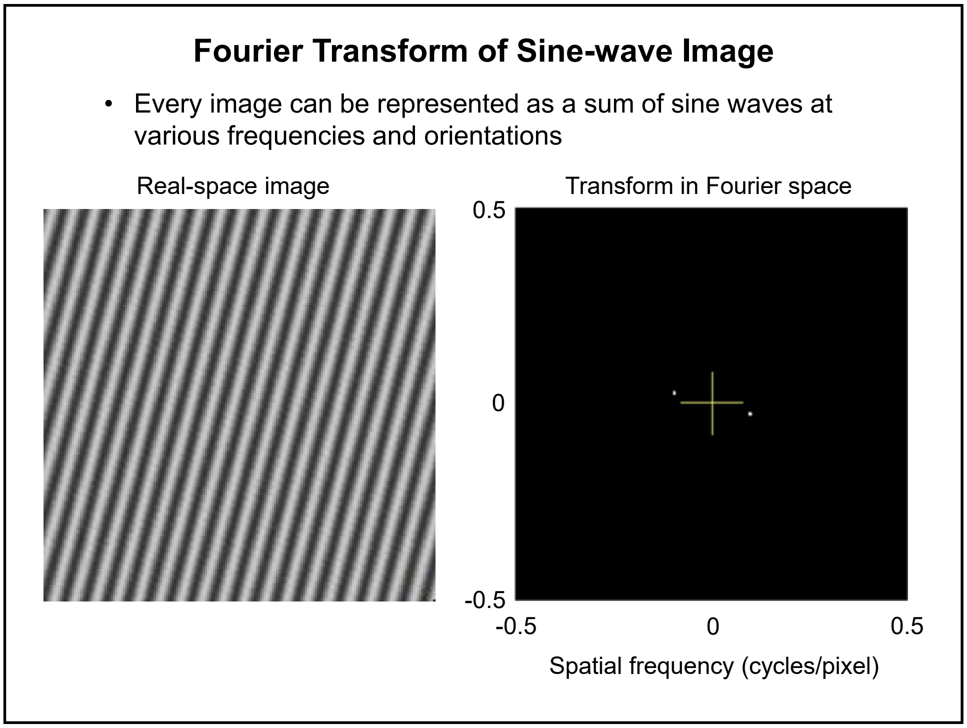


Backprojection without and with R-weighting

Nearby lines of tilt series image at 0º

X/Z slice of backprojection without R-weighting

X/Z slice of backprojection with R-weighting

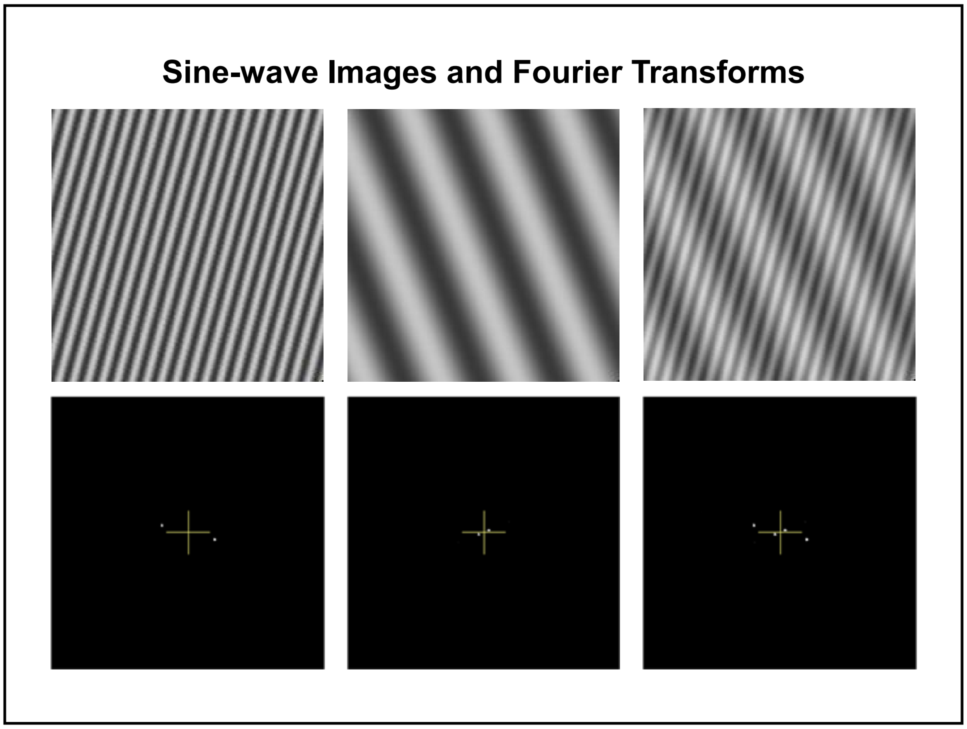


Fourier Transform of Sine-wave Image

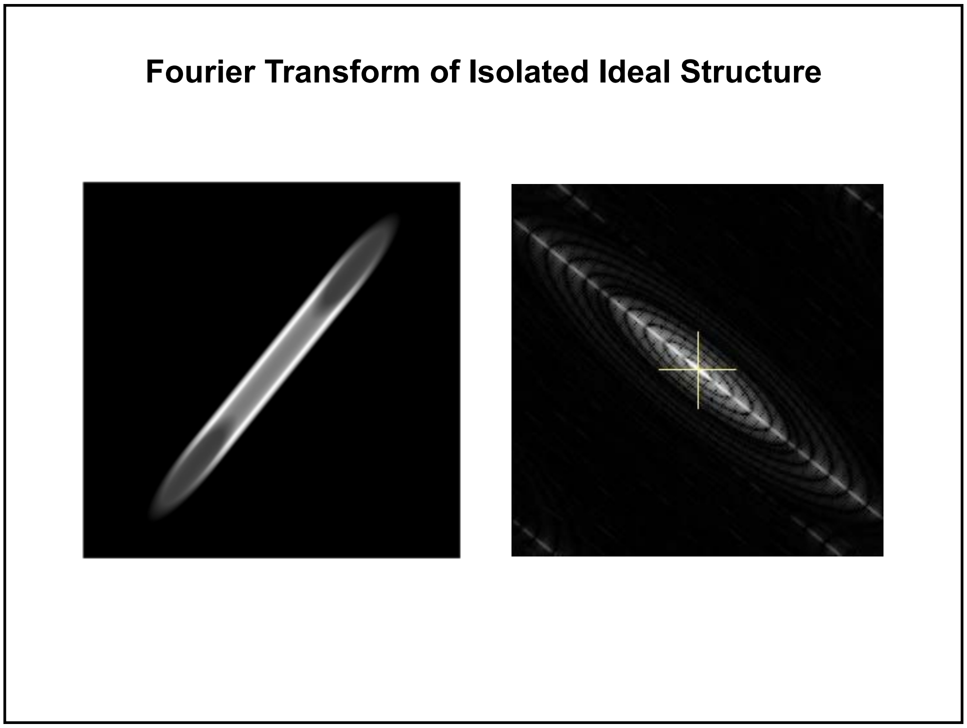
- Every image can be represented as a sum of sine waves at various frequencies and orientations

Real-space image

Transform in Fourier space

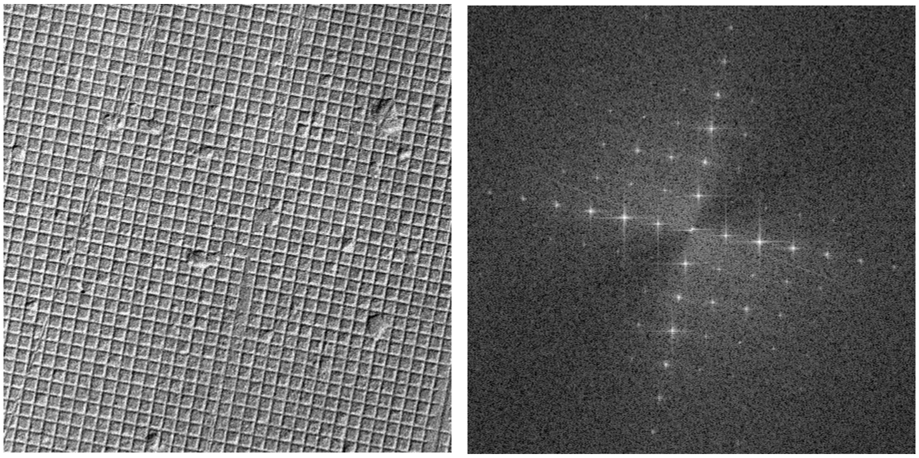


Sine-wave Images and Fourier Transforms

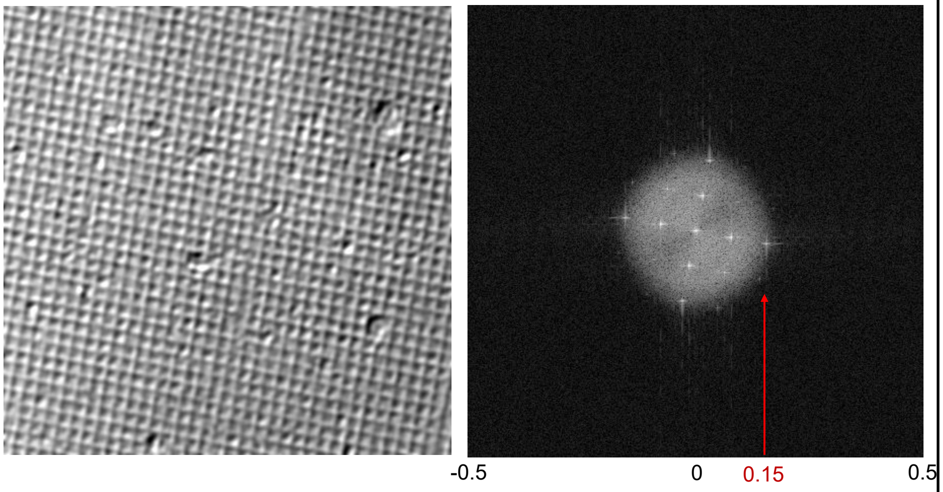


Fourier Transform of Isolated Ideal Structure

Fourier Transform of Periodic Structure

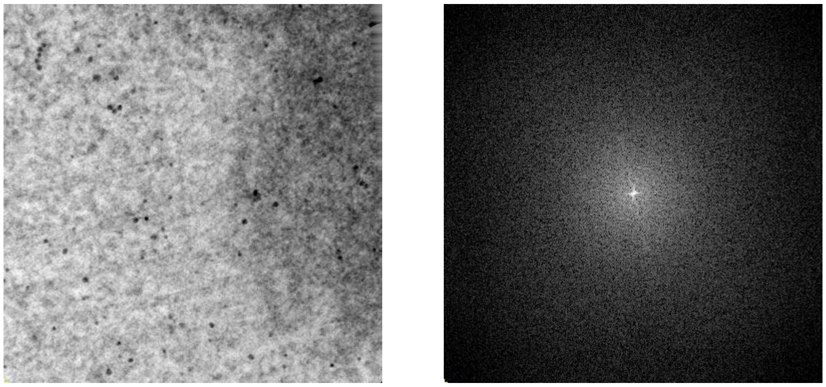


“Low-pass” Filtering Removes High Spatial Frequencies (Fine Details)

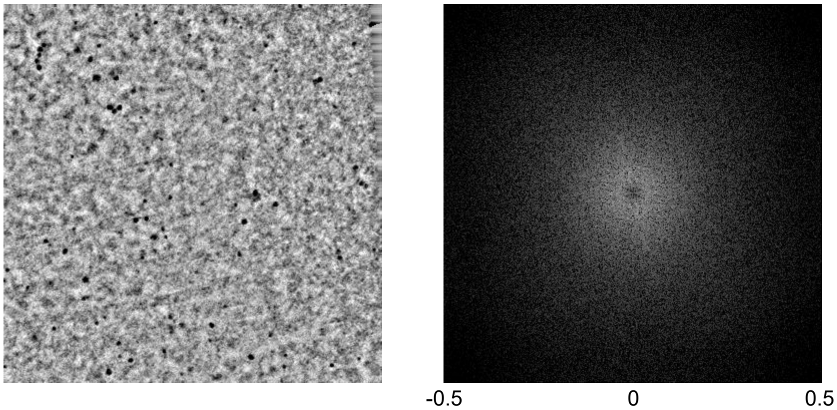


Frequencies above ~0.15 cycles/pixel filtered out

Fourier Transform of Image without Periodic Structure



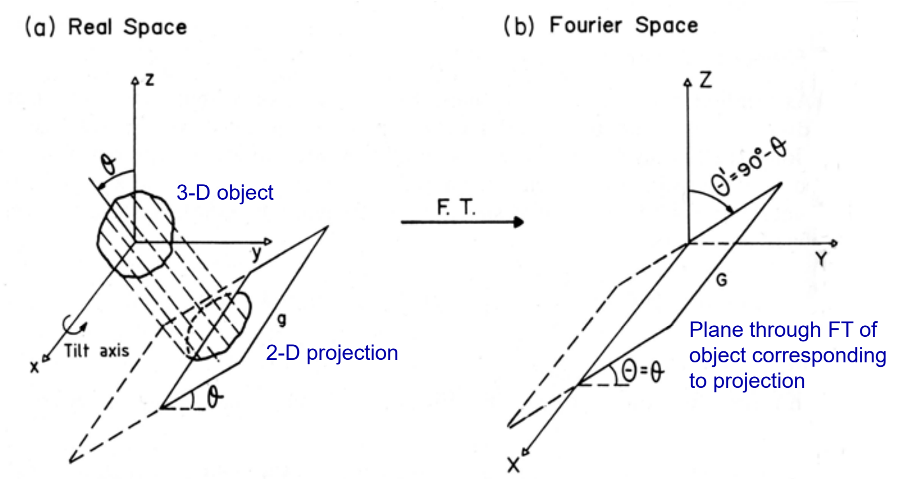
“High-pass” Filtering Removes Low Spatial Frequencies (Large-scale Intensity Changes)



Frequencies below ~0.03 cycles/pixel filtered out

The Central Section Theorem

The 2-D Fourier transform of a projection image is a central plane through the 3-D Fourier transform of the object



3-D object

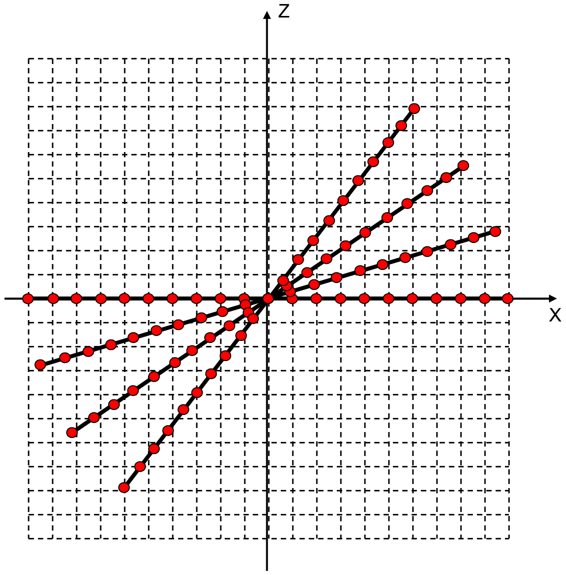
2-D projection

Plane through FT of object corresponding to projection

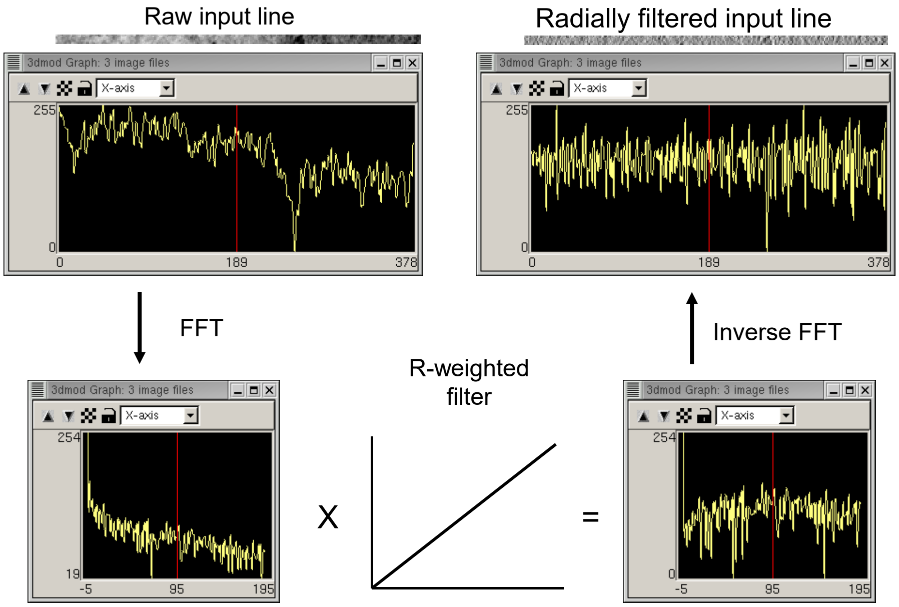
(From Carazo, 1992)

Problems Revealed by Projection Data in Fourier Space

- Sample points from projection lines do not fall on rectangular lattice -> interpolation needed for inverse Fourier method
- Fourier space is oversampled near the origin -> R-weighting needed in backprojection
- Samples are too sparse at high frequencies -> resolution is limited by number of projections
- Samples are nonexistent above highest angle -> missing wedge of data



Radial Filtering of Projection Line before Backprojection



Raw input line

Radially filtered input line

FFT

Inverse FFT

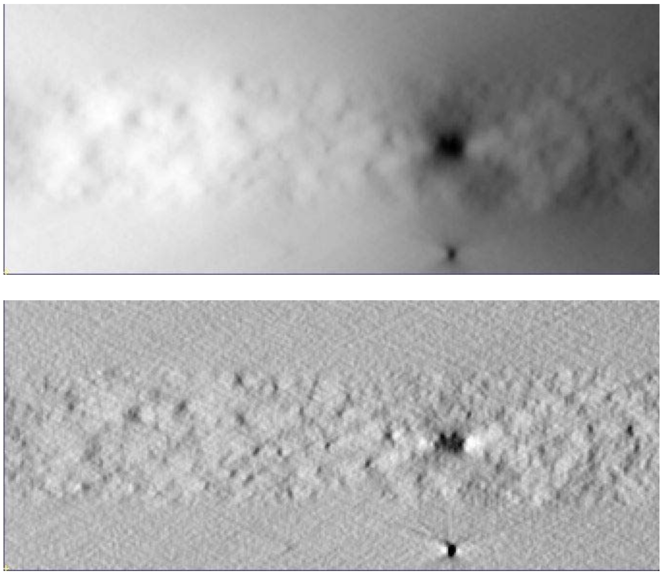
R-weighted filter

X

=

3dmod Graph: 3 image files
X-axis
255
0
0
189
378
3dmod Graph: 3 image files
X-axis
255
0
0
189
378
3dmod Graph: 3 image files
X-axis
254
19
-5
95
195
3dmod Graph: 3 image files
X-axis
254
0
-5
95
195

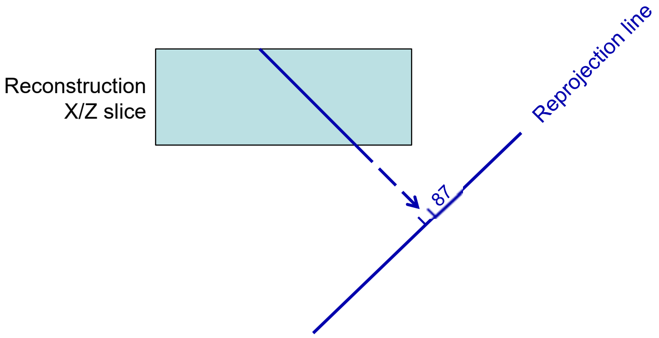
Backprojection Without and With R-weighting



R-weighting keeps low frequencies from dominating

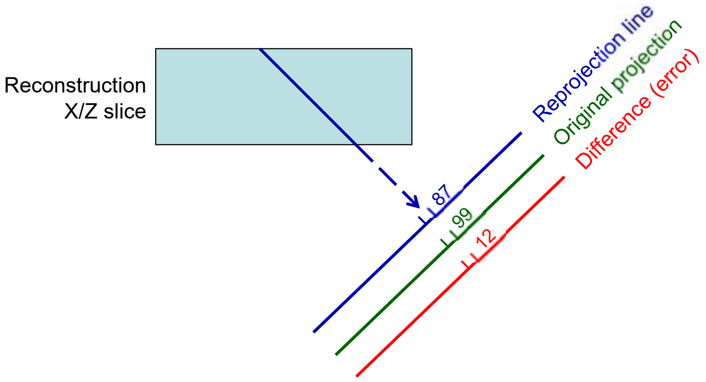
Iterative Reconstruction

- In an iterative method, each iteration involves:
  - Reprojecting from the current estimate of the tomogram (i.e., adding up the densities along each ray line at an given angle)



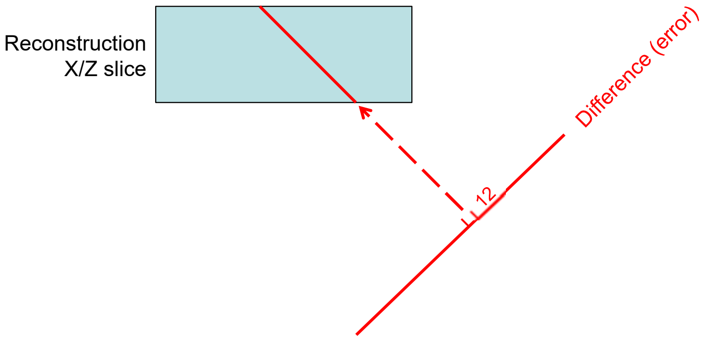
Iterative Reconstruction

- In an iterative method, each iteration involves:
  - Reprojecting from the current estimate of the tomogram (i.e., adding up the densities along each ray line at an given angle)
  - Taking difference between reprojection and original data at each pixel



Iterative Reconstruction

- In an iterative method, each iteration involves:
  - Reprojecting from the current estimate of the tomogram (i.e., adding up the densities along each ray line at an given angle)
  - Taking difference between reprojection and original data at each pixel
  - Adjusting tomogram by distributing difference for each projection pixel into the pixels along the ray through reconstruction



Iterative Reconstruction

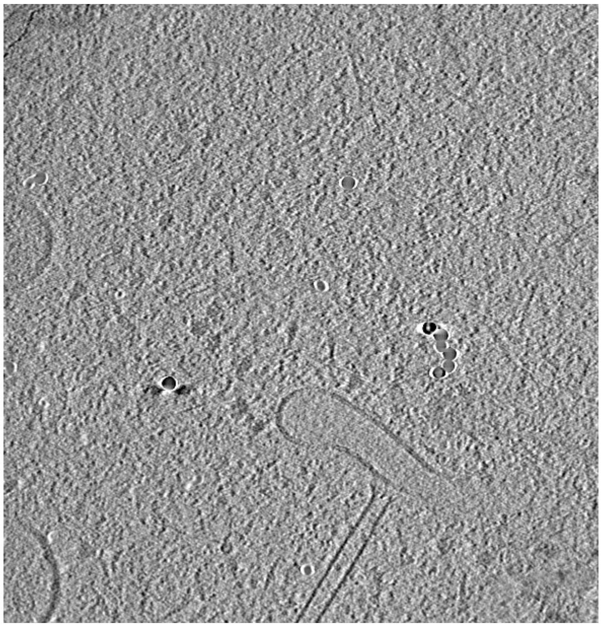
- In an iterative method, each iteration involves:
  - Reprojecting from the current estimate of the tomogram (i.e., adding up the densities along each ray line at an given angle)
  - Taking difference between reprojection and original data at each pixel
  - Adjusting tomogram by distributing difference for each projection pixel into the pixels along the ray through reconstruction
- In ART (Algebraic Reconstruction Technique) this operation is done one angle at a time
- In SIRT (Simultaneous Iterative Reconstruction Technique) the reprojections are computed for all angles first, then the tomogram is adjusted by all the differences
  - Requires more iterations than ART but is more resistant to noise

Iterative Reconstruction

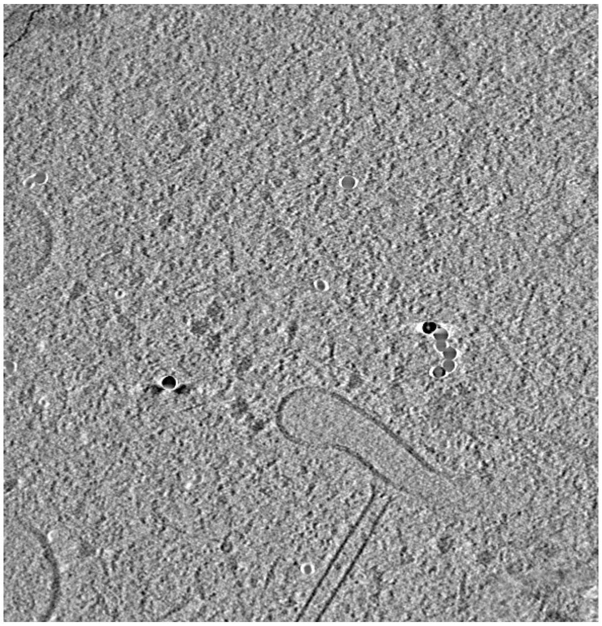
- In an iterative method, each iteration involves:
  - Reprojecting from the current estimate of the tomogram (i.e., adding up the densities along each ray line at an given angle)
  - Taking difference between reprojection and original data at each pixel
  - Adjusting tomogram by distributing difference for each projection pixel into the pixels along the ray through reconstruction
- In ART (Algebraic Reconstruction Technique) this operation is done one angle at a time
- In SIRT (Simultaneous Iterative Reconstruction Technique) the reprojections are computed for all angles first, then the tomogram is adjusted by all the differences
  - Requires more iterations than ART but is more resistant to noise
- The starting reconstruction is generally an unweighted backprojection, so it starts dominated by low frequencies, and higher frequencies are added in through the iterations

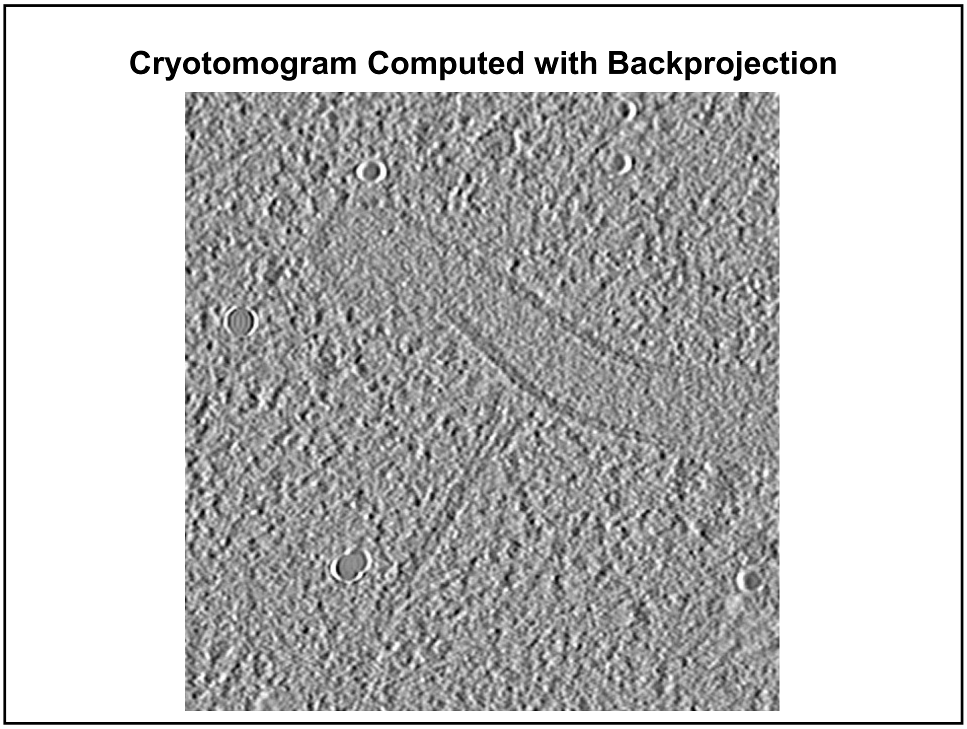
Cryotomogram Computed with Backprojection

Cryotomogram Computed with SIRT (16 iterations)

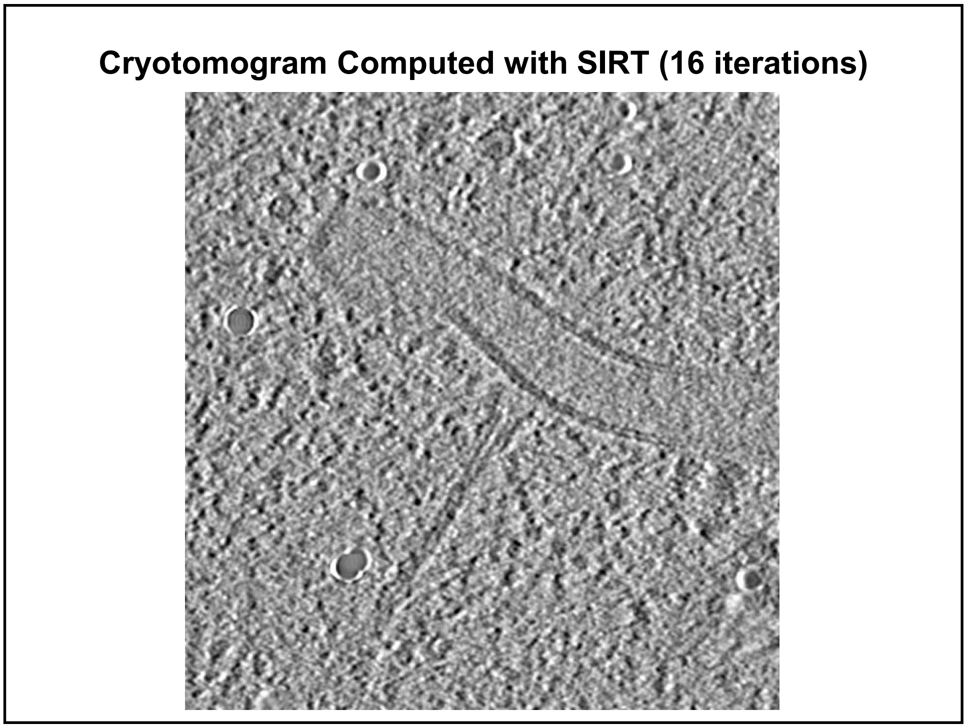


Cryotomogram Computed with SIRT (8 iterations)



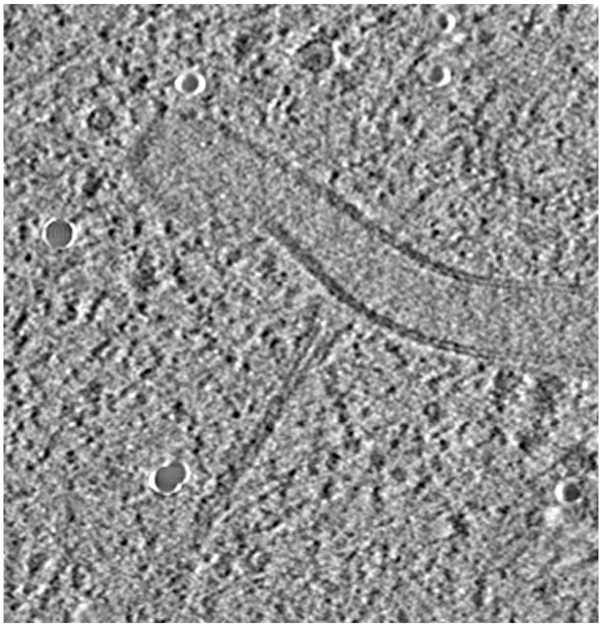


Cryotomogram Computed with Backprojection



Cryotomogram Computed with SIRT (16 iterations)

Cryotomogram Computed with SIRT (8 iterations)



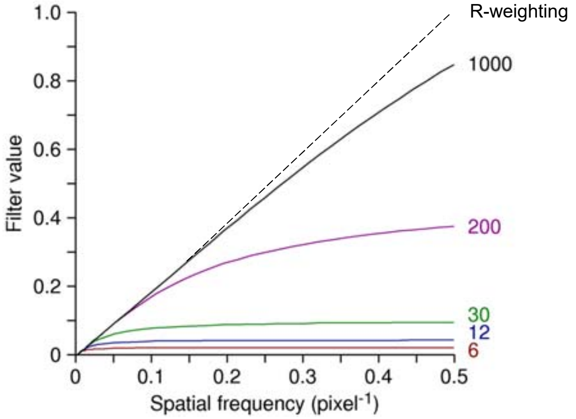
SIRT Is Just an Expensive Filter

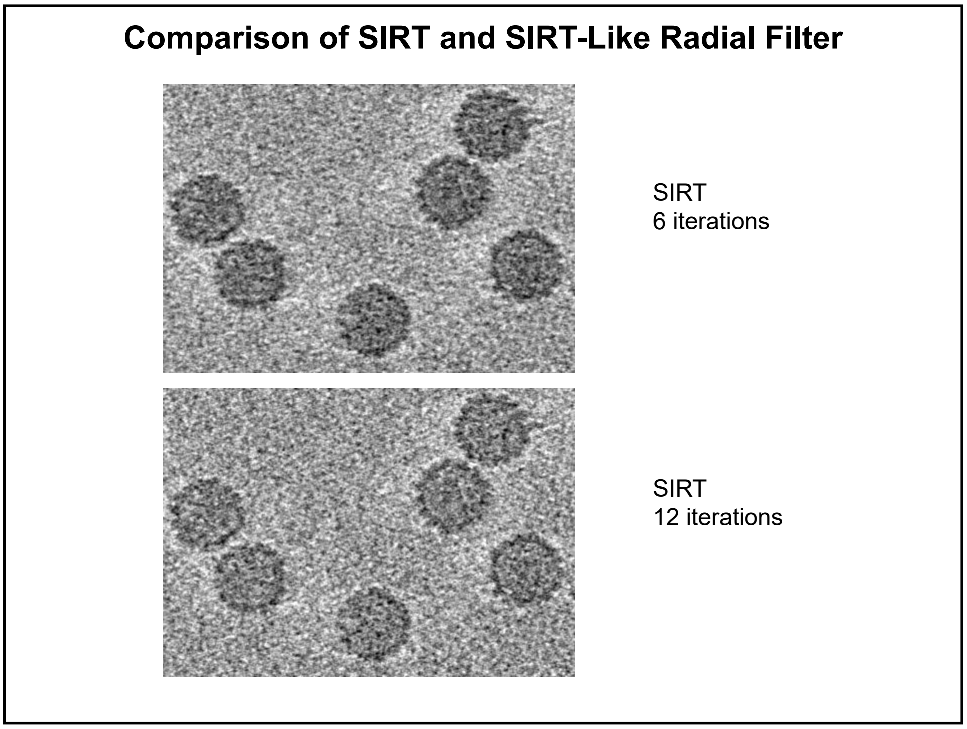
- N iterations of SIRT is mathematically equivalent to doing back-projection with a radial weighting function proportional to

  $f * (1 – (1 – a/f)^N)$ instead of f

  – Zeng, G.L., 2012, A filtered backprojection algorithm with characteristics of the iterative landweber algorithm, Med. Phys. 39: 603-607

  – The constant “a” and an adjusted “N” value are determined by matching output to that from SIRT

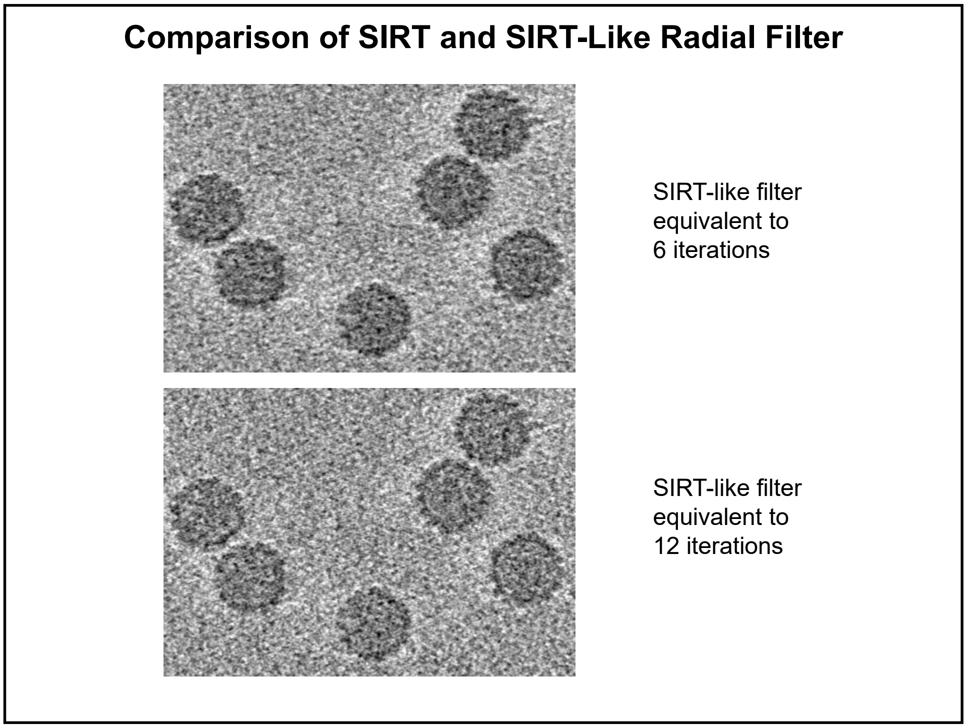




Comparison of SIRT and SIRT-Like Radial Filter

SIRT
6 iterations

SIRT
12 iterations



Comparison of SIRT and SIRT-Like Radial Filter

SIRT-like filter
equivalent to
6 iterations

SIRT-like filter
equivalent to
12 iterations

Factors Limiting Tomogram Resolution

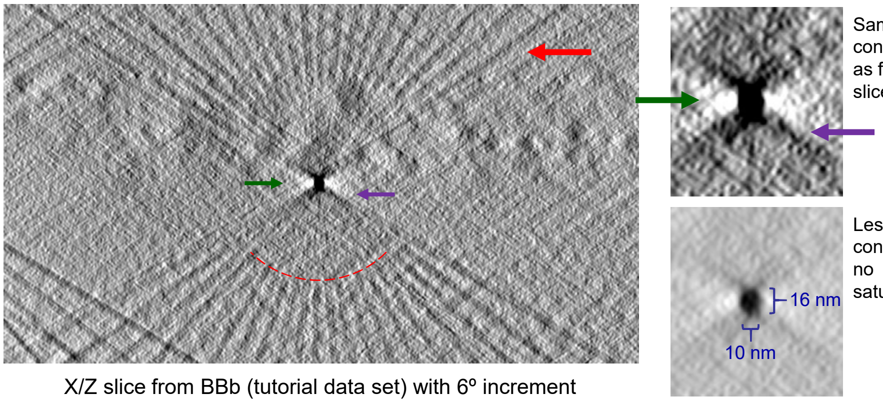
Resolution here refers to the ability to resolve features within the tomogram reliably; this is independent of the resolution that can be achieved by averaging information above the tomogram resolution.

Factors are:

- Number of projections relative to thickness of material
- Density of material within the volume
- Signal-to-noise ratio of input images, determined by electron dose and efficiency of camera
- Resolution of imaging system (microscope and camera)
- Quality of alignment of data entering into backprojection

The Point Spread Function from Single-Axis Tilting

- Elongation in Z from limited tilt range (1.6 x for ±60º range)
- Fringes (overshoots) in X from limited tilt range
- Artefactual rays at each tilt angle starting at a distance that depends on tilt increment
  - These start closer and are more prominent at higher increment
  - Stronger rays start right at the density for the terminal angles

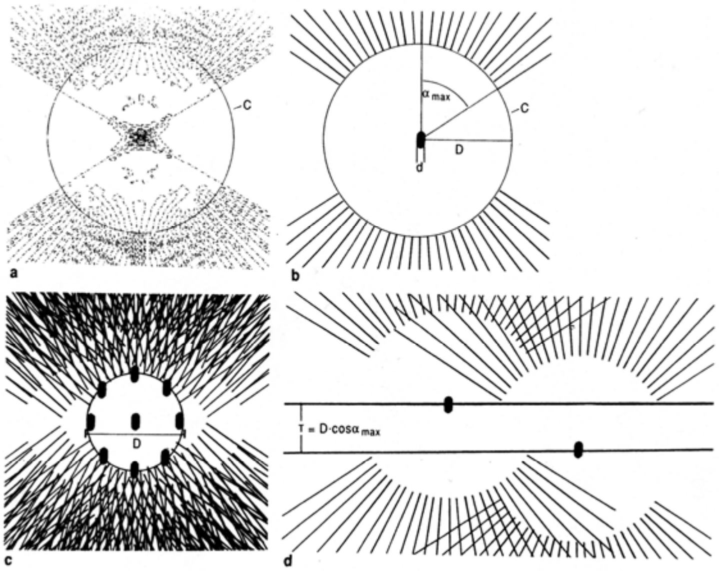


Same contrast as full slice

Less contrast – no saturation

X/Z slice from BBb (tutorial data set) with 6º increment

The Point Spread Function and its Effects on Resolution



(From Radermacher, 1992)

The Crowther Resolution Formula

(Crowther, DeRosier, and Klug 1970)

1. For D = diameter of reconstructed volume
   $\Delta\theta$ = tilt increment (radians)
   resolution $d = D\,\Delta\theta$

2. For $\Delta\theta$ = tilt increment (degrees)
   f = resolution in frequency (reciprocal space) units
   $f = 57.3 / (D\,\Delta\theta)$

3. For n = number of views
   θ_{max} = maximum tilt angle
   $f = 28.5\,n / (D\,\theta_{max})$

4. For $\theta_{max} = \pm 60^{\circ}$
   $f = 0.48\,n / D$

Tomograms Often Beat the Formulas

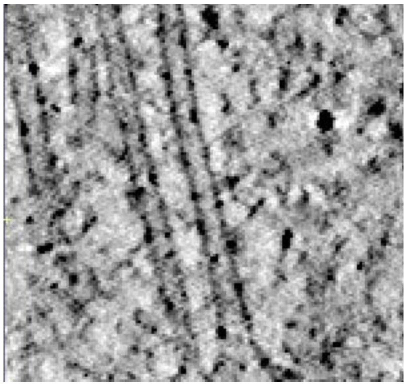
- Stained and especially frozen-hydrated material may be sparse – the “object” of diameter D is not the whole specimen but structural components in it
- The artefactual rays are not of paramount importance at relatively fine tilt intervals – they are just another source of noise

Factors Limiting Tomogram Resolution

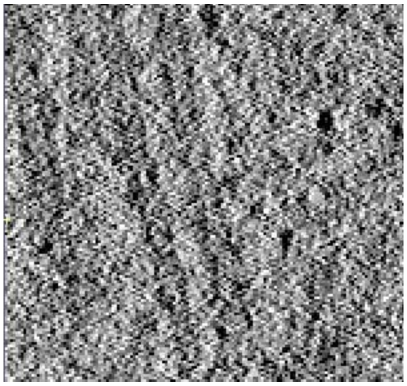
- Number of projections relative to thickness of material
- Density of material within the volume
- Signal-to-noise ratio of input images, determined by electron dose and efficiency of camera

Effect of Noise on Resolution

There is no averaging of high frequency data in the reconstruction process: the SNR of the input data determines the SNR of the reconstruction



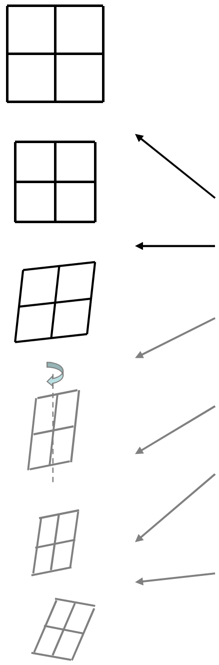
Reconstruction from original data



Noise added equivalent to 100 electrons/pixel

Factors Limiting Tomogram Resolution

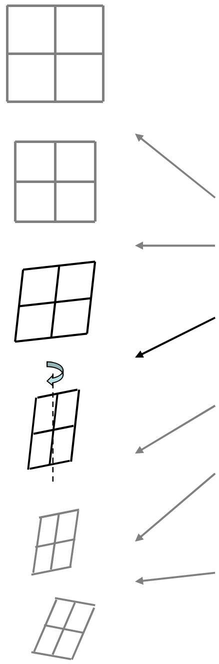
- Number of projections relative to thickness of material
- Density of material within the volume
- Signal-to-noise ratio of input images, determined by electron dose and efficiency of camera
- Resolution of imaging system (microscope and camera)
- Quality of alignment of data entering into backprojection



Tiltalign’s Model for Fiducial Positions

Fiducial positions in every projection view are derived from a single set of positions in 3D by the following changes or operations:

1. The specimen shrinks uniformly in 3D
2. The specimen shrinks along one axis in the plane of the specimen
3. The specimen is tilted around the Y axis, which need not be perpendicular to the beam axis, and the tilt axis could change with tilt direction
4. An image is projected with a magnification that may vary among the views
5. The image is rotated during projection (the tilt axis is not vertical on the camera) and may also stretch along an axis during projection
6. The image is shifted due to variations in specimen position

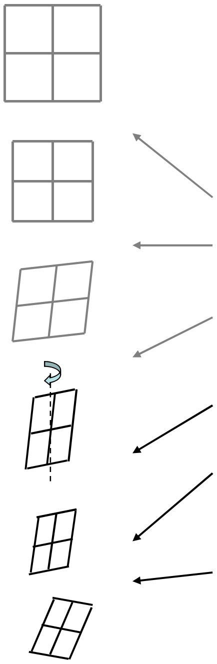


Tiltalign’s Model for Fiducial Positions

Fiducial positions in every projection view are derived from a single set of positions in 3D by the following changes or operations:

1. The specimen shrinks uniformly in 3D
2. The specimen shrinks along one axis in the plane of the specimen
3. The specimen is tilted around the Y axis, which need not be perpendicular to the beam axis, and the tilt axis could change with tilt direction
4. An image is projected with a magnification that may vary among the views
5. The image is rotated during projection (the tilt axis is not vertical on the camera) and may also stretch along an axis during projection
6. The image is shifted due to variations in specimen position

Tiltalign’s Model for Fiducial Positions



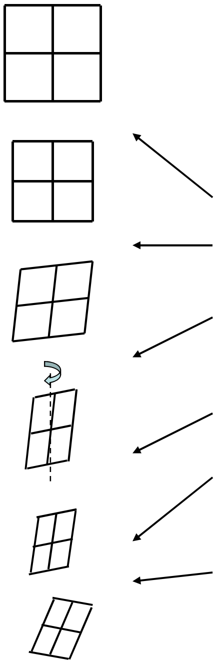
Fiducial positions in every projection view are derived from a single set of positions in 3D by the following changes or operations:

1. The specimen shrinks uniformly in 3D
2. The specimen shrinks along one axis in the plane of the specimen
3. The specimen is tilted around the Y axis, which need not be perpendicular to the beam axis, and the tilt axis could change with tilt direction
4. An image is projected with a magnification that may vary among the views
5. The image is rotated during projection (the tilt axis is not vertical on the camera) and may also stretch along an axis during projection
6. The image is shifted due to variations in specimen position

The Tiltalign Variables

- The specimen changes and imaging operations are expressed in terms of 5 variables that Tiltalign can solve for at each tilt:
  1. **Mag**: a uniform change in specimen size or microscope magnification
  2. **Tilt**: the tilt angle
  3. **Rotation**: the rotation of the tilt axis from the vertical
  4. **X-stretch (Dmag)**: a shrinkage/stretch along the X-axis in the plane of the specimen
  5. **Skew**: a change in the angle between X and Y axes
- X-stretch and Skew together represent a linear shrinkage along an arbitrary axis (distortion)
- Tiltalign can also solve for three variables that are the same for all views:
  1. **Beam tilt**: the angle between the tilt axis and the perpendicular to the beam axis
  2. **Projection skew**: a change in the angle between X and Y axes resulting from stretch during projection
  3. **X-axis tilt**: a tilt around the X axis between two halves of a bidirectional tilt series

Tiltalign’s Model for Fiducial Positions

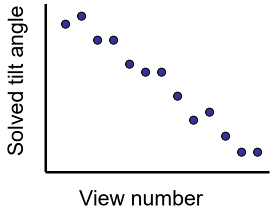


Fiducial positions in every projection view are derived from a single set of positions in 3D by the following changes or operations:

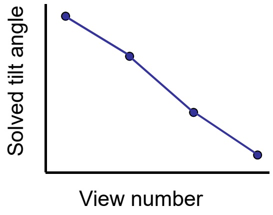
1. The specimen shrinks uniformly in 3D (**Mag**)
2. The specimen shrinks along one axis in the plane of the specimen (**X-stretch** and **Skew**)
3. The specimen is tilted around the Y axis, which need not be perpendicular to the beam axis (**Tilt**, **Beam tilt**), and the tilt axis could change with tilt direction (**X-axis tilt**)
4. An image is projected with a magnification that may vary among the views (**Mag**)
5. The image is rotated during projection (the tilt axis is not vertical on the camera) and may also stretch along an axis during projection (**Rotation**, **Projection skew**)
6. The image is shifted due to variations in specimen position

Grouping of Views to Reduce Variables

When grouping is used, a variable is constrained to change linearly over a set of contiguous views, or to have the same value for all of the views in the set.

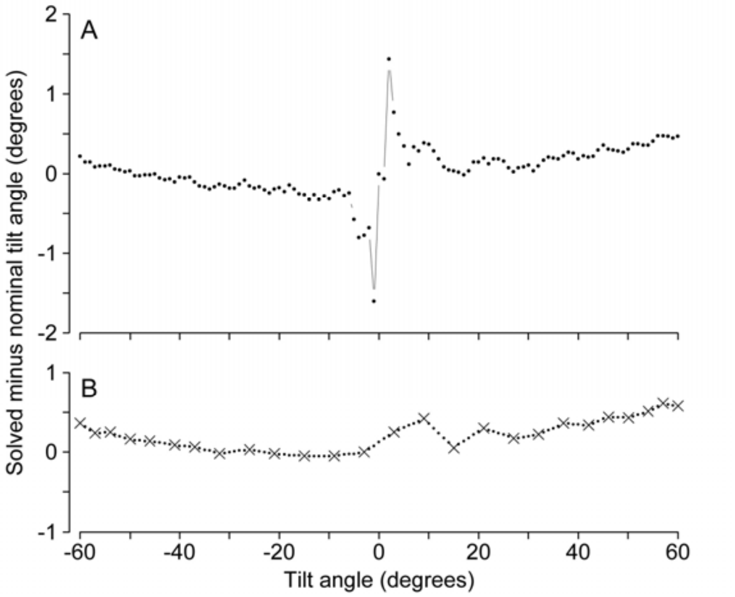


No grouping: Tiltalign solves for an independent tilt angle at each view



Grouping by 4: Tiltalign solves for tilt angle only at a subset of views (circles)
At other views, the tilt angle is constrained to be a linear combination of two variables
Fiducial positions from all views are used to find these tilt angles

Benefits of Grouping for Reducing Random Errors and Giving an Accurate Tilt Angle Solution

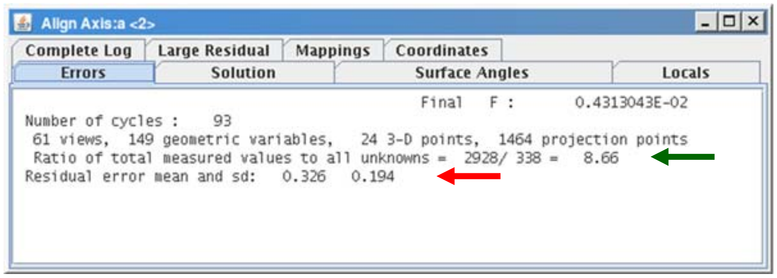


Grouping

- Grouping can dramatically reduce the number of variables being solved for.
- Grouping provides more averaging over errors in fiducial positions and keeps the solution from accommodating to random errors. (Mean residual may be higher but the solution should be more accurate.)
- Grouping is appropriate for slowly changing variables, especially for ones that are hard to solve for.

Key Measures in Alignment Output

- Mean of residual errors (distances between actual and predicted positions)
- Ratio of measurements to values being solved for (reflects amount of averaging over random position errors)



```
                                         Final    F :        0.4313043E-02
 Number of cycles :    93
  61 views,  149 geometric variables,   24 3-D points,  1464 projection points
  Ratio of total measured values to all unknowns =  2928/ 338 =   8.66
 Residual error mean and sd:   0.326    0.194
```

Measured position ●

Position predicted by fit ✖ Residual error

Alignment and Other Challenges in Reconstructing Cryotomograms with IMOD

Challenges in Cryotomography

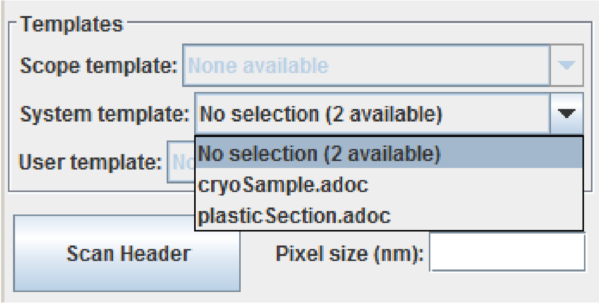
- Alignment, alignment, alignment
  - It can be hard to get fiducials onto/in the sample
  - The low SNR makes the fiducial positions more uncertain
  - Fiducials are often essentially in a plane, restricting the kind of alignment solution that can be sought
  - Energy filtering, good for cryoimaging, introduces geometric distortions that impair alignment (correctable in IMOD, minimal in new filters)
  - The ice does change (deform) over course of tilt series
  - Cryosections are even harder to get fiducials on, tend to change more under the beam, and fiducialless alignment works worse because of crevasses

Challenges in Cryotomography

- Alignment, alignment, alignment
  - It can be hard to get fiducials onto/in the sample
  - The low SNR makes the fiducial positions more uncertain
  - Fiducials are often essentially in a plane, restricting the kind of alignment solution that can be sought
  - Energy filtering, good for cryoimaging, introduces geometric distortions that impair alignment (correctable in IMOD)
  - The ice does change (deform) over course of tilt series
  - Cryosections are even harder to get fiducials on, tend to change more under the beam, and fiducialless alignment works worse because of crevasses.
- Contrast is low and artifacts from gold become more prominent
  - Erasing gold may be particularly helpful
- SNR is low and filtering/denoising may be needed to see features of interest

Use a Template in Etomo

- The cryoSample template distributed with IMOD has ~10 useful default values for cryo reconstructions, e.g.,
  - Filter setting for centering gold beads better when tracking
  - Patch size for tracking image patches for alignment
- You can substitute a system template or make a user template with even better values, but in any case, you should use something



Templates

Scope template: None available

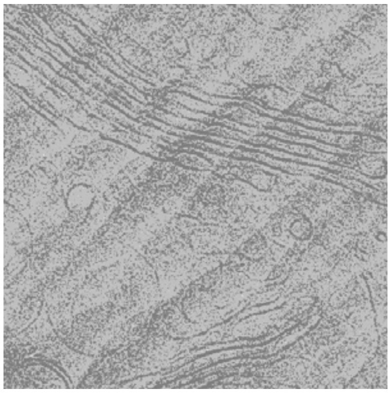
System template: No selection (2 available)

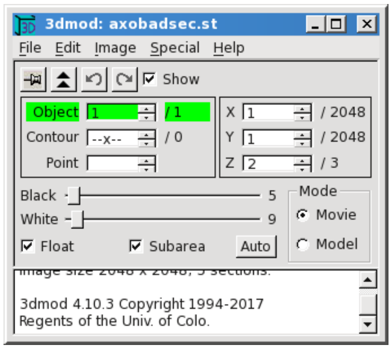
User template: No

Pixel size (nm):

Looking at the Raw Stack

- Before preprocessing to remove X-rays (extreme values), the dynamic range of cryo data often appears poor when the data are loaded into 3dmod as bytes
  - 3dmod stores the data as bytes by default, without truncating extreme values





Object 1 / 1
Contour --x-- / 0
Point

X 1 / 2048
Y 1 / 2048
Z 2 / 3

Black 5
White 9
☑ Float ☑ Subarea Auto

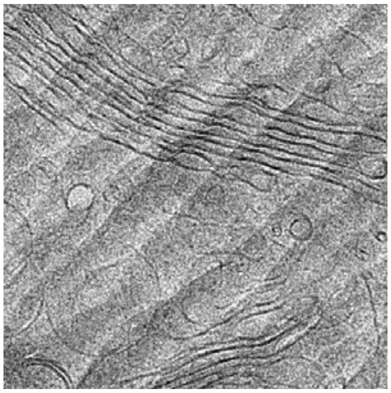
Mode
● Movie
○ Model

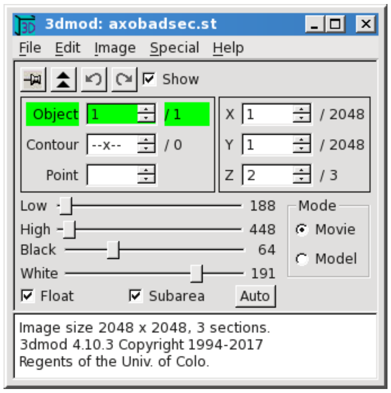
Image size 2048 x 2048, 3 sections.
3dmod 4.10.3 Copyright 1994-2017
Regents of the Univ. of Colo.

Default loading as bytes

Looking at the Raw Stack

- Before preprocessing to remove X-rays (extreme values), the dynamic range of cryo data often appears poor when the data are loaded into 3dmod as bytes
  - 3dmod stores the data as bytes by default, without truncating extreme values
- Loading data into 3dmod as integers preserves the number of gray levels when contrast is stretched





Object 1 / 1
Contour --x-- / 0
Point

X 1 / 2048
Y 1 / 2048
Z 2 / 3

Low 188
High 448
Black 64
White 191
☑ Float ☑ Subarea Auto

Mode
● Movie
○ Model

Image size 2048 x 2048, 3 sections.
3dmod 4.10.3 Copyright 1994-2017
Regents of the Univ. of Colo.

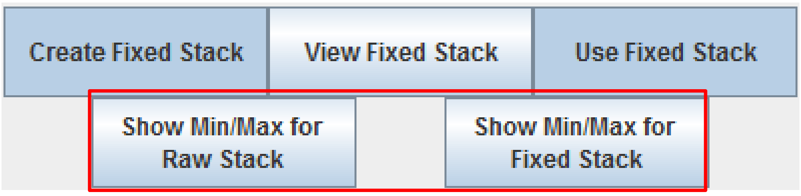
Loading as integers

Looking at the Raw Stack

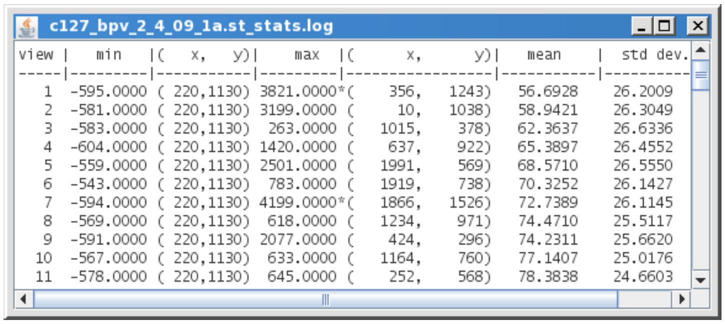
- Before preprocessing to remove X-rays (extreme values), the dynamic range of cryo data often appears poor when the data are loaded into 3dmod as bytes
  - 3dmod stores the data as bytes by default, without truncating extreme values
- Loading data into 3dmod as integers preserves the number of gray levels when contrast is stretched
  - Etomo will load a raw stack this way to avoid initial problems
  - You can set default to load integers
- Better long-term solution: remove extreme values
  - This also avoids artefactual rays through tomogram

Extreme Value Removal

- X-ray events and extreme values in images are found in two ways:
  - Looking for pixels higher than background by extreme amounts (controlled by “peak criterion”)
  - Looking for pixels that differ from adjacent ones by extreme amounts (controlled by “difference criterion”)
- The default criteria in the cryo template are a good starting point.
  - If more points need to be removed, lower both criteria by 1
- The best way to judge if removal is sufficient is from output of Min/Max values



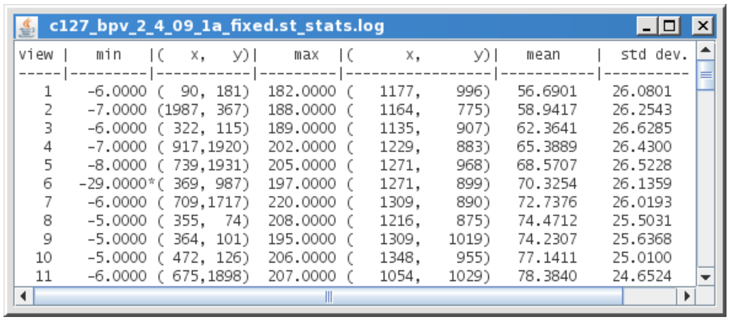
Statistics Before and After X-ray Removal



```
view |    min    |(    x,    y)|    max    |(        x,        y)|    mean     |  std dev.
-----|-----------|-------------|-----------|---------------------|-------------|----------
   1  -595.0000 ( 220,1130) 3821.0000*(     356,     1243)   56.6928     26.2009
   2  -581.0000 ( 220,1130) 3199.0000 (      10,     1038)   58.9421     26.3049
   3  -583.0000 ( 220,1130)  263.0000 (    1015,      378)   62.3637     26.6336
   4  -604.0000 ( 220,1130) 1420.0000 (     637,      922)   65.3897     26.4552
   5  -559.0000 ( 220,1130) 2501.0000 (    1991,      569)   68.5710     26.5550
   6  -543.0000 ( 220,1130)  783.0000 (    1919,      738)   70.3252     26.1427
   7  -594.0000 ( 220,1130) 4199.0000*(    1866,     1526)   72.7389     26.1145
   8  -569.0000 ( 220,1130)  618.0000 (    1234,      971)   74.4710     25.5117
   9  -591.0000 ( 220,1130) 2077.0000 (     424,      296)   74.2311     25.6620
  10  -567.0000 ( 220,1130)  633.0000 (    1164,      760)   77.1407     25.0176
  11  -578.0000 ( 220,1130)  645.0000 (     252,      568)   78.3838     24.6603
```

Before

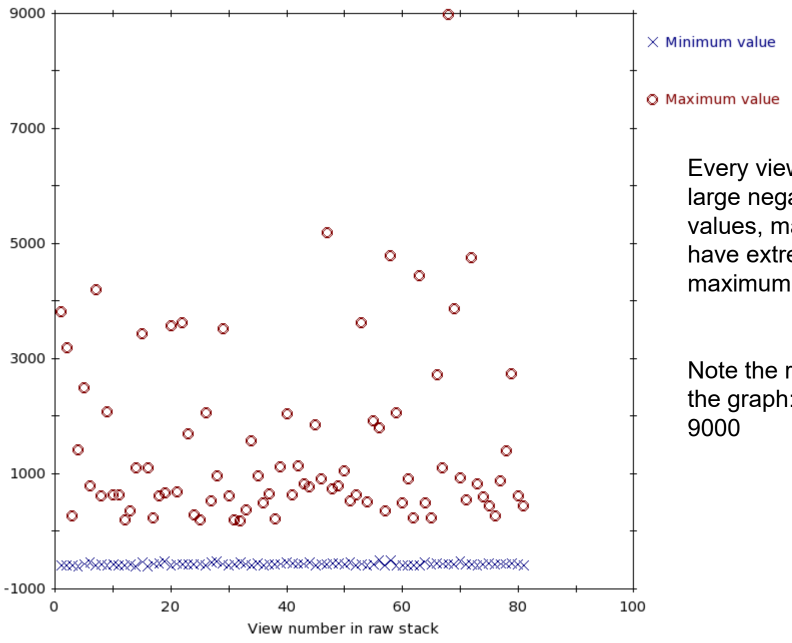
Min and max values that stand out are marked with *



```
view |    min    |(    x,    y)|    max    |(        x,        y)|    mean     |  std dev.
-----|-----------|-------------|-----------|---------------------|-------------|----------
   1     -6.0000 (  90, 181)  182.0000 (    1177,      996)   56.6901     26.0801
   2     -7.0000 (1987, 367)  188.0000 (    1164,      775)   58.9417     26.2543
   3     -6.0000 ( 322, 115)  189.0000 (    1135,      907)   62.3641     26.6285
   4     -7.0000 ( 917,1920)  202.0000 (    1229,      883)   65.3889     26.4300
   5     -8.0000 ( 739,1931)  205.0000 (    1271,      968)   68.5707     26.5228
   6    -29.0000*( 369, 987)  197.0000 (    1271,      899)   70.3254     26.1359
   7     -6.0000 ( 709,1717)  220.0000 (    1309,      890)   72.7376     26.0193
   8     -5.0000 ( 355,  74)  208.0000 (    1216,      875)   74.4712     25.5031
   9     -5.0000 ( 364, 101)  195.0000 (    1309,     1019)   74.2307     25.6368
  10     -5.0000 ( 472, 126)  206.0000 (    1348,      955)   77.1411     25.0100
  11     -6.0000 ( 675,1898)  207.0000 (    1054,     1029)   78.3840     24.6524
```

After

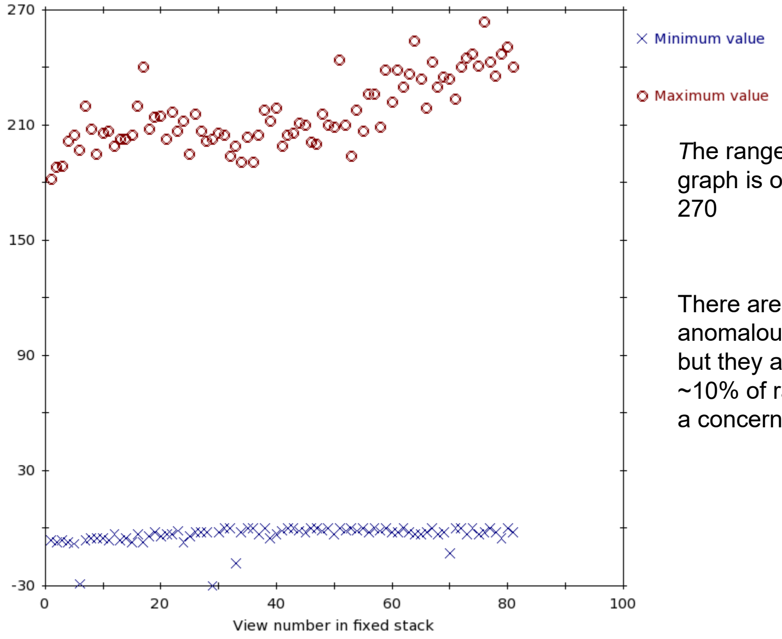
Min/Max in Raw Stack



Every view has large negative values, many views have extreme maximum values

Note the range of the graph: -1000 to 9000

Min/Max in Fixed Stack

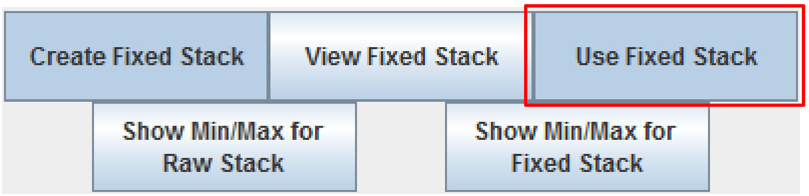


The range of the graph is only -30 to 270

There are still some anomalous values, but they are only ~10% of range: not a concern

Extreme Value Removal

- X-ray events and extreme values in images are found in two ways:
  - Looking for pixels higher than background by extreme amounts (controlled by “peak criterion”)
  - Looking for pixels that differ from adjacent ones by extreme amounts (controlled by “difference criterion”)
- The default criteria in the cryo template are a good starting point.
  - If more points need to be removed, lower both criteria by 1
- The best way to judge if removal is sufficient is from output of Min/Max values
- You need to push “Use Fixed Stack” to replace original stack with the fixed stack
- You could then iterate, but the program now does 3 iterations, so manual iteration should not be needed much



Points on Alignment Specific to Cryo

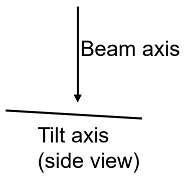
- Solving for distortion (stretching)
- Solving for beam tilt
- Local alignments
- Dealing with few beads

Distortion (Stretching) Solution is Rarely Helpful

- Solving for stretch requires a good distribution of fiducials in Z, which is usually not the case for cryoET
- There are fewer of the kind of changes that it can correct for than with plastic sections
  - Section thinning is corrected by this solution

Beam Tilt

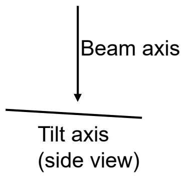
- Beam tilt is shorthand for nonperpendicularity between beam axis and tilt axis
- Its effects are very similar to those of stretch along an oblique axis in specimen
  - In “aligned” images, features move up and down in Y through the tilt series
  - Solving for stretch will correct for beam tilt effects
  - Solving for variable rotation angles partially corrects for them



Beam Tilt

- Beam tilt is shorthand for nonperpendicularity between beam axis and tilt axis
- Its effects are very similar to those of stretch along an oblique axis in specimen
  - In “aligned” images, features move up and down in Y through the tilt series
  - Solving for stretch will correct for beam tilt effects
  - Solving for variable rotation angles partially corrects for them

- Including beam tilt can significantly improve the alignment solution when variable rotation angles are not being solved for
- Including beam tilt only adds one unknown, not one per group of views
- Data sets from a given microscope tend to have similar values



Local Alignments

- The alignment equations can account for linear changes in the specimen (the same everywhere)
- Nonlinear changes in specimen have worse effects on alignment the larger the area
- With local alignments, different alignments are obtained from subsets of fiducials, thus allowing correction for nonlinear changes

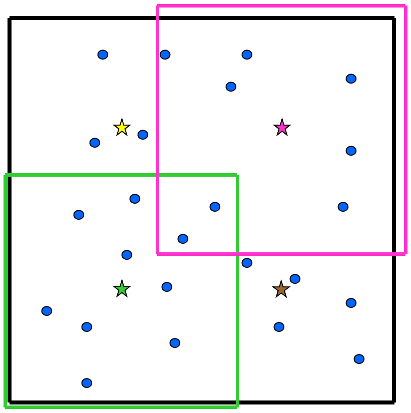
Local Alignments in Overlapping Local Areas

The centers of the local areas are defined by the number of overlapping areas.

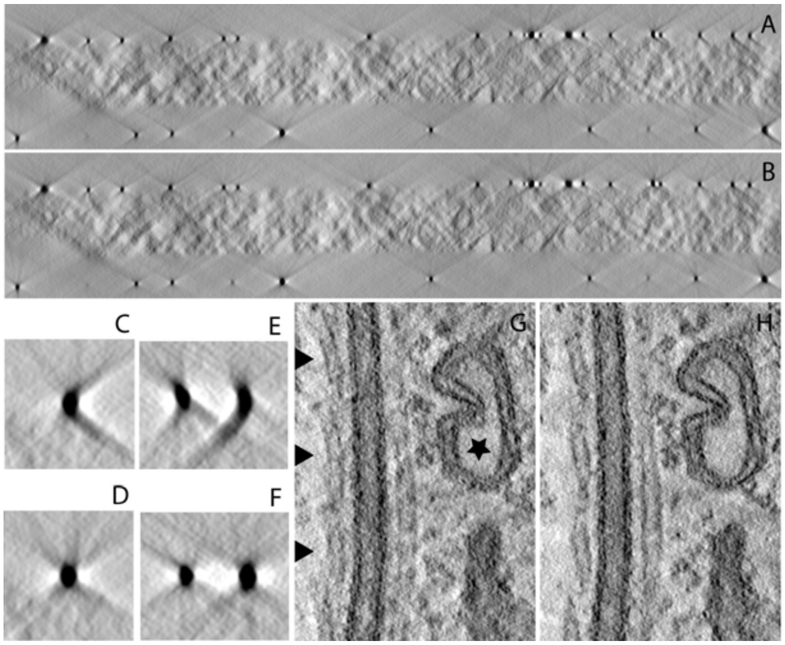
Here each local area is required to have 8 fiducials.

Area 1,1 does not need to be expanded from the minimum size to include 8 fiducials

Area 2,2 grows from its center to include 8 fiducials

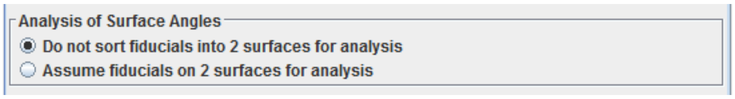


Improvements with Local Alignments



Local Alignments in CryoET

- Local alignments can be helpful with cryoET if there are enough fiducials, particularly for larger areas acquired with direct detectors
  - There should be at least 20 fiducials, preferably ~40-50
  - The average mean residual of local areas can be 20-40% less than that of global solution
- Just be sure not to solve for stretching and not to analyze for beads on two surfaces (use a template!)

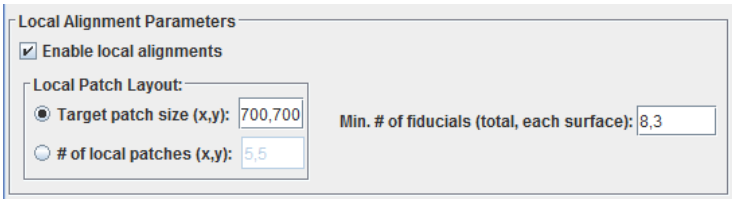


Analysis of Surface Angles

● Do not sort fiducials into 2 surfaces for analysis

○ Assume fiducials on 2 surfaces for analysis

- For 4K areas, increase the default target patch size or switch to specifying # of local patches



Local Alignment Parameters

☑ Enable local alignments

Local Patch Layout:

● Target patch size (x,y): 700,700

○ # of local patches (x,y): 5,5

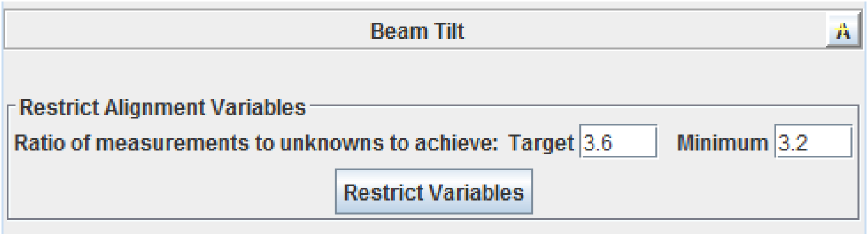
Min. # of fiducials (total, each surface): 8,3

What to Do with Few Fiducials

- In general, try to keep the ratio of measured/unknowns >= 4; if points are well-distributed and well-centered, the ratio can go down to 3 or even lower
  - When the ratio is too low, there is insufficient averaging over the errors in fiducial positions and images become misaligned by these errors
  - How low a ratio you can get away with also depends on how good the distribution of fiducials is: do they cover the field, are some of them close together?

Restricting Alignment Variables

- The script Restrictalign will restrict variables automatically to reach a target value for the ratio of measurements to unknowns
  - It first groups variables that are not grouped, then fixes variables
  - It will start to solve for beam tilt if it drops to solving for one rotation
  - It handles all the way down to one point and solving for shifts only
  - It is used in the batch processing



Beam Tilt

Restrict Alignment Variables

Ratio of measurements to unknowns to achieve: Target 3.6 Minimum 3.2

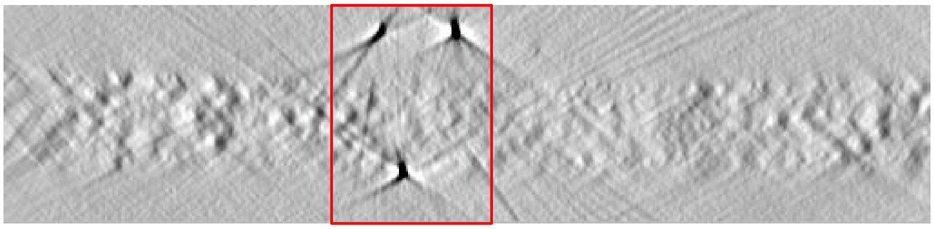
Guidelines for Adjusting Variables by Hand

- 5 Fiducials:
  - Group all variables (magnification and rotation).
  - Or solve for one rotation and beam tilt to get M/U near 4.
- 4 Fiducials:
  - You should be able to solve for one grouped variable, e.g., group magnification and solve for one rotation and beam tilt.
  - You may be able to solve for two grouped variables instead.
- 3 Fiducials:
  - You can certainly solve for one rotation angle and beam tilt (M/U near 3)
  - With well-distributed points you may be able to solve for one grouped variable (magnification or rotation would be more reliable).
- 2 Fiducials:
  - If points are well-separated you should be able to solve for one rotation angle and beam tilt (M/U near 2)
- 1 Fiducial:
  - You can solve for translation only – fix all variables (M/U = 1)

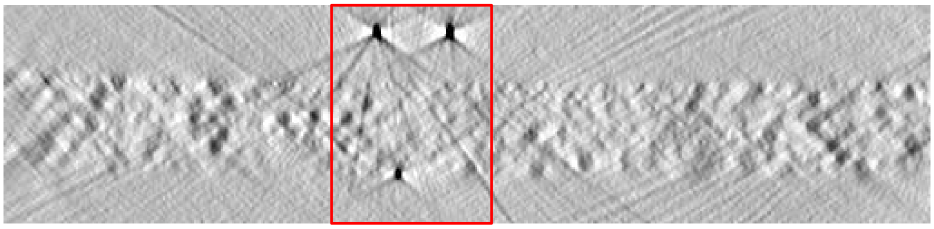
Avoid Overfitting with Few (or Many) Fiducials

- When comparing results with different variable selections:
  - A selection that give a substantial reduction in mean residual with only a small drop in M/U ratio is good
  - If a selection drops the M/U ratio substantially but gives only a small reduction in mean residual, the reduction is a direct consequence of fitting to more variables, which can do more harm than good

Initial Cross-Correlation Alignment is Inadequate



Cross-section of plastic section tomogram aligned by correlation alone

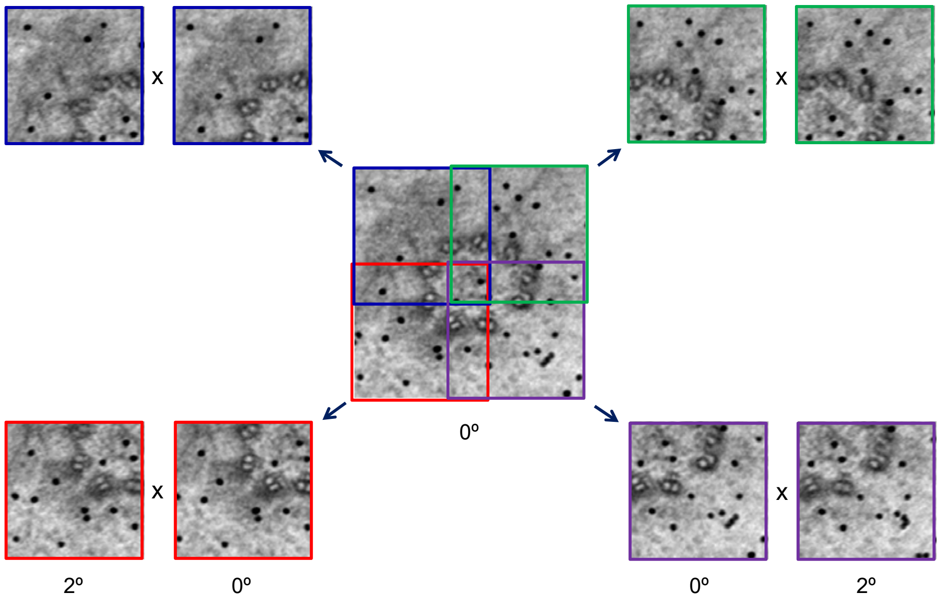


Cross-section of tomogram aligned with fiducial markers

Initial Cross-Correlation Alignment is Inadequate

- It fails to correct for effects like shrinkage of plastic-embedded specimens
- Errors can build up when aligning one image to the next, so that one end of tilt series is out of register with the other
- The angle of the tilt axis has to be determined accurately by other means
- There is no guarantee that it is aligning the same features through the whole series, especially for thick specimens

Local Patch Tracking from One View to Next



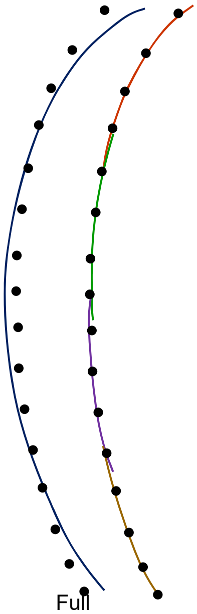
Local Patch Tracking from One View to Next

- Multiple subregions are independently correlated from one view to the next
- The positions of the patch centers are saved as a model that can be used for alignment
- The correlations are of a whole image area, do not localize or center any features in the area, and are vulnerable to the same problems of wandering off as whole image correlations
- Because there are multiple patches, we can derive alignment parameters such as rotation, and assess quality of fit

Local Patch Tracking Fiducial(less) Alignment

- The size of patches needed depends on the richness of image detail and its SNR in the images
  - For cryospecimens, 500-1000 pixels is typically needed (template default value is 680)
- Areas can be excluded from the coarsely aligned stack
- Run initially without breaking contours of tracked points into pieces
  - Mean residual from alignment with full-length contours can be validly compared to residual from true fiducial alignment

Breaking Tracking Contours into Pieces



Full contour

Four separate contours overlapping in Z, made from same set of points

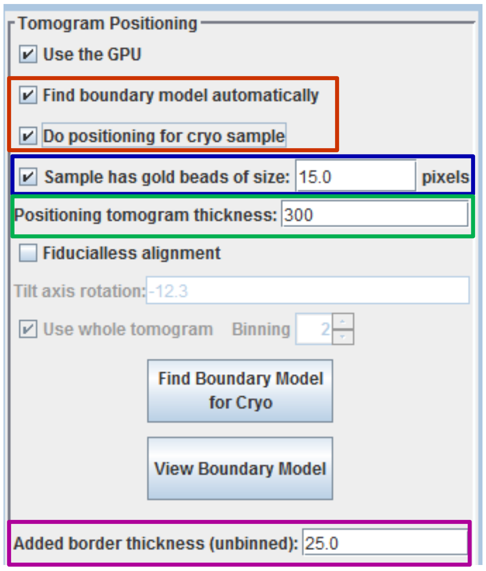
- Breaking patch tracking contours up may help in two ways
  - To extent that feature being tracked changes through series, each segment of contour tracks a feature more consistently
  - Subsets of the tracking that are least consistent can be identified and eliminated in Bead Fixer or with robust alignment
- But whenever there is a systematic misfit between fiducials and alignment model, fitting over shorter extents will reduce the error
  - So the **much** lower residual that occurs from breaking the contours is largely due to fitting over shorter segments

Tomogram Positioning

- Tomogram positioning (setting angles and thickness) is challenging for cryo tilt series because of low contrast and sparseness
- There is now automated positioning in Etomo
  - Works great for plastic sections
  - Completely different procedure implemented for cryo – worth trying
- With cryosamples, artifacts from the high-density structures (gold) interfere with the detection of ice boundaries
- The Cryoposition script builds a tomogram from tilt images with high-density features erased and analyzes that for structure to find surfaces of the ice

Automated Cryopositioning

- It is selected by turning on both the automatic and cryo options
- The procedure needs to know if there are gold beads even if they were not used for alignment
- It needs a generous thickness set, as usual for positioning
- Etomo automatically switches to a larger border to add when computing the thickness from the boundary model



Tomogram Positioning

☑ Use the GPU

☑ Find boundary model automatically

☑ Do positioning for cryo sample

☑ Sample has gold beads of size: 15.0 pixels

Positioning tomogram thickness: 300

□ Fiducialless alignment

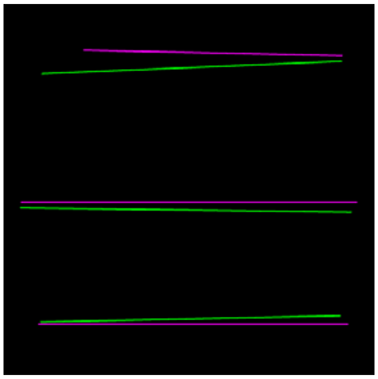
Tilt axis rotation: -12.3

☑ Use whole tomogram Binning 2

Added border thickness (unbinned): 25.0

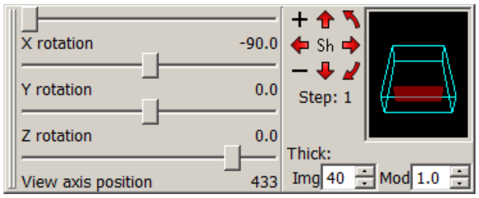
Whole Tomogram Positioning

- The whole tomogram allows you to draw 2 or more pairs of lines at arbitrary positions in Y, instead of 3 pairs of lines at fixed positions
  - The lines need not be perfectly horizontal (viewed from top) and need not match up exactly between top and bottom surfaces
  - But it should be obvious how they are paired – open model view window to assess this
- The tomogram should be built with binning
  - This will be quick
  - Binning increases SNR a lot and makes it easier to see the surface features in a cryotomogram



The Easiest Way to Do Whole Tomogram Positioning

- Use Slicer to view X/Z slices as in conventional sampling
  - Set X angle to +90° or -90°
  - Scroll with the “View axis position” slider to a likely level in Y
  - Average slices until you can see the top and bottom surface
  - Repeat at 1-2 more places in Y



X rotation -90.0

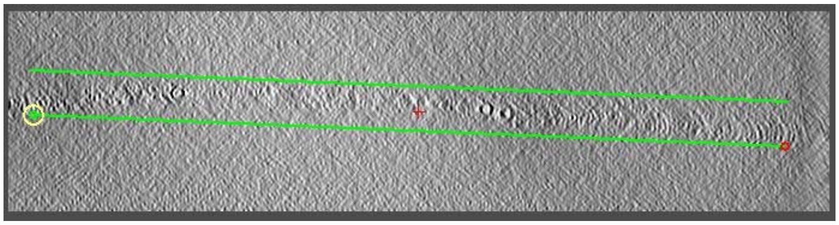
Y rotation 0.0

Z rotation 0.0

View axis position 433

Step: 1

Thick: Img 40 Mod 1.0

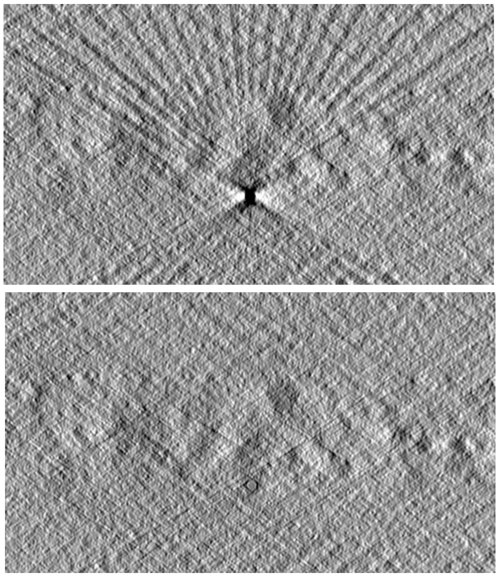


Harder Ways to Do Whole Tomogram Positioning

- View X/Y slices in Zap window
  - Looking at one Y level, scroll in Z until the surface becomes visible on left or right side of image, and add a point there
  - Step in Z until surface is reached on the other side and add point there
  - Scroll in Z to other surface of the material and add two points there
  - Repeat at 1-2 more Y levels
- View slices parallel to surface in Slicer
  - Scroll in Z until a surface is visible in middle
  - Rock angles until the surface appears parallel to the slice
  - Draw lines at surface, step in Z to bring each point to consistent location relative to surface if necessary

Erasing Gold Particles

- Erasing gold beads will eliminate undesirable rays in tomograms, whose effect is worse in cryotomograms due to lower contrast

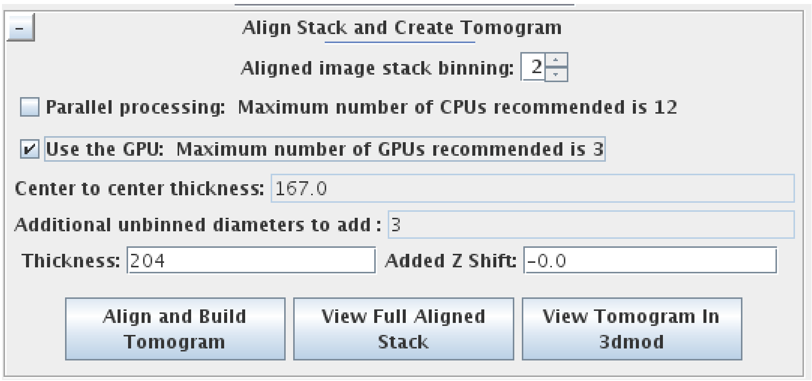


Erasing Gold Particles

- Erasing gold beads will eliminate undesirable rays in tomograms, whose effect is worse in cryotomograms due to lower contrast
- There are two different methods of getting positions for erasing gold beads
  - Transform the fiducial model itself
    - This is simple to do but only includes fiducials
    - A “completed” model with missing points filled in is used
  - Find gold in tomogram, project positions onto aligned stack
    - This can be the easiest way to remove all gold, but checking the 3D model is more difficult
    - If a position is correct in the 3D model, its projections will be present and (nearly) correct in all views of aligned stack

Erasing Gold Particles after Detecting in Tomogram (1)

- Build a tomogram specifically for detecting gold beads
  - Can and should be binned for speed, as long as beads are still >= 5 pixels
    - Etomo will pick the right binning
  - May need to be thicker, or shifted in Z, to hold all the gold
    - Etomo will initialize based on analysis of range of fiducial positions reported in align.log



Align Stack and Create Tomogram

Aligned image stack binning: 2

□ Parallel processing: Maximum number of CPUs recommended is 12

☑ Use the GPU: Maximum number of GPUs recommended is 3

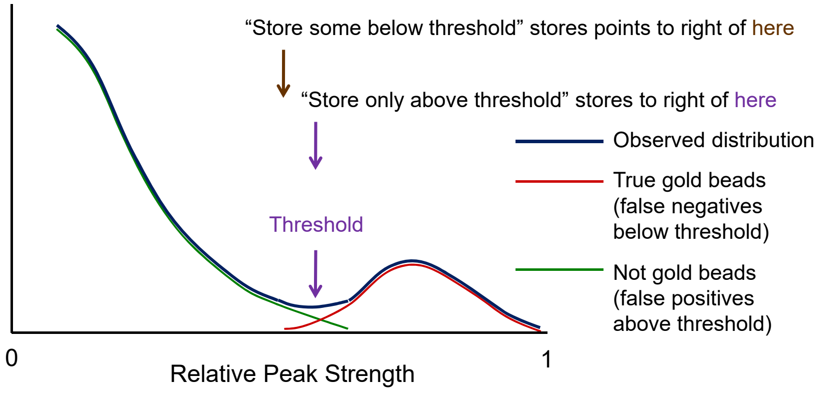
Center to center thickness: 167.0

Additional unbinned diameters to add : 3

Thickness: 204 Added Z Shift: -0.0

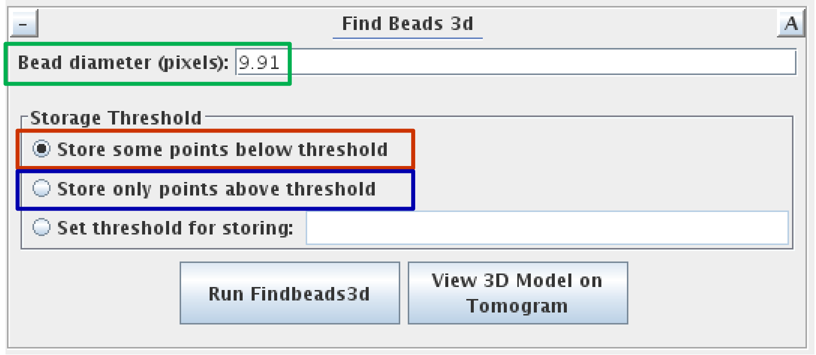
Findbeads3d Operations

- For every possible bead, an integrated density relative to background is computed
- Integrated densities are scaled from 0 to 1 to obtain “relative peak strengths”
- Histogram of peak strengths is analyzed to find dip between two peaks
- This dip is the threshold value that best separates densities that are probably gold from densities that are probably not



Erasing Gold Particles after Detecting in Tomogram (2)

- Detect beads in tomogram with Findbeads3d using the true diameter
  - “Store some points below threshold” will save points in model that are probably not beads and will need to be deleted
  - Use “Store only points above threshold” if you do not plan to work with model or have trouble remembering to delete points below threshold



Bead diameter (pixels): 9.91

Storage Threshold

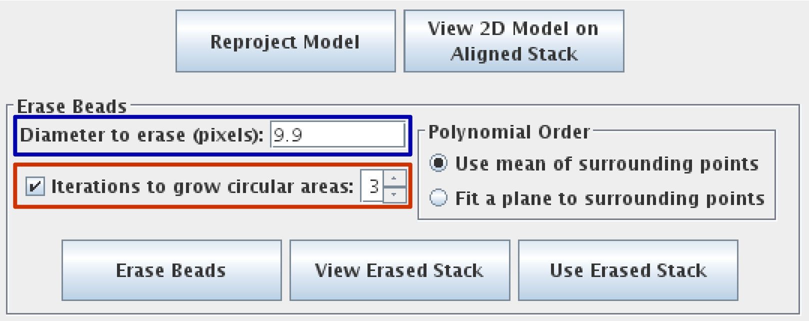
● Store some points below threshold

○ Store only points above threshold

○ Set threshold for storing:

Erasing Gold Particles after Detecting in Tomogram (4)

- Project model, erase stack, and view erased stack
  - It may be easier to see problems here; if so, adjust the 3D model, project, erase, and view again
  - Points are often not centered at high tilt; “Growing circular areas” will erase dark pixels on one side
  - Increase the diameter both to handle remaining deficiencies in centering and to remove underfocus fringes



Diameter to erase (pixels): 9.9

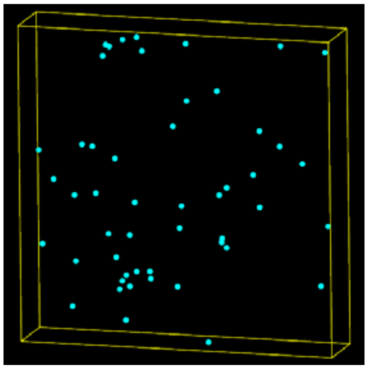
☑ Iterations to grow circular areas: 3

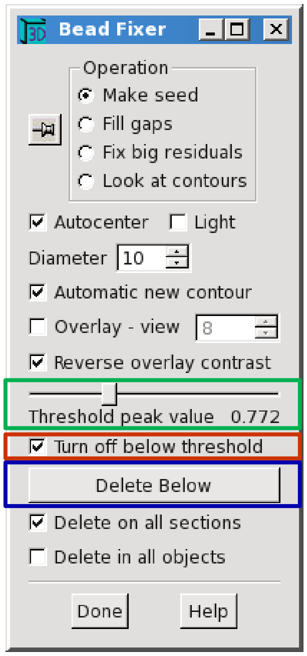
● Use mean of surrounding points

○ Fit a plane to surrounding points

Erasing Gold Particles after Detecting in Tomogram (3)

- “View 3D Model on Tomogram” opens tomogram, model view, and Bead Fixer with extra controls for this situation
  - The threshold slider lets you adjust the selection of points
  - Toggle “Turn off below threshold” to see the ones you could add
  - Right click on point to bring it up in Zap window
  - Press “Delete Below” to remove points below current threshold once yet have set it optimally





● Make seed

○ Fill gaps

○ Fix big residuals

○ Look at contours

☑ Autocenter ☐ Light

Diameter 10

☑ Automatic new contour

☐ Overlay - view 8

☑ Reverse overlay contrast

Threshold peak value 0.772

☑ Turn off below threshold

☑ Delete on all sections

☐ Delete in all objects

Manual Reconstruction of a Cryo Tomogram

RECONSTRUCTING A CRYO TOMOGRAM USING FIDUCIALS

In this exercise, you will build a tomogram from scratch using Etomo.

1) `cd $WORKSHOP_HOME/IMOD_Labs/cryo-manual`

2) `etomo`

3) In this initial step, we define some features of the data set and create the files needed for processing. Press **Build Tomogram** in the Etomo **Front Page**.

   Press the yellow file chooser icon on the **Datatset name** line and select *cryo.st*. For **SystemTemplate**, select *cryoSample.adoc*. This will set several parameters appropriately for cryo reconstructions; you should generally select a template.

   Press **Scan Header** to retrieve the pixel size and rotation angle of the tilt axis from the image file. Enter **15** for **Fiducial diameter (nm).** Select **Parallel Processing** to use all of the CPU cores when possible. These laptops do not have GPUs, however, it is best to check this box if you do have an IMOD compatible GPU. For **Image distortion field file**, select *27.5kGIF2007-03-24.idf*. These data were taken with an energy filter which introduces significant geometric distortions. The distortions were measured and can be corrected with this file.

Press **View Raw Image Stack** and scroll through the images to see that they are not well-aligned. Notice the 4 contrast control sliders on the IMOD info window, which are present because the data have been loaded as integers. The top two sliders are very close together, which means the image data occupy a very small fraction of the full data range; the rest of the data range is taken up by image artifacts due to X-rays. In general, if you see bad views that you know that you want to exclude from alignment and reconstruction, you can list these in the Exclude views box. Close 3dmod. Press **Create Com Scripts.**

4) Click on **Pre-processing Not Started**. Pre-processing is needed to remove artifacts in the images, generally produced by X-ray events in the camera. These artifacts will produce streaks in a reconstruction and can also make it harder to see the image features, which have a much smaller dynamic range than the artifacts.

   Press **Show Min/Max for Raw Stack** to see the range of the data; both a plot of minimum and maximum values and a table with more detailed information will open. Every view has a large negative minimum because there was an X-ray artifact in the dark reference.

   Press Create Fixed Stack to run the program that finds and erases artifacts. Press **View Fixed Stack**. Now there are only 2 contrast sliders because data can be loaded as bytes after removing most of the artifacts.

   Press **Show Min/Max for Fixed Stack** to see the new range of data. Deviations of 50-100 from the rest of the data will not matter in the reconstruction. Close all 3dmod, graph, and table windows.

   Press **Use Fixed Stack**. Press **Done** to advance to the next step.

5) In the Course Alignment step, we use image cross-correlation to align successive images, which makes it easier to track fiducial markers.

   Press **Calculate Cross-Correlation**
   When done, press **Generate Coarse Aligned Stack**. When done, press **View Aligned Stack in 3dmod**. Scroll through the images to see that they now look aligned. Press **Done** to go on.

6) In the Fiducial Model Gen. step, the positions of selected gold markers are found on all of the images, which allows a more accurate alignment to be obtained.

   Check that **Make seed and track** is selected. On the **Seed Model** tab, select **Make seed model manually**. The option to make the seed model automatically works quite well; we are not using it so that you learn how to do this step manually if necessary. Press **Seed Fiducial Model**, which opens the Bead Fixer window. Turn on **Automatic new contour** in the Bead Fixer window if it is not on already. A contour is a set of connected points. We need to put these points in separate contours because a) we don't want to see them connected and b) each contour will be added to with the corresponding points on other views.

   Make sure you are on the central view (view **31**) and place the cursor very near each bead and add a point by middle-clicking. It will be automatically centered (the Autocenter option is turned on by Etomo in seeding mode). Press the **s** key to save the model.

   Switch to the **Track Beads** tab and press **Track Seed Model**. The **Project Log** window shows the number of missing points when done. It may be 0. Press **Fix Fiducial Model**, which will load the tracked model into 3dmod and switch the Bead Fixer to Fill gaps mode. You can scroll through the views and also press the **v** key to see the tracks of the beads in 3D.

If there are missing points, you can now use the Bead Fixer to step from one gap to the next (using the space bar as a hot key) and add a point, if appropriate. A bead does not need to be marked on every view, and you should not add a point if the bead's position is not clear. If you add points, be sure to save the model with the s key. Another way to complete the model is to press Track with Fiducial Model as Seed. It is a good idea to look at the model in 3D first and make sure there are not deviant points at the ends of contours that should be fixed first. Press **Done** to go on.

7) During Fine Alignment, the bead positions are fit to a mathematical model of specimen movements. The model predicts a position for each bead on each view, and the mean distance between the predicted and actual positions is referred to as the "mean residual error". These errors will let you find and correct badly modeled points. The need to do so has been much reduced by the recent addition of a method called "robust fitting", which automatically gives less weight or even eliminates the points most likely to be at incorrect positions. However, it is good to learn how positions can be fixed manually.

   Select **Do not sort fiducials into 2 surfaces for analysis** and press **Compute Alignment**. When it is done, the Mean residual error will appear in the **Project Log**.

   Press **View 3D Model** to open the model of solved positions in 3D and assess whether to solve for distortion (stretching and skew). Press the **r** key in the Model View window for a side-view of the model. If you middle-click and move the mouse to the right, the model will rotate until the three lower beads are very close together. To solve for distortion, fiducials should be well-distributed in Z: not all on one plane, more than a few at a different Z height, and the ones at a different height distributed over the area. This is not often the case for cryo data sets, and this one fails the latter two tests.

Press **View/Edit Fiducial Model**. The Bead Fixer will open in (or be switched to) Fix big residuals mode and read in the log from alignment. Zoom up in the ZaP window to 2 or so. It is also helpful to switch the ZaP window to keeping the current model point centered, by pressing the concentric squares in the toolbar.

On the **Bead Fixer** window, press **Go to Next Big Residual** or the single quote (') key, it will move to each point with a big residual in order by decreasing value. If a point is not centered on the gold bead, you can move it to the correct position by right-clicking to the correct position. The red arrow points to the position predicted by the alignment solution; it is not based on any knowledge of what is in the image. This predicted position may not be correct, but if it is, you can move the point to that predicted position with **Move Point by Residual** or the semicolon (**;**) key. All of the biggest residuals occur at very high tilt, and each point has been correctly modeled, so there is nothing to fix. The reason for this is evident if you examine the output from the alignment program.

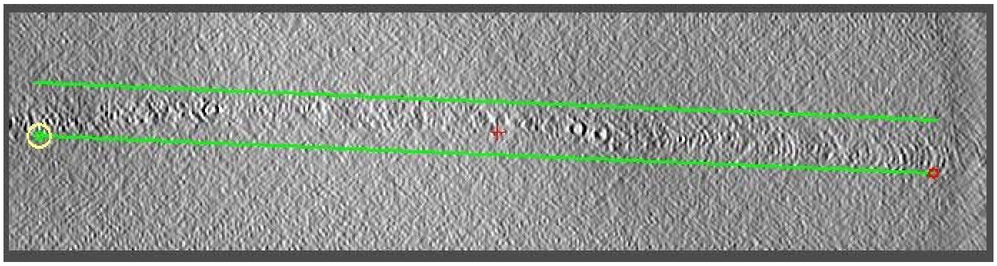
Right-click in Etomo over the **Fine Alignment** panel and select **Align log file**. The error summary and the ratio of measured values to variables solved for are in the **Errors** tab. Switch to the **Solution** tab to see the alignment parameters. Various columns from the solution table can be plotted by right-clicking in Etomo and selecting a particular parameter. Right-click and select **Plot global mean residual**. Notice that the mean residual gets much higher at high tilt, which is why we saw only those points having the biggest residuals. To find possible erroneous points at lower tilts in this case, in Etomo, under **Residual Reporting**, select **Relative to Neighboring views**. Press **Compute Alignment** again. There are no new points to fix as reported in the 3dmod info window. Press **Done** to go on. When Etomo asks "coarsely aligned stack is open in 3dmod. Should file be closed?" Press **Yes**. Etomo knows when a file is no longer needed and will prompt you to close them as necessary.

8) The goal of Tomogram Positioning is to set angles and an offset in Z so that the specimen is flat and centered in Z in the computed volume, thus minimizing the computational effort.

   Set **Positioning tomogram thickness** to **500**. Press **Create Whole Tomogram** to build a whole, binned-down tomogram.

   Press **Create Boundary Model** to open the tomogram. Open a Slicer window with **Image / Slicer** from the 3dmod info window. Set **X-rotation** to **90** and **Thick: Img** to **20**, which makes it average 20 slices. The goal is to draw a pair of lines enclosing the sample at ~3 different Y locations in the tomogram. For this specimen we can see the boundaries of the sample well in cross-section, so this is the easiest way to draw them.

   Use the **View axis position** slider to scroll through Y and get a sense of how the boundaries change. Go to the middle (**~340**) and draw two lines along the top and bottom surfaces of the sample. 3dmod will automatically start a new contour after two points. Move the current point by right-clicking to the new position; left-click to select a new current point. See the image below.



   Do the same near the lower end of the tomogram (axis position **~70**) and the upper end (**~610**). Your final model should contain 6 contours with 2 points each. Save the model with the **s** key.

Press **Compute Z Shift & Pitch Angles**. The Angle offset, Z shift, X-axis tilt, and tomogram thickness are shown; these will give the thinnest, most level tomogram containing the area you outlined. The required Final Tomogram Thickness is also shown.

Close 3dmod windows, and press **Create Final Alignment**, which reruns the fine alignment. Press **Done** to go on.

9) Next we will create a Final Aligned Stack, which applies a rotation to make the tilt-axis vertical. Press **Create Full Aligned Stack**. When done, press **View Full Aligned Stack**. Notice how the image has rotated and the corners are now cut off. Close 3dmod windows and press **Done** to move on.

10) On the Tomogram Generation page, you will at last compute the tomogram. In the **Parallel Processing** box, change the # **of CPUs Used** to **4** and press **Save As Defaults**.

    Press **Generate Tomogram**. Press **View Tomogram in 3dmod** when it is done. After reviewing the tomogram, press **Done** to go on.

11) In the Post-processing step, you can trim away unneeded regions, convert the tomogram to bytes to save time and space, and reorient the tomogram so that the slices stored in the file are in X/Y planes instead of X/Z planes. Even if you do not want to trim or convert to bytes, you should always go through this step to get a reoriented tomogram, which will work better with other programs.

Turn on the rubber band (rectangle to the left of the lasso) in the ZaP window and select an area of about 1500x1500 pixels, i.e., at a window zoom of **0.25**, the rubber band size on the screen of about **375x375** pixels, as shown in the ZaP toolbar. You can move an edge or corner of the rubber band by left-clicking, or shift the whole band by middle-clicking with the cursor over an edge. Scroll to the first slice that you want to keep and press **Lo**, then find the last slice you want to keep and press **Hi**. In Etomo, press **Get XYZ Volume Range from 3dmod** to import these limits.

In the ZaP window, move to where you can clearly see the gold particles. Reposition the rubber band square so that no gold particles or associated rays are within the box, then select a narrower range of slices with **Lo** and **Hi** that will be used to determine scaling; exclude the dense contamination on the surface. In Etomo, press **Get XYZ Sub-Area from 3dmod** in the **Scaling** section to import these limits. Close all 3dmod windows.

Press **Trim Volume**. When done, press **3dmod Trimmed Volume** to see the result. Press **Done** to go on.

12) Next we will Clean Up the directory. In most cases, there is no need for the intermediate files from processing. This step allows you to remove these files and leave all of the information from which they could easily be recreated if necessary.

The original raw tilt series stack can also "archived" by compressing its difference from the current stack; this operation is reversible. Press **Archive Original Stack** and press **Delete** to confirm the deletion of the original stack when it is done.

All intermediate files are now shown in the box. Click in the box and type **Ctrl+A** to select all files. You could then click on individual files while holding down the Ctrl key to unselect them. You might wish to leave *cryo.preali* (coarse aligned stack), *cryo.ali* (final aligned stack), or *cryo_full.rec* (raw reconstruction, which could be trimmed differently). Press **Delete Selected** to remove the intermediate files that are still selected. Press **Done** to finish and close Etomo.

RECONSTRUCTING A CRYO TOMOGRAM USING FIDUCIALLESS ALIGNMENT

13)

```
cd $WORKSHOP_HOME/IMOD_Labs/cryosection
```

14)

```
etomo
```

15) Press **Build Tomogram**. Press the file chooser icon on the **Datatset name** line and select *cryosection.st*. For **System template**, select *cryoSample.adoc*. Press **Scan Header** and enter **0** for **Fiducial diameter**. Some parameters for X-ray removal are set based on the fiducial size to avoid erasing fiducials, so it is important to enter 0 rather than an arbitrary size when there are no fiducials. Select **Parallel Processing** to use multiple CPUs to compute the reconstruction. In the **Axis A** group box, select **Series was bidirectional from** and set the angle to **0**. Press **Create Com Scripts.**

16) Select **Pre-processing** and press **Create Fixed Stack**. Press **Show Min/Max for Fixed Stack**. Press **Use Fixed Stack**, then **Done**.

17) Press **Calculate Cross-Correlation**. When done, press **Generate Coarse Aligned Stack** and then **View Aligned Stack in 3dmod**. Notice the slight ‘shift’ at zero degrees. Press **Done**.

18) In the Fiducial Model Generation step, patches of image will be tracked from one view to the next, and the center positions of the patches will be placed into a model file. This model can be used to solve for the tilt series alignment, much as a model of true fiducial markers can. Near the top of the window, select **Use patch tracking to make fiducial model**. Change the **Size of patches(X, Y)** to **680,680** (if it is not already set to that); this will give 4x4 patches. Press the **Advanced** button at the bottom and change the **High frequency cutoff radius** to **0.125** (if it is not already set to that). The parameters for patch tracking are mostly based on the ones for coarse alignment since the same program is being run. However, the correlations for coarse alignment are done on almost the full image and are generally binned down, so the high frequency filter cutoff of 0.25 is adequate for filtering out noise. The patches being correlated are smaller and will not be binned, so stronger filtering is needed. Press **Basic t**o make it easier to see the basic controls. Press **Track Patches**.

When done, press **Open Tracked Patches** to see the tracked locations on the image and in Model View (use hotkey **v** to open). No specific feature has been tracked, just the whole patch area. Notice that there are just 16 contours going all the way through the tilt series. Press **Done**.

19) For this data set, the **Fine Alignment** process will involve cycling back to the Patch Tracking section on the Fiducial Model Gen. page.

Press **Compute Alignment**. Look at your **Project Log** window, the mean residual is fairly high, ~1.5 pixels. Such a high residual occurs because some of the positions tracked through the patches do not correspond to projections from single points in 3D. A better fit to the points can be gotten by breaking the contours into pieces.

Return to the **Fiducial Model Generation** page and select **Break contours into pieces with overlap 4**. Press **Recut or Restore Contours**. Return to the **Fine Alignment** page and press **Compute Alignment**. The mean residual is much lower because much smaller errors build up over the shorter tilt ranges in each contour.

Press **View/Edit Fiducial Model**. In the Model View window, rotate the model so that you can see all the tracks well by pressing the middle mouse button and moving the mouse at a steep diagonal. Each track consists of 7 overlapping segments, which you can see by right-clicking near a track to select a contour; only a small segment is highlighted.

Right-click in Etomo to bring up the **Align log file** and switch to the **Coordinates** tab. The "mean resid" column shows the mean residual in each contour or segment of the track. The range is large, from 0.2 to 1.2. When patch tracking was first developed, the approach was simply to delete the contours with the highest residuals, and the Bead Fixer has a Look at contours mode to assist this process. Not only is this somewhat arbitrary, but it would lead to trouble in this case, since 10 of the highest 11 residuals occur for contours passing through zero degrees. The robust fitting that is now available is a better approach. Close the log file. In Etomo, turn on **Do robust fitting with tuning factor 1.0** and then turn on **Find weights for contours, not points**. Press **Compute Alignment**.

Open the **Align log** and go to **Coordinates** again. Each contour has been given a weight between 0 and 1, where a weight of 0 means that the points in the contour are ignored in the fitting. On the **Robust** tab, a summary line indicates 6-8 contours have weights under 0.5. On the **Coordinates** tab, there is a new column with the weights. The contours that have small weights are ones that have large residuals relative to other comparable contours (i.e., ones at similar tilt angles).

For your own data sets with patch tracking, you will probably want to select the robust fitting whenever you break contours into pieces. If contours are not broken into pieces, there are usually too few contours to allow robust fitting. Close all 3dmod windows and log window. Press **Done** in Etomo.

20) This sample can be positioned by viewing X/Z slices in the binned-down tomogram. Set **Positioning tomogram thickness** to **800**. Press **Create Whole Tomogram** to build a whole, binned-down tomogram. Press **Create Boundary Model** to open the tomogram

    Open a Slicer window with **Image / Slicer**. Set **X-rotation** to **90** and **Thick: Img** to **20**. Draw 3 pairs of lines: in the middle of the tomogram (axis position ~**380**) and at axis positions near **115** and **555**. Save the model with the **s** key. Close 3dmod.

    Press **Compute Z Shift & Pitch Angles**. Note the large angle offset (~11), which means that the tilt angles assumed during patch tracking were significantly different from the actual angles, enough to affect the correlations. Do the following steps to incorporate this knowledge. Return to the **Fiducial Model Generation** page and press **Advanced**. Enter the **Tilt angle offset** from the positioning page in the **Tilt angle offset** field. Press **Track Patches**. Go to the **Fine Alignment** page and do **Compute Alignment**. Note the substantial improvement in the weighted mean residual, from ~0.41 to ~0.37. You now have a final alignment that incorporates the positioning information, so you can go back to **Tomogram Positioning** and press **Create Final Alignment**. Press **Done**.

21) Press **Create Full Aligned Stack**. When done, press **View Full Aligned Stack**. Close 3dmod and press **Done**.

22) Press **Generate Tomogram**. Press **View Tomogram in 3dmod** when the program is done. Press **Done**.

23) Turn on the rubber band in the ZaP window and select a subarea area if desired. Scroll to the first slice that you want to keep and press **Lo**, then find the last slice you want to keep and press **Hi**. You should be able to trim off most of the crevasses on the bottom. In Etomo, press **Get XYZ Volume Range from 3dmod** to import these limits.

    In the ZaP window, select a narrower range of slices with **Lo** and **Hi** that will be used to determine scaling. In Etomo, press **Get XYZ Sub-Area from 3dmod in the Scaling** section to import these limits.

    Press **Trim Volume**. When done, press **3dmod Trimmed Volume** to see the result.

24) Press **Archive Original Stack** and **Delete** when asked. Select all **Intermediate files, SIRT Iteration** files and press **Delete Selected**. Press **Done** and close Etomo.

Batch Processing of Tomograms with IMOD

Batch Processing Can Be Useful for a Wide Range of Tilt Series

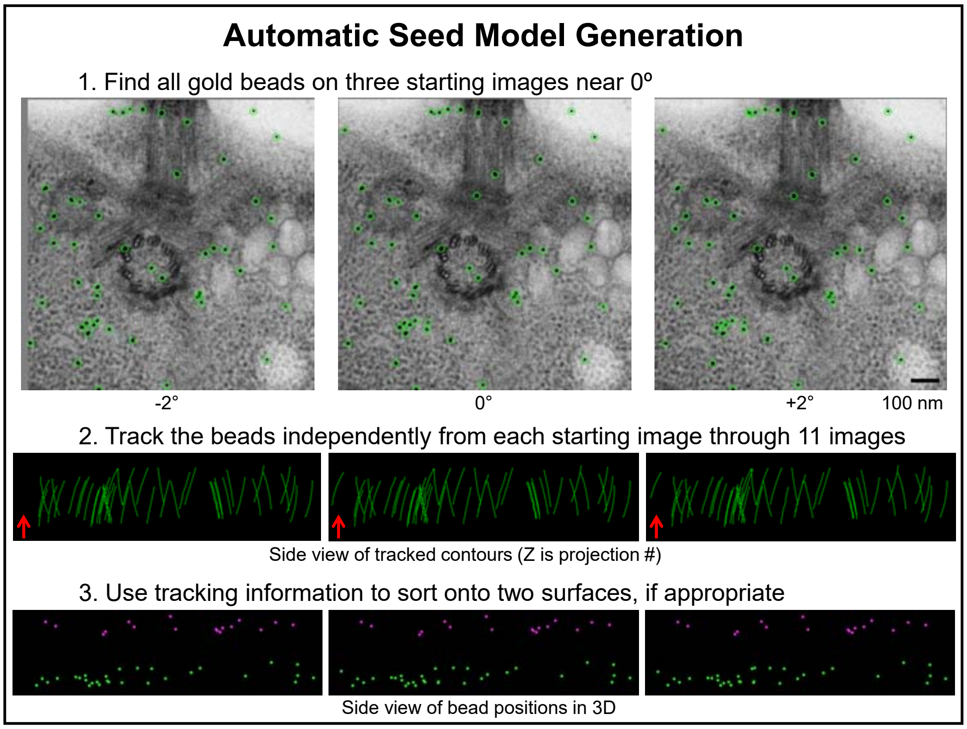
- "Routine" plastic section tilt series can be fully reconstructed automatically with 80-95% success rate
- Some kinds of cryoET data can also be reconstructed automatically up to the final trimming
- Both the method of processing and the batch interface are designed to allow interactive steps to be intermixed with batch processing
  - Even difficult data that require intervention at several points can be handled more efficiently by processing them through the batch interface

Major Developments to Support Automated Processing

- Automatic seed model selection to get a well-distributed set of fiducials suitable for tracking

Automatic Seed Model Generation

The goal is to find a well-distributed set of seed points suitable for tracking



Automatic Seed Model Generation

1. Find all gold beads on three starting images near 0º

-2°

0°

+2°

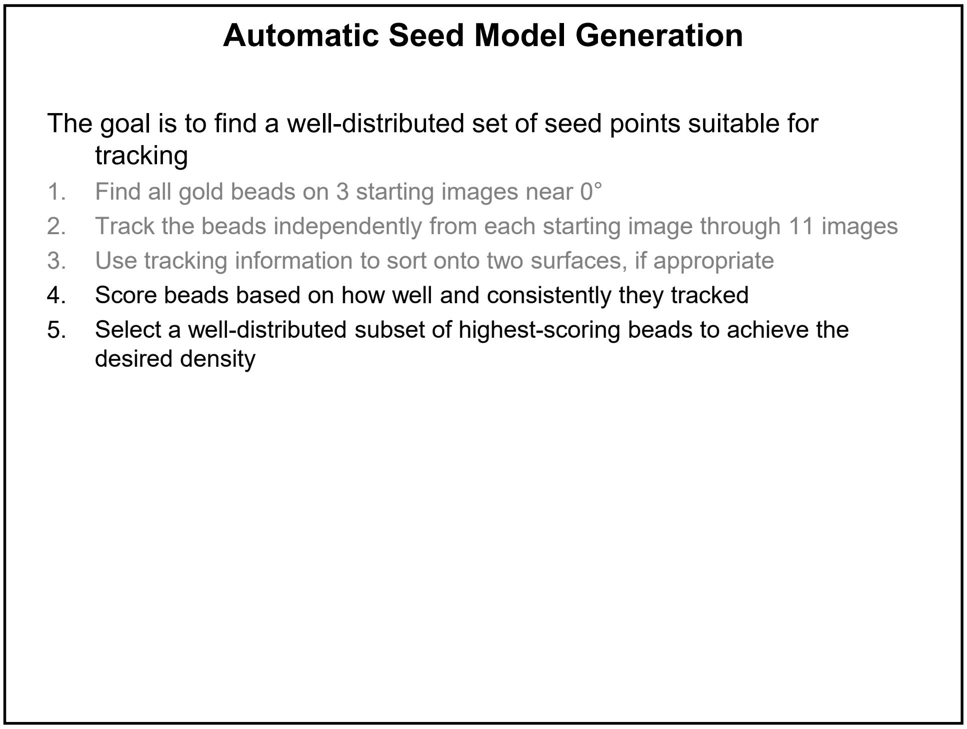
100 nm

2. Track the beads independently from each starting image through 11 images

Side view of tracked contours (Z is projection #)

3. Use tracking information to sort onto two surfaces, if appropriate

Side view of bead positions in 3D



Automatic Seed Model Generation

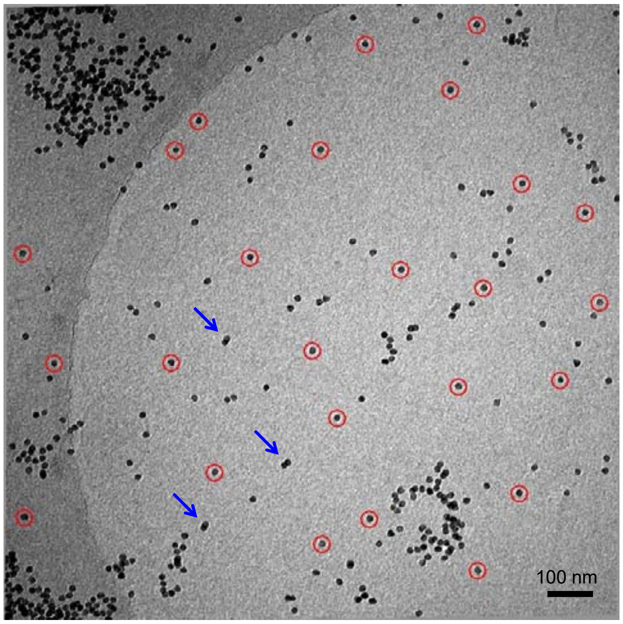
The goal is to find a well-distributed set of seed points suitable for tracking

1. Find all gold beads on 3 starting images near 0°
2. Track the beads independently from each starting image through 11 images
3. Use tracking information to sort onto two surfaces, if appropriate
4. Score beads based on how well and consistently they tracked
5. Select a well-distributed subset of highest-scoring beads to achieve the desired density

Seed Model for Cryo Tilt Series

- Requested 25 beads
- It avoided clusters, distinct beads too close to others, and elongated (overlapping) beads

Tilt series of frozen-hydrated mammalian cell infected with bovine papilloma virus, from Mary Morphew



Major Developments to Support Automated Processing

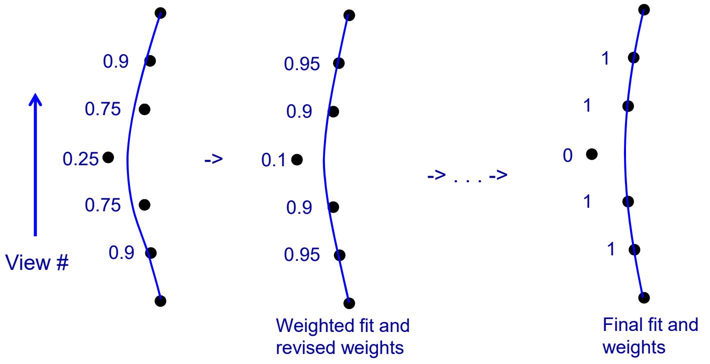
- Automatic seed model selection to get a well-distributed set of fiducials suitable for tracking
- Improvements in bead centering by “Sobel filtering” to enhance bead edges
  - For plastic section data, Sobel centering reduced error by a mean of 20% (0 – 0.25 pixel, n = 15)
  - For cryo data, it reduced error by a mean of 10% (0 – 0.1 pixel, n = 15)

Major Developments to Support Automated Processing

- Automatic seed model selection to get a well-distributed set of fiducials suitable for tracking
- Improvements in bead centering by “Sobel filtering” to enhance bead edges
  - For plastic section data, Sobel centering reduced error by a mean of 20% (0 – 0.25 pixel, n = 15)
  - For cryo data, it reduced error by a mean of 10% (0 – 0.1 pixel, n = 15)
- Robust fitting to reduce/eliminate effects of inevitable bad points

Robust Fitting – A Substitute for Fixing Bad Points

- Robust fitting follows these steps:
  - Fit as usual and get residual for each point
  - Give each point a weight based on how extreme the residual is
  - Fit again, minimizing *weighted* sum of errors
  - Get new residuals, new weights, repeat until convergence
- About 2% of points end up with zero weights, ~5% have weights < 0.5 weight
- If there is a sufficient excess of points, this should give as good an alignment as correcting positions manually

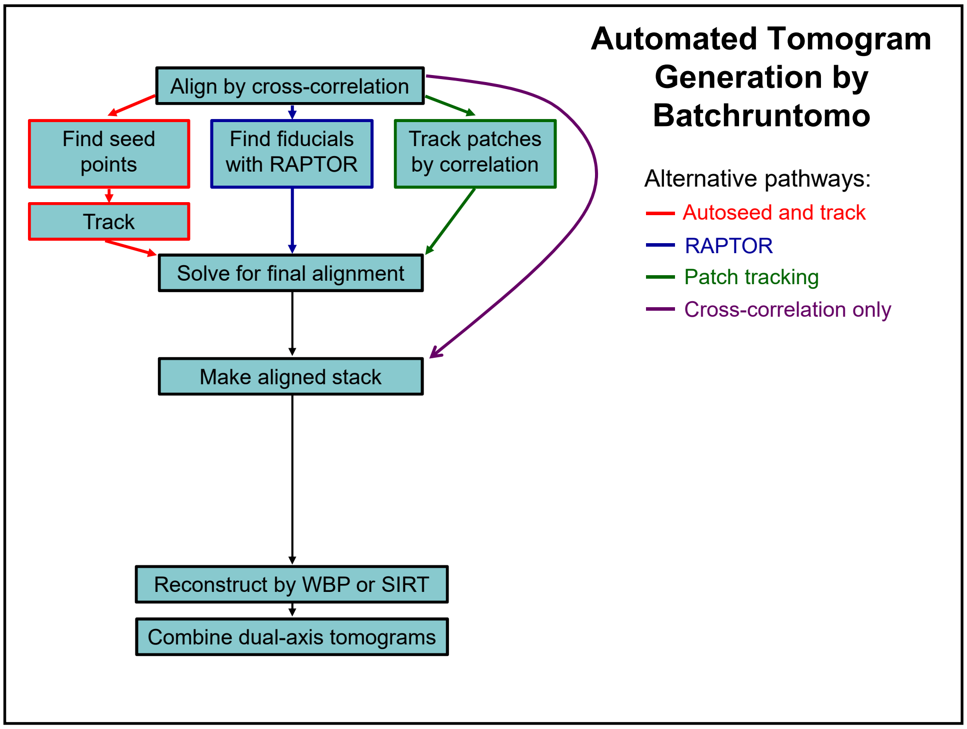


Major Developments to Support Automated Processing

- Automatic seed model selection to get a well-distributed set of fiducials suitable for tracking
- Improvements in bead centering by “Sobel filtering” to enhance bead edges
  - For plastic section data, Sobel centering reduced error by a mean of 20% (0 – 0.25 pixel, n = 15)
  - For cryo data, it reduced error by a mean of 10% (0 – 0.1 pixel, n = 15)
- Robust fitting to reduce/eliminate effects of inevitable bad points
- Detection of structure in specimen, for automating tomogram positioning, the alignment of dual-axis tomograms, and trimming of stained specimens

Major Developments to Support Automated Processing

- Automatic seed model selection to get a well-distributed set of fiducials suitable for tracking
- Improvements in bead centering by “Sobel filtering” to enhance bead edges
  - For plastic section data, Sobel centering reduced error by a mean of 20% (0 – 0.25 pixel, n = 15)
  - For cryo data, it reduced error by a mean of 10% (0 – 0.1 pixel, n = 15)
- Robust fitting to reduce/eliminate effects of inevitable bad points
- Detection of structure in specimen, for automating tomogram positioning, the alignment of dual-axis tomograms, and trimming of stained specimens
- A framework, script, and interface for running tracking, alignment, and reconstruction automatically



Automated Tomogram Generation by Batchruntomo

Align by cross-correlation

Find seed points

Find fiducials with RAPTOR

Track patches by correlation

Track

Solve for final alignment

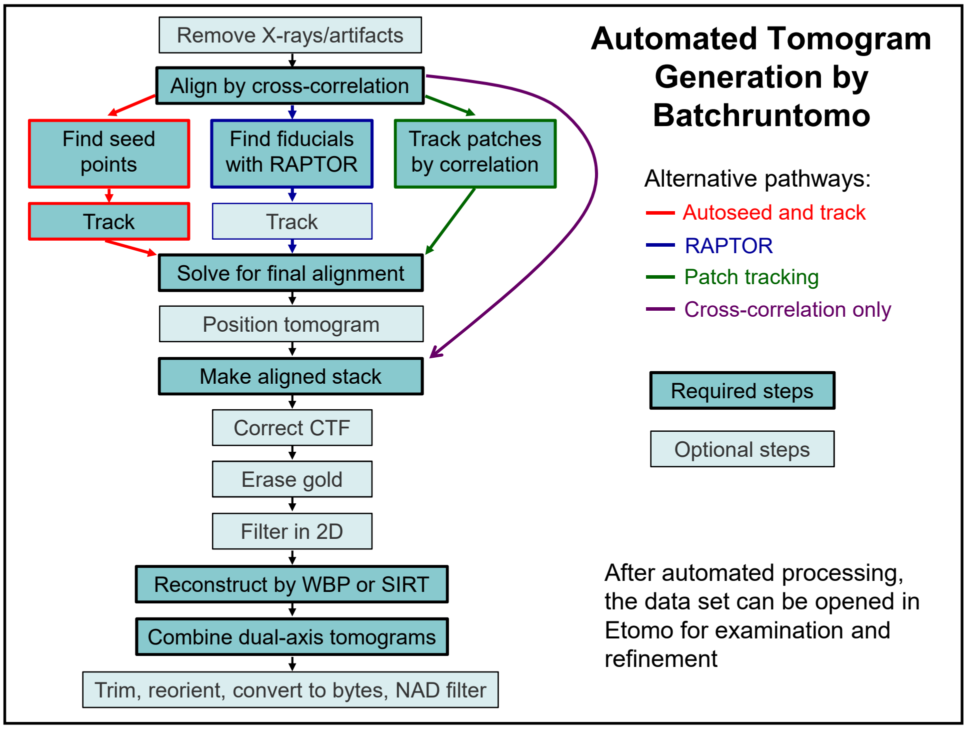
Make aligned stack

Reconstruct by WBP or SIRT

Combine dual-axis tomograms

Alternative pathways:

- Autoseed and track
- RAPTOR
- Patch tracking
- Cross-correlation only



Automated Tomogram Generation by Batchruntomo

Remove X-rays/artifacts

Align by cross-correlation

Find seed points

Find fiducials with RAPTOR

Track patches by correlation

Track

Track

Solve for final alignment

Position tomogram

Make aligned stack

Correct CTF

Erase gold

Filter in 2D

Reconstruct by WBP or SIRT

Combine dual-axis tomograms

Trim, reorient, convert to bytes, NAD filter

Alternative pathways:

- Autoseed and track
- RAPTOR
- Patch tracking
- Cross-correlation only

Required steps

Optional steps

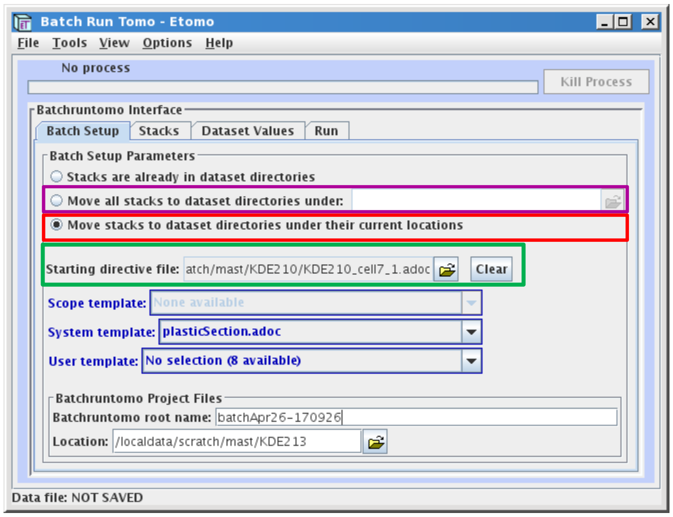
After automated processing, the data set can be opened in Etomo for examination and refinement

It's All Done With Directives

- Directives to control Batchruntomo are key-value pairs in a text file
- Can be options to programs, which can be handled generically
  - comparam.track.beadtrack.RoundsOfTracking = 4
- Can be directions interpreted by Batchruntomo or Etomo
  - runtime.BeadTracking.any.numberOfRuns = 2
- These are the same directives used in templates

Batch Interface: Initial Setup

- Stacks can be moved from where they are into separate directories under one directory, or under their current location(s)



○ Stacks are already in dataset directories

○ Move all stacks to dataset directories under:

● Move stacks to dataset directories under their current locations

Starting directive file: atch/mast/KDE210/KDE210_cell7_1.adoc

Scope template: None available

System template: plasticSection.adoc

User template: No selection (8 available)

Batchruntomo root name: batchApr26-170926

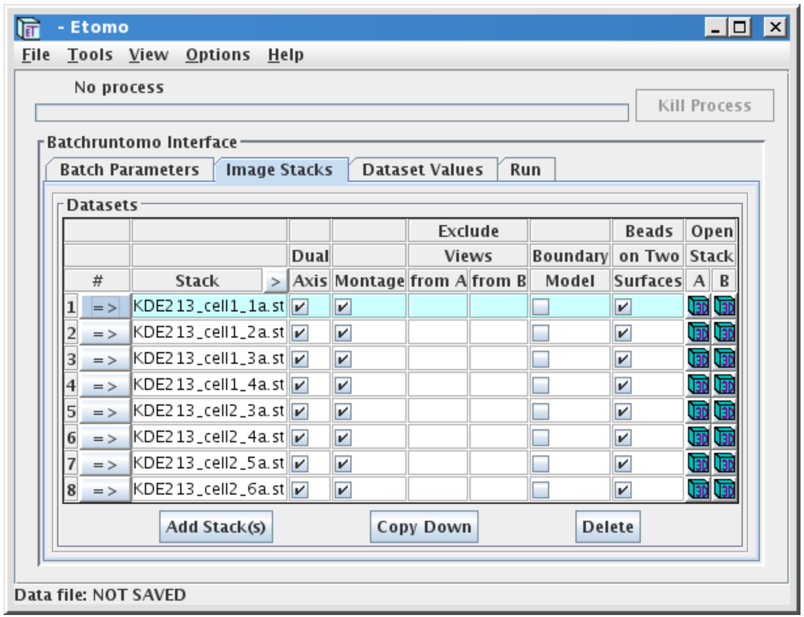
Location: /localdata/scratch/mast/KDE213

Data file: NOT SAVED

- A directive file from a previous run can be used as starting point
- Template files can be chosen

Batch Interface: Adding Stacks

The table contains some parameters most likely to vary among datasets



- Etomo

File Tools View Options Help

No process

Batchruntomo Interface

Batch Parameters | Image Stacks | Dataset Values | Run

Datasets

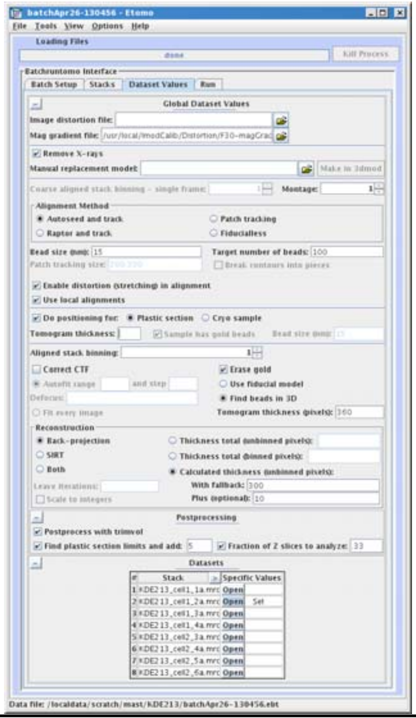
						Exclude			Beads	Open	
				Dual		Views		Boundary	on Two	Stack	
#		Stack	>	Axis	Montage	from A	from B	Model	Surfaces	A	B
1	=>	KDE213_cell1_1a.st		☑	☑			☐	☑	3D	3D
2	=>	KDE213_cell1_2a.st		☑	☑			☐	☑	3D	3D
3	=>	KDE213_cell1_3a.st		☑	☑			☐	☑	3D	3D
4	=>	KDE213_cell1_4a.st		☑	☑			☐	☑	3D	3D
5	=>	KDE213_cell2_3a.st		☑	☑			☐	☑	3D	3D
6	=>	KDE213_cell2_4a.st		☑	☑			☐	☑	3D	3D
7	=>	KDE213_cell2_5a.st		☑	☑			☐	☑	3D	3D
8	=>	KDE213_cell2_6a.st		☑	☑			☐	☑	3D	3D

Add Stack(s)
Copy Down
Delete

Data file: NOT SAVED

Batch Interface: Setting Parameters

- Trying to keep it as simple as possible, so it fits on a screen
- Relying on templates and an "advanced" directive editor for other parameters



batchApr26-130456 - Etomo

File Tools View Options Help

Loading Files

done

Batchruntomo Interface

Batch Setup | Stacks | Dataset Values | Run

Global Dataset Values

Image distortion file:

Mag gradient file: /usr/local/ImodCalib/Distortion/F30-magGrad

☑ Remove X-rays

Manual replacement model:

Coarse aligned stack binning - single frame: 1 Montage: 1

Alignment Method

● Autoseed and track ○ Patch tracking

○ Raptor and track ○ Fiducialless

Bead size (nm): 15 Target number of beads: 100

Patch tracking size: ☐ Break contours into pieces

☑ Enable distortion (stretching) in alignment

☑ Use local alignments

☑ Do positioning for: ● Plastic section ○ Cryo sample

Tomogram thickness: ☑ Sample has gold beads Bead size (nm):

Aligned stack binning: 1

☐ Correct CTF ☑ Erase gold

● Autofit range and step ○ Use fiducial model

Defocus: ● Find beads in 3D

○ Fit every image Tomogram thickness (pixels): 360

Reconstruction

● Back-projection ○ Thickness total (unbinned pixels):

○ SIRT ○ Thickness total (binned pixels):

○ Both ● Calculated thickness (unbinned pixels):

Leave iterations: With fallback: 300

☐ Scale to integers Plus (optional): 10

Postprocessing

☑ Postprocess with trimvol

☑ Find plastic section limits and add: 5 ☑ Fraction of Z slices to analyze: 33

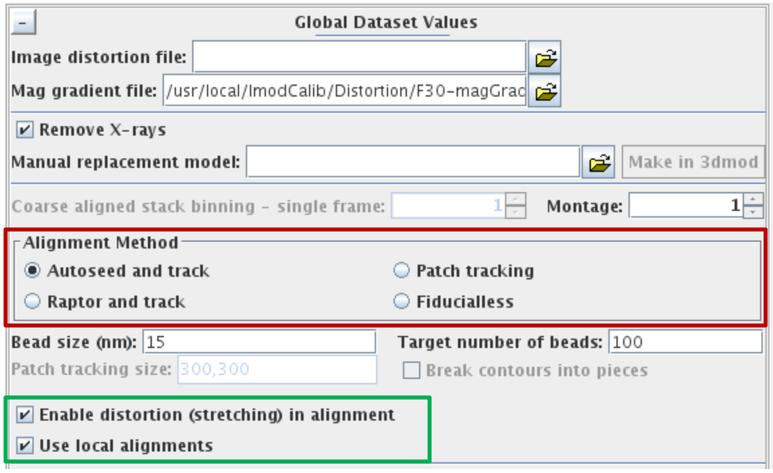
Datasets

#	Stack	>	Specific Values	
1	KDE213_cell1_1a.mrc		Open	
2	KDE213_cell1_2a.mrc		Open	Set
3	KDE213_cell1_3a.mrc		Open	
4	KDE213_cell1_4a.mrc		Open	
5	KDE213_cell2_3a.mrc		Open	
6	KDE213_cell2_4a.mrc		Open	
7	KDE213_cell2_5a.mrc		Open	
8	KDE213_cell2_6a.mrc		Open	

Data file: /localdata/scratch/mast/KDE213/batchApr26-130456.ebt

Batch Interface: Setting Parameters (Top)

- It allows all the choices for alignment method
- The stretching solution and local alignments are **enabled** by the choices here, but Batchruntomo will decide whether the number of fiducials is sufficient for each



Global Dataset Values

Image distortion file:

Mag gradient file: /usr/local/ImodCalib/Distortion/F30-magGrad

☑ Remove X-rays

Manual replacement model: Make in 3dmod

Coarse aligned stack binning - single frame: 1 Montage: 1

Alignment Method

● Autoseed and track ○ Patch tracking

○ Raptor and track ○ Fiducialless

Bead size (nm): 15 Target number of beads: 100

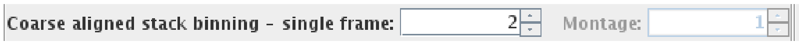
Patch tracking size: 300,300 □ Break contours into pieces

☑ Enable distortion (stretching) in alignment

☑ Use local alignments

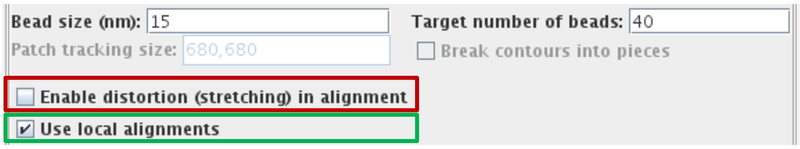
Parameter Settings for CryoET (Top)

- Using binning for the coarse aligned stack is a common way to improve bead tracking with a small pixel size and/or low SNR



Coarse aligned stack binning - single frame: 2 Montage: 1

- Stretching would generally not be enabled but local alignments can be helpful for large areas with enough beads



Bead size (nm): 15 Target number of beads: 40

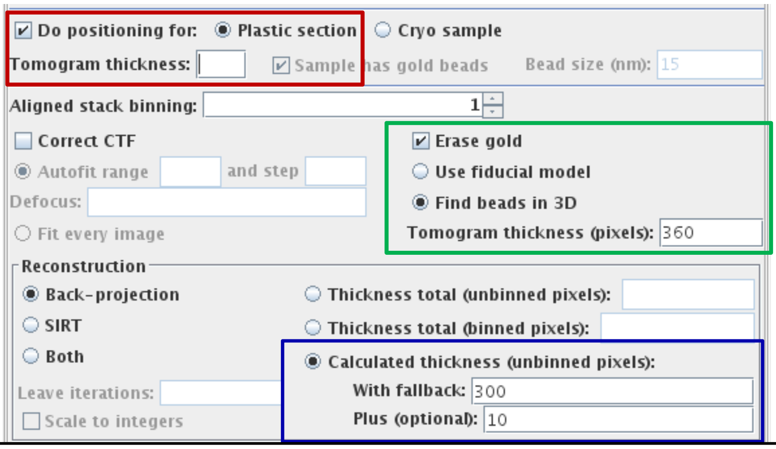
Patch tracking size: 680,680 □ Break contours into pieces

□ Enable distortion (stretching) in alignment

☑ Use local alignments

Batch Interface: Setting Parameters (Middle)

- With positioning for a plastic section, the thickness is optional
- But thickness is required when finding beads in 3D for erasing gold
- The "calculated thickness" is usually best for plastic sections
  - It is based on distance between beads on surfaces if no positioning is done
  - The fallback is used if either analysis gives < 40% of this thickness



☑ Do positioning for: ● Plastic section ○ Cryo sample

Tomogram thickness: ☑ Sample has gold beads Bead size (nm): 15

Aligned stack binning: 1

□ Correct CTF

● Autofit range and step

Defocus:

○ Fit every image

☑ Erase gold

○ Use fiducial model

● Find beads in 3D

Tomogram thickness (pixels): 360

Reconstruction

● Back-projection

○ SIRT

○ Both

Leave iterations:

□ Scale to integers

○ Thickness total (unbinned pixels):

○ Thickness total (binned pixels):

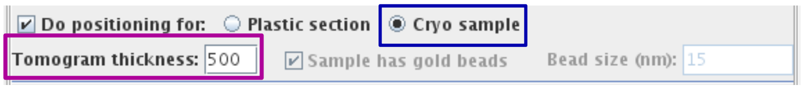
● Calculated thickness (unbinned pixels):

With fallback: 300

Plus (optional): 10

Parameter Settings for CryoET (Middle)

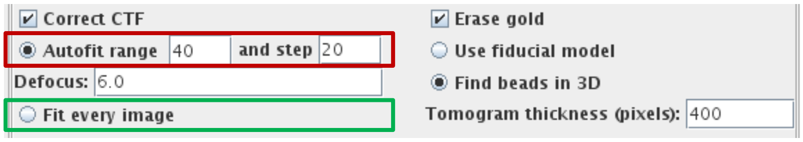
- Positioning must be done with the Cryo sample option and a thickness must be entered
  - If patch tracking is used and there are gold beads, this should be indicated



☑ Do positioning for: ○ Plastic section ● Cryo sample

Tomogram thickness: 500 ☑ Sample has gold beads Bead size (nm): 15

- CTF correction can be done by having Ctfplotter autofit to ranges of angles or fit every image
  - Voltage, spherical aberration, and noise configuration file must be set in a template or starting batch file



☑ Correct CTF

● Autofit range 40 and step 20

Defocus: 6.0

○ Fit every image

☑ Erase gold

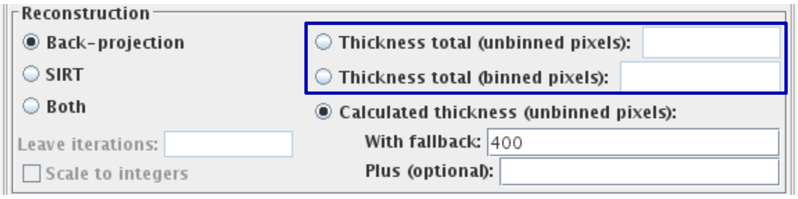
○ Use fiducial model

● Find beads in 3D

Tomogram thickness (pixels): 400

Parameter Settings for CryoET (Middle)

- A total reconstruction thickness must be entered if positioning is not done



Reconstruction

● Back-projection

○ SIRT

○ Both

Leave iterations:

□ Scale to integers

○ Thickness total (unbinned pixels):

○ Thickness total (binned pixels):

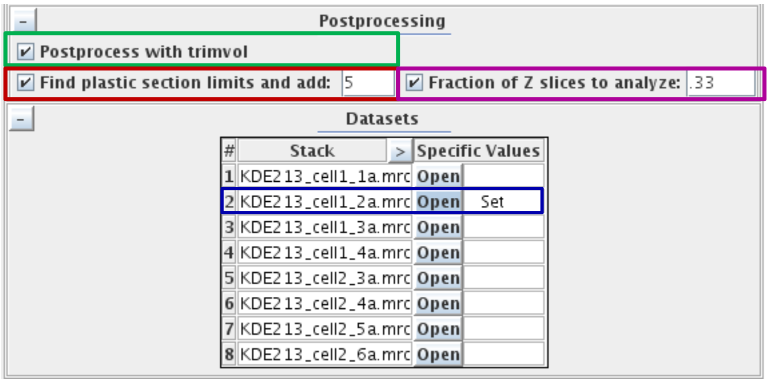
● Calculated thickness (unbinned pixels):

With fallback: 400

Plus (optional):

Batch Interface: Setting Parameters (Bottom)

- With plastic sections, postprocessing can trim the tomogram if “Find plastic section limits” is checked, and convert to bytes if “Fraction of Z slices” is checked
- For cryo, postprocessing will just rotate the full tomogram so slices are in X/Y
- In the Datasets table, you can press Open to open a complete Dataset Values dialog specific to one data set; “Set” indicates it has its
  - “Set” indicates there are specific values for a data set



Postprocessing

☑ Postprocess with trimvol

☑ Find plastic section limits and add: 5

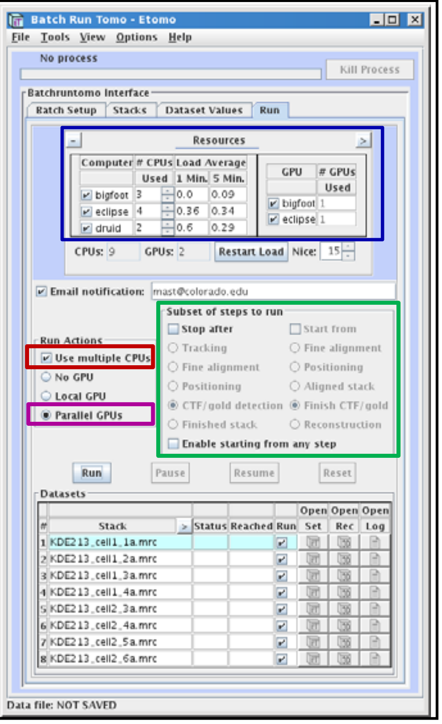
☑ Fraction of Z slices to analyze: .33

Datasets

#	Stack	Specific Values	
1	KDE213_cell1_1a.mrc	Open	
2	KDE213_cell1_2a.mrc	Open	Set
3	KDE213_cell1_3a.mrc	Open	
4	KDE213_cell1_4a.mrc	Open	
5	KDE213_cell2_3a.mrc	Open	
6	KDE213_cell2_4a.mrc	Open	
7	KDE213_cell2_5a.mrc	Open	
8	KDE213_cell2_6a.mrc	Open	

Batch Interface: Running

- In the Resources area, a table for computers with CPUs opens if Use multiple CPUs is selected, and table for GPUs opens if Parallel GPUs is checked
- The default is to run every set to completion, but you can stop, interact, and restart



Computer	# CPUs Used	Load Average 1 Min.	Load Average 5 Min.
☑ bigfoot	3	0.0	0.09
☑ eclipse	4	0.36	0.34
☑ druid	2	0.6	0.29

GPU	# GPUs Used
☑ bigfoot	1
☑ eclipse	1

CPUs: 9 GPUs: 2 Nice: 15

☑ Email notification: mast@colorado.edu

Run Actions
- ☑ Use multiple CPUs
- ○ No GPU
- ○ Local GPU
- ● Parallel GPUs

Subset of steps to run

□ Stop after	□ Start from
○ Tracking	○ Fine alignment
○ Fine alignment	○ Positioning
○ Positioning	○ Aligned stack
● CTF/gold detection	● Finish CTF/gold
○ Finished stack	○ Reconstruction

□ Enable starting from any step

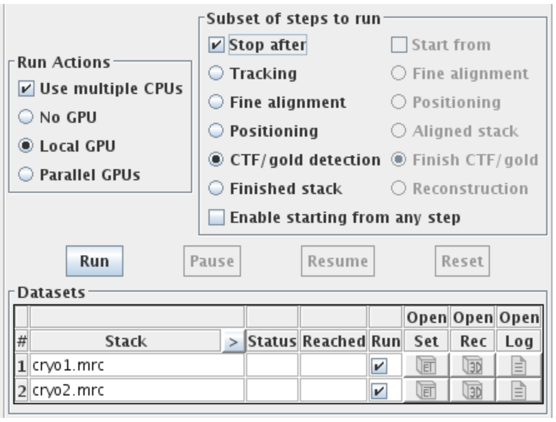
Datasets

#	Stack	Status	Reached	Run	Open Set	Open Rec	Open Log
1	KDE213_cell1_1a.mrc			☑			
2	KDE213_cell1_2a.mrc			☑			
3	KDE213_cell1_3a.mrc			☑			
4	KDE213_cell1_4a.mrc			☑			
5	KDE213_cell2_3a.mrc			☑			
6	KDE213_cell2_4a.mrc			☑			
7	KDE213_cell2_5a.mrc			☑			
8	KDE213_cell2_6a.mrc			☑			

Data file: NOT SAVED

Batch Processing for CryoET: Running

- A common stopping point would be after CTF estimation and detection of gold
  - Both can be checked and then processing can be restarted to do CTF correction and gold erasing



Run Actions
- ☑ Use multiple CPUs
- ○ No GPU
- ● Local GPU
- ○ Parallel GPUs

Subset of steps to run

☑ Stop after	□ Start from
○ Tracking	○ Fine alignment
○ Fine alignment	○ Positioning
○ Positioning	○ Aligned stack
● CTF/gold detection	● Finish CTF/gold
○ Finished stack	○ Reconstruction

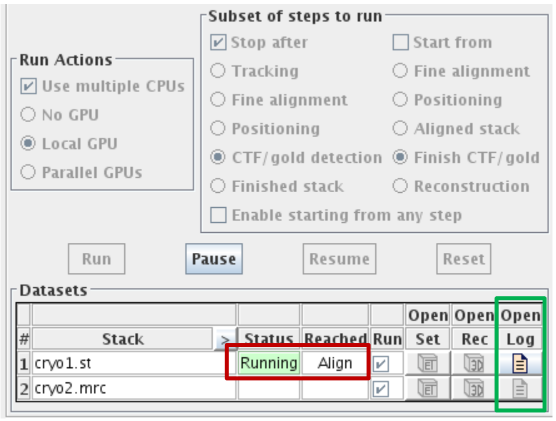
□ Enable starting from any step

Datasets

#	Stack	Status	Reached	Run	Open Set	Open Rec	Open Log
1	cryo1.mrc			☑			
2	cryo2.mrc			☑			

Batch Processing for CryoET: Running

- The status table shows the current state of the running job including the last step completed
- The Batchruntomo log can be opened with the Open Log button as soon as it is available



Run Actions

☑ Use multiple CPUs
○ No GPU
● Local GPU
○ Parallel GPUs

Subset of steps to run

☑ Stop after	□ Start from
○ Tracking	○ Fine alignment
○ Fine alignment	○ Positioning
○ Positioning	○ Aligned stack
● CTF/gold detection	● Finish CTF/gold
○ Finished stack	○ Reconstruction

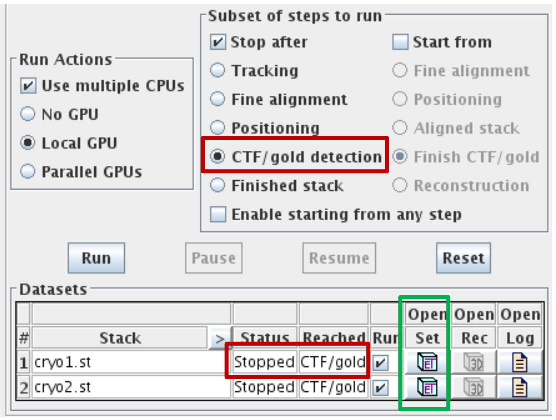
□ Enable starting from any step

Datasets

#	Stack	>	Status	Reached	Run	Open Set	Open Rec	Open Log
1	cryo1.st		Running	Align	☑			
2	cryo2.mrc				☑			

Batch Processing for CryoET: Running

- All jobs are taken to the selected step for stopping
- Data sets can be opened in the reconstruction interface with Open Set when processing is stopped



Run Actions

☑ Use multiple CPUs
○ No GPU
● Local GPU
○ Parallel GPUs

Subset of steps to run

☑ Stop after	□ Start from
○ Tracking	○ Fine alignment
○ Fine alignment	○ Positioning
○ Positioning	○ Aligned stack
● CTF/gold detection	● Finish CTF/gold
○ Finished stack	○ Reconstruction

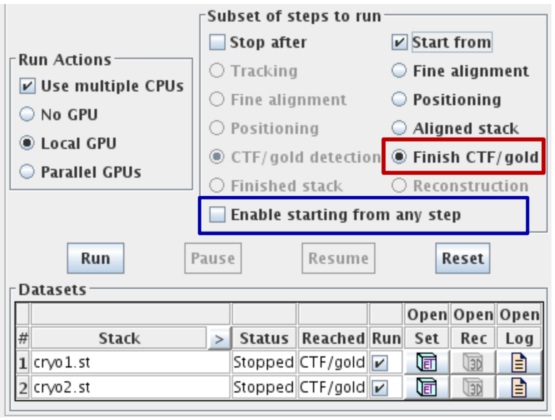
□ Enable starting from any step

Datasets

#	Stack	>	Status	Reached	Run	Open Set	Open Rec	Open Log
1	cryo1.st		Stopped	CTF/gold	☑			
2	cryo2.st		Stopped	CTF/gold	☑			

Batch Processing for CryoET: Running

- After interactive work, you can start from the next step or repeat an earlier step
  - You can skip ahead, over steps done interactively, with Enable starting from any step



Run Actions
☑ Use multiple CPUs
○ No GPU
● Local GPU
○ Parallel GPUs

Subset of steps to run

□ Stop after	☑ Start from
○ Tracking	○ Fine alignment
○ Fine alignment	○ Positioning
○ Positioning	○ Aligned stack
● CTF/gold detection	● Finish CTF/gold
○ Finished stack	○ Reconstruction

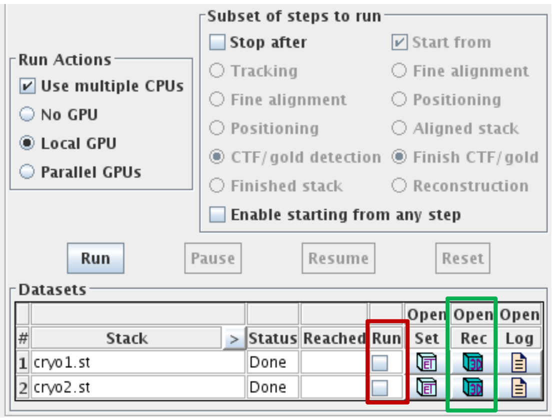
□ Enable starting from any step

Datasets

						Open	Open	Open
#	Stack	>	Status	Reached	Run	Set	Rec	Log
1	cryo1.st		Stopped	CTF/gold	☑			
2	cryo2.st		Stopped	CTF/gold	☑			

Batch Processing for CryoET: Running

- When a data set is finished, the Run button is turned off
  - You have to turn Run back on for at least one set to enable starting from a chosen point
- The tomogram can be opened in 3dmod with Open Rec when one is available



Run Actions
☑ Use multiple CPUs
○ No GPU
● Local GPU
○ Parallel GPUs

Subset of steps to run

□ Stop after	☑ Start from
○ Tracking	○ Fine alignment
○ Fine alignment	○ Positioning
○ Positioning	○ Aligned stack
● CTF/gold detection	● Finish CTF/gold
○ Finished stack	○ Reconstruction

□ Enable starting from any step

Datasets

						Open	Open	Open
#	Stack	>	Status	Reached	Run	Set	Rec	Log
1	cryo1.st		Done		□			
2	cryo2.st		Done		□			

Customizing Directives

- Almost every program option can be controlled in batch processing with a directive
- The master table of directives is available with:

  ```
  imodhelp –d directives
  ```

  - But for programs listed there, any option can be specified with a comparam, not just the ones in the table
- To do CTF correction with the batch interface, you will need to supply microscope parameters as directives
  - A Scope template is a logical way to do this

Two Ways to Make a Scope Template for CTF Correction

1. Through the template editor:
   - In any data set open in Etomo, select File – Templates – Save Scope Template
   - Check “Show unchanged” in the Control Panel section
   - Turn on “Arguments to Copytomocoms” to open that section if necessary
   - Select “Microscope voltage” and “Spherical aberration” and fill in values
   - Select “Noise configuration file” and choose the file
   - Turn off other selected items if any – check “Show included” to see them better
   - Press the Save button and save it to a file with extension “.adoc”
2. With a text editor:
   - Make a file with extension “.adoc” with these entries modified for your case:

     ```
     setupset.copyarg.voltage=300
     setupset.copyarg.Cs=2.0
     setupset.copyarg.ctfnoise=full_path_to_noise_file
     ```

- In either case, the file goes in /usr/local/ImodCalib/SystemTemplate

THIS PAGE INTENTIONALLY LEFT BLANK

Reconstructing a Cryo Tomogram using BRT

In this exercise, you will build a single tomogram to learn how to use the BatchRunTomo interface in Etomo.

1) `cd $WORKSHOP_HOME/IMOD_Labs/cryo-BRT`

2) `etomo`

3) Press **Batch Tomograms** on the **Front Page**. The interface opens with the Batch Setup tab. The initial radio buttons allow stacks to be moved from elsewhere into automatically created data set directories. Your **stacks are already in dataset directories**, so there is no need to change the default selection.

   Select *cryoSample.adoc* as the **System Template**. This template turns on specific settings that work well for most cryo data sets.

   At the bottom, two fields allow one to change the root name that will be used as a prefix for all the batch processing files, and to select a different Location in which various batch project files will be kept. Again, there is no need to change the defaults here.

4) Switch to the **Stacks** tab. Press **Add Stack(s)**. Select *cryo.st* and press **Open**. Notice how the program automatically detects this is not Dual Axis or a Montage. Press the **3D** icon under **Open Stack A**. Here is where you check to see if you need a Boundary Model or to Exclude Views. In this simple case, neither option is needed.

5) Switch to the **Dataset Values** tab. We need to choose the **Image distortion File** *27.5kGIF2007-03-24.idf*. Select **Remove X-Rays**. Enter **15** for **Bead size (nm)** and **15** for **Target number of beads**. There are only 12 beads in this case, but it doesn't hurt to estimate a little high. Select **Do positioning for Cryo sample**. Enter a tomogram thickness of **250**. In the Reconstruction section, the selection of the **Calculated thickness (unbinned pixels)** is appropriate because it will be doing positioning, and an entry for **With fallback** is required with this choice in case the positioning comes out with too low a thickness. Enter **250** here. Also enter **16** for **Plus (optional)**. This will allow some extra thickness that can be trimmed away in the postprocessing step, which we will do manually once the tomogram is built automatically.

6) Switch to the **Run** Tab. Increase the **Resource** table # **CPUs Used** to **4**, then press **Save As Defaults**. Press **Run**. Directive files will be written for each data set and the processing will begin. Watch the **Status** output in the table at the bottom. Pausing, killing, and resuming are possible but more complicated than for ordinary processing; see Running the Data Sets in the Batch Guide for details.

   Essential items like mean residual values and warnings appear in the **Project Log** during the processing. At any time after a set begins being processed, you can see the full output from the batch program by pressing the button in the Open Log column. Once a set has finished or reached a stopping point, it can be opened in the Etomo reconstruction interface with the button in the Open Set or Open Rec column. You can see that if you turned on Stop after, the default choice would be to stop after the gold detection, which might be useful if you were erasing gold. All data sets would run to that point—regardless of whether those steps were actually run for a particular data set. When you restart, both sets would restart after that point, unless an earlier point were selected.

7) Once the **Status** of the tomogram has changed to **Done**, click on **Open Set**. This will open the Etomo interface. Etomo has no a priori knowledge about what has happened during batch processing, so all status values show Not Started. Go to the **Post-processing** section and press **3dmod Full Volume**. Page through the tomogram and choose the lowest and highest slices you would like to keep. Put them into the **Volume Trimming** table under **Z min** and **Z max**. Use the rubber band tool and the **Lo** and **Hi** boxes to select a small region in X, Y, and Z for **Scaling.** Press **Get XYZ Sub-Area from 3dmod**. Press **Trim Volume** and when done, **3dmod Trimmed Volume**. Press **Done**.

8) On the Clean Up section, you will notice that you cannot Archive Original Stack. This is because the stack gets archived automatically after X-ray removal in BatchRunTomo. Instead, select all intermediate files and press **Delete Selected** and then **Done**. Exit Etomo.

THIS PAGE INTENTIONALLY LEFT BLANK

June 27 (Tues)

8:00 Security

Questions

Reconstruct Difficult Cryo Tomograms Using BatchRunTomo

9:00

CTF Correction with IMOD

10:00 David

Coffee Break

11:00 CTF Correction

12:00

Lunch

1:00 Optional Tour of Rocky Mountain Labs: Ken Pekoc

Denoising Cryotomograms with IMOD

2:00 David

Walkthrough Nonlinear Anisotropic Diffusion

Cindi

3:00

Coffee Break

Basic Modeling Tips/Tricks Demo

4:00 David

SerialEM for Cryo Tomography: Cindi

Using Direct Detectors with SerialEM

5:00 David

6:00

7:00

8:00

9:00

THIS PAGE INTENTIONALLY LEFT BLANK

Reconstructing Difficult Cryo Data using BRT

In this exercise, you will use BatchRunTomo (BRT) to build 3 cryo tomograms that have typical issues found in cryo data. Normally, BRT would be used to run many datasets of the same sample type and imaging conditions. Most people will run up one dataset to find appropriate parameters and then process the remaining data in BRT using those parameters. This practice lab is intended to introduce issues that may crop up now and then and is not realistic to how BRT is used on a day-to-day basis.

1) `cd $WORKSHOP_HOME/IMOD_Labs/cryo-practice`

2) `etomo`

3) Press **Batch Tomograms** on the **Front Page**. These data are all within this single directory, so we need to change to **Move stacks to dataset directories under their current location**.

   Select the file chooser for **Starting directive file** and use *Cryo-Practice-Tutorial.adoc*. This file contains information about the data being collected in two directions about zero degrees, which can affect alignment. Select *cryoSample.adoc* as the **System Template**.

4) Switch to the **Stacks** tab. Press **Add Stack(s)**. Highlight all 3 stacks (*Series4.mrc*, *Series5.mrc*, and *Series6.mrc*) and press **Open**.

For *Series4.mrc*, press the **3D** icon under **Open Stack A**. Many people choose not to use gold fiducials found on the carbon. To exclude these fiducials, check **Boundary Model** in the **Datasets table** for *Series4.mrc*. Go to the central slice (**21**) and middle-click around the region of the hole so that you exclude the carbon. You do not need to be exact and you can do this with about 10 points. Press **s** to save your model.

Open *Series5.mrc* and movie through the series. There is a little bit of carbon on the central slice, but not enough to worry about. Instead, notice at high tilt that some of the views are no longer on the area of interest. In the **Datasets** table under **Exclude Views from A**, put **59-61** on the *Series5.mrc* line.

Open *Series6.mrc.* You'll notice these data have extremely good contrast. This is because they were acquired using the Volta Phase Plate. Check Boundary Model and draw a model to exclude the carbon at the bottom of the image. Save your model.

5) Switch to the **Dataset Values** tab. You may have noticed that these data are relatively different from each other. They have different gold fiducial sizes and fiducial numbers. We will first set up the Global Dataset Values and then Specific Values for each dataset as needed. Select **Remove X-Rays**. Change **Coarse aligned stack binning – single frame** to **2**. This will help with the automatic bead finding. Enter **10** for **Bead size (nm)** and **50** for **Target number of beads**. In the **Positioning** area, enter a **Tomogram Thickness** of **1500**. Because these data are relatively large, to save computing time, we are going to bin the final tomogram by changing **Aligned stack binning** to **4**. In the **Reconstruction** section, select **Calculated thickness (unbinned pixels),** set **With Fallback** to **1500,** and set **Plus (optional)** to **20**.

In the **Datasets** table at the bottom, press **Open** next to *Series4.mrc*. Look at the tilt-series you still have open. This data set has a lot of gold, and the size of the fiducials is 5 nm. Change **Bead size (nm)** to **5** and **Target number of beads** to **75**. Check **Use local alignments**. We cannot **Enable distortion (stretching)** in alignment because the fiducials are clearly not distributed in Z. Press **OK** and close all *Series4.mrc* 3dmod windows.

In the **Datasets** table at the bottom, press **Open** next to *Series5.mrc*. Look at the tilt-series you still have open. Check **Use local alignments**. We cannot **Enable distortion (stretching)** in alignment because the fiducials are clearly not distributed in Z. Press **OK** and close all *Series5.mrc* 3dmod windows.

Series6.mrc has much less than 50 gold beads, so change **Target number of beads** to **15**. Press **OK** and close *Series6.mrc* 3dmod windows.

6) Switch to the **Run** Tab. Make sure the **Resources** table **shows # CPUs Used** as **4** and under **Run Actions** check **Use multiple CPUs**. Under the **Subset of steps to run** area, check **Stop after** and select **Fine alignment**. Press **Run**. It may take a while on these laptops, be patient.

7) Once the **Status** of *Series4.st* has changed to **Stopped** and **Reached** has changed to **Align**, press **Open Set**. This will open the Reconstruction interface. Go to **Fine Alignment** and press **Compute Alignment** followed by **View/Edit Fiducial Model.** You should see 2-3 instances where the contour points do not follow the gold fiducial properly. Delete these contours by left-clicking to select and pressing **Shift+D** on the keyboard (You should see 2-3 of these). Save your model (**s**). On **Bead Fixer**, press **Save & Run Tiltalign**. Notice the improvement with your mean residual values as reported in the **Project Log.** Press **Done** and in Etomo, go to **File / Close** and select **Yes** to close 3dmod. You should now have only the BatchRunTomo interface open.

8) Once the **Status** of *Series5.st* has changed to **Stopped** and **Reached** has changed to **Align**, press **Open Set**. Go to **Fine Alignment** and press **Compute Alignment** followed by **View/Edit Fiducial Model.** You'll notice that the excluded views (59-61) are still present, but are being ignored during alignment. A way to avoid this would be to add "runtime.Preprocessing.any.removeExcludedViews=1" to the starting directive file. This would create a new stack that no longer contains the views you wish to ignore.
   Movie through the views and if you see any contours where the model is not actually following a gold bead, select it by left-clicking and then use the hotkey **Shift+D** to delete the entire contour. On **Bead Fixer**, press **Save & Run Tiltalign**. Notice your residual errors are quite good. Press **Done** and close 3dmod windows. In Etomo, go to **File / Close** and you should now have only the BatchRunTomo interface open.

9) Once the **Status** of *Series6.st* has changed to **Stopped** and **Reached** has changed to **Align**, press **Open Set**. Go to **Fine Alignment** and press **Compute Alignment**. In the **Project Log**, you will see that there are no local alignment values because we cannot use that feature with such few gold beads. Press **View/Edit Fiducial Model.** You should see that several of the fiducials fall off on views **1** and **2**. This can be fixed by left-clicking on each point and then right-clicking to move it to the appropriate gold bead. Since there are only 15 fiducials, move them all now to their correct positions. Press **Save & Run Tiltalign** and notice the improvement in your residual error. Press **Done** and close 3dmod windows. In Etomo, go to **File / Close** and you should now have only the BatchRunTomo interface open.

10) In the BatchRunTomo Interface press **Start from** and select **Positioning**. The program should already be set to **Stop after Positioning**. We will likely have to manually change positioning values. Press **Run**.

For each data set, press **Open Set** and go to **Tomogram Positioning**. Press **Create Whole Tomogram**. When done, press **Create Boundary Model**. We will now use Slicer to model the minimum of 2 sets of lines to orient the tomogram in space. In ZaP, left-click on the bacteria cell near the top of the tomogram and open Slicer. Make **X rotation 90** and **Img 20**. Place 2 lines on the Slicer window (\) that encompass the top and bottom surfaces. It is easy when there is a cell in that region of the volume, but may be difficult if there is only ice. You may need to adjust the **Img** setting for best visualization. Once those lines are drawn, use **View axis position** in Slicer to move to about the center of the volume. Draw 2 lines here. Use **View axis position** to move to the bottom of the volume and draw 2 lines there. Sometimes it is impossible to draw 3 sets of lines. If that is the case, you can use 2 sets of lines, but they must be a reasonable distance apart. Save your model (**s**). In Etomo, press **Compute Z Shift & Pitch Angles**. When done, press **Create Final Alignment**, then **Done**. Close Etomo tab and all 3dmod windows. The BatchRunTomo interface should still be open.

Repeat this process for the other 2 datasets. On *Series6*, you will notice that the tomogram is not thick enough to contain the entire bacterium. Change **Positioning tomogram thickness** to **2500** and try again.

11) In the BatchRunTomo Interface select **Start from Aligned Stack** and turn off **Stop after**. Press **Run**. The tomograms should be reconstructed through to the end with no problems.

12) For each data set, go to the **Post-processing** section and press **3dmod Full Volume**. Page through the tomogram and choose the lowest and highest slices you would like to keep and enter those values in **Z min** and **Z max** or use the rubber band tool to **Get XYZ Volume Range from 3dmod** for the **Volume Trimming** section. Use the rubber band tool and the **Lo** and **Hi** boxes to select a small region in X, Y, and Z for **Scaling** that contains representative material but that does not contain any gold particles or other non-cellular contamination. Press **Get XYZ Sub-Area from 3dmod**. Press **Trim Volume** and close the 3dmod window. When done press **3dmod Trimmed Volume**. Press **Done**.

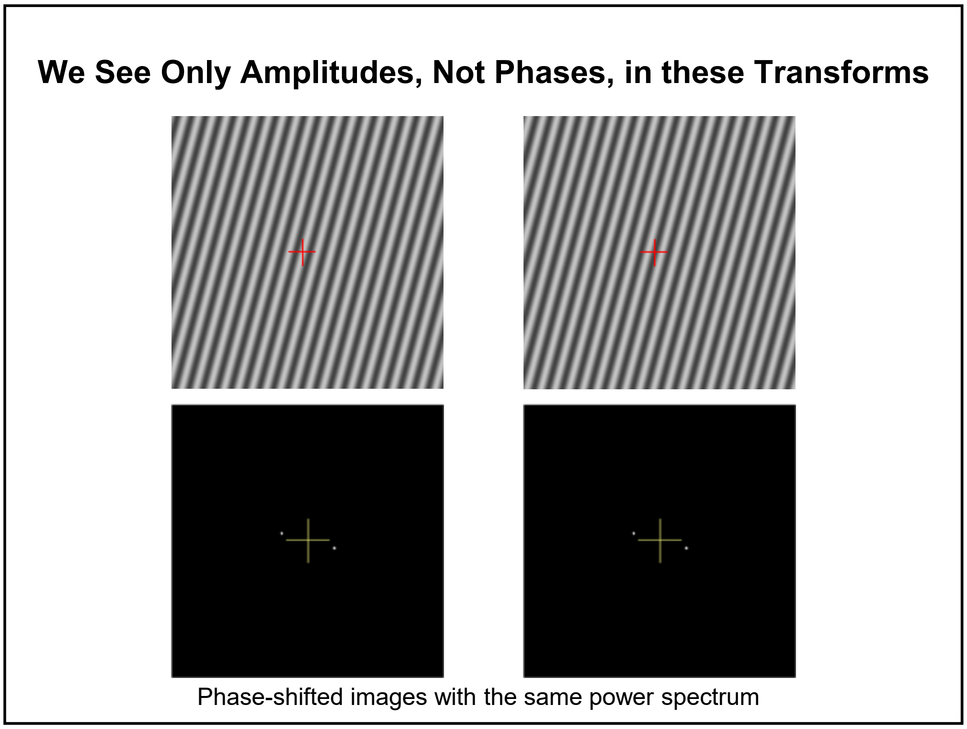
    On the **Clean Up** section, you will notice that you cannot Archive Original Stack. This is because the stack gets archived automatically after X-ray removal in BatchRunTomo. Instead, select all intermediate files and press **Delete Selected** and then **Done**.

    After you have finished all 3 data sets, you can close Etomo completely.

CTF Correction with IMOD

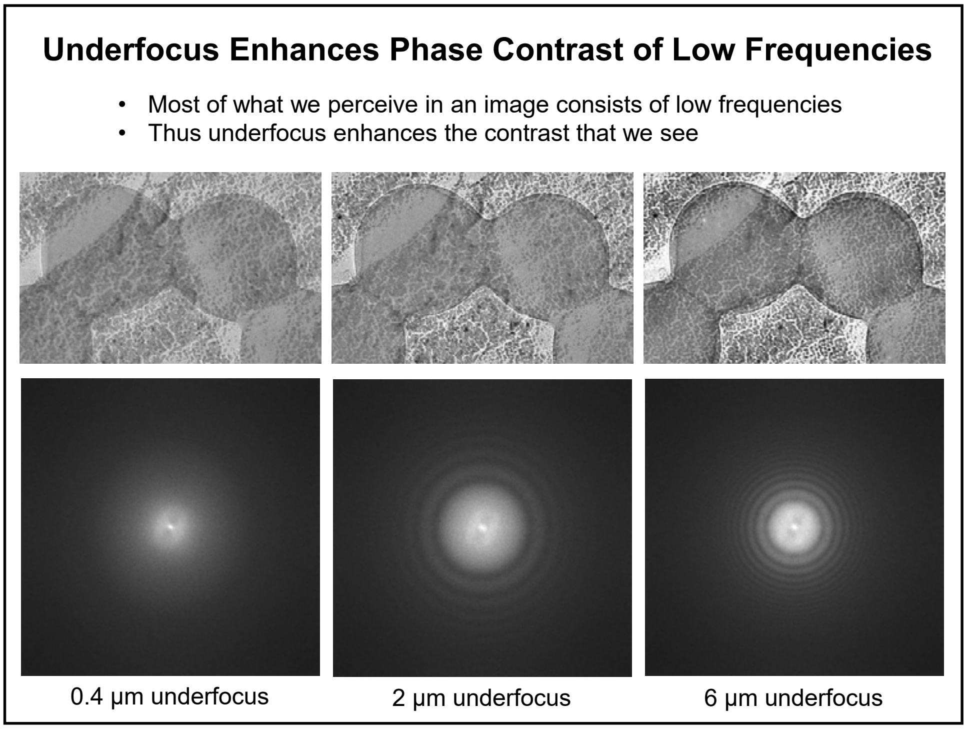
CTF Correction

- When microscope is operated in underfocus to produce phase contrast, the contrast is inverted in some spatial frequency ranges



We See Only Amplitudes, Not Phases, in these Transforms

Phase-shifted images with the same power spectrum



Underfocus Enhances Phase Contrast of Low Frequencies

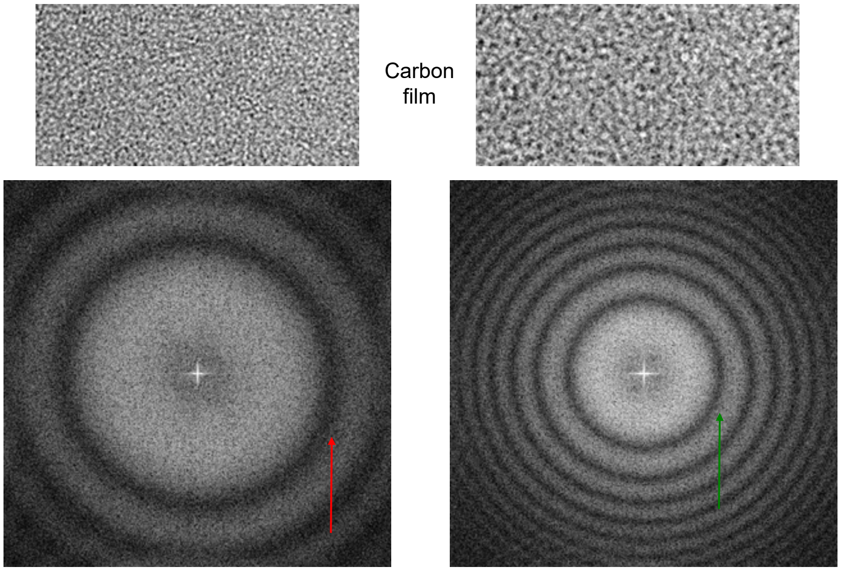
- Most of what we perceive in an image consists of low frequencies
- Thus underfocus enhances the contrast that we see

0.4 µm underfocus

2 µm underfocus

6 µm underfocus

Underfocus Also Produces Thon Rings in Power Spectra

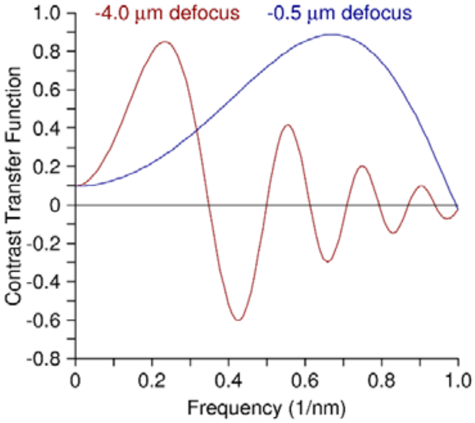


1.5 µm underfocus
First zero at 0.56/nm (1/1.8 nm)

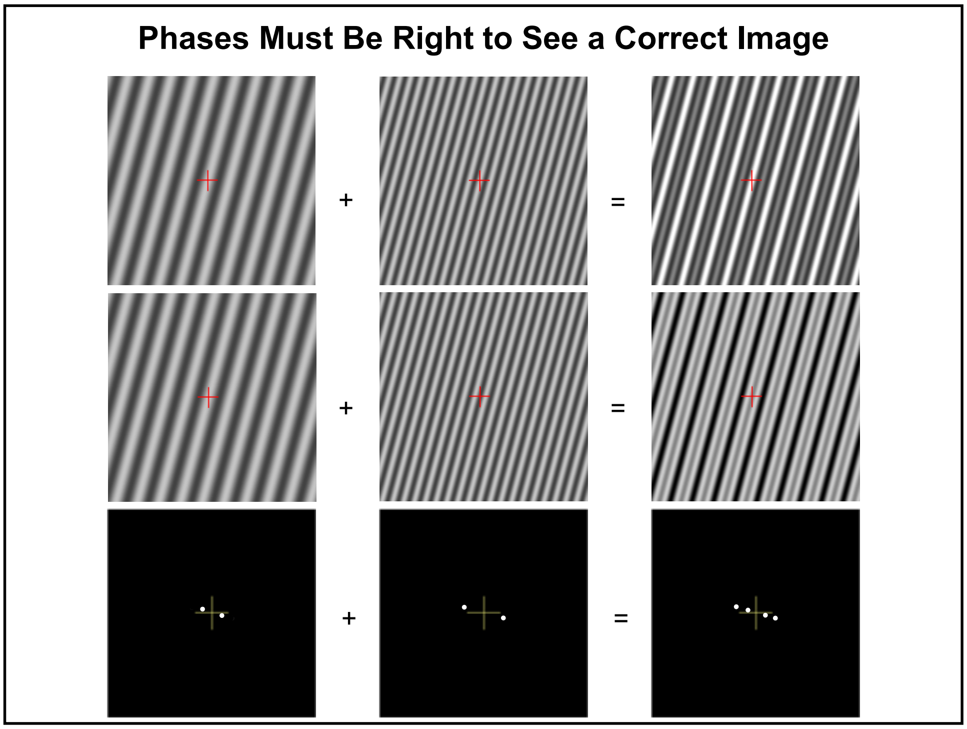
5 µm underfocus
First zero at 0.32/nm (1/3.1 nm)

The CTF Produces Contrast Inversions

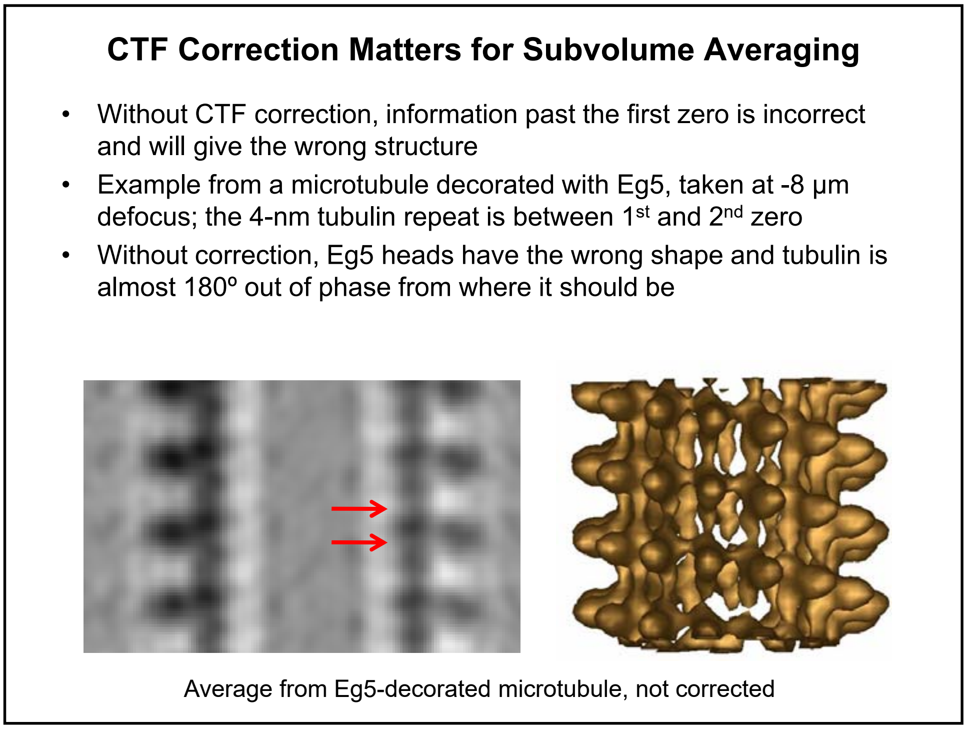
- Thon rings from oscillations in the CTF are an intrinsic result of the physics of phase image formation with underfocus
- In fact the CTF goes THROUGH zero, so contrast is inverted for frequencies between the first and second zero, third and fourth zero, etc.



Example:
300 kV FEG
0.5 nm pixel size
Amplitude contrast 0.1
First inversion for -4 µm:
1/2.8 nm to 1/2.0 nm



Phases Must Be Right to See a Correct Image



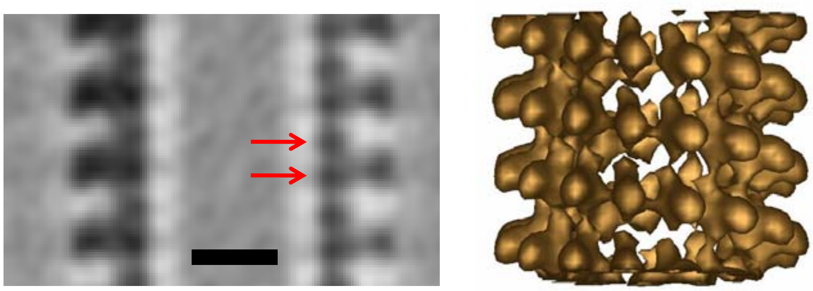
CTF Correction Matters for Subvolume Averaging

- Without CTF correction, information past the first zero is incorrect and will give the wrong structure
- Example from a microtubule decorated with Eg5, taken at -8 µm defocus; the 4-nm tubulin repeat is between 1st and 2nd zero
- Without correction, Eg5 heads have the wrong shape and tubulin is almost 180º out of phase from where it should be

Average from Eg5-decorated microtubule, not corrected

CTF Correction Matters for Subvolume Averaging

- Without CTF correction, information past the first zero is incorrect and will give the wrong structure
- Example from a microtubule decorated with Eg5, taken at -8 µm defocus; the 4-nm tubulin repeat is between 1st and 2nd zero
- Without correction, Eg5 heads have the wrong shape and tubulin is almost 180º out of phase from where it should be

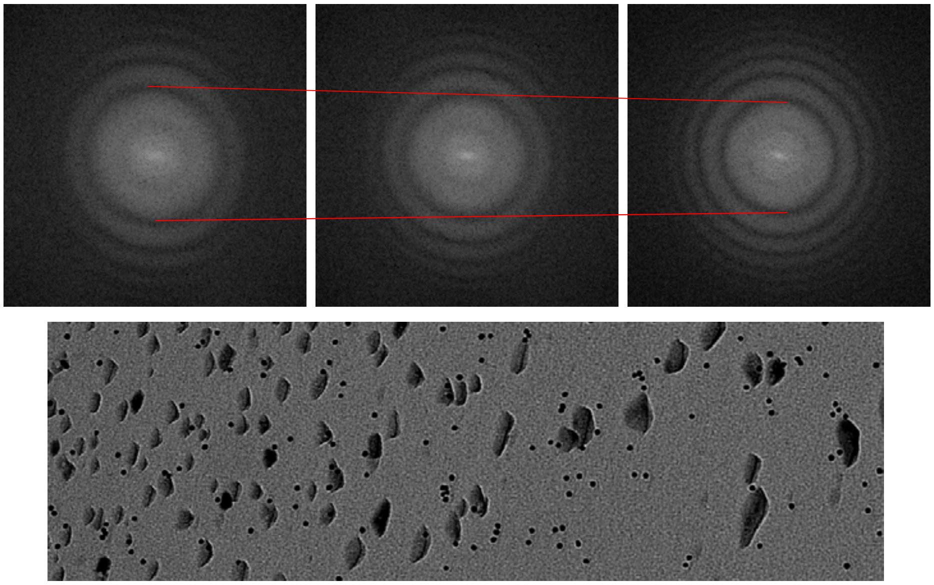


Average from Eg5-decorated microtubule, CTF corrected

What Makes CTF Correction of Tilt Series Hard?

- There is a different defocus gradient across each tilt series image
  - Not straightforward to invert phases in an FFT
  - Adding together data from different defoci will blur the power spectrum, make it hard to detect zeros

Defocus and Thon Ring Variation in Tilted Sample



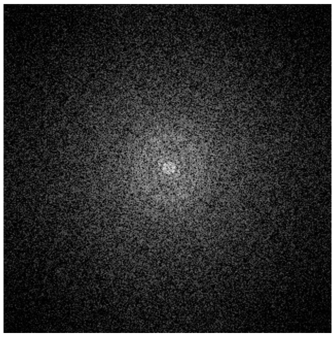
5 nm gold on carbon film, tilted to 65°

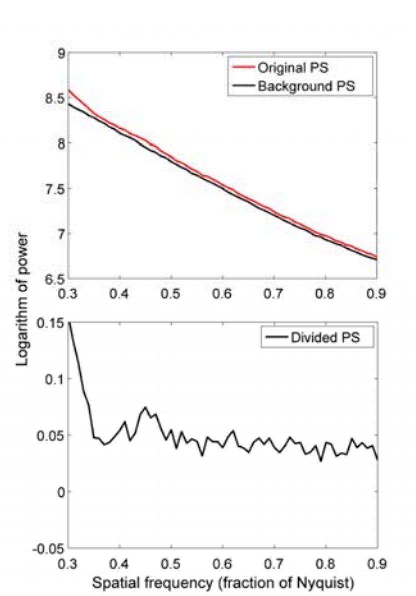
4.9 µm underfocus 6 µm underfocus 7.1 µm underfocus

What Makes CTF Correction of Tilt Series Hard?

- There is a different defocus gradient across each tilt series image
  - Not straightforward to invert phases in an FFT
  - Adding together data from different defocuses will blur the power spectrum, make it hard to detect zeros
- The dose is ~10-20 fold lower per micrograph than for single-particle averaging
  - Data must be averaged from multiple images to see CTF effect in power spectrum, unless camera is very efficient
- There may be little information past the second zero (or even the first zero with a CCD camera)
  - This depends on relationship between pixel size and defocus
  - For cryoET with relatively low defocus, there may be few zeros in spectrum, unless very high resolution is being targeted

Rotational Averaging and Background Subtraction Are Essential for Seeing CTF Zeros in Tilt Series



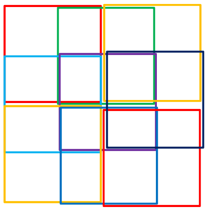


Rotational Averaging and Background Subtraction Are Essential for Seeing CTF Zeros in Tilt Series

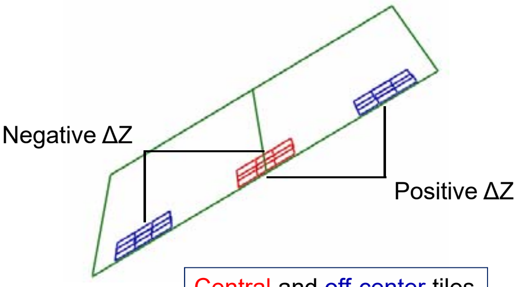
- The Ctfplotter program uses noise images to estimate the background
  - These are specific to camera, microscope, KV, binning.
  - Take series of blank images increasing in counts by factor of 1.5-2
  - Place these in /usr/local/ImodCalib/CTFnoise directory
  - See Ctfplotter man page for instructions
  - Do this once, use images on many data sets
- Ctfplotter interpolates between the nearest pair of noise images to find the noise background for a given image

To Compute Power Spectra, Each Image Is Divided into Overlapping Tiles

- This is periodogram averaging, standard method used to get power spectra for CTF correction
  - Tiles are typically 256 pixels square and overlap by 50%
  - Fourier transforms are taken separately and averaged
- For tilt series images, tiles are classified by distance from tilt axis in center:
  - Close to axis (Z height within 200 nm of center): central tiles
  - Farther from axis (Z height difference > 200 nm): off-center tiles



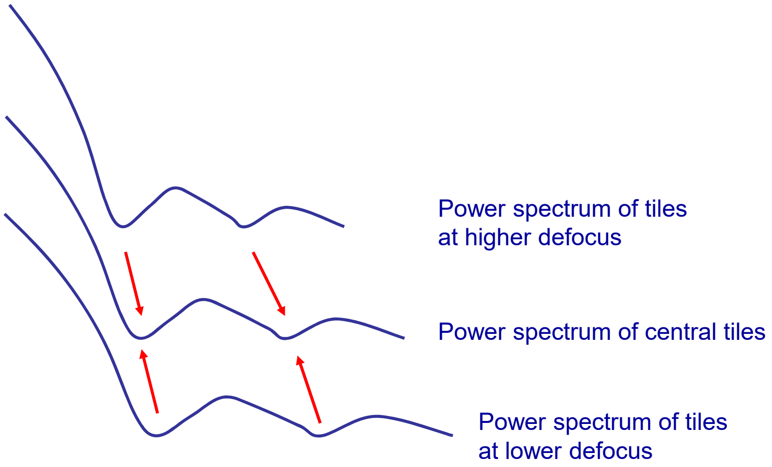
Overlapping tiles



Central and off-center tiles

Adding Together Spectra from Different Defoci

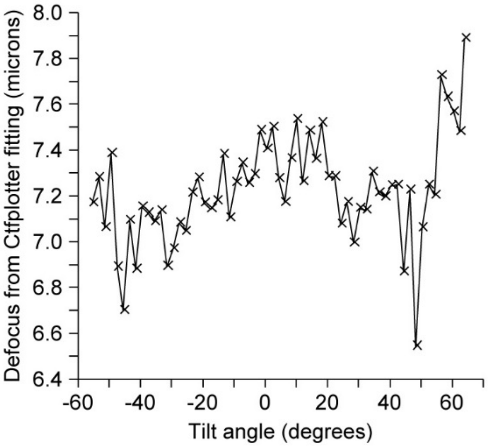
- Power spectra from off-center tiles are scaled and shifted so that the first and second zeros are reinforced when they are added into spectra from central tiles



Power spectrum of tiles at higher defocus
Power spectrum of central tiles
Power spectrum of tiles at lower defocus

Goals in CTF Correction

- Ultimate goal is to find defocus for as many individual views as possible
  - Defocus can vary 0.2-0.4 µm from one tilt to next
  - Reliable fitting to single views is possible for data from direct detectors, especially K2; may be possible with CCD data too.

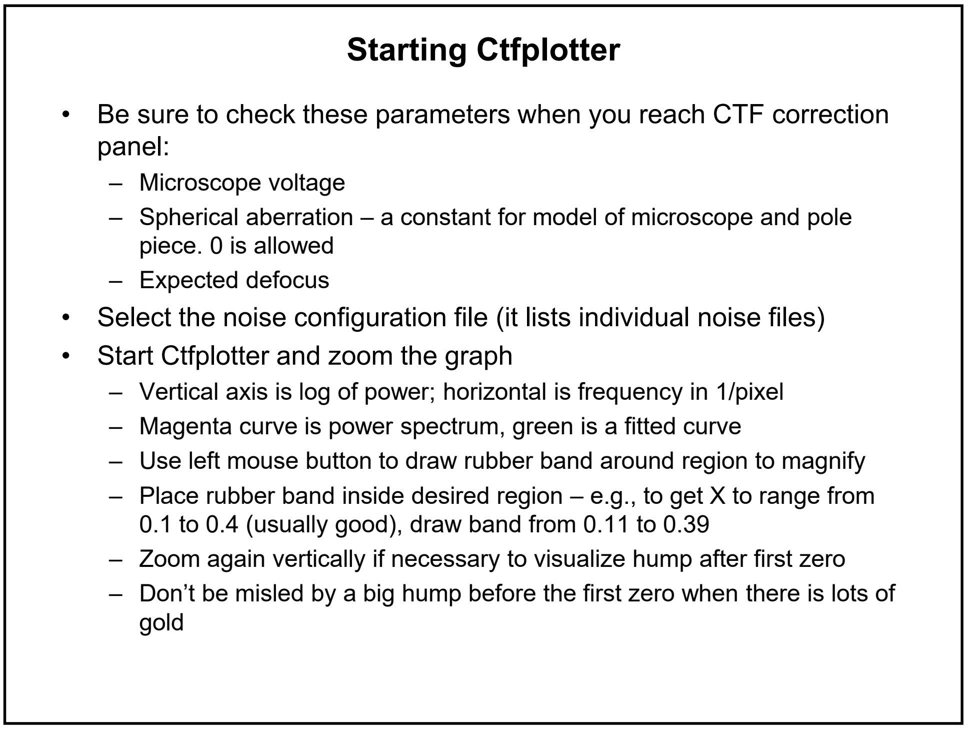


Defocus measured from single views

- Tilt series of Giardia ventral disk
- Acquired on Krios with energy filter and K2 camera
- Total dose 27 $e^-/Å^2$

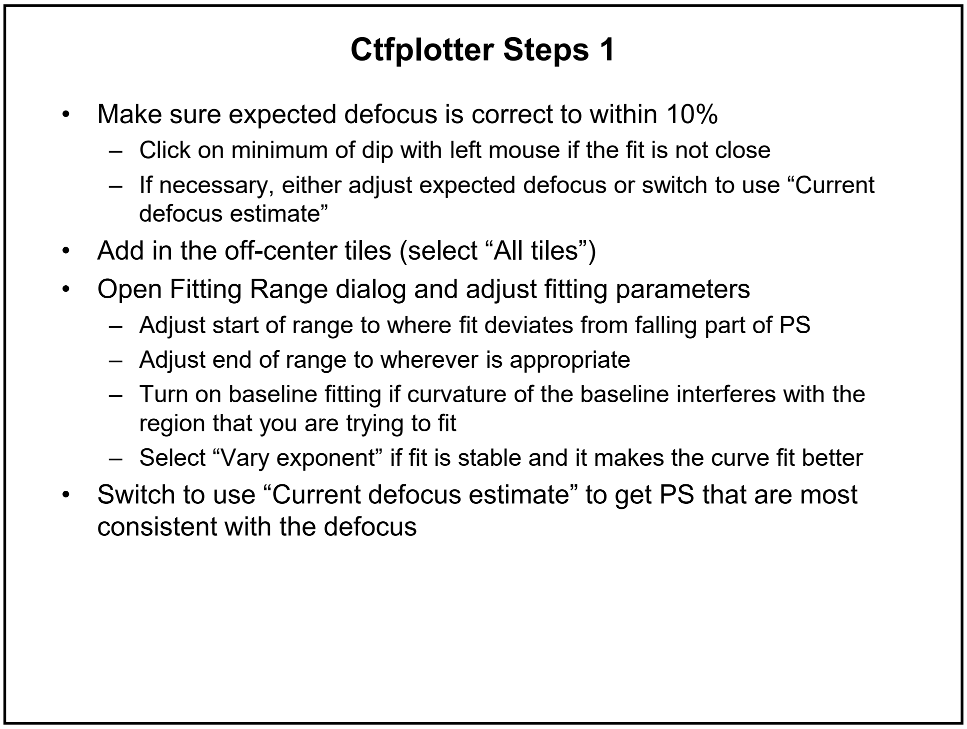
Goals in CTF Correction

- Ultimate goal is to find defocus for as many individual views as possible
  - Defocus can vary 0.2-0.4 µm from one tilt to next
  - Reliable fitting to single views is possible for data from direct detectors, especially K2; may be possible with CCD data too.
  - If signal is not strong enough to allow reliable fitting, fitting to every view will do more harm than good
- More modest goal, almost always achievable, is to find systematic change in defocus through series
  - Defocus can vary by up to ~1 µm from one end of series to other
  - Fitting to ranges of 10º to 40º can be useful for this
- Fallback goal: fit to whole series and find better estimate than the nominal defocus



Starting Ctfplotter

- Be sure to check these parameters when you reach CTF correction panel:
  - Microscope voltage
  - Spherical aberration – a constant for model of microscope and pole piece. 0 is allowed
  - Expected defocus
- Select the noise configuration file (it lists individual noise files)
- Start Ctfplotter and zoom the graph
  - Vertical axis is log of power; horizontal is frequency in 1/pixel
  - Magenta curve is power spectrum, green is a fitted curve
  - Use left mouse button to draw rubber band around region to magnify
  - Place rubber band inside desired region – e.g., to get X to range from 0.1 to 0.4 (usually good), draw band from 0.11 to 0.39
  - Zoom again vertically if necessary to visualize hump after first zero
  - Don’t be misled by a big hump before the first zero when there is lots of gold



Ctfplotter Steps 1

- Make sure expected defocus is correct to within 10%
  - Click on minimum of dip with left mouse if the fit is not close
  - If necessary, either adjust expected defocus or switch to use “Current defocus estimate”
- Add in the off-center tiles (select “All tiles”)
- Open Fitting Range dialog and adjust fitting parameters
  - Adjust start of range to where fit deviates from falling part of PS
  - Adjust end of range to wherever is appropriate
  - Turn on baseline fitting if curvature of the baseline interferes with the region that you are trying to fit
  - Select “Vary exponent” if fit is stable and it makes the curve fit better
- Switch to use “Current defocus estimate” to get PS that are most consistent with the defocus

Ctfplotter Steps 2

- Determine an angular range that can be fit through the whole series
  - Reduce the angular range and go to high tilt at both ends of series: see what works
  - Set the step size between ranges to half of the angular range
- Step through the series, storing values in the table as you go
- Or, autofit to all steps, then check the fits by double-clicking each line in the table

THIS PAGE INTENTIONALLY LEFT BLANK

Finding Defocus with Ctfplotter and Gold Erasing

ESTIMATING CTF ON A K2 CAMERA

This tilt series is of a flagellum of a Giardia cell, taken with a K2 camera on a Krios microscope at Janelia Farm. The total dose was 26 electrons/square Angstrom. Images were taken in superresolution mode with an exposure time of 0.5 sec to avoid having to save and align subframes, and reduced by a factor of 4 with antialiasing. With this protocol, they may have somewhat better high-frequency information than a typical tilt series taken in counting mode without binning, so the power spectra may be particularly good here.

1)
```
cd $WORKSHOP_HOME/IMOD_Labs/K2-ctf
```

2)
```
etomo *.edf
```

3) Go to the **Final Aligned Stack** page and press **Create Full Aligned Stack**. When done, switch to the **Correct CTF** tab. Notice Spherical Aberration can be 0. The Expected defocus (microns) has already been set to 6.0 in this data set. The Config file has already been selected to access a configuration file listing noise files in the *Janelia* subdirectory of the data set directory (*K2-ctf*).

4) Press **Run Ctf Plotter**. The magenta curve shows the rotationally averaged power spectrum plotted versus spatial frequency. The green curve is a fitted curve, which in general may not fit well until fitting parameters have been adjusted. The units of spatial frequency along the X axis are reciprocal pixels and range from 0 to 0.5/pixel.

5) Power is always very high at low frequencies, so the first operation is to zoom up the part of the curve that shows the CTF effect. Click the mouse just under the hump in the curve after 0.1/pixel and drag the zoom area out to the right edge, just under the curve, so that the Y axis range is about -0.1 to 0.5.

6) Since the fitting looks good already, on the **Angle Range & Tile Selection** window, switch to use the **Current defocus estimate**. In the **Initial tiles to include** section, select **All tiles** in and Press **Apply**.

7) In the upper left corner of the **ctfplotter** window, press **Fitting** to open the dialog for setting fitting parameters. The frequency of the first zero determined from the fit is shown at the top of the plotter window after Z:, and the corresponding defocus is shown. It is ~4.6 microns instead of the nominal 6 microns. The fitting range starts at a low frequency appropriate for the higher nominal defocus but too early for this lower actual defocus. It is best if the fitting is done to a linearly falling part of the curve and excludes the portion before that curving away from a line. Change **X1 Starts** to **0.16.** The fitting can go to the third zero or even the fourth, so change **X2 Ends** to **0.38** or **0.43**. Press the **Enter** key or **Apply** to fit with a changed value. Turn on **Vary exponent of CTF function**; the fit looks better, so that is an appropriate parameter to include when fitting these data.

8) To assess whether fitting to a single image is possible, go back to the **Angle Range and Tile Selection** window, change the **Starting tilt angle** to **0** and the **Ending tilt angle** to **1.8**, and press **Apply**. The curve looks good. To see whether this is still the case at high tilt, change the **Starting tilt angle** to **60** and the **Ending tilt angle** to **62** and press **Apply**. The curve still looks good. Turn on **Fit each view separately** and press **Autofit All Single Views**. Resize the **Angle Range & Tile Selection** window so that you can see more of the table and scroll through the values to see how much defocus varies. Note the changes of over a micron up and down above 40 degrees. Double-click a series of lines in

the table to check the curve-fitting in this region. It is clearly correct; the defocus changes from image to image have been measured accurately. Press **Save to File** and exit ctfplotter.

9) To apply the CTF correction, in Etomo, press **Correct CTF.** When done, **right-click** on the **View CTF Correction** button and select **Open with startup window**. **In Image file(s)**, press **Select** and choose both *WTI042413_1series4_ctfcorr.ali* and *WTI042413_1series4.ali* and press **Open**, then press **OK**. Compare the 2 files in the ZaP window by toggling the **4th D**. They look very similar, although there are some subtle changes. Close 3dmod and press **Use CTF Correction** in Etomo. Close Etomo.

10) `header WTI042413_1series4.ali`

At the bottom of the printout, you will see that ctfPhaseFlip has been used on these data.

GOLD ERASING USING FINDBEADS3D

11) Go to the **Erase Gold** tab. Under **Model Creation Method**, make sure **Use findbeads3d** is selected. Change **Aligned image stack binning** to **3** and press **Align and Build Tomogram**. When done, press **View Tomogram in 3dmod**. Verify that it appears that both surfaces containing gold particles are completely within the tomogram. In this case, a thickness of **350** worked, but in other cases, you may have to increase the Thickness value and try again.

12) Press **Run Findbeads3d**. When done, press **View 3D Model on Tomogram**. Bead Fixer will automatically pop up. Move through the tomogram and look for gold beads that do not have model points and model points that are not actually on a gold bead. If the model looks reasonable, press **Delete Below** on the **Bead Fixer** window and then **Yes**. This action deletes points whose correlation score was below the value set with the **Threshold** slider and were likely to be incorrect. Be sure to save your model (**s**).

13) Press **Reproject Model**. When done, press **View 2D Model on Aligned Stack**. Movie through the aligned stack and watch for any gold beads you may have missed that do not have model points. Also, there may be contours that do not follow gold beads. In this 2D model, you can easily delete contours that are not on gold beads and save the model. However, if there are any gold beads that are missing contours, you need to add them to the 3D model on the tomogram, save the model, and reproject that model onto the aligned stack. This can become an iterative process. It is often difficult to tell if gold beads are modeled properly when there are clumps of gold as in these data.

14) In the **Erase Beads** section of Etomo, **change Iterations to grow circular areas** to **3**. Press **Erase Beads**. When done, press **View Erased Stack**. Movie through the stack and you will probably notice some gold beads within the clumps that did not get erased. At this point, you would need to return to View 3D Model on Tomogram and add model points to those gold beads (centered in X, Y, and Z) and continue through until when you view the erased stack, you no longer see any gold beads. You may also determine that the gold beads are not over an area of interest and therefore it is OK if they remain. When you are satisfied, press **Use Erased Stack**. We are finished with this exercise, so close Etomo.

ESTIMATING CTF ON A DE-12 CAMERA

This tilt series is of a preparation of microtubules decorated with the motor protein Eg5, taken with a DE-12 camera during a demo on the F20 microscope in Boulder. The tilt series had a 2 degree increment and the total dose was 79 electrons/square Angstrom. Parts of the series have good signal for determining CTF, but not all of it.

15)
```
cd $WORKSHOP_HOME/IMOD_Labs/DE-ctf/
```

16)
```
etomo MTEg5series6D.edf
```

17) Open the **Final Aligned Stack** page press **Create Full Aligned Stack**. When done, switch to the **Correct CTF** tab. The Expected defocus has already been set to 6.0 in this data set. The Config file has already been selected to access a configuration file listing noise files in the *DE12-Div2* subdirectory of the data set directory (*De-ctf*).

18) Press **Run Ctf Plotter**.

    Zoom the power spectrum by clicking on the magenta curve to the left of 0.1/pixel and dragging the selection region to a point before a frequency of 0.4/pixel and just below the baseline. The power spectrum shows a clear signal out to the third zero and the green fitted curve matches the location of the zeros fairly well, so the defocus estimate is good. On the **Angle Range & Tile Selection** window, select both **All Tiles** and **Current defocus estimate** and press **Apply**.

19) Press **Fitting** to open the Fitting Range dialog. Set **X1 Starts** to **0.1**, since the fitted curve deviates before this point. Set **X2 Ends** to **0.25** to fit out to the third zero, and press the Enter key or **Apply**. Turn on **Vary exponent of CTF function**; the fit does not look any better so there is no reason to leave this option on. Turn it off for now; below you will see how it is inappropriate in some cases.

20) To see if single images can be fit, set the **Ending tilt angle** to **-18** and press Enter or **Apply**. This curve is noisy but the fitting still seems reasonable. To see if fitting is still good at high tilt, change the **Starting tilt angle** to **-60** and the **Ending tilt angle** to **-58**. The curve is even noisier but fitting is still plausible. Turn on **Fit each view separately** and press **Autofit All Single Views**.

21) Resize the **Angle Range** dialog so that you can see more of the table and scroll through the values. Notice that there are some big jumps at positive tilt angles, particularly from **25** to **27** and from **41** to **43**. Double-click these lines in the table to see these fits. The power spectra at **27** and **43** are particularly low in signal, so fitting to the noisy data from single views is just not reliable. To get a better sense for how often the fitting looks good, double-click other lines through the series. At most (but not all) negative tilt angles, the fit looks fairly reliable; but at positive tilt angles it often is not. Since inaccurate defocus values can do more harm than good, we need to fit to multiple views instead, with the reduced goal of estimating the trend in defocus through the series.

22) Now switch to fitting sets of 4 views (8 degree ranges) by entering a **Starting tilt angle** of **0**, an **Ending tilt angle** of **8**, and **4** for **Step angle range by**. Turn off **Fit each view separately** and press **Autofit All Steps**. The program will ask you to confirm that you want to replace all of the existing values in the table; press **Yes**.

23) Double-click the lines at positive tilt angles to see how reliable these fits look. The signal gets rather low at the highest tilts, but the noise is low enough to allow you to see that the defocus is being found adequately. To see the potential problems with adding the exponent of the CTF function as a variable in the fit, double-click the line for **21** to **29** degrees. Press **Apply** 5-10 times and watch the defocus value (**D:**) in the plotter window. It varies from 5.26 to 5.29, which means that fitting is self-consistent and not too sensitive to the current assumed defocus value. On the **Fitting Range and Method** window, turn on **Vary exponent of CTF function** and press **Apply** many times. The estimated defocus jumps around between 5.15 and 5.3, a 5-fold bigger range. When the signal is too weak, adding this fifth parameter to the fit can significantly reduce the reliability and stability of the fits. Turn off **Vary exponent of CTF function** one more time and press **Apply**. Press **Save to File** and close ctfplotter.

24) Press **Correct CTF**. When done, press **Use CTF Correction**.

GOLD ERASING USING EXISTING FIDUCIAL MODEL

25) Go to the **Erase Gold** tab. Under **Model Creation Method**, make sure **Use the existing fiducial model** is selected, and press **Transform Fiducial Model.** When done, press **View Transformed Model**. Notice the fiducial model used the 2-3 gold beads found very near the microtubules among others within the tilt-series.

26) In the **Erase Beads** section of Etomo, **change Iterations to grow circular areas** to **3**. Press **Erase Beads**. When done, press **View Erased Stack**. Press **Use Erased Stack** and then **Done**. We are finished with this exercise, close Etomo.

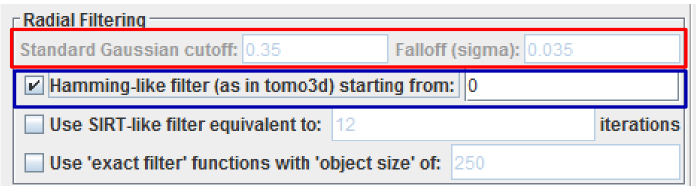
Denoising Cryotomograms with IMOD

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



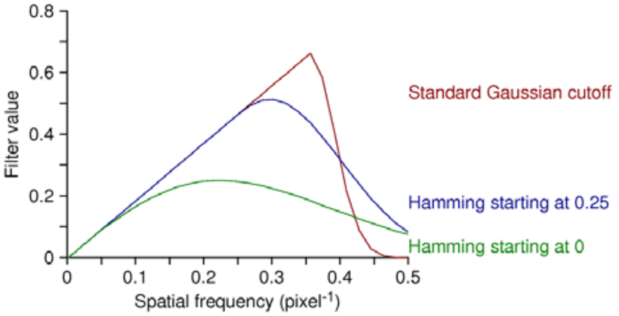
Radial Filtering

Standard Gaussian cutoff: 0.35 Falloff (sigma): 0.035

☑ Hamming-like filter (as in tomo3d) starting from: 0

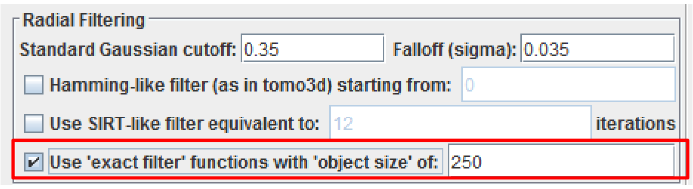
□ Use SIRT-like filter equivalent to: 12 iterations

□ Use 'exact filter' functions with 'object size' of: 250



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



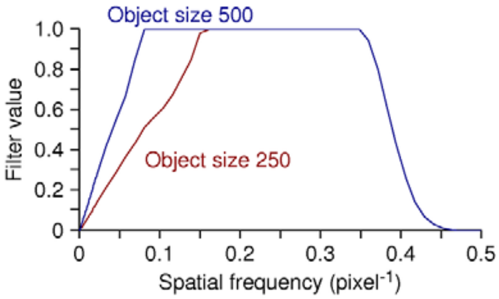
Radial Filtering

Standard Gaussian cutoff: 0.35 Falloff (sigma): 0.035

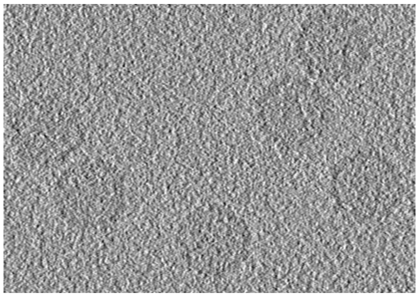
□ Hamming-like filter (as in tomo3d) starting from: 0

□ Use SIRT-like filter equivalent to: 12 iterations

☑ Use 'exact filter' functions with 'object size' of: 250

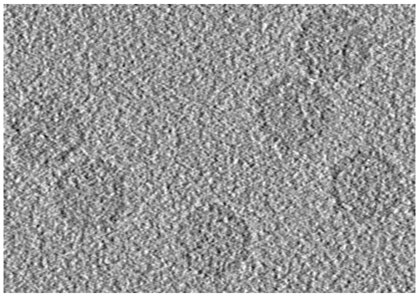


Standard Gaussian Cutoff



Default radius and sigma

Hamming-like Filter

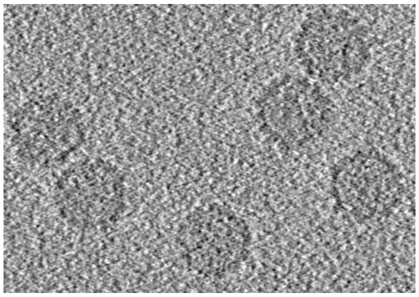


Starting at zero

Exact Filters

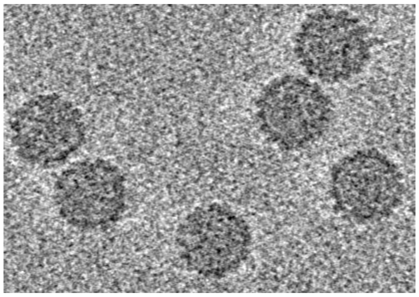
Object size 250

Exact Filters



Object size 500

SIRT-like Filter



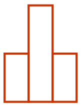
Equivalent to 12 iterations

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

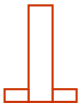
1	2	1
2	4	2
1	2	1

Top view
(all divided by 16)

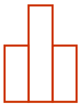


Side view through
central pixel

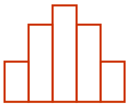
- For more flexibility, weights can be set from a real-space Gaussian with a chosen sigma. The standard filter corresponds to σ = 0.85.
- The kernel is 3x3 pixels for σ ≤ 1, 5x5 for 1 < σ ≤ 2, or 7x7 for σ > 2



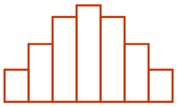
σ = 0.5



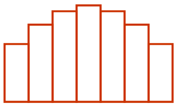
σ = 1.0



σ = 1.5



σ = 2.0

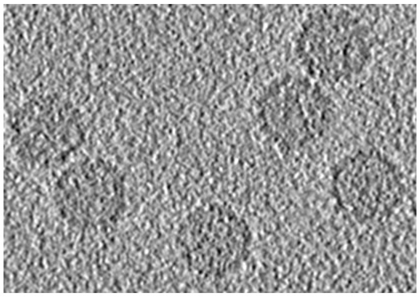


σ = 3.0

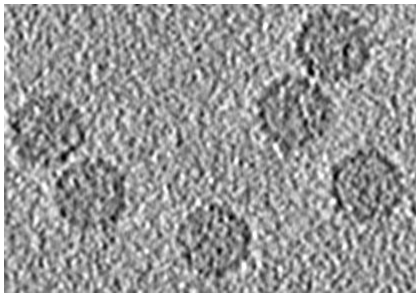
- For even more filtering than σ = 3, you would have to iterate – but σ = 3 already smooths a lot!

Kernel Smoothing Filter

Default



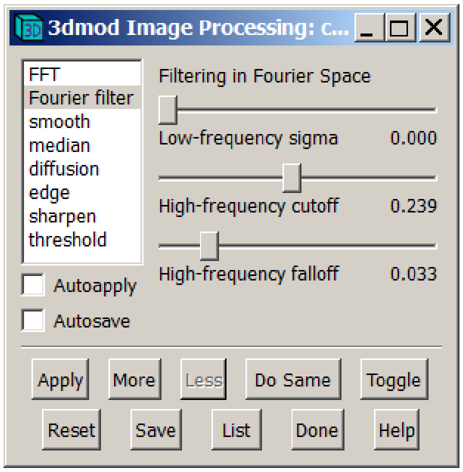
Sigma 1.7



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



Filtering in Fourier Space

Low-frequency sigma 0.000

High-frequency cutoff 0.239

High-frequency falloff 0.033

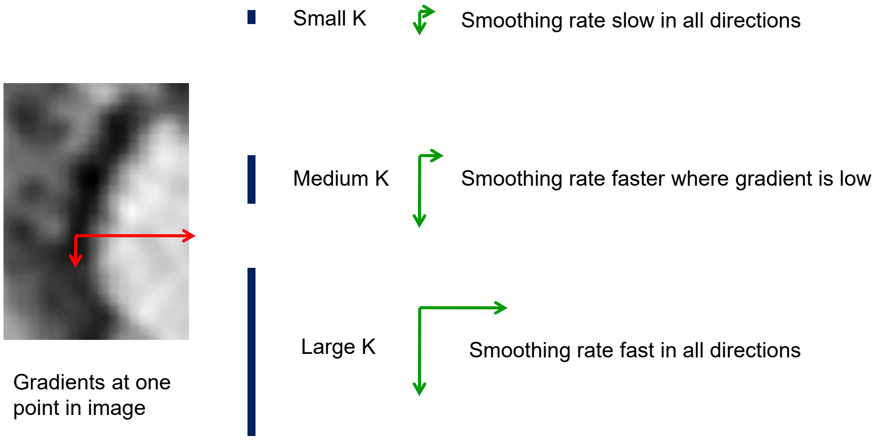
☐ Autoapply

☐ Autosave

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



Small K — Smoothing rate slow in all directions

Medium K — Smoothing rate faster where gradient is low

Large K — Smoothing rate fast in all directions

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.

THIS PAGE INTENTIONALLY LEFT BLANK

Using Nonlinear Anisotropic Diffusion (NAD)

Filtering with NAD is relatively slow and involves adjusting two parameters to give the desired amount of filtering. The strategy is to do the operation on a small test volume and compare the results from different parameter settings in order to pick the right setting. After that, the full volume is filtered by breaking it into chunks so that multiple processors can be used.

1) `cd $WORKSHOP_HOME/IMOD_Labs/cryo-subvolume`

2) `etomo`

3) Press **Nonlinear Anisotropic Diffusion.**

4) First, we extract a test volume. Press the **file chooser** icon on the **Pick a volume** line and select *cryoSubvol.rec*. Press **View Full Volume**.

   Zoom the ZaP window up to 1 if necessary, and draw a rubber band in the middle that is about 300x300 pixels (see the size in the ZaP toolbar). Scroll through the slices and set the **Lo** and **Hi** limits to extract 25-30 slices. When choosing a test area, try and pick a region that exemplifies your region of interest. For this example, you might be interested in microtubules, actin, or ribosomes.

   Press **Get Test Volume Range from 3dmod** to fetch the range into Etomo. Press **Extract Test Volume**. Press **View Test Volume** to make sure you got an appropriate selected area.

5) Now, we need to find the first parameter, the **K Value for the Test Volume**. For **List of K values**, enter **0.1,1,5,10,15,25,30,50,75.** And keep the **Iterations** at **10**. Press **Run with Different K Values**. The program computes a volume for each K value, using multiple processors. The K value controls how the diffusion of density occurs on each iteration. Where the image gradient is higher than K, this is considered to represent an edge, and density flows along the edge and not across it. Where the image gradient is lower than K, density diffuses uniformly.

6) When done, press **View Different K Values Test Results** to load the multiple volumes into 3dmod. The ZaP window has a second toolbar with **4th D** left and right arrows for stepping between the volumes (the keys **1** and **2** can also be used). The toolbar also shows the name of the file being displayed.

   One way to compare the volumes is to step between them in one window; this is the best way to see how they change from one to the next. To see the volumes side-by-side, select **Image / Linked Slicers**. A slicer will open for each volume, each one locked to that volume (note the red lock icon in the toolbar). The slicers will be kept at the same orientation and position, controlled by a single floating toolbar.

   Images look strange for intermediate K values (10 - 25), with regions of uniform intensity. Above this range, K is too high to stop the diffusion across places of high gradient, and diffusion is essentially isotropic. Within this range, diffusion is blocked across medium to high gradients, so the density probably gets trapped in medium-sized regions bounded by these gradients and becomes evenly distributed. Below this K range, there are many, more closely-spaced, gradients to block diffusion, and this probably prevents the noticeable uniform regions. Close all 3dmod windows once you know the value you will choose.

7) Next, we need to choose the second parameter for NAD by **Finding Iteration Number for the Test Volume.**

   Enter your preferred K value in the **K value** field. For **List of iterations**, enter **2,5,8,11,15,21**. Press **Run with Different Iterations**. This time the nad eed 3d program runs once, saving the results at each selected iteration.

   Press **View Different Iteration Test Results**. This time it is probably easier to assess the preferred filtering by stepping through the volumes in the ZaP window.

8) Now we can apply the 2 parameters by **Filtering the Full Volume**. Insert your selected K value in the **K value** field and number of iterations in the **Iterations** field. Press **Filter Full Volume**.

   When it is done, you can open it with **View Filtered Volume**, and delete the test volumes with **Clean Up Subdirectory**. Close Etomo and 3dmod windows when you are finished. The final volume is named by adding the extension *.nad* to the name of the original volume. Note that the extensions *.rec* and *.nad* are simply conventions and do not specify a file format; this volume is an MRC file.

Modeling Tips & Tricks

Space is provided here for you to take notes during the Demo

Modeling Tips & Tricks

Space is provided here for you to take notes during the Demo

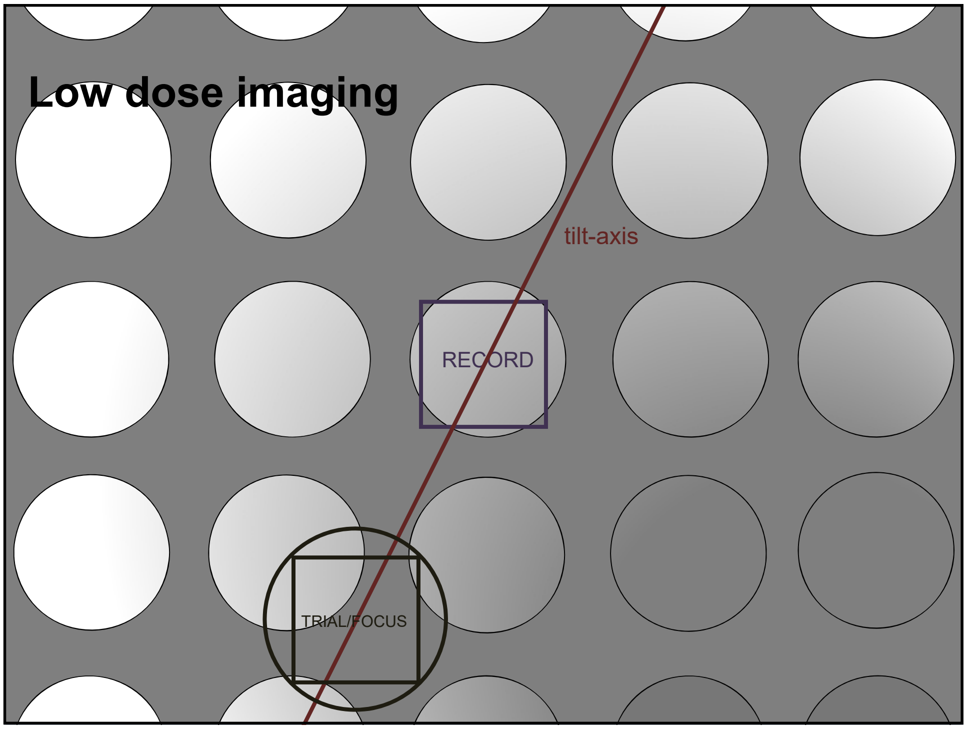
SerialEM for Cryo Tomography

Types of samples

- Very small whole cells (bacteria, some algae)
- FIB milled lamellae
- Vitreous sections
- Isolated organelles (mitochondria, microtubules)
- Viruses and virus-like-particles
- Individual proteins

Setting up Low Dose

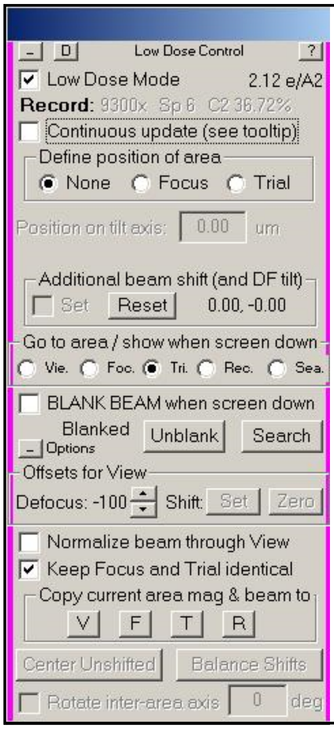
- Remember your record image is the most important and all other images should 'revolve' around that image
- When changing exposure times, make sure ALL images have 0 drift settling!!!
- Maximize the Low Dose Control window, it may be helpful to float it and put it on a second monitor
- Check 'Low Dose Mode'



Low dose imaging

Image types for Cryo

- Record: Final image that goes into tilt-series
  - Try to keep dose around 1 e/A2
- Trial/Focus: Used for tasks and tracking during tilt-series and for autofocus
  - Make beam just bigger than the area of the camera
  - Can be higher dose than record
- View: Lower mag search mode
  - Use a mag that allows you to see several holes at a time
  - Can add some defocus to see particles
  - Make extremely low dose <<<1 e/A2
- Preview: Exact area/mag/C2%/spot size as record, but binned x4 or x8
  - Used to 'see' the record area, but with really low dose
  - Use exposure time and binning to lower dose but have decent image
  - Check dose in Camera Parameters dialogue box



Low Dose Control

☑ Low Dose Mode 2.12 e/A2

Record: 9300x Sp 6 C2 36.72%

□ Continuous update (see tooltip)

Define position of area
● None ○ Focus ○ Trial

Position on tilt axis: 0.00 um

Additional beam shift (and DF tilt)
□ Set Reset 0.00, -0.00

Go to area / show when screen down
○ Vie. ○ Foc. ● Tri. ○ Rec. ○ Sea.

□ BLANK BEAM when screen down
Blanked Unblank Search
Options

Offsets for View
Defocus: -100 Shift: Set Zero

□ Normalize beam through View
☑ Keep Focus and Trial identical

Copy current area mag & beam to
V F T R

Center Unshifted Balance Shifts

□ Rotate inter-area axis 0 deg

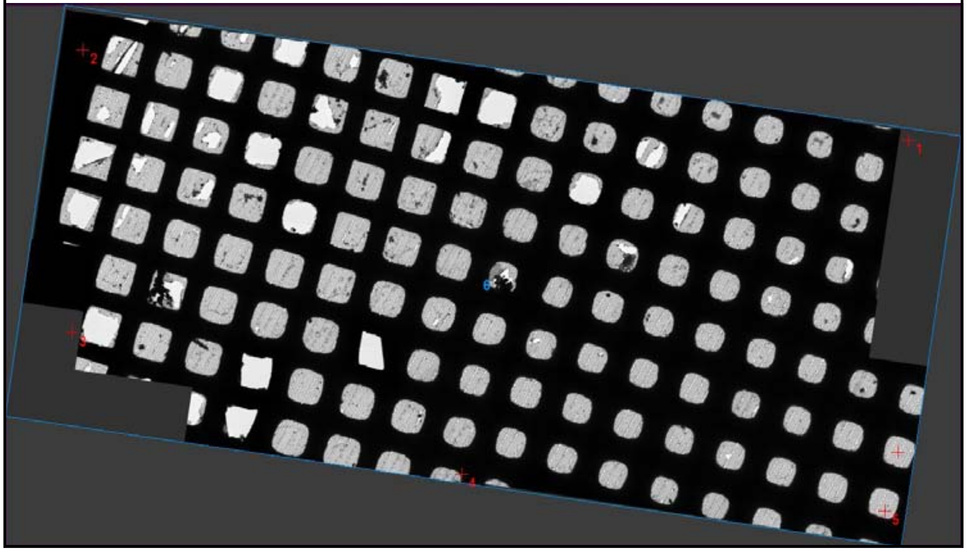
This allows you to change the beam for each image type

Lets you correct beam position if it is not centered the same in different modes (or offset it from the center away from the record area in focus/trial)

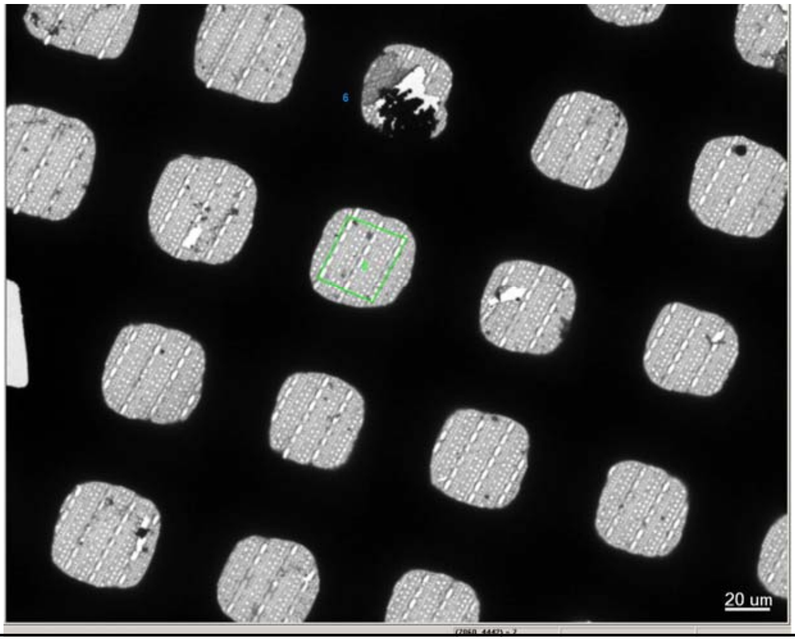
Lets you put a high defocus in view mode so you can see small particles in the holes

Typically focus and trial are identical unless you are at very high mag, where it could be advantageous to have a lower tracking mag

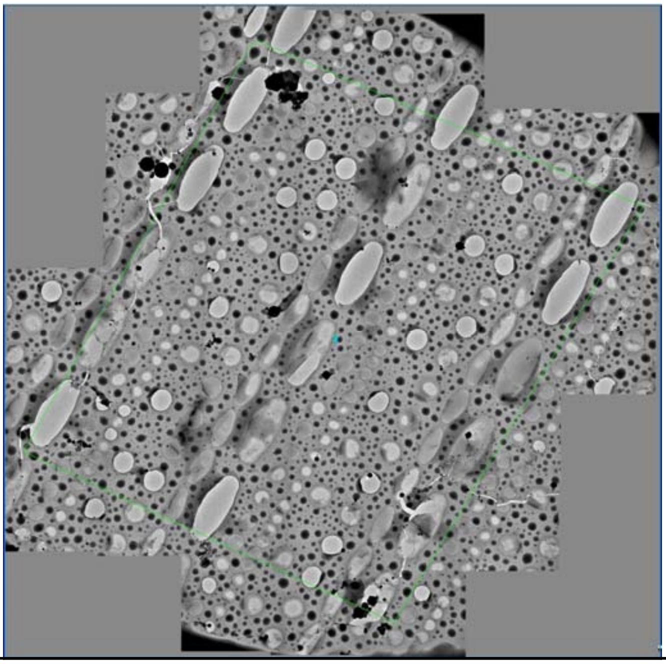
Generate a map of the grid where a tilt-series is possible. Red points are 'corners' of stage positions to tell SerialEM where to collect the map



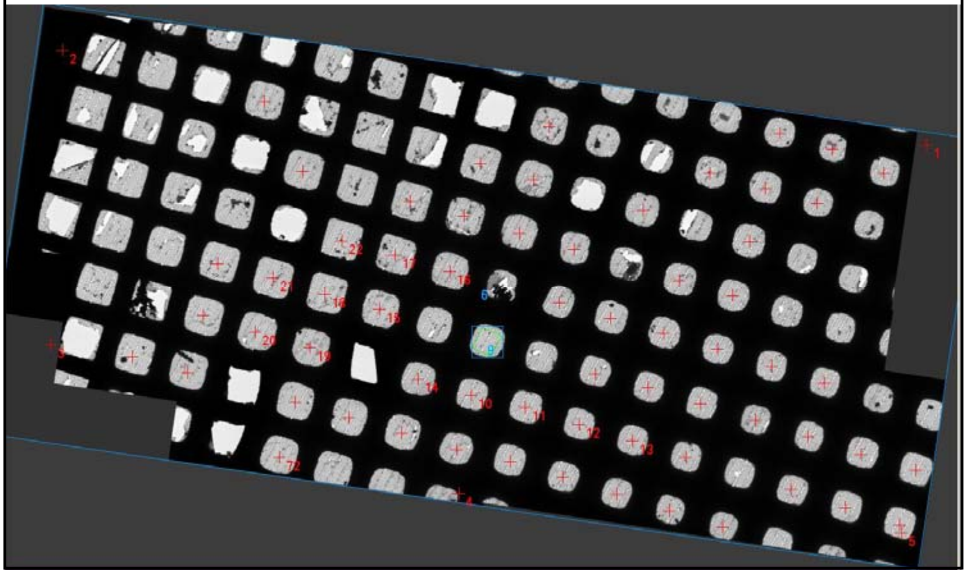
Zoom in and place a polygon within a grid square—best to size it to fit the region where a tilt-series is possible (basic center of the square)



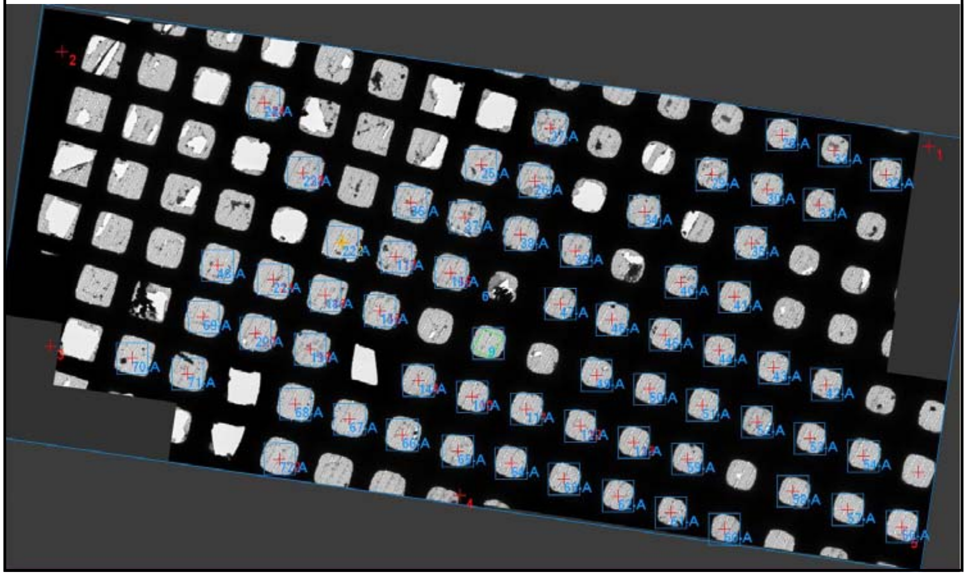
Completed polygon map for the grid square



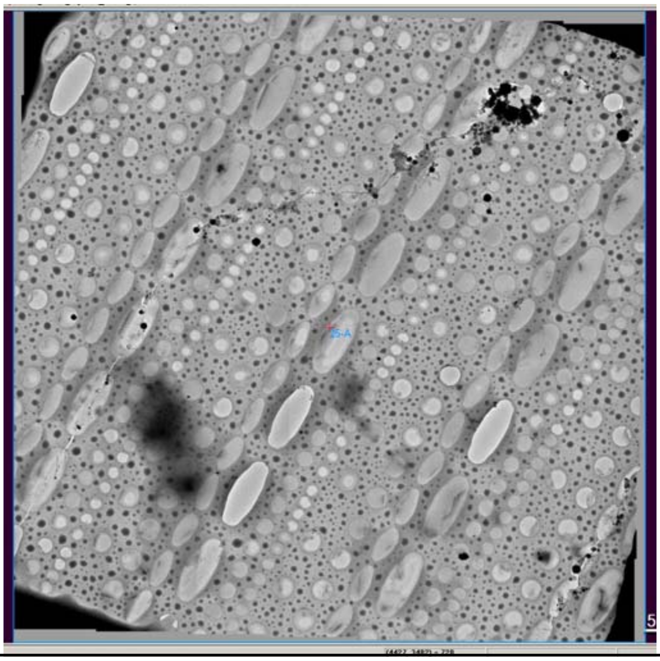
Mark points in the center of each grid square of interest and use Navigator to automatically acquire polygon maps around those points



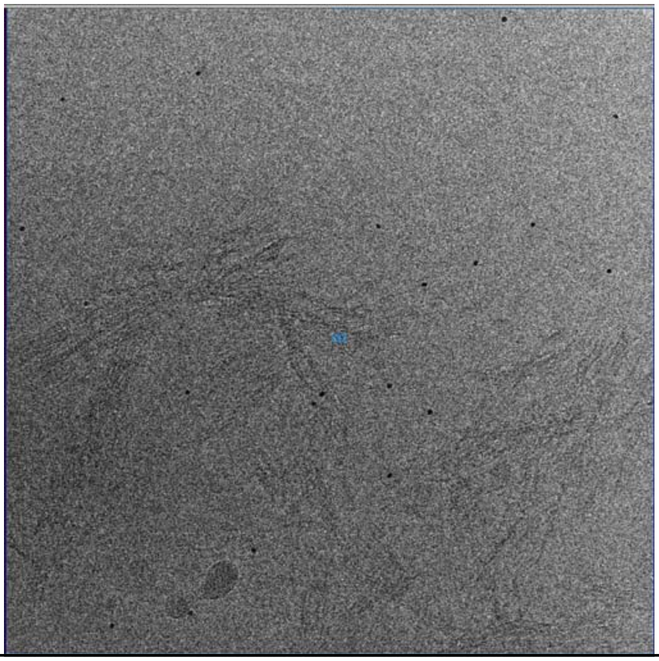
All blue squares are now maps that were acquired automatically with the same settings as the original green polygon



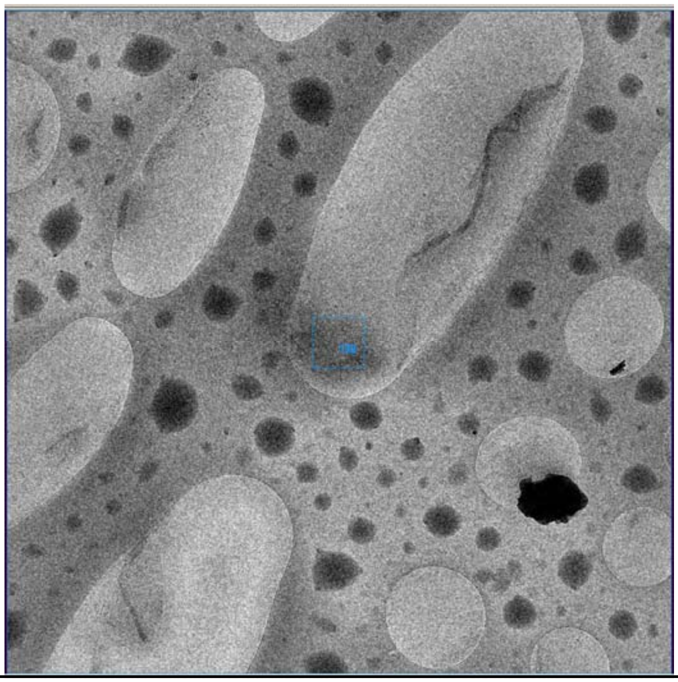
Example of automatically acquired map



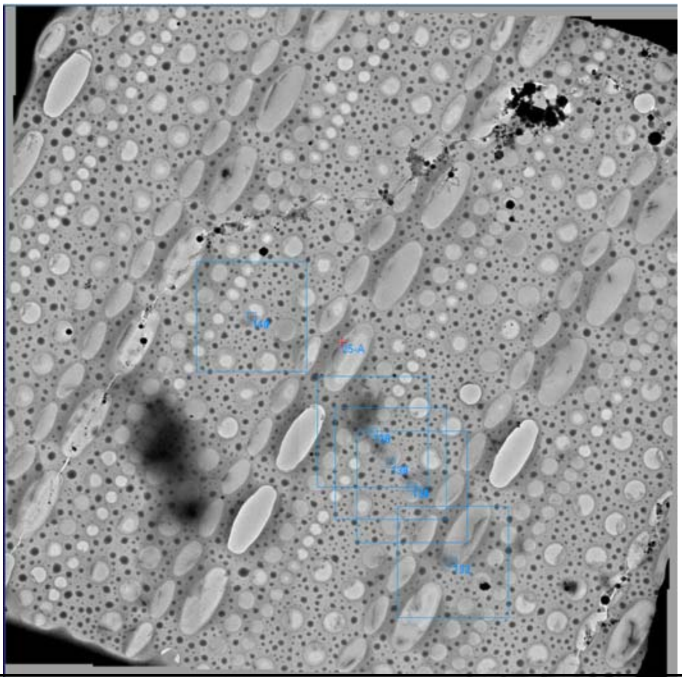
Preview image used to center TS Record area



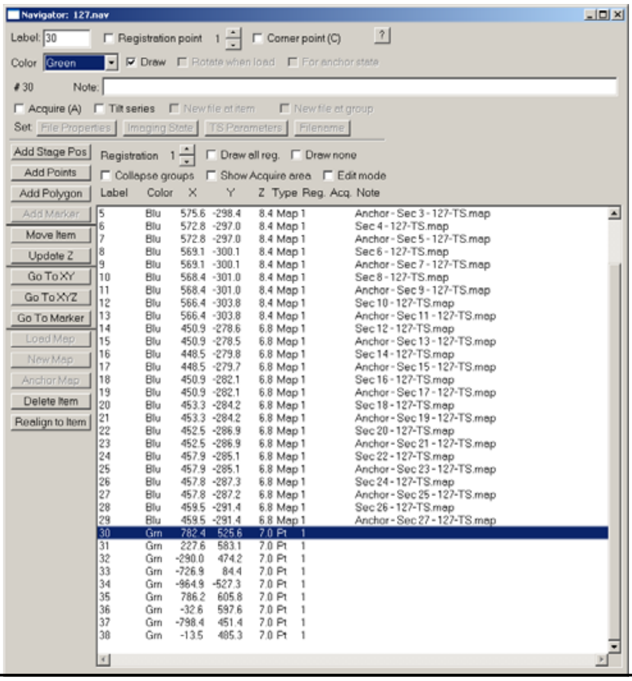
Anchor image used to center at low mag



Map with tilt-series marked for acquisition.



List of tilt-series to acquire overnight



Navigator: 127.nav

Label: 30 □ Registration point 1 □ Corner point (C) ?

Color: Green ☑ Draw □ Rotate when load □ For anchor state

30 Note:

□ Acquire (A) □ Tilt series □ New file at item □ New file at group

Set: File Properties | Imaging State | TS Parameters | Filename

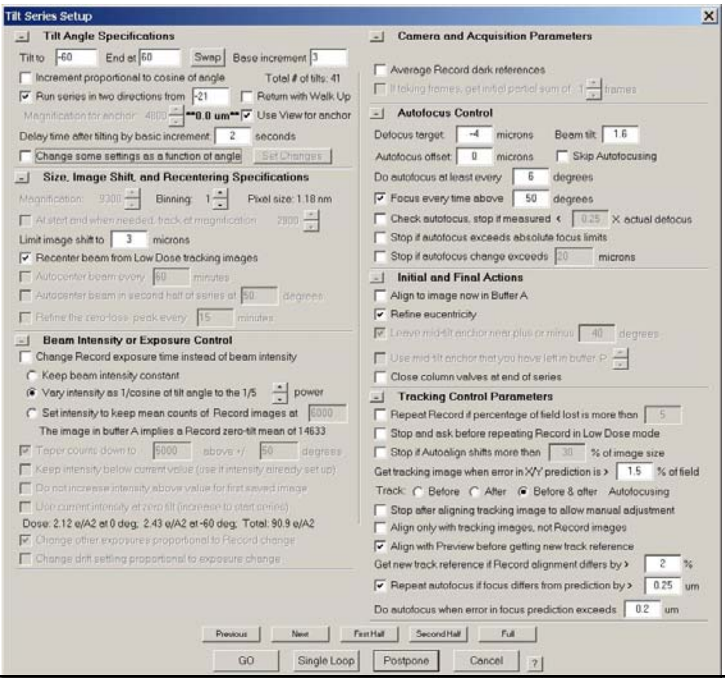
Add Stage Pos | Registration 1 □ Draw all reg. □ Draw none

Add Points | □ Collapse groups □ Show Acquire area □ Edit mode

Add Polygon

Label	Color	X	Y	Z	Type	Reg.	Acq.	Note
5	Blu	575.6	-298.4	8.4	Map	1		Anchor - Sec 3 - 127-TS.map
6	Blu	572.8	-297.0	8.4	Map	1		Sec 4 - 127-TS.map
7	Blu	572.8	-297.0	8.4	Map	1		Anchor - Sec 5 - 127-TS.map
8	Blu	569.1	-300.1	8.4	Map	1		Sec 6 - 127-TS.map
9	Blu	569.1	-300.1	8.4	Map	1		Anchor - Sec 7 - 127-TS.map
10	Blu	568.4	-301.0	8.4	Map	1		Sec 8 - 127-TS.map
11	Blu	568.4	-301.0	8.4	Map	1		Anchor - Sec 9 - 127-TS.map
12	Blu	566.4	-303.8	8.4	Map	1		Sec 10 - 127-TS.map
13	Blu	566.4	-303.8	8.4	Map	1		Anchor - Sec 11 - 127-TS.map
14	Blu	450.9	-278.6	6.8	Map	1		Sec 12 - 127-TS.map
15	Blu	450.9	-278.5	6.8	Map	1		Anchor - Sec 13 - 127-TS.map
16	Blu	448.5	-279.8	6.8	Map	1		Sec 14 - 127-TS.map
17	Blu	448.5	-279.7	6.8	Map	1		Anchor - Sec 15 - 127-TS.map
18	Blu	450.9	-282.1	6.8	Map	1		Sec 16 - 127-TS.map
19	Blu	450.9	-282.1	6.8	Map	1		Anchor - Sec 17 - 127-TS.map
20	Blu	453.3	-284.2	6.8	Map	1		Sec 18 - 127-TS.map
21	Blu	453.3	-284.2	6.8	Map	1		Anchor - Sec 19 - 127-TS.map
22	Blu	452.5	-286.9	6.8	Map	1		Sec 20 - 127-TS.map
23	Blu	452.5	-286.9	6.8	Map	1		Anchor - Sec 21 - 127-TS.map
24	Blu	457.9	-285.1	6.8	Map	1		Sec 22 - 127-TS.map
25	Blu	457.9	-285.1	6.8	Map	1		Anchor - Sec 23 - 127-TS.map
26	Blu	457.8	-287.3	6.8	Map	1		Sec 24 - 127-TS.map
27	Blu	457.8	-287.2	6.8	Map	1		Anchor - Sec 25 - 127-TS.map
28	Blu	459.5	-291.4	6.8	Map	1		Sec 26 - 127-TS.map
29	Blu	459.5	-291.4	6.8	Map	1		Anchor - Sec 27 - 127-TS.map
30	Grn	782.4	525.6	7.0	Pt	1		
31	Grn	227.6	583.1	7.0	Pt	1		
32	Grn	-290.0	474.2	7.0	Pt	1		
33	Grn	-726.9	84.4	7.0	Pt	1		
34	Grn	-964.9	-527.3	7.0	Pt	1		
35	Grn	786.2	605.8	7.0	Pt	1		
36	Grn	-32.6	597.6	7.0	Pt	1		
37	Grn	-798.4	451.4	7.0	Pt	1		
38	Grn	-13.5	485.3	7.0	Pt	1		

Tilt-series dialog



Tilt Series Setup

Tilt Angle Specifications

Tilt to -60 End at 60 Swap Base increment 3

□ Increment proportional to cosine of angle Total # of tilts: 41

☑ Run series in two directions from -21 □ Return with Walk Up

Magnification for anchor: 4800 **0.0 um** ☑ Use View for anchor

Delay time after tilting by basic increment 2 seconds

□ Change some settings as a function of angle Set Changes

Size, Image Shift, and Recentering Specifications

Magnification: 9300 Binning: 1 Pixel size: 1.18 nm

□ At start and when needed, track at magnification 2800

Limit image shift to 3 microns

☑ Recenter beam from Low Dose tracking images

□ Autocenter beam every 60 minutes

□ Autocenter beam in second half of series at 50 degrees

□ Refine the zero-loss peak every 15 minutes

Beam Intensity or Exposure Control

□ Change Record exposure time instead of beam intensity

○ Keep beam intensity constant

● Vary intensity as 1/cosine of tilt angle to the 1/5 power

○ Set intensity to keep mean counts of Record images at 6000

The image in buffer A implies a Record zero-tilt mean of 14633

☑ Taper counts down to 5000 above +/ 50 degrees

□ Keep intensity below current value (use if intensity already set up)

□ Do not increase intensity above value for first saved image

□ Use current intensity at zero tilt (increase to start series)

Dose: 2.12 e/A2 at 0 deg; 2.43 e/A2 at -60 deg; Total: 90.9 e/A2

☑ Change other exposures proportional to Record change

□ Change drift setting proportional to exposure change

Camera and Acquisition Parameters

□ Average Record dark references

□ If taking frames, get initial partial sum of 1 frames

Autofocus Control

Defocus target -4 microns Beam tilt 1.6

Autofocus offset 0 microns □ Skip Autofocusing

Do autofocus at least every 6 degrees

☑ Focus every time above 50 degrees

□ Check autofocus, stop if measured < 0.25 X actual defocus

□ Stop if autofocus exceeds absolute focus limits

□ Stop if autofocus change exceeds 20 microns

Initial and Final Actions

□ Align to image now in Buffer A

☑ Refine eucentricity

☑ Leave mid-tilt anchor near plus or minus 40 degrees

□ Use mid tilt anchor that you have left in buffer P

□ Close column valves at end of series

Tracking Control Parameters

□ Repeat Record if percentage of field lost is more than 5

□ Stop and ask before repeating Record in Low Dose mode

□ Stop if Autoalign shifts more than 30 % of image size

Get tracking image when error in X/Y prediction is > 1.5 % of field

Track: ○ Before ○ After ● Before & after Autofocusing

□ Stop after aligning tracking image to allow manual adjustment

□ Align only with tracking images, not Record images

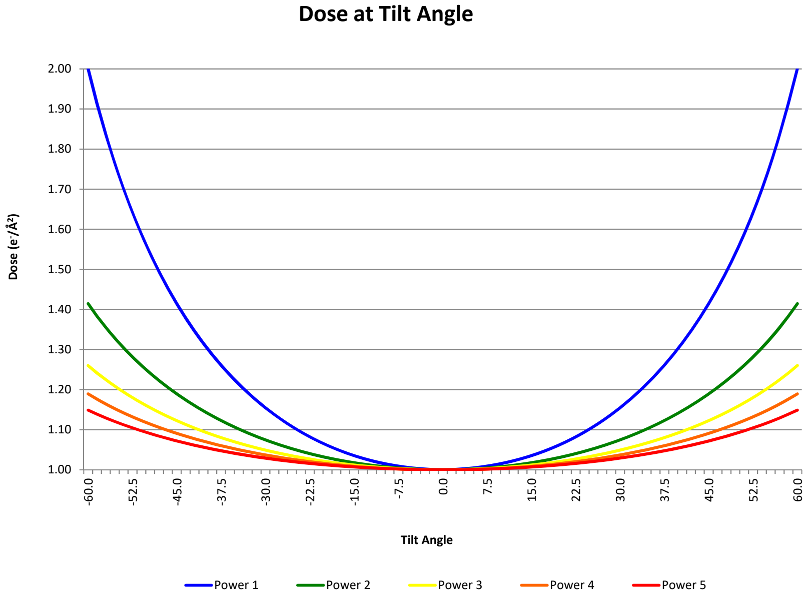
☑ Align with Preview before getting new track reference

Get new track reference if Record alignment differs by > 2 %

☑ Repeat autofocus if focus differs from prediction by > 0.25 um

Do autofocus when error in focus prediction exceeds 0.2 um

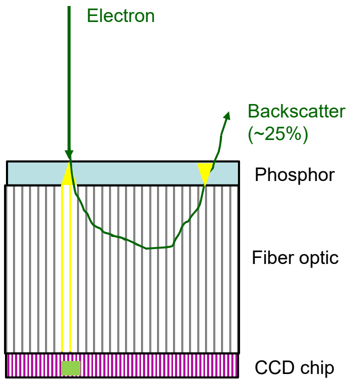
Using 1/cos of tilt angle to the 1/X power (x=1-5)



Using Direct Detectors with SerialEM

What Was Wrong with CCD Cameras?

- Spreading of signal in phosphor
- A bit more spreading and loss in fiber optic
- Variability in number of photons caught by CCD
  - Reduces ability to know how many primary electrons contributed to integrated signal
- At higher voltages, 20-30% of electrons backscatter and give strong signal at wrong place
- Readout is slow: accumulate whole image and read out once

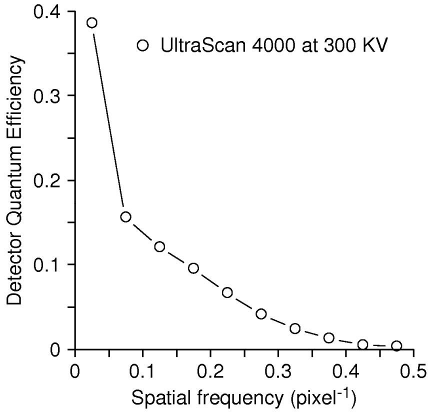


Camera Efficiency: DQE

- DQE (detector quantum efficiency) is a factor (0 to 1) measuring how well a camera detects electrons compared to an ideal detector
- Inverse of DQE is how much extra dose is needed to get the same signal-to-noise ratio (SNR) as an ideal detector would give
  - DQE of 33% => 3 times as much dose

Camera Efficiency: DQE

- DQE (detector quantum efficiency) is a factor (0 to 1) measuring how well a camera detects electrons compared to an ideal detector
- Inverse of DQE is how much extra dose is needed to get the same signal-to-noise ratio (SNR) as an ideal detector would give
  - DQE of 33% => 3 times as much dose
- DQE is actually a function of spatial frequency



Direct Electron Detection with CMOS “Monolithic Active Pixel Sensors”

- Primary electron generates 100-300 electrons in P- epilayer
- Electrons collect in closest well and have to be read out frequently
- Backscatter still a big problem unless substrate is severely thinned (from 700 µm to 30-50 µm

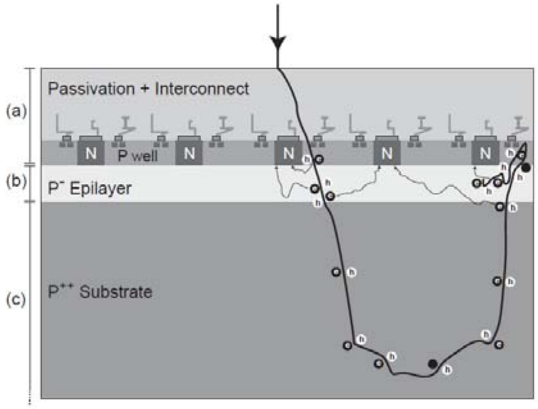
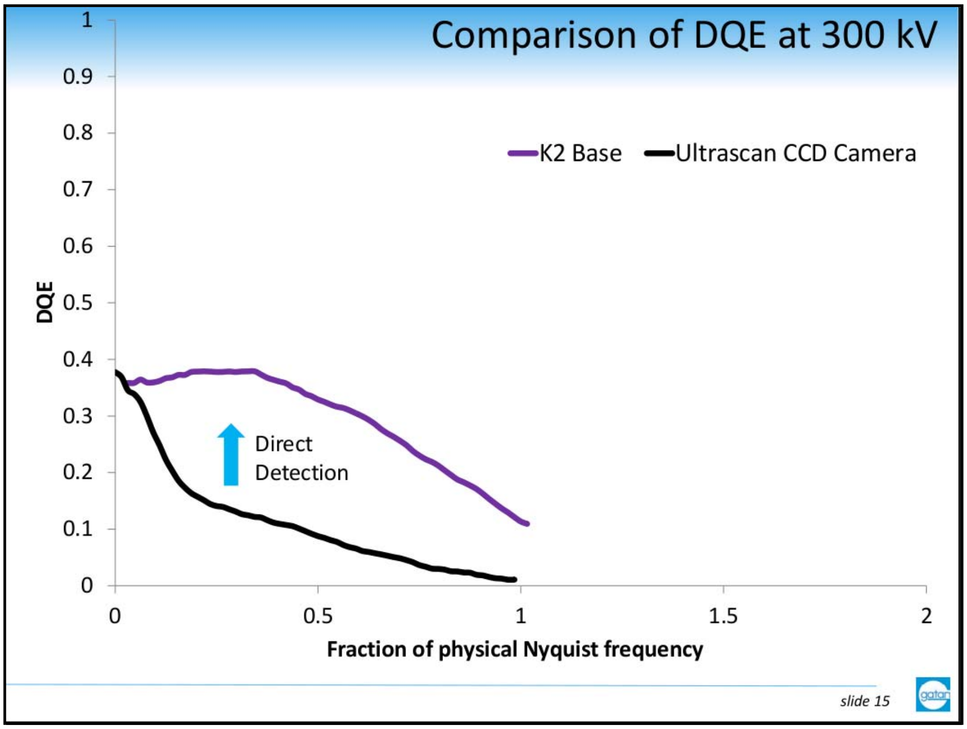


Fig. 1. Schematic of MAPS detector shown in cross-section. The detector has three main regions: (a) about 5-µm-thick passivation layer plus interconnections for readout electronics in the P well, (b) a few microns of lightly doped epilayer where the useful signal is generated, and drifts on to N wells prior to being read out, and (c) the main bulk of the detector, the substrate, which is heavily doped and which does not play a significant role in the detection process. A possible path for a single incident high-energy electron is shown to illustrate the problem with backscatter from the substrate.

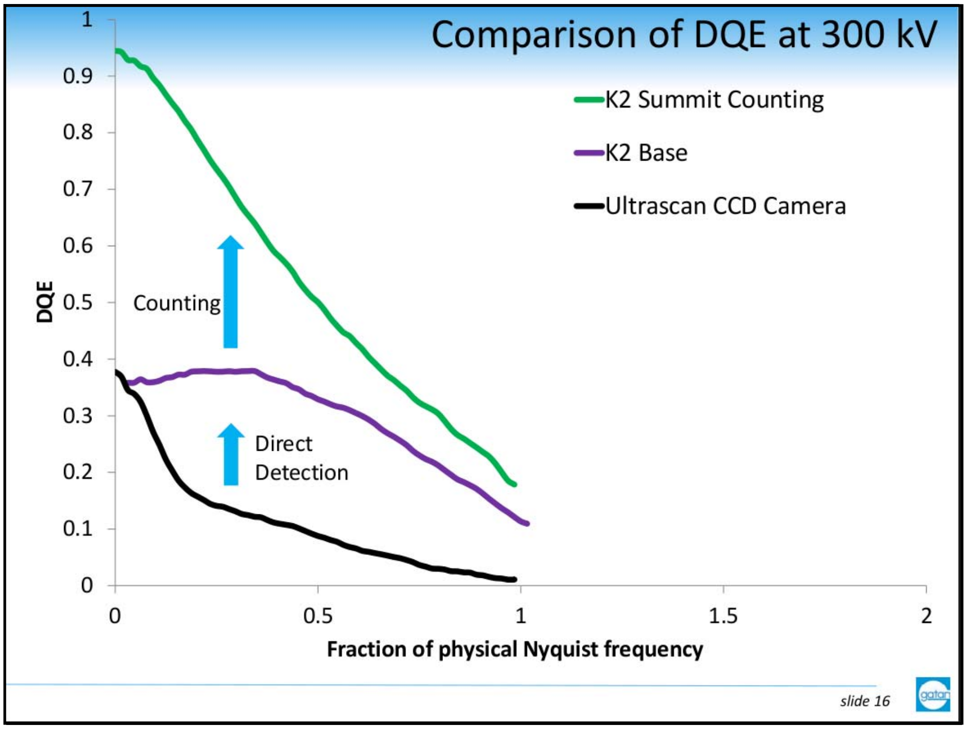
Why Is Electron Counting Good?

- Integrating variable-sized packets reduces the DQE
- Packets spread over several pixels and this reduces the resolution of an integrated signal
- Deducing where each electron occurs eliminates variability in measured events and spread of signal
- Using counting also reduces effects of noise and crap in signal read out from chip



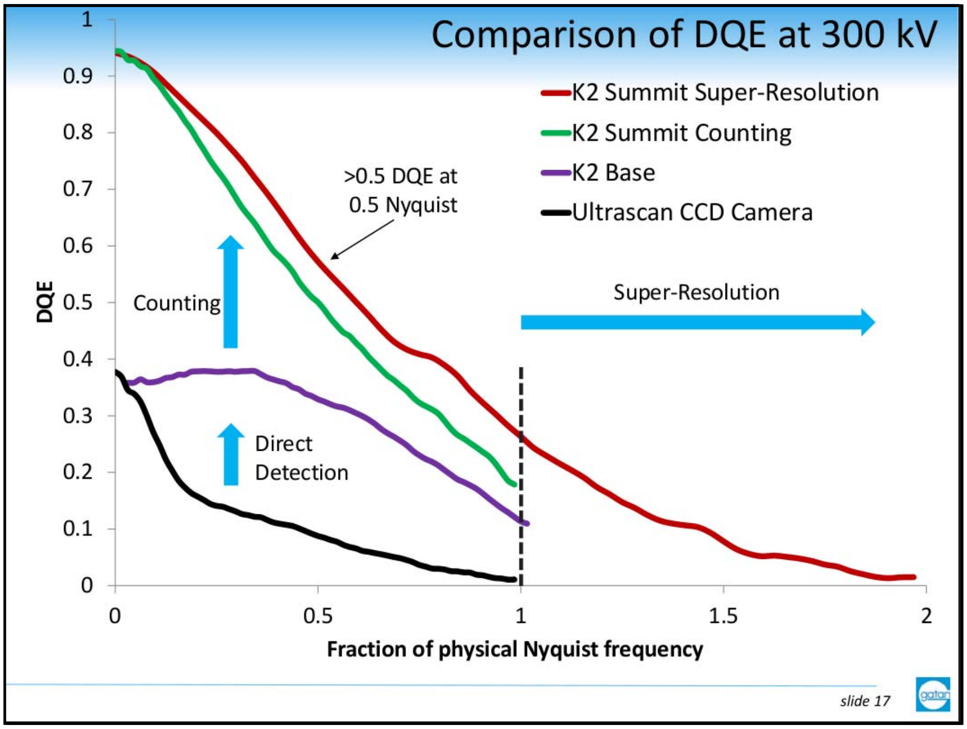
Comparison of DQE at 300 kV

slide 15



Comparison of DQE at 300 kV

slide 16



Comparison of DQE at 300 kV

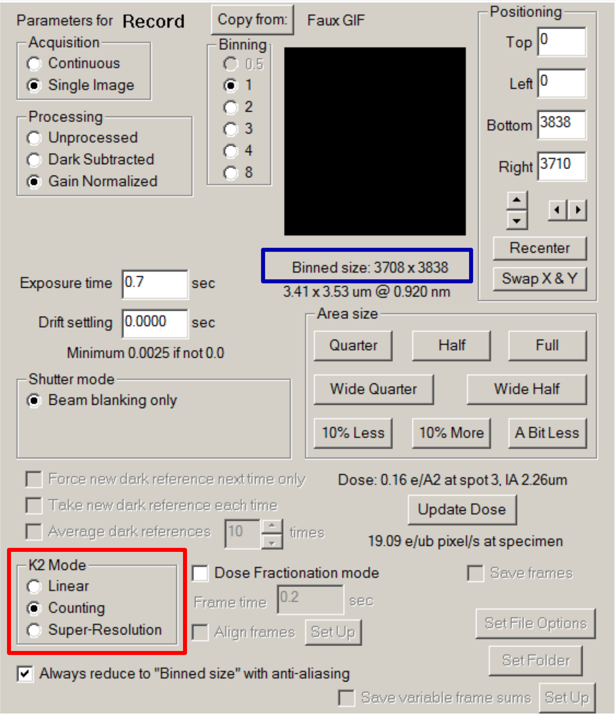
slide 17

Alignment of Multiple Frames

- The other major feature of direct detectors is the ability to break one image acquisition into multiple frames
- When these frames are aligned, the effects of drift can be reduced or eliminated and higher-resolution information is preserved
- There are several options for frame alignment
  - Motioncorr and Motioncor2 from UCSF
  - Unblur from Grigorieff group
  - Alignframes in IMOD
    - Incorporates features from Motioncorr and Unblur
    - Should be good for tilt series data AND more convenient

Counting Mode

- The mode can be set to Linear, Counting, or Super-resolution
- In counting mode, the full size is the number of physical pixels


Parameters for Record

Copy from: Faux GIF

Acquisition: ○ Continuous ● Single Image

Binning: ○ 0.5 ● 1 ○ 2 ○ 3 ○ 4 ○ 8

Processing: ○ Unprocessed ○ Dark Subtracted ● Gain Normalized

Positioning: Top 0, Left 0, Bottom 3838, Right 3710

Binned size: 3708 x 3838

3.41 x 3.53 um @ 0.920 nm

Exposure time 0.7 sec

Drift settling 0.0000 sec

Minimum 0.0025 if not 0.0

Shutter mode: ● Beam blanking only

□ Force new dark reference next time only

□ Take new dark reference each time

□ Average dark references 10 times

Dose: 0.16 e/A2 at spot 3, IA 2.26um

19.09 e/ub pixel/s at specimen

K2 Mode: ○ Linear ● Counting ○ Super-Resolution

□ Dose Fractionation mode

Frame time 0.2 sec

□ Align frames

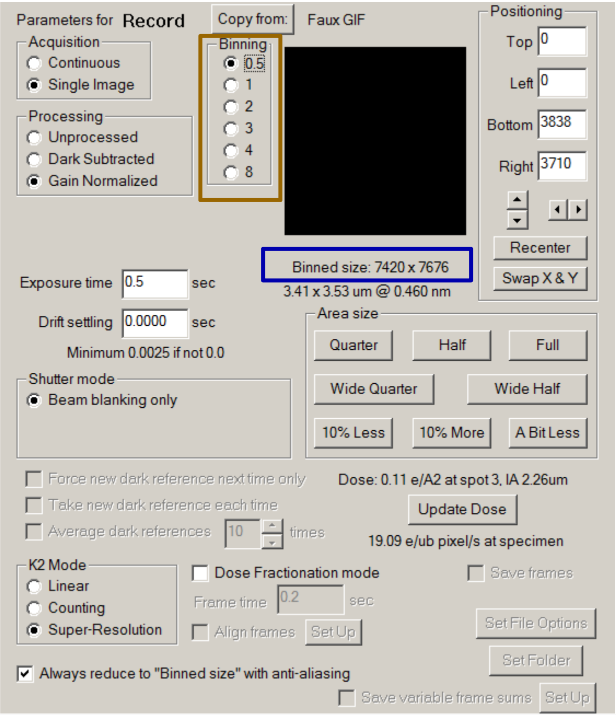
□ Save frames

☑ Always reduce to "Binned size" with anti-aliasing

□ Save variable frame sums

Super-Resolution Mode

- The binning is shown as 0.5 for the full super-resolution image.
- Now the size is twice as big as the physical number of pixels.


Parameters for Record

Copy from: Faux GIF

Acquisition: ○ Continuous ● Single Image

Binning: ● 0.5 ○ 1 ○ 2 ○ 3 ○ 4 ○ 8

Processing: ○ Unprocessed ○ Dark Subtracted ● Gain Normalized

Positioning: Top 0, Left 0, Bottom 3838, Right 3710

Binned size: 7420 x 7676

3.41 x 3.53 um @ 0.460 nm

Exposure time 0.5 sec

Drift settling 0.0000 sec

Minimum 0.0025 if not 0.0

Shutter mode: ● Beam blanking only

□ Force new dark reference next time only

□ Take new dark reference each time

□ Average dark references 10 times

Dose: 0.11 e/A2 at spot 3, IA 2.26um

19.09 e/ub pixel/s at specimen

K2 Mode: ○ Linear ○ Counting ● Super-Resolution

□ Dose Fractionation mode

Frame time 0.2 sec

□ Align frames

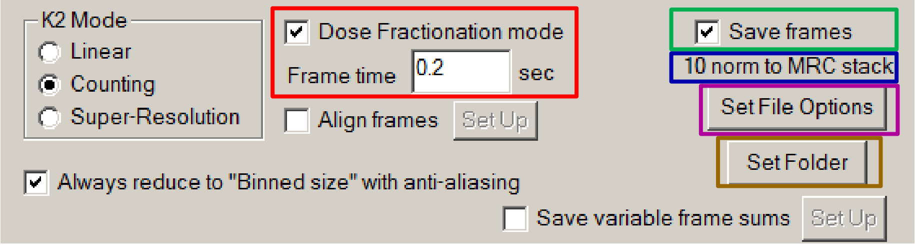
□ Save frames

☑ Always reduce to "Binned size" with anti-aliasing

□ Save variable frame sums

Dose Fractionation Mode and Saving Frames

- Dose Fractionation mode must be turned on to have multiple frames with the given frame time
- Save frames can then be turned on have them saved
- The summary line shows the number of frames and how they will be saved
- Set File Options opens a dialog for file name and format control
  - Unnormalized counting mode data (small integer counts) can be saved in TIFF files with good compression
- A folder must be defined: anywhere accessible from K2 computer



K2 Mode

○ Linear

● Counting

○ Super-Resolution

☑ Dose Fractionation mode

Frame time 0.2 sec

□ Align frames Set Up

☑ Always reduce to "Binned size" with anti-aliasing

☑ Save frames

10 norm to MRC stack

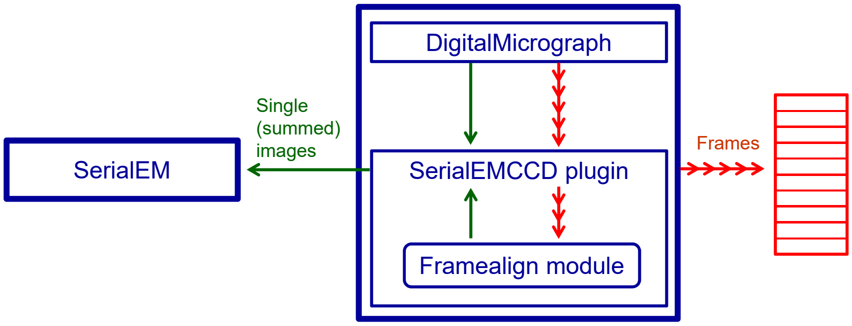
Set File Options

Set Folder

□ Save variable frame sums Set Up

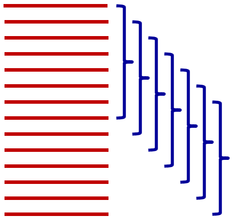
K2 Frame Handling in Dose Fractionation Mode

- All frames are handled by the SerialEMCCD plugin to DigitalMicrograph; SerialEM receives only single images (sums of frames)
- The plugin can save frames and/or align them with the Framealign module shared with IMOD



The Framealign Module in Alignframes and SerialEMCCD

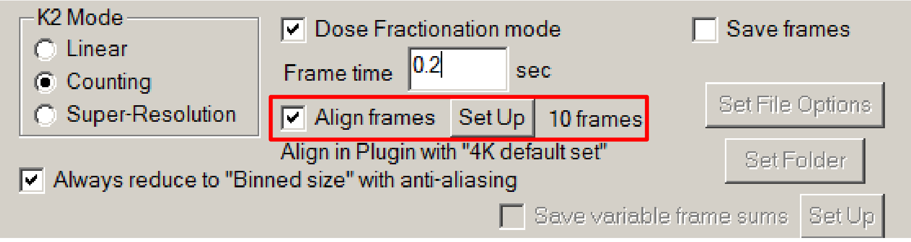
- Alignment is found by correlating many pairs of frames with each other and solving for best shifts of individual frames
  - Robust regression is used to reject effects of some bad alignments
  - All pairs are aligned in successive subsets of frames to avoid dependence on square of number of frames
- It processes data as it is available, leaving as little computation as possible until the end
- It can use the GPU of an NVIDIA card
- Aligning is slightly faster than saving frames even without a GPU



In each subset of 8, all 28 pairs are aligned to solve for 7 shifts

Aligning K2 Frames

- When Align frames is selected, you can then set the parameters and options for alignment
- Frames can be aligned in plugin and aligned image returned to SerialEM, or a command file can be written for running Alignframes



K2 Mode
○ Linear
● Counting
○ Super-Resolution

☑ Dose Fractionation mode

Frame time 0.2 sec

☑ Align frames Set Up 10 frames

Align in Plugin with "4K default set"

☑ Always reduce to "Binned size" with anti-aliasing

□ Save frames

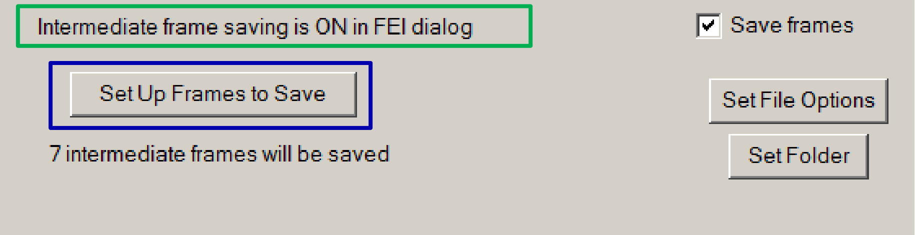
Set File Options

Set Folder

□ Save variable frame sums Set Up

Saving Frames from Falcon 2

- SerialEM may or may not be able to control or even know whether frame saving is turned on in the separate FEI dialog, so there can be either a checkbox to tell SerialEM it is on, a status line like here, or no text at all
- The FEI software can save up to 7 (or up to 40) sums of sequential frames (camera readouts) and the Set Up dialog lets you control that



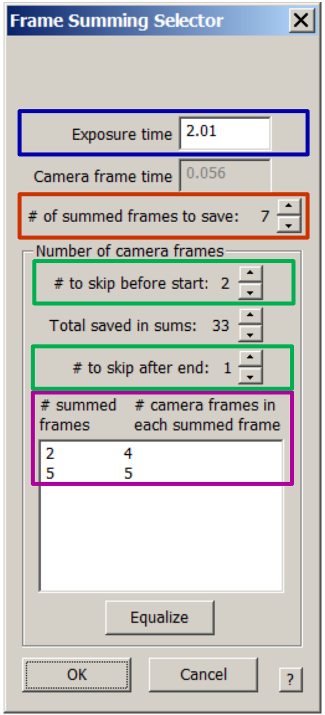
Intermediate frame saving is ON in FEI dialog

☑ Save frames

7 intermediate frames will be saved

Controlling Which Frames Are Saved

- In the simplest case, you set the exposure time and set the selector to save 7 summed frames
- It is often necessary to skip a camera frame at the start and/or end, so there are selectors for that
- When you change exposure in the main dialog, the # of camera frames in each summed frame is automatically redistributed



Frame Summing Selector

Exposure time 2.01

Camera frame time 0.056

# of summed frames to save: 7

Number of camera frames

to skip before start: 2

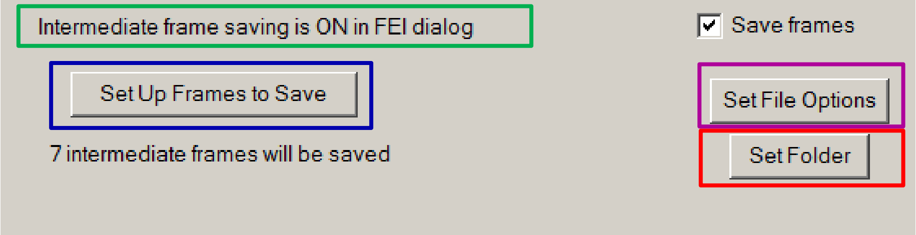
Total saved in sums: 33

to skip after end: 1

# summed frames	# camera frames in each summed frame
2	4
5	5

Saving Frames from Falcon 2

- SerialEM may or may not be able to control or even know whether frame saving is turned on in the separate FEI dialog, so there can be either a checkbox to tell SerialEM it is on, a status line like here, or no text at all
- The FEI software can save up to 7 (or up to 40) sums of sequential frames (camera readouts) and the Set Up dialog lets you control that
- A folder must be defined: anywhere accessible from SerialEM
- Set File Options opens a dialog just for file name and folder control



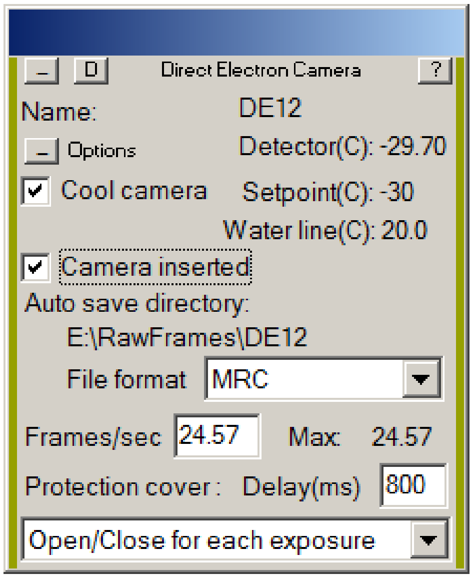
Intermediate frame saving is ON in FEI dialog

☑ Save frames

7 intermediate frames will be saved

The Control Panel for DE Cameras

- It has controls for the frame file format, frame rate, and a few other features



Name: DE12

Detector(C): -29.70

☑ Cool camera

Setpoint(C): -30

Water line(C): 20.0

☑ Camera inserted

Auto save directory:

E:\RawFrames\DE12

File format: MRC

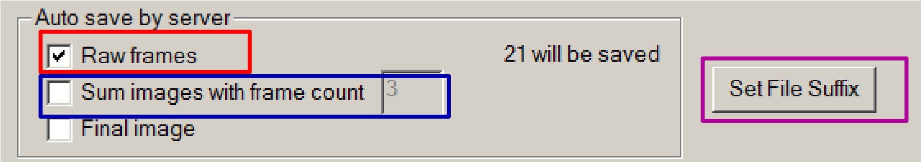
Frames/sec: 24.57 Max: 24.57

Protection cover : Delay(ms) 800

Open/Close for each exposure

Saving Frames from Direct Electron Cameras

- Frames are saved by the DE server and are not currently accessible from SerialEM
- Both raw frames and summed frames can be saved
- These frames need dark-subtraction and gain-normalization
- Set File Suffix opens a dialog just to control a portion of the file name



Auto save by server

☑ Raw frames

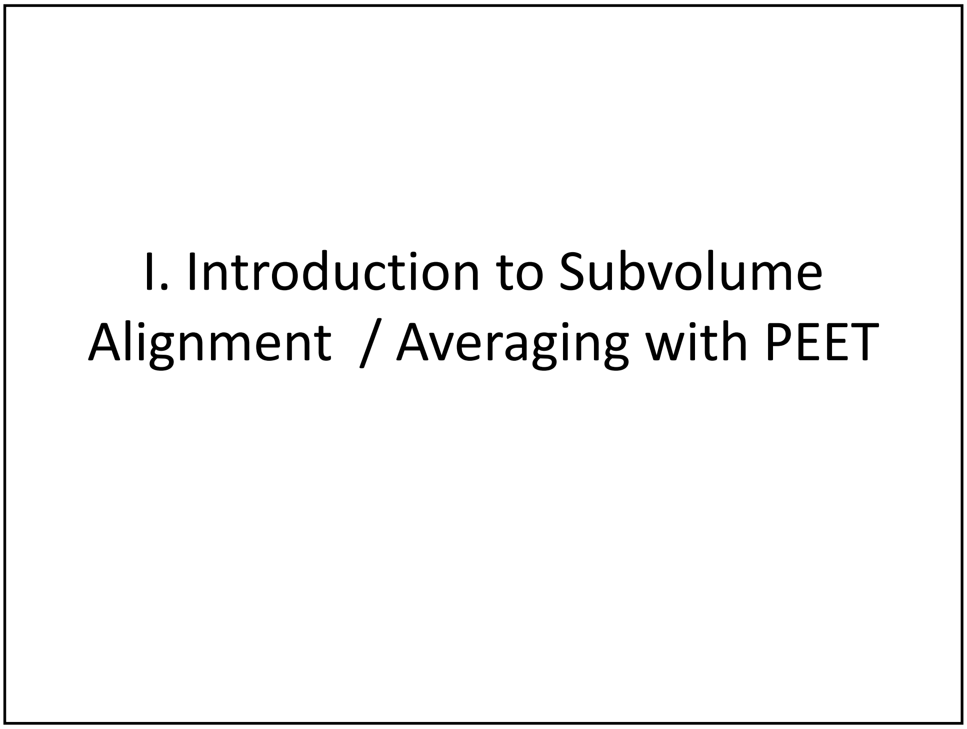
□ Sum images with frame count 3

□ Final image

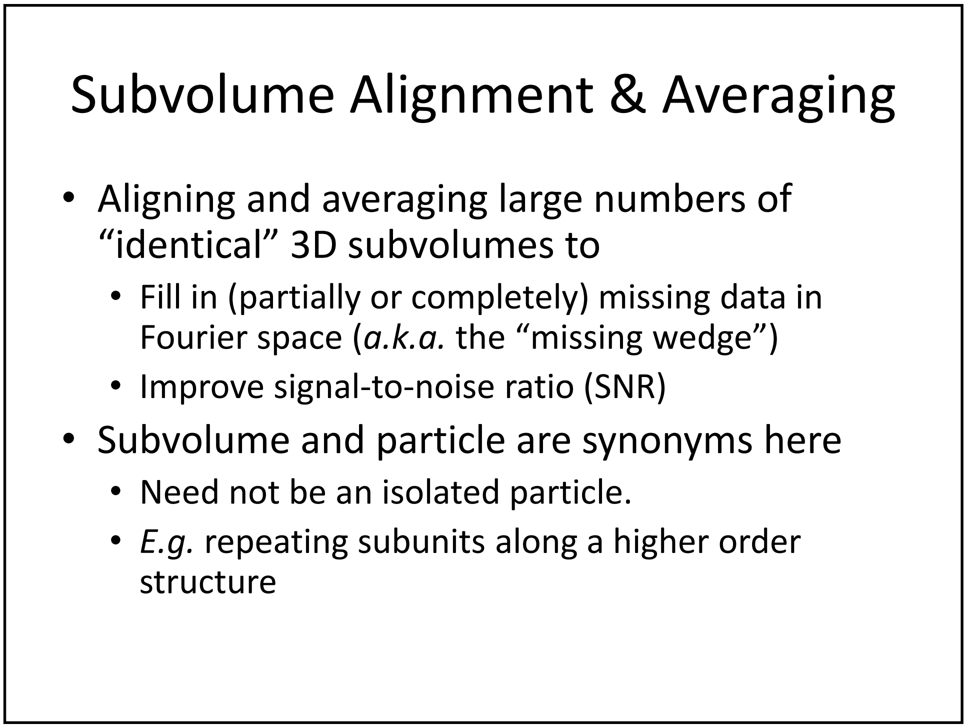
21 will be saved

THIS PAGE INTENTIONALLY LEFT BLANK

	June 28 (Wed)
8:00	Security
	Questions
	Introduction to PEET
9:00	John
	Walkthrough Introduction to PEET
	John
10:00	
	Coffee Break
11:00	BPV: Icosahedral Virus
	BPV
12:00	
	Lunch and Poster Session
1:00	
	Microtubules: Filaments
2:00	John
	Microtubules
3:00	
	Coffee Break
	Symmetrization
4:00	John
	Symmetrization with Demo First
	John
5:00	
6:00	
7:00	
8:00	
9:00	



I. Introduction to Subvolume Alignment / Averaging with PEET



Subvolume Alignment & Averaging

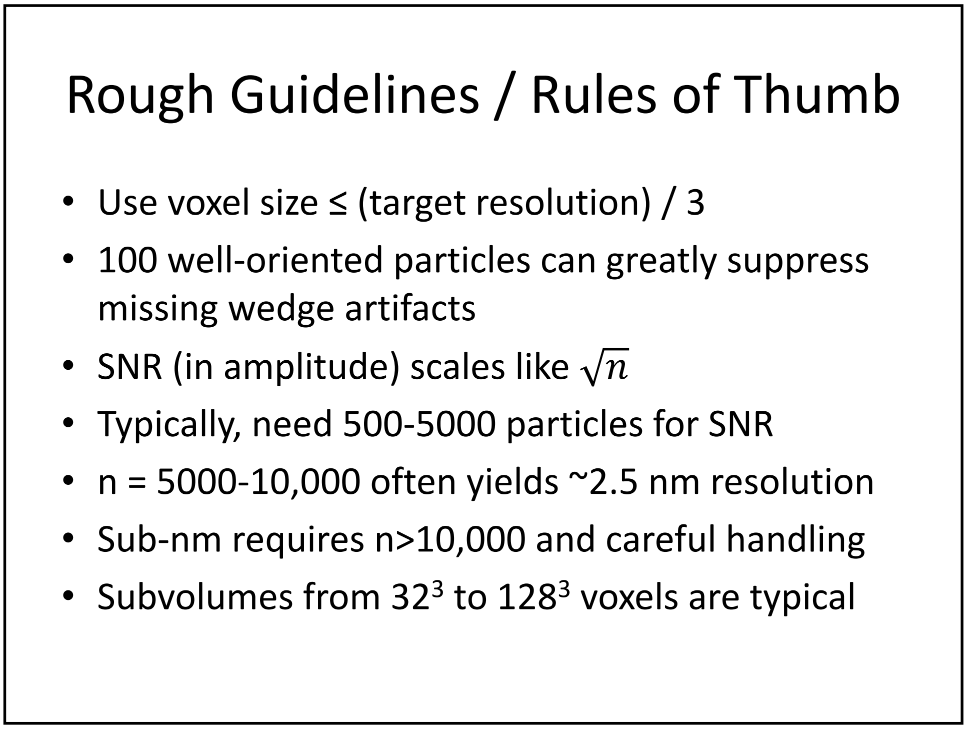
- Aligning and averaging large numbers of “identical” 3D subvolumes to
  - Fill in (partially or completely) missing data in Fourier space (*a.k.a.* the “missing wedge”)
  - Improve signal-to-noise ratio (SNR)
- Subvolume and particle are synonyms here
  - Need not be an isolated particle.
  - *E.g.* repeating subunits along a higher order structure

“Identical” Subvolumes?

- Biological samples are usually heterogeneous
  - We will discuss checking for and handling this
- Eliminate unnecessary variation beforehand!
  - Mixed samples
  - Imaging conditions… magnification, HV, etc.
  - Contrast
  - Ice thickness
  - Filament polarity
  - And so on…

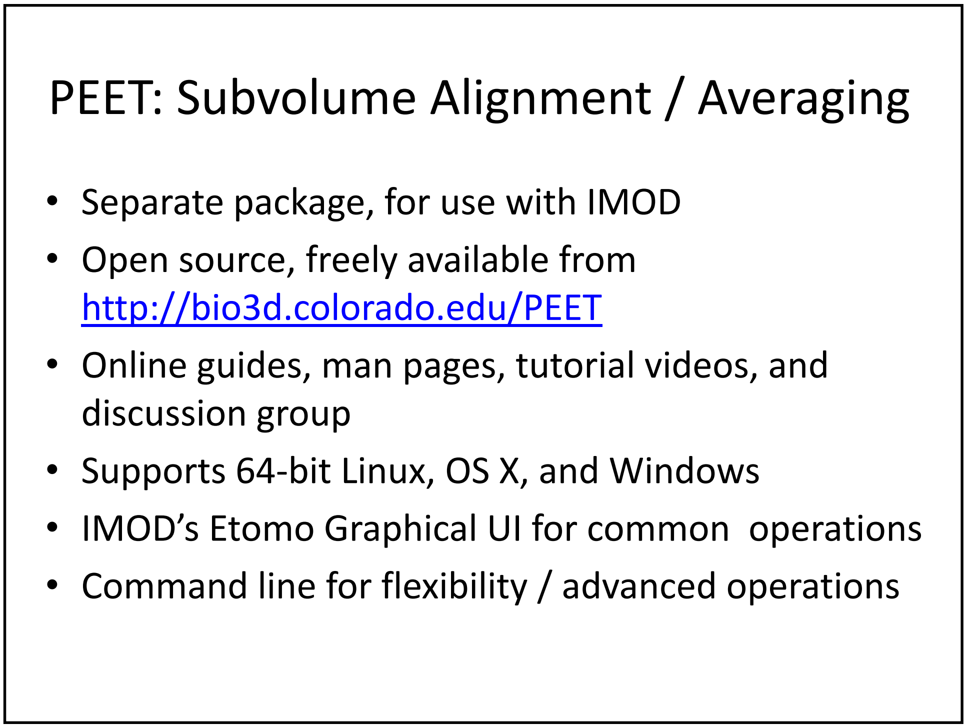
SVA Compared to SPA

- SVA
  - Uses 3D volumetric data from each article
  - Is good for *in vivo* or *in situ* data and portions of higher order structures, where unobstructed projections of individual particles are difficult or impossible to obtain
  - Typically yields lower resolution results than SPA for a given amount of input data
- Note on terminology: SVA ≈ STA ≈ SPT



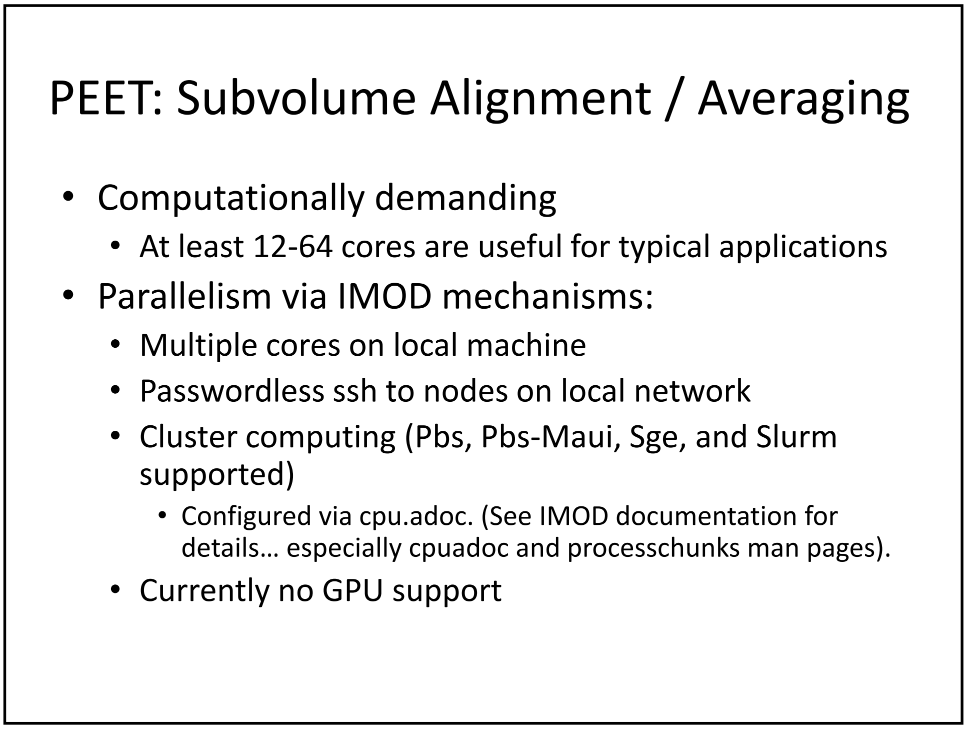
Rough Guidelines / Rules of Thumb

- Use voxel size ≤ (target resolution) / 3
- 100 well-oriented particles can greatly suppress missing wedge artifacts
- SNR (in amplitude) scales like $\sqrt{n}$
- Typically, need 500-5000 particles for SNR
- n = 5000-10,000 often yields ~2.5 nm resolution
- Sub-nm requires n>10,000 and careful handling
- Subvolumes from 32^3 to 128^3 voxels are typical



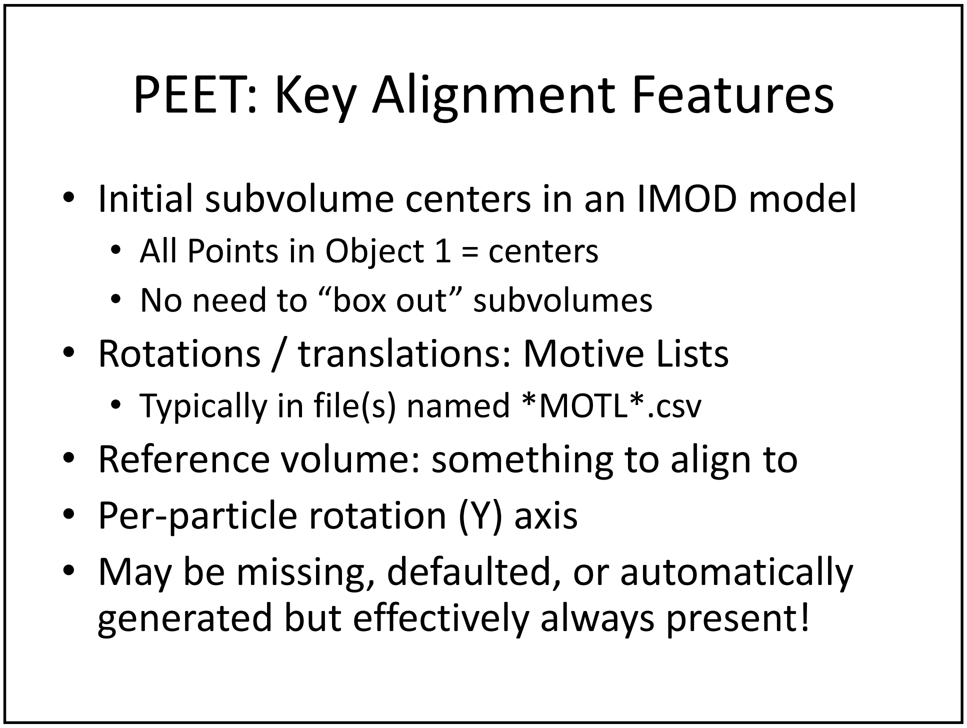
PEET: Subvolume Alignment / Averaging

- Separate package, for use with IMOD
- Open source, freely available from http://bio3d.colorado.edu/PEET
- Online guides, man pages, tutorial videos, and discussion group
- Supports 64-bit Linux, OS X, and Windows
- IMOD’s Etomo Graphical UI for common operations
- Command line for flexibility / advanced operations



PEET: Subvolume Alignment / Averaging

- Computationally demanding
  - At least 12-64 cores are useful for typical applications
- Parallelism via IMOD mechanisms:
  - Multiple cores on local machine
  - Passwordless ssh to nodes on local network
  - Cluster computing (Pbs, Pbs-Maui, Sge, and Slurm supported)
    - Configured via cpu.adoc. (See IMOD documentation for details… especially cpuadoc and processchunks man pages).
  - Currently no GPU support



PEET: Key Alignment Features

- Initial subvolume centers in an IMOD model
  - All Points in Object 1 = centers
  - No need to “box out” subvolumes
- Rotations / translations: Motive Lists
  - Typically in file(s) named *MOTL*.csv
- Reference volume: something to align to
- Per-particle rotation (Y) axis
- May be missing, defaulted, or automatically generated but effectively always present!

Multiple Coordinate Systems

- Global / Tomogram coordinates
  - Fixed
  - Used for subvolume centers, motive list rotations / translations, and search distances
- Per-particle coordinates
  - Attached to and move with each particle
  - Particle Y = particle rotation axis
  - Used to specify angular searches

3D Rotations In PEET

- Etomo angular search ranges: Y(ϕ)-Z(Θ)- X(Ψ) in particle coordinates (fixed at start of search)
- Motive list entries: Z-X-Z Euler angles in global coordinates
- IMOD Slicer angles: Z-Y-X angles in global coordinates
- Confusing variety!
- You will seldom have to deal with this directly

Questions?

A Few Words on Logistics…

- Presentations will cover concepts
- Lab exercises will explore details
  - We will announce time available for each lab
  - Try to pace yourself and complete topics of greatest interest
  - Okay if you can’t complete all the exercises
  - Labs designed and available for your later use
  - We will skip most alignments to save time

THIS PAGE INTENTIONALLY LEFT BLANK

Introduction

MODELING WITH AN ACCESSORY PROGRAM (STALKINIT); EXPLORING PEET FILES AND STRUCTURE.

In this exercise, we'll use unrealistically simple data (no noise, no missing wedge) to explore the Etomo GUI, some of the files used by PEET, and use of an aid to simplify the model task and speed alignment.

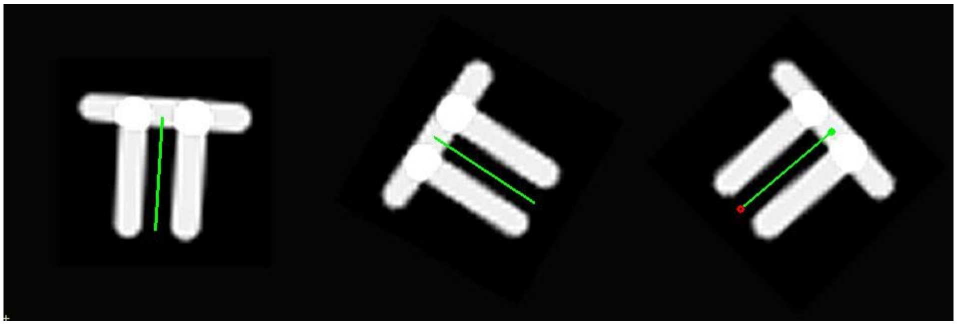
1) At a shell prompt execute

   ```
   cd $WORKSHOP_HOME/PEET_Labs/Intro-1
   ```

2) Open *pi-a.rec* in 3dmod by running `3dmod pi-a.rec`. Adjust the zoom factor to **4** by pressing the **=** or **–** keys or by clicking on the up or down arrow icons, and then press **Shift+R** to re-size the ZaP window accordingly. You'll see 3 cartoons of the Greek letter pi, with differing orientations but all conveniently (and unrealistically!) centered in the XY plane.

3) Rather than simply placing a model point near the center of each particle, we'll create an initial, accessory model, which will define the rotation axis for each particle. In 3dmod, select **Model** mode, press **Edit / Object/ Type** and select **Open** as the Object Type. Then close the Object Type dialog.

4) Middle-click in the middle of the crossbar at the top (the "head") of the left-most pi to place a model point there, then middle-click in the middle of the base of this letter pi to place the "tail" there.



This simple 2-point contour (actually the vector from tail to head) defines the rotation axis (Y) for this particle. Press **N** or **Edit / Contour / New** to create a new contour, and similarly place 2 points in the middle pi, again placing the head first. Repeat for the right-most pi. In the 3dmod info window go to **File / Save Model As** *pi-stalks.mod*. You should end up with 3 open contours with 2 points in each contour as shown above. Your first point of each contour should be the head and the second point should be the tail. Exit 3dmod.

5) This model is not suitable for use as input to a PEET alignment, since there are 2 points per particle. Instead, we'll use it as input to a program stalkInit to generate the final model in addition to a RotAxes file and an initial motive list. To learn more about stalkInit run `PEETHelp` and select the entry for stalkInit or run `man stalkInit.` Use "**q**" to exit a man page invoked at the command prompt. PEETHelp does not require Internet access, although the results will be displayed in a web browser.

6) Suppose you were familiar with stalkInit, but had forgotten the arguments it needs. Run `stalkInit` with no arguments, and it will print a brief usage message summarizing the available arguments. Most PEET programs follow this convention. PEETHelp and plotFSC, which work with no arguments, are exceptions.

7) Run `stalkInit pi-stalks.mod 1 0`
   The first argument, 1, means that we will be using particle 1 (the pi on the left) as our reference particle. The 3d argument, 0, says to suppress random rotations about the particle Y-axes. Often you can just run stalkInit with only the name of the input model. In this case, since the particles are already in the XY plane, random axial rotation would be counter-productive.

8) stalkInit will list the names of the three output models it generates. They differ only in whether the subvolume will be centered at the head (*head.mod*), the tail (*tail.mod*), or their midpoint (*centroid.mod*). We want the midpoint, so we'll rename that model to *pi.mod* and delete the other two. Execute:

   ```
   mv centroid.mod pi.mod
   rm head.mod tail.mod
   ```

9) stalkInit also generated a rotation axes file, *pi-stalks_RotAxes.csv*, and an initial motive list, *pi-stalks_InitMOTL.csv*. Notice that both types of files end in .csv, which stands for "comma-separated-value". CSV files are text files containing simple values separated by commas. They're used for several purposes in PEET, since they're human readable and editable with text editors or spreadsheet programs. Please keep in mind that while all motive lists in PEET are CSV files, not all CSV files are motive lists!

10) Examine the contents of the RotAxes file by running `cat pi-stalks_RotAxes.csv`. You will see that it contains 3 lines (one for each particle), and each line contains 3 numbers; these are the X, Y, and Z components of the rotation axis vector for the corresponding particle in tomogram coordinates.

11) PEET runs generate many output files, so we require each run to be in its own "project" directory. We've already created a subdirectory, *PEET/run1* for you to use. Move the rotation axis file and initial motive list into that directory, renaming them as you do so:

```
mv pi-stalks_InitMOTL.csv PEET/run1/initMOTL.csv
mv pi-stalks_RotAxes.csv PEET/run1/pi_Tom1_RotAxes.csv
```

PEET can only use rotation axes files in the project directory with the above naming convention or with a different naming convention in the directory where the tomogram is located. See yaxisType on the PEET man page for more details. Putting them in the project directory, as we've done here, is recommended.

12) Next, we'll set up and do a PEET run using the Etomo GUI. Execute `cd PEET/run1`, and then run `etomo`. (Note the command must be all lower case). Select **Subvolume Averaging (PEET).** Normally, at this point, you would create a new project and enter project settings manually. We'll save some time by instead copying the settings from a project I've already set up. In the resulting **Starting PEET** dialog, select **Copy project from**, use the File Chooser to go up one directory and then into **run1Done**. Select **pi.epe**, and press **OK**. The newly copied project will initially be set up to use the same input files as the project we copied from, which is typically not what you want, so we'll correct that next.

13) On the Etomo **Setup** tab, press the **>** symbols above the **Volume**, **Model**, and **Initial MOTL** columns to display the full path to the corresponding files. You'll see that the volume and model files are from 2 directories up (**../../**), which happens to be correct in this case. The initial motive list is wrong, however. Either remove the leading **../run1Done/** prefix manually, or use the File Chooser to browse to and select the *initMOTL.csv* file in the *PEET/run1* project directory.

14) Review the various settings on the **Setup** and **Run** tabs to see if you understand them. Hover your mouse over a field or a control in the Etomo GUI to view a pop-up tool tip. You can also right-click on a blank area to access the PEET Users Guide.

15) Switch to the **Run** Tab and press **Run** at the bottom left.

16) This small project will not take long to run, even with only a few cpus. While it's running, we'll explore what's going on and look at some of the intermediate and output files. In the terminal, type `ls` to display the contents of the directory. You'll notice files *pi.epe* and *pi.prm*. The epe file, called the project file, is what you'll use to re-open this same project at a later time: `etomo *.epe`. You'll rarely need to be concerned about its contents, although it is readable text. The prm or parameter file is where settings you make in the Etomo GUI are stored, in readable MATLAB syntax. Some advanced options are available only by manually editing this file; we'll see examples of this later in the workshop.

17) When the run was started, Etomo ran a PEET program prmParser to split this job up into multiple tasks or "chunks". In the project directory, there are a series of files *pi-start.com*, *pi-001.com*, … *pi-finish.com*. Each of these contains the commands for a single chunk. Run `cat pi-start.com` to examine the contents of this file. You will see that it executes 3 programs: PEETCleanup removes output from any previous runs, while prepareRef and prepareEM, respectively, generate the initial reference and motive list for the first iteration. Once the alignment has finished, most **.com* files will be automatically deleted, but you can always re-create them if desired by running `prmParser *.prm.` Examining the com and log files in various scenarios is a good way to learn more about individual programs used in a PEET run. *pi-001* through *pi-003.com* each run alignSubset to align a single particle at iteration 1. A corresponding log file (*e.g. pi-start.log, pi-002.log, etc.*) will be created as each chunk is executed. Run `gedit  pi-002.log` to examine one of these log files. As each chunk completes successfully, CHUNK DONE is appended to the end of its log file. Exit gedit when finished.

18) After the run finishes, press **Open averages in 3dmod** in Etomo. This will automatically open the Isosurface and ZaP views. Alternatively, you can run `3dmod *AvgVol*.mrc` at the command line, and open the Isosurface view using the hotkey Shift+U or from the Image menu. Adjust the Isosurface Threshold so that the pi is visible. Examine the resulting averages with 1, 2, and 3 particles in the ZaP window, using the **4th D** arrows, and in the Model View. Not surprisingly, they're nearly identical in this toy example. Exit 3dmod when finished.

Run `3dmod  pi_Ref2.mrc  unMaskedpi_Ref2.mrc` to examine the references used as input to search iteration 2. Notice that the references are considerably larger than the requested Volume Size (found on the Setup tab). PEET automatically chooses an appropriate reference size for you except when you supply an external reference. Notice how masking eliminates the ghostly copies of neighboring particles that are present in the unmasked references at iterations after the 1st. Appropriate masking can reduce interference from noise and extraneous structures leading to better alignments. Exit 3dmod when finished.

19) Run `cat  pi_MOTL_Tom1_Iter*.csv` and examine the initial, intermediate, and final motive lists. The file named **Iter1** is the input to the first iteration (the initial motive list), while **Iter3** is the output motive list from the 2nd, final search iteration. (*i.e.* it would be input to iteration 3 if there were one). Motive lists and references are special in terms of this off-by-one numbering. Most other files are numbered by the iteration at which they are produced.

Compare the shifts (in columns **11-13**) and the Euler angles (in columns **17-19**) in the initial motive list with those in the final output. In this case, the initial motive list created by stalkInit was quite good and only minor adjustments to position and orientation are found at later iterations. Also of interest are the cross-correlation coefficients in column 1. In this case, they are close to 1 indicating nearly perfect correlation. If only real data behaved like this!

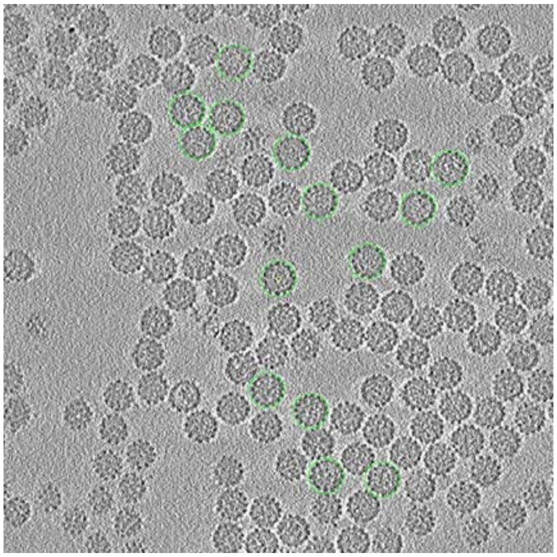
20) After alignment, we have an initial model giving starting positions, and the final motive list containing changes to those locations as well as rotations to be applied. In many case, it will be desirable to create new model(s) containing the final, corrected positions and new motive list(s) containing only the rotations. This is easy to do. Run `createAlignedModel  *.prm` and note the names of the resulting output files: *pi_Tom1_Iter2.csv, pi_Tom1_Iter2_Summary.csv, pi_Tom1_Iter2.mod, pi_Tom1_Iter2_RotAxes.csv*. If you wish, examine the newly generated motive list by running `cat pi_Tom1_Iter2.csv` and notice that the shifts are now all 0's since the new model contains the aligned locations. CreateAlignedModel has several useful properties, which we will explore in later exercises.

21) When you have finished examining your results, exit Etomo, close any remaining 3dmod windows, and clean up the directory by executing `PEETCleanup  *.prm`. If you wish, `rm  *~` will also remove backups; please be careful not to put a space between the * and the ~ however, or you will remove everything!

When run with only a prm file, PEETCleanup deletes only intermediate files. When invoked with an optional 1 following the prm file, it also deletes output files, saving a backup copy of them. The later form is invoked automatically each time you start a new run.

Aligning Isolated Particles: BPV

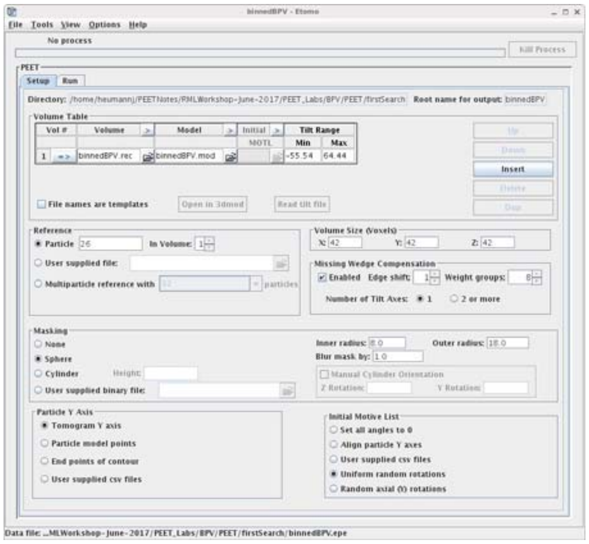
2X Binned BPV Sample -8 µm Defocus



Modeling BPV

- Always try to model accurately
    - improves the odds of a good alignment / average
- Center points in XY and Z as well as you can
- Setting “Sphere Radius for Points” and paging up / down helps with centering points in Z
- May wish to start with binned or binned and NAD-filtered data

Alignment Settings – Setup Tab



Directory: /home/heumannj/PEETNotes/RMLWorkshop-June-2017/PEET_Labs/BPV/PEET/firstSearch Root name for output: binnedBPV

Volume Table

Vol #	Volume	Model	Initial MOTL	Tilt Range Min	Tilt Range Max
1	binnedBPV.rec	binnedBPV.mod		-55.54	64.44

□ File names are templates

Reference
● Particle 26 In Volume: 1
○ User supplied file:
○ Multiparticle reference with particles

Volume Size (Voxels)
X: 42 Y: 42 Z: 42

Missing Wedge Compensation
☑ Enabled Edge shift: 1 Weight groups: 8
Number of Tilt Axes: ● 1 ○ 2 or more

Masking
○ None
● Sphere
○ Cylinder Height:
○ User supplied binary file:
Inner radius: 8.0 Outer radius: 18.0
Blur mask by: 1.0
□ Manual Cylinder Orientation
Z Rotation: Y Rotation:

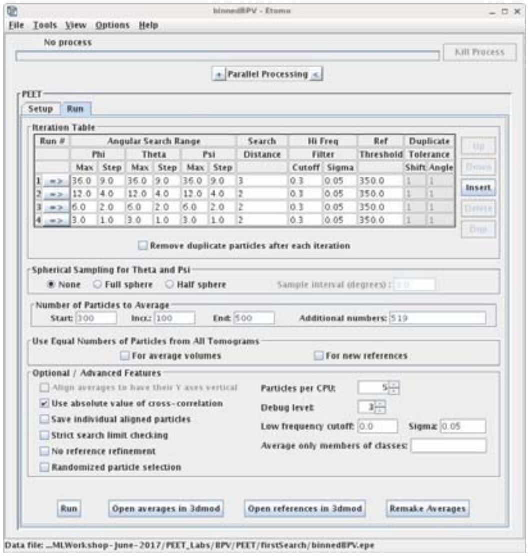
Particle Y Axis
● Tomogram Y axis
○ Particle model points
○ End points of contour
○ User supplied csv files

Initial Motive List
○ Set all angles to 0
○ Align particle Y axes
○ User supplied csv files
● Uniform random rotations
○ Random axial (Y) rotations

Data file: ...MLWorkshop-June-2017/PEET_Labs/BPV/PEET/firstSearch/binnedBPV.epe

- Volume(s)
- Model(s)
- Tilt range
- Reference
- Size to average
- Mask Definition
- Particle Y Axis
- Initial Motive List

Alignment Settings – Run Tab



Iteration Table

Run #		Angular Search Range						Search	Hi Freq		Ref	Duplicate	
		Phi		Theta		Psi		Distance	Filter		Threshold	Tolerance	
		Max	Step	Max	Step	Max	Step		Cutoff	Sigma		Shift	Angle
1	=>	36.0	9.0	36.0	9.0	36.0	9.0	3	0.3	0.05	350.0	1	1
2	=>	12.0	4.0	12.0	4.0	12.0	4.0	2	0.3	0.05	350.0	1	1
3	=>	6.0	2.0	6.0	2.0	6.0	2.0	2	0.3	0.05	350.0	1	1
4	=>	3.0	1.0	3.0	1.0	3.0	1.0	2	0.3	0.05	350.0	1	1

□ Remove duplicate particles after each iteration

Spherical Sampling for Theta and Psi

● None ○ Full sphere ○ Half sphere Sample interval (degrees): 1.0

Number of Particles to Average

Start: 300 Incr.: 100 End: 500 Additional numbers: 519

Use Equal Numbers of Particles from All Tomograms

□ For average volumes □ For new references

Optional / Advanced Features

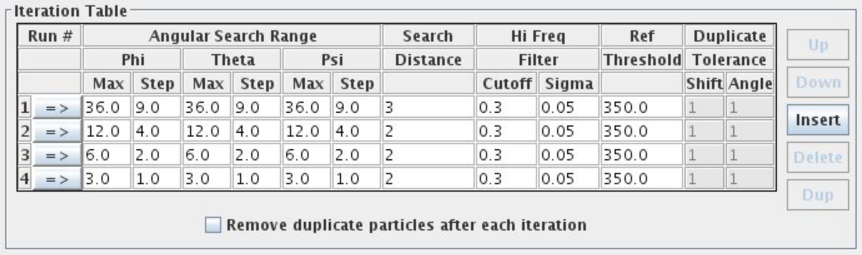
□ Align averages to have their Y axes vertical
☑ Use absolute value of cross-correlation
□ Save individual aligned particles
□ Strict search limit checking
□ No reference refinement
□ Randomized particle selection

Particles per CPU: 5
Debug level: 3
Low frequency cutoff: 0.0 Sigma: 0.05
Average only members of classes:

Data file: ...MLWorkshop-June-2017/PEET_Labs/BPV/PEET/firstSearch/binnedBPV.epe

- Parallel Processing
- Iteration Table
- Spherical Sampling
- Number to Average
- Equal Numbers?
- Optional / Advanced Features

Iteration Table



Iteration Table

Run #		Angular Search Range						Search	Hi Freq		Ref	Duplicate	
		Phi		Theta		Psi		Distance	Filter		Threshold	Tolerance	
		Max	Step	Max	Step	Max	Step		Cutoff	Sigma		Shift	Angle
1	=>	36.0	9.0	36.0	9.0	36.0	9.0	3	0.3	0.05	350.0	1	1
2	=>	12.0	4.0	12.0	4.0	12.0	4.0	2	0.3	0.05	350.0	1	1
3	=>	6.0	2.0	6.0	2.0	6.0	2.0	2	0.3	0.05	350.0	1	1
4	=>	3.0	1.0	3.0	1.0	3.0	1.0	2	0.3	0.05	350.0	1	1

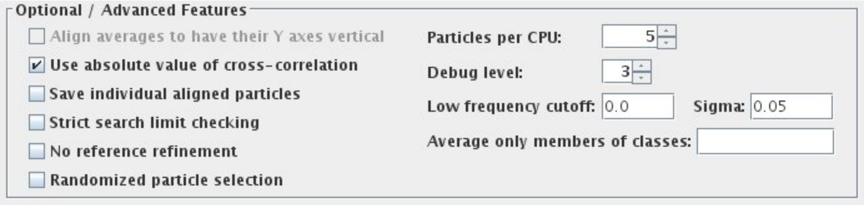
□ Remove duplicate particles after each iteration

- Search settings based on model uncertainty
- Iteration n Max ~ 1.5X iteration n-1 Step
- 3:1 ratio Max:Step good for speed / accuracy
- Angular settings decrease 2X row-to-row

Finest Angular Resolution?

- R sin(dΘ) ≈ R dΘ = 1 [voxels]
- dΘ ≈ 1 / R [radians] = 180 / (π R) [degrees]
- In this case, mask radius = 18
  so dΘ ≈ 3°
- Actually searched down to 1°
- Last 2 iterations shouldn't change much
- This is the case, as we'll verify in the lab

Optional / Advanced Features



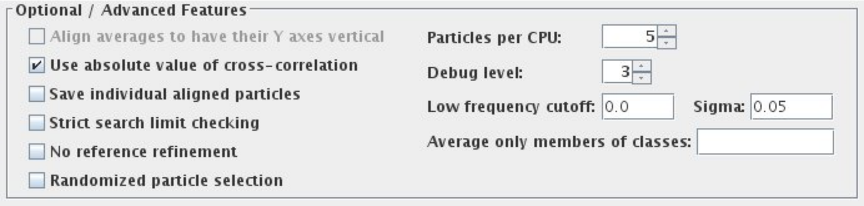
Optional / Advanced Features

☐ Align averages to have their Y axes vertical
☑ Use absolute value of cross-correlation
☐ Save individual aligned particles
☐ Strict search limit checking
☐ No reference refinement
☐ Randomized particle selection

Particles per CPU: 5
Debug level: 3
Low frequency cutoff: 0.0 Sigma: 0.05
Average only members of classes:

- Align vertical: not available with Tomogram Y
- Absolute value:|CCC| instead of CCC
  - avoid Einstein-from-the-noise
  - problematical with some pattern… *e.g.* Zebra stripes
- Strict search limits: never exceed max step
- No reference refinement: template matching

Optional / Advanced Features



Optional / Advanced Features

- □ Align averages to have their Y axes vertical
- ☑ Use absolute value of cross-correlation
- □ Save individual aligned particles
- □ Strict search limit checking
- □ No reference refinement
- □ Randomized particle selection

Particles per CPU: 5

Debug level: 3

Low frequency cutoff: 0.0 Sigma: 0.05

Average only members of classes:

- Randomized selection: not based on CCC
- Particles per cpu: determines number of chunks
- Debug level: amount of detail, seldom used
- Low frequency cutoff: attenuate shading gradients, seldom used
- Average only: for use with heterogeneous data

Questions?

THIS PAGE INTENTIONALLY LEFT BLANK

BPV

SIMPLE MODELING WITH SPHERICAL PARTICLES

1. `cd $WORKSHOP_HOME/PEET_Labs/BPV_-3`

2. Open the tomogram and create a new model by typing:
   ```
   3dmod bpv_bin2.rec.nad myBPV.mod
   ```

3. Select **Model** mode

4. Press **Edit / Object / Type** and select **Scattered**. Leave the Object Edit window open.

5. Adjust zoom and the ZaP window size to settings you find comfortable for modeling. Notice how particle diameters change as you **PgUp / PgDn**, reaching a maximum when the slice cuts through the center of the particle.

6. Adjust Z slice for maximum diameter of some particle near the center of your view, which is not touching other particles or obscured by gold or contamination. Place a model point near the center of this particle by clicking with the middle mouse button.

7. In the **Object Edit** dialog, increase **Sphere radius for points** until the resulting green circle is just large enough to just include the entire virus particle (~**19**). This circle makes it easier to see if your model point is correctly centered. If you find the green circle difficult to see, try switching to the Slicer window (hotkey \) and increasing **Mod** to ~**10** or increase the **line width** in the **Edit / Object / Type** dialog. In the Slicer window, adjust the position of the point by right clicking until you are satisfied that it is well-centered (in all 3 dimensions!). In both the Slicer and ZaP windows, left, middle, and right mouse clicks, respectively, select an existing model point, add a new one, or move an existing one. In ZaP movement by right-clicking is limited to the plane containing the point, while in Slicer it is not.

8. Try adding at least 10 more points, leaving the sphere radius as you've set it. You'll find this makes accurate modeling in all 3 dimensions much easier. Choose particles that are not touching other particles, gold, or contamination and are at least half a particle diameter away from any edge of the volume. When finished, you can save your model (**File / Save Model**). Examine the supplied model by going to **File / Open Model** and choosing *bpv_bin2.mod*. Exit 3dmod when finished.

CHECKING MASK SIZE AND LOCATION

9. *PEET/firstSearch* contains an already completed PEET run on the 2X binned data. PEETCleanup has already been run to remove intermediate files, leaving important output files. Recall that when using a mask, one of the first things to do is verify that the mask is correctly sized, located, and (in the case of non-spherical masks) oriented.
   ```
   cd PEET/firstSearch
   etomo *.epe
   ```
   On the Setup tab, notice that we've chosen a spherical mask with inner and outer radii of 10 and 22 voxels to be blurred with a Gaussian of standard deviation 1 voxel. Close Etomo.

10. `3dmod bpv*Ref*.mrc`
    This opens all of the masked references and we will go through them one-by-one using the **4th D** arrows (hotkeys **1** and **2**) in the ZaP window.

11. Examine *bpv_Ref1.mrc* and notice that the mask is well centered and that the inner and outer radii seem appropriately set to include all the capsid, with minimal interference from the central nucleic acid. Note also the soft mask edges. When using masking, it is a good idea to check the mask settings shortly after starting the run. Recall that the **Ref1*.mrc* initial reference files are created very quickly. If the mask settings are incorrect, it makes sense to kill the run and fix them rather than waiting for the run to complete.

VERIFY ANGULAR SEARCH STEPS

12. The angular resolution calculation we discussed in the lecture suggested that the last 2 iterations, with search steps of 2 degrees and 1 degree, were probably superfluous and unlikely to improve results. Verify that this is the case by comparing references **Ref3.mrc* through **Ref5.mrc* and observing that they are almost unchanged from **Ref2.mrc*. Exit 3dmod when finished.

SET UP A SPHERICAL SEARCH

For BPV, icosahedral symmetry allowed us to limit our initial angular search range. In this exercise, we'll examine how the settings would be different if that were not true. First we'll copy the existing firstSearch project settings, and then modify them appropriately for randomly oriented particles with no symmetry.

13.
```
mkdir ../mySphericalSearch
cd ../mySphericalSearch
```

14. As in the Introductory lab, run etomo and copy project settings from **../firstSearch/bpv.epe** (step 12 in the Introductory exercise).

15. On the lower right corner of the **Setup** tab, change **Initial motive list** from Uniform random rotations to **Set all angles to 0**. Since we are now assuming a full spherical search will be required, random rotations serve no purpose in suppressing missing wedge artifacts; the full search would allow the artifacts to simply re-align. Randomization is typically only helpful when a limited angular search can be used.

16. Switch to the **Run** tab.

17. Under **Spherical Sampling for Theta and Psi**, select **Full sphere**. Notice how the Theta and Psi fields for Iteration 1 in the Iteration Table are now greyed out and unavailable. These will be handled automatically by the spherical search algorithm.

18. We will use the same Phi step (9°) as previously. Since we're doing a full spherical search, we need to search from -180° to +180°; change the **Phi Max** setting at Iteration 1 from 36.0 to **180.0**.

19. Search parameters for Theta and Psi will be set automatically, but we need to specify the **Sample interval (degrees)** or finest search step to be used. Normally, this is just set equal to the Phi step at Iteration 1. In this case, that's **9.0** which happens to be the default, so we don't need to change anything.

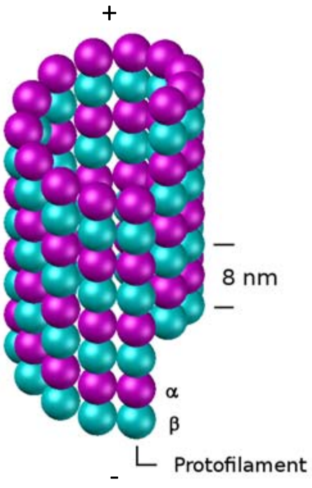
20. We found above that possibly Iteration 3 and certainly Iteration 4 weren't really necessary. In the **Iteration Table**, Click on the **=>** to the right of the Iteration number (**Run** #) 4, and then press the **Delete** button at the right side of window to remove this row. Similarly, delete the row for Iteration 3 if you wish.

21. Exit Etomo and your settings will be automatically saved.

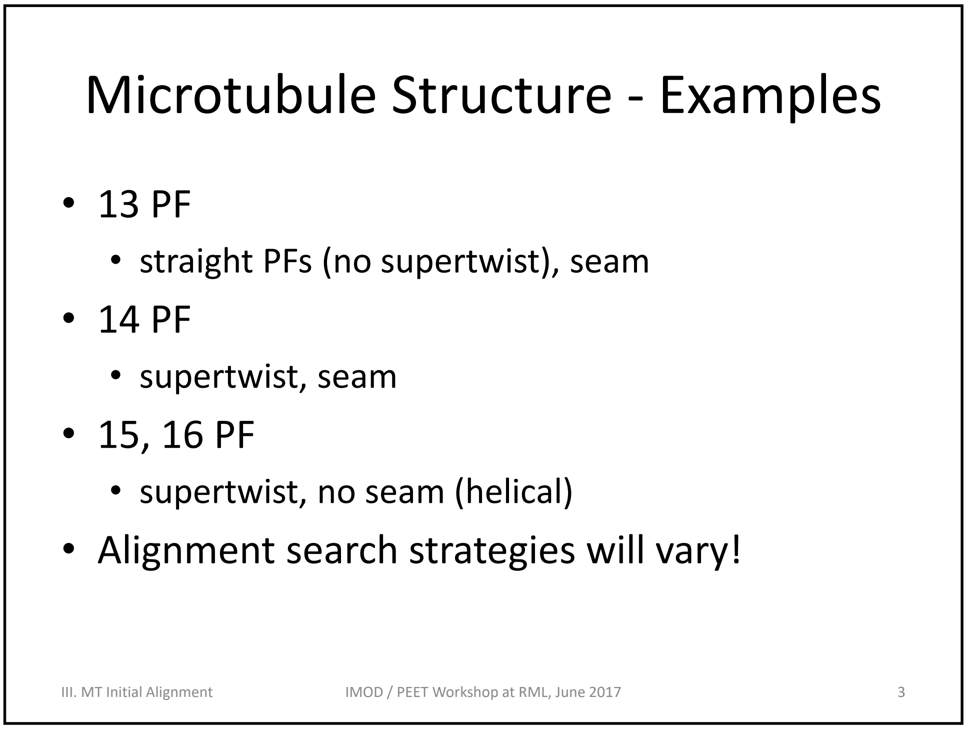
22. `cd  ../sphericalSearch` to see the results of a completed spherical search run that we've already done. Examine the results in 3dmod if you like by typing `3dmod  *AvgVol*.mrc`. You may also wish to open the Isosurface view (**Shift+U** or **Image / Isosurface**). Command line options are available to specify in advance which 3dmod views to open; consult the 3dmod man page for details. A spherical search run will take longer and may suffer more from missing wedge artifacts than the limited search range approach. Close 3dmod when finished.

Aligning Filaments: Microtubules

Microtubule Structure

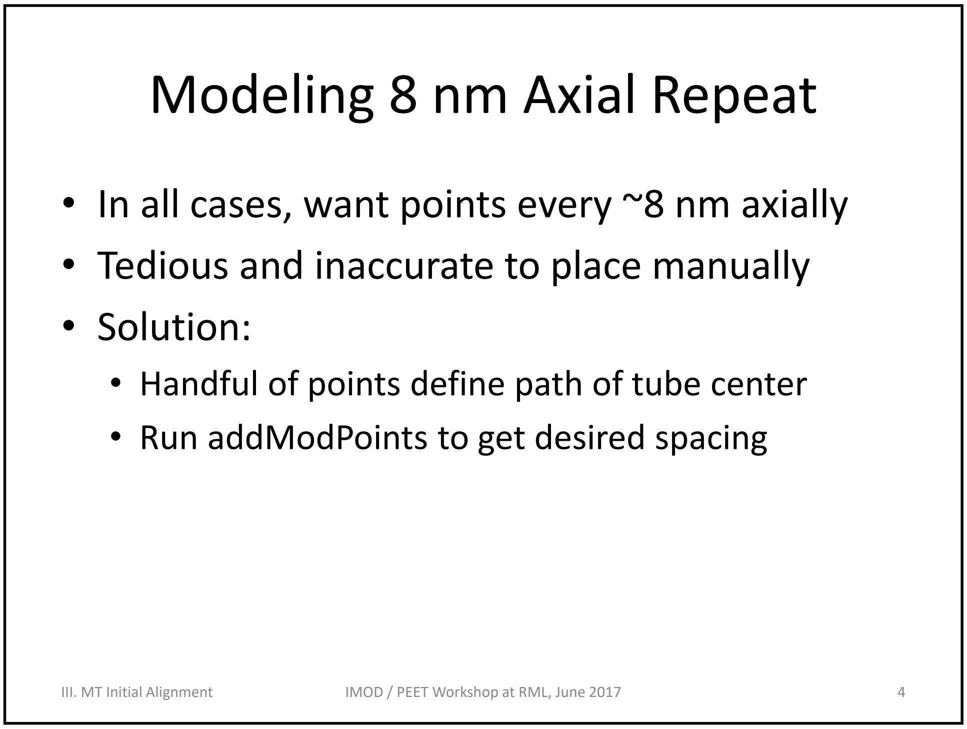


A 13 PF Microtubule



Microtubule Structure - Examples

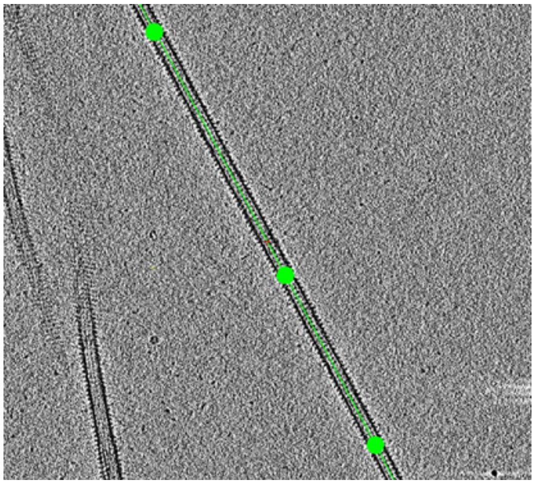
- 13 PF
  - straight PFs (no supertwist), seam
- 14 PF
  - supertwist, seam
- 15, 16 PF
  - supertwist, no seam (helical)
- Alignment search strategies will vary!



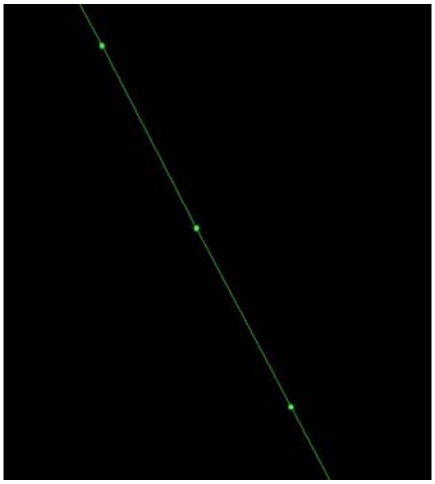
Modeling 8 nm Axial Repeat

- In all cases, want points every ~8 nm axially
- Tedious and inaccurate to place manually
- Solution:
  - Handful of points define path of tube center
  - Run addModPoints to get desired spacing

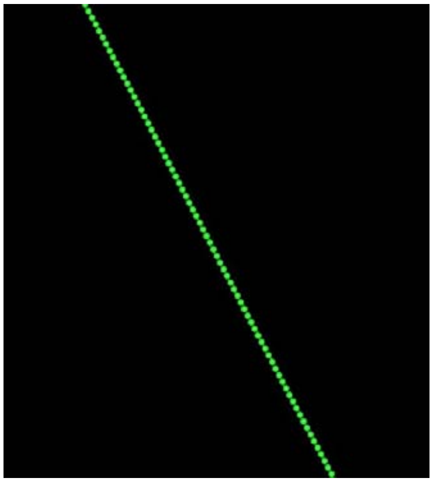
Initial Model: Tube Center



Use addModPts For Desired Spacing



Initial Model



Points Added

13 PF Search Strategy

- Straight (no supertwist), seam
- No axial randomization
- Align particle Y axes
- Missing wedge artifacts will be present
- Goal: visualize seam well enough to latter combine tubes with varying orientations to reduce / eliminate missing wedge artifacts

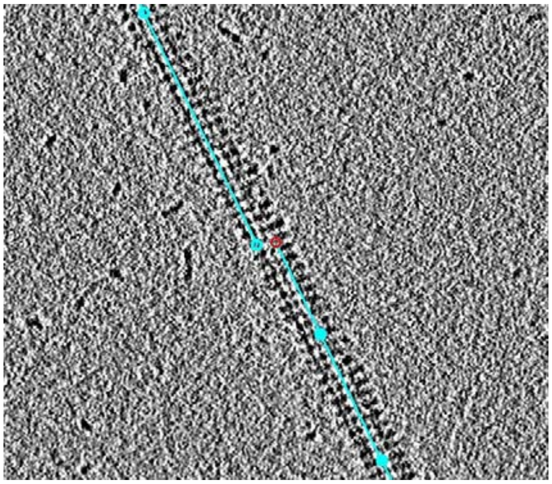
15-16 PF Search Strategy

- Slight supertwist, helical 16 nm pitch (no seam)
- Align particle Y axes + axial randomization
- 360 / 15 = 24° ϕ (Phi) search range
- Will be able to exploit helical symmetry
- Can also use modTwist2EM as described next for twisted, non-helical tubes

Other Numbers of PFs

- Supertwist and seam
- Twist can help reduce missing wedge artifacts
- How to handle the twisted seam?
  - Trace protofilaments (follow supertwist)
    - Object 1: points follow center of tube
    - Object 2: points follow protofilaments on surface
    - Contours correspond to protofilaments
    - Object 1 and 2 points in 1-to-1 correspondence (axial posn)
  - modTwist2EM generates an initial motive list which compensates for the supertwist
- Model twisted PFs before running addModPts (special addModPts mode for this case)

Sample Model Following PFs



Questions?

THIS PAGE INTENTIONALLY LEFT BLANK

Microtubules

SIMPLE FILAMENT MODELING

In this exercise, we'll look at modeling a simple, untwisted microtubule using an accessory program, addModPts.

1) `cd $WORKSHOP_HOME/PEET_Labs/MT`

2) Open the tomogram along with a new model.
   `3dmod series4-8um-cor.rec mySimpleModel.mod`

3) Switch to **Model** mode, select **Edit / Object / Type**, and choose **open**.

4) In the ZaP window, set zoom to 1.0 and center the microtubule in Z (Z = ~45). Drag the image with the mouse so the top left corner of the microtubule is in view. Starting at the upper left, and working down and to the right, middle-click to place 5-7 model points sequentially along the length of the tube near the center of the lumen. You will need to change Z height to follow the center of the tube since it is not quite in the XY plane, especially near the bottom right corner. Do this relatively quickly, not spending too much time to accurately center the model points, since we'll be correcting their location in the next step. You should end up with all your points in Contour 1. Save the model with hotkey **s**.

5) Open a Slicer window (**Image / Slicer** or hotkey \). Select **interpolation** (checkerboard) and **centering** (box within a box) modes by clicking the **1st** and **3rd** buttons at the top of the Slicer window. You may need to change the point selection for centering to take effect. Select either the first or last model points. Set the zoom to **2.5** and adjust the **Z rotation** until the tube is vertical (approximately **-27**°).

6) Set **X rotation** in the Slicer window to **-90**° so you are looking down on an XZ cross-section of the tube. Set **Img** to **30** to display a 30-slice thick view (approximately 27 nm). If the current model point does not appear well centered, right-click in the center of the tube to move the point location. You can right-click repeatedly until you are satisfied. As in a previous lab, you may optionally wish to set **Mod** to **10** and **Sphere radius for points** to ~**12** as an aid to more easily visualize when the point is well centered.

7) Adjust the remaining points by using the **scroll buttons** in the 3dmod info window or move to the previous or next point using the **[** and **]** hotkeys. With tubes more curved than this one, you might need to adjust the Z rotation when changing points. This tube is sufficiently straight so that no such adjustment is necessary. Accurate modeling is worth the effort! In difficult cases, it can make the difference between success and failure. Save your model when finished.

8) Next we want to insert model points approximately every 8 nm along the center of the tube, since this is the pitch of the α-β tubulin dimers. The voxel size is 9.06 Å, so we'll place a point every round(8 / 0.906) = 9 voxels, where "round" denotes the nearest integer.

```
addModPts mySimpleModel.mod 9
```

Notice that addModPts prints both the name of the output file (*mySimpleModel_PtsAdded.mod*) and the final number of points.

9) Open *mySimpleModel_PtsAdded.mod* on your tomogram by going to **File / Open Model** via the 3dmod info window to verify that you now have points evenly spaced along the length of the microtubule. Adjusting Sphere radius for points will make visualization easier. Exit 3dmod when finished. You've now completed a simple model suitable for aligning and averaging either a helical microtubule, such as the 15 PF sample we're currently working with, or a 13 PF tube with a seam. We'll look at the settings you might use for such alignments momentarily. First, let's see how to model a more complicated case in which we need to account for twisting of the tube about its axis.

MODELING A TWISTED FILAMENT

We'll consider how to model a twisted microtubule using programs addModPts and modTwist2EM. We need to create a model with 2 objects in point-to-point correspondence, with Object 1 following to the center of the microtubule and Object 2 tracing paths following contours indicating twist on the tube surface. We'll use protofilaments on the upper surface of the tube for the latter and will trace a given protofilament only until it starts to wrap around either side of the tube and becomes hard to follow in an XY cross section; then we'll start a new contour and switch to another protofilament. This process is not difficult, but is slightly tedious, so we've done this portion of the modeling for you to save time. Let's open the starting model, so you can see this initial model and how you would create such a model yourself.

10) `3dmod series4-8um-cor.rec stage1.mod` and select **Model** mode.

11) Open a Slicer window, set **zoom** to **1.0**, select **interpolation** (checkerboard) and **centering** (box within a box) modes, and set **Img** to **15** and **Mod** to **5**. Rotate the microtubule to be vertical and adjust the size of the Slicer window so that it fills your screen from top to bottom, allowing you to visualize a significant fraction of the microtubule's length. The 15-section thick slice makes the protofilaments readily visible when focused on the upper (or lower) surface of the microtubule. Additionally, we've already set Line thickness and Sphere radius for points to aid visualization of neighboring points and contours. This is often not necessary when modeling, but it's good to know that you can always make similar adjustments when you find them helpful. Open the Model View window by using the hotkey, **v**. Right-click on any model line in the Model View window to center that same line in the Slicer window.

12) Examine several points / contours. Notice:
    a) How the first point of a new contour is always placed at approximately the same axial position as the last point of the previous contour, and
    b) how we can skip one or more protofilaments when starting a new contour.

    PEET uses the single point overlap between neighboring contours to distinguish rotation due to supertwist from that caused by the jump from one protofilament to another.

13) Next, we will make a second object, with the same settings as Object 1. Select **Edit / Object / New**, press **Copy from object 1**, and **Done**. Somewhat counter-intuitively, this creates a new, but still empty Object 2. Select Object 1 by left clicking on any existing model point or by choosing **Object 1** in the 3dmod info window. **Press Edit / Contour / Copy**, select **Copy to Object#**, enter **2**, choose **All contours in object**, press **Apply**, and then press **Done**. You now have two identical objects, 1 and 2, following protofilaments along the surface of the tube. Recall, however, that we want the points in Object 1 to follow the center of the tube. We'll next modify Object 1 to accomplish this.
14) Once again, set the Slicer window **Z rotation** to **-27**° to orient the tube vertically, and **X rotation** to **-90**° to obtain a cross-sectional view. Adjust slice thickness as desired using the **Img** field.

15) Select **Object 1, Contour 1, Point 1** on the 3dmod info window. As before, right-click in the center of the tube to move this point to the center of the tube. When satisfied, proceed to the next point, continuing until all the points in Object 1, Contour 1 have been centered. **Do *not* center the points in Object 2!** Those should remain at the surface of the tube where we initially placed them. Repeat this process for the remaining contours. Use the model view window to verify that you moved all the points.

16) When finished, save your model as *myStage2.mod* by going to **File / Save Model As**. If you wish, compare your results with the provided *stage2.mod* by typing `3dmodv stage2.mod`. They should be very similar. Close all 3dmod windows.

17) Next run `addModPts myStage2.mod 9 T`. The final "T" argument tells addModPts to run in a special mode in which it will process both center points in Object 1 and surface points in Object 2.

18) Run `3dmod series4-8um-cor.rec myStage2_PtsAdded.mod`, and select **Model** mode. Next we need to select a nice-looking, but representative contour / point near the middle of the tube to use as a reference. For this example, we will use point 36 of contour 3. Exit 3dmod.

19) Run `modTwist2EM myStage2_PtsAdded.mod 36 3` and notice the information printed by modTwist2EM. The "EM" in the name comes from an old format previously used instead of csv files for motive lists. The name has persisted, even though the format has changed.

    This yields both a model, *myStage2_PtsAdded_Twisted.mod*, from which overlaps between contours have been eliminated, and a corresponding initial motive list, which compensates for the supertwist named *myStage2_PtsAddedRefP129_initMOTL.csv*. These are suitable as inputs into a PEET alignment for a twisted tube.

EXPLORING THE ALIGNMENT SETTINGS AND RESULTS

20) `cd PEET/firstSearch`

21) `etomo *.epe`

22) Examine the settings chosen for the alignment search, and see if they make sense to you. Specifically, these settings are for the 15 PF helical case using the first, simpler modeling strategy from step 8 above. Axial randomization is used with a Phi search range limited to 360 / 15 = 24°.

    The model used here actually contains 2 objects and is also suitable as input to modTwist2EM. PEET ignores objects other than the first when doing alignments, however, and the few overlapping points between adjacent contours, while not desirable, do not cause any significant problems in this case.

23) One of the first things you should do during an alignment run is to verify that the reference, and if applicable, masking is appropriate by running `3dmod  *_Ref1.mrc` and checking in appropriate views. Depending on your situation, you may need to check multiple slices and in several orientations. In this case, because we're examining output files from an already completed run, you can simply press **Open references in 3dmod** at the bottom of the Etomo **Run** tab. It's also instructive to scroll through the references from the various iterations. Typically, you should see monotonically improving images, ideally with rapid convergence. References that get worse are often an indication of an incorrectly set up alignment, even if things eventually converge.

24) Next, consider how you might change the alignment settings for a 13 PF microtubule, again assuming use of the simple search strategy without tracing protofilaments. You would simply change the Initial Motive List selection from Random axial (Y) rotations to Align particle Y axes, and, optionally, choose a more restricted initial search range for φ (Phi), reflecting only the uncertainty in tracing protofilaments rather than the 24° spacing between protofilaments for a 15 PF microtubule. Close all Etomo and 3dmod windows.

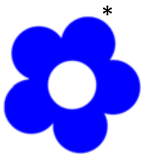
25) Similarly, to use the files generated by modTwist2EM from step 19 for the more complex strategy described above you would

   a. copy / rename the model and initial motive list generated by modTwist2EM if desired (often directly into the new project directory, although this is not required),
   b. change the Initial Motive List setting to User supplied csv files, and
   c. add the path to the initial motive list csv file in the Initial MOTL column of the volume table.

As above, a more restricted φ search would suffice.

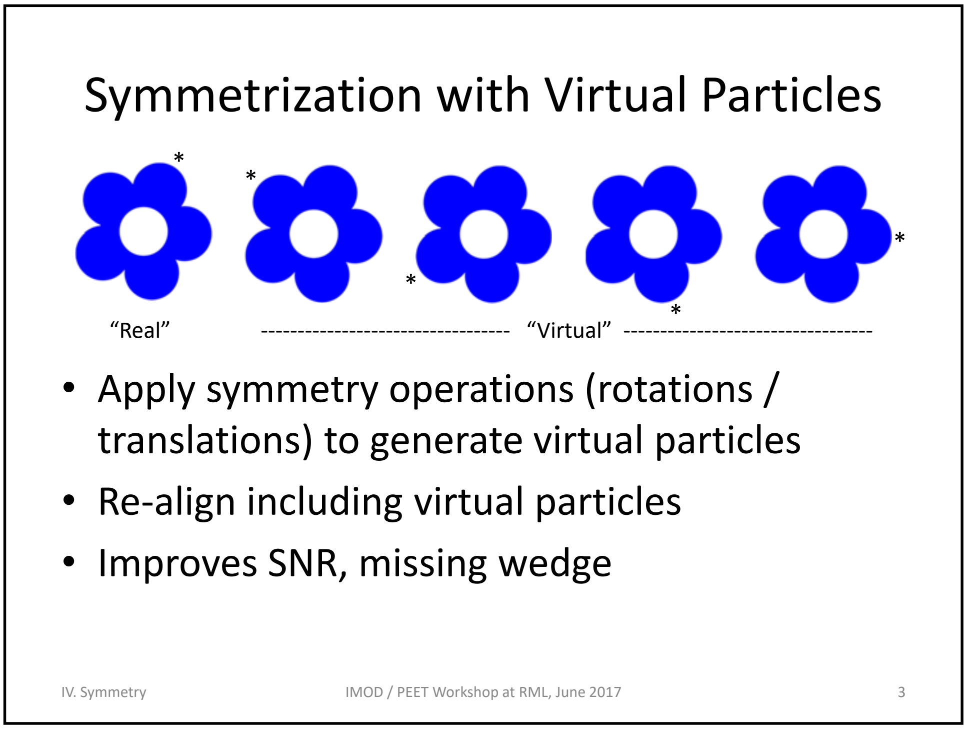
Using Symmetry

Symmetrization with Virtual Particles



"Real"

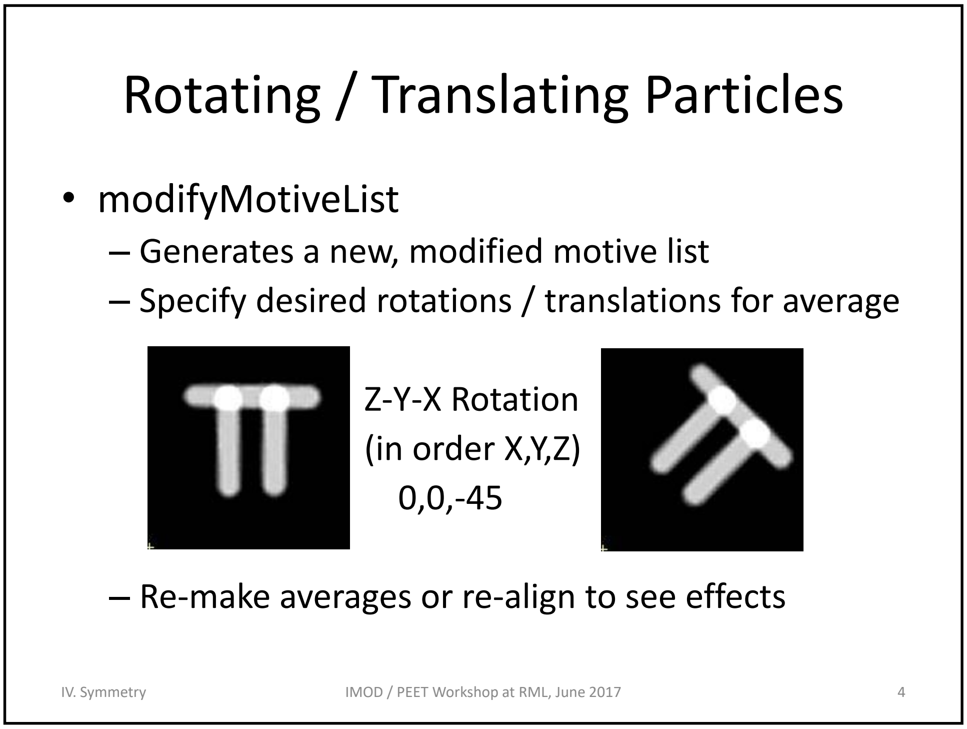
- Initial average of real particles



Symmetrization with Virtual Particles

*
*
*
*
*
"Real"
---------------------------------- "Virtual" ----------------------------------

- Apply symmetry operations (rotations / translations) to generate virtual particles
- Re-align including virtual particles
- Improves SNR, missing wedge



Rotating / Translating Particles

- modifyMotiveList
  – Generates a new, modified motive list
  – Specify desired rotations / translations for average

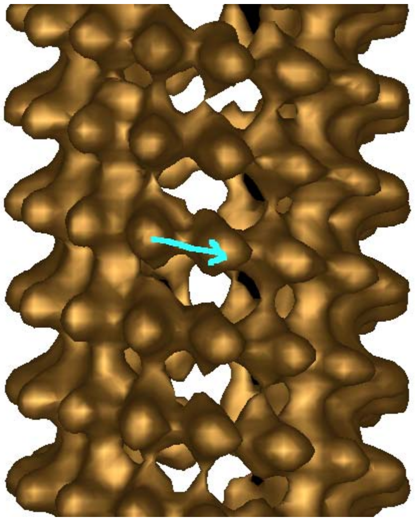
Z-Y-X Rotation
(in order X,Y,Z)
0,0,-45

  – Re-make averages or re-align to see effects

Before Symmetrization…

- Center and align the starting average!
  - Use modifyMotiveList for this too
  - Okay to use repeatedly before aligning / averaging
- Align averages vertically (used in some labs)
  - Affects final averages but not in final motive list
  - Get vertical alignment angles from *finish.log and apply with modifyMotiveList before symmetrizing

15 PF Microtubule Symmetrization



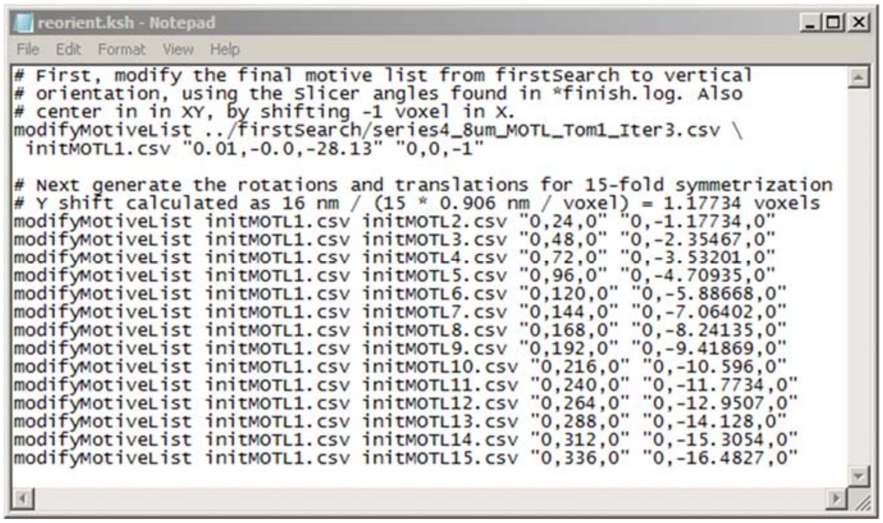
360/15=24° Y rotation

-16 /15=-1.07 nm shift

(1.07 nm ≈ 1.18 voxels)

15X effective particles

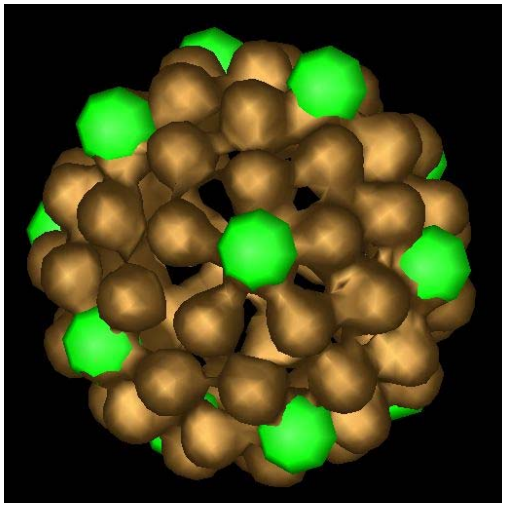
15 PF Microtubule Script



```
# First, modify the final motive list from firstSearch to vertical
# orientation, using the Slicer angles found in *finish.log. Also
# center in in XY, by shifting -1 voxel in X.
modifyMotiveList ../firstSearch/series4_8um_MOTL_Tom1_Iter3.csv \
 initMOTL1.csv "0.01,-0.0,-28.13" "0,0,-1"

# Next generate the rotations and translations for 15-fold symmetrization
# Y shift calculated as 16 nm / (15 * 0.906 nm / voxel) = 1.17734 voxels
modifyMotiveList initMOTL1.csv initMOTL2.csv "0,24,0" "0,-1.17734,0"
modifyMotiveList initMOTL1.csv initMOTL3.csv "0,48,0" "0,-2.35467,0"
modifyMotiveList initMOTL1.csv initMOTL4.csv "0,72,0" "0,-3.53201,0"
modifyMotiveList initMOTL1.csv initMOTL5.csv "0,96,0" "0,-4.70935,0"
modifyMotiveList initMOTL1.csv initMOTL6.csv "0,120,0" "0,-5.88668,0"
modifyMotiveList initMOTL1.csv initMOTL7.csv "0,144,0" "0,-7.06402,0"
modifyMotiveList initMOTL1.csv initMOTL8.csv "0,168,0" "0,-8.24135,0"
modifyMotiveList initMOTL1.csv initMOTL9.csv "0,192,0" "0,-9.41869,0"
modifyMotiveList initMOTL1.csv initMOTL10.csv "0,216,0" "0,-10.596,0"
modifyMotiveList initMOTL1.csv initMOTL11.csv "0,240,0" "0,-11.7734,0"
modifyMotiveList initMOTL1.csv initMOTL12.csv "0,264,0" "0,-12.9507,0"
modifyMotiveList initMOTL1.csv initMOTL13.csv "0,288,0" "0,-14.128,0"
modifyMotiveList initMOTL1.csv initMOTL14.csv "0,312,0" "0,-15.3054,0"
modifyMotiveList initMOTL1.csv initMOTL15.csv "0,336,0" "0,-16.4827,0"
```

BPV Symmetrization

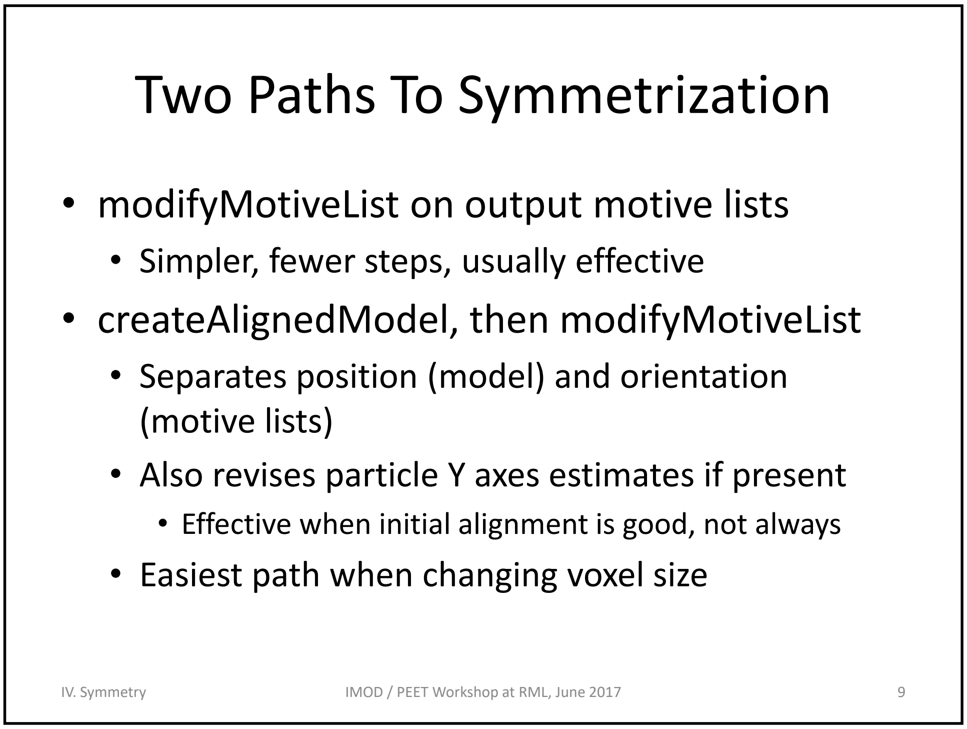


12 5-fold (c5) sites

Each to top center

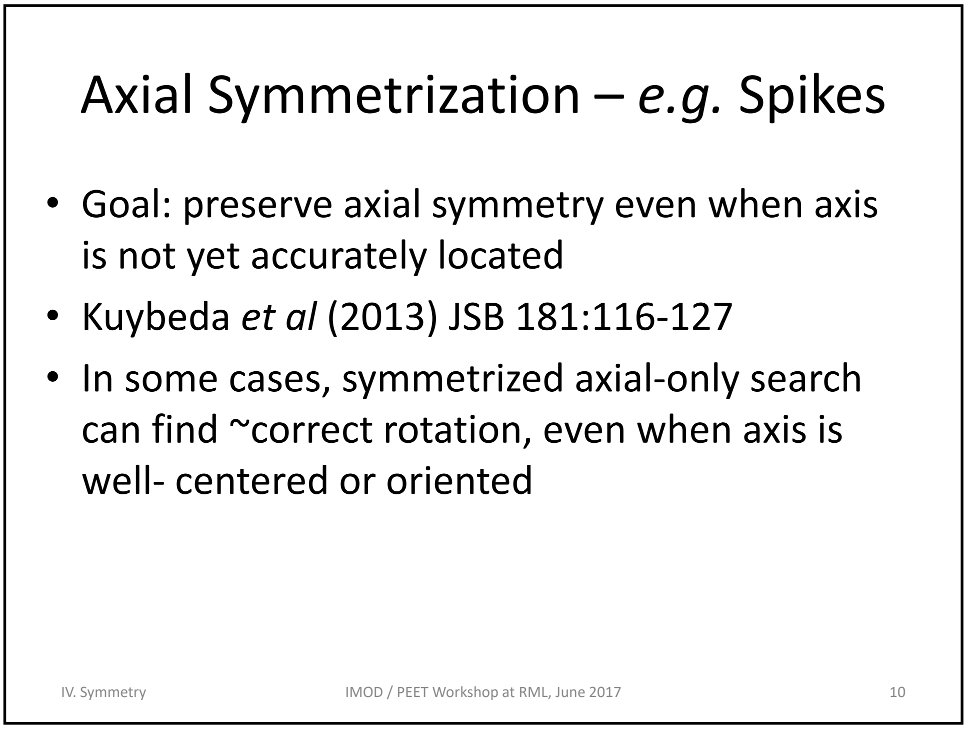
5 orientations each

60X effective particles



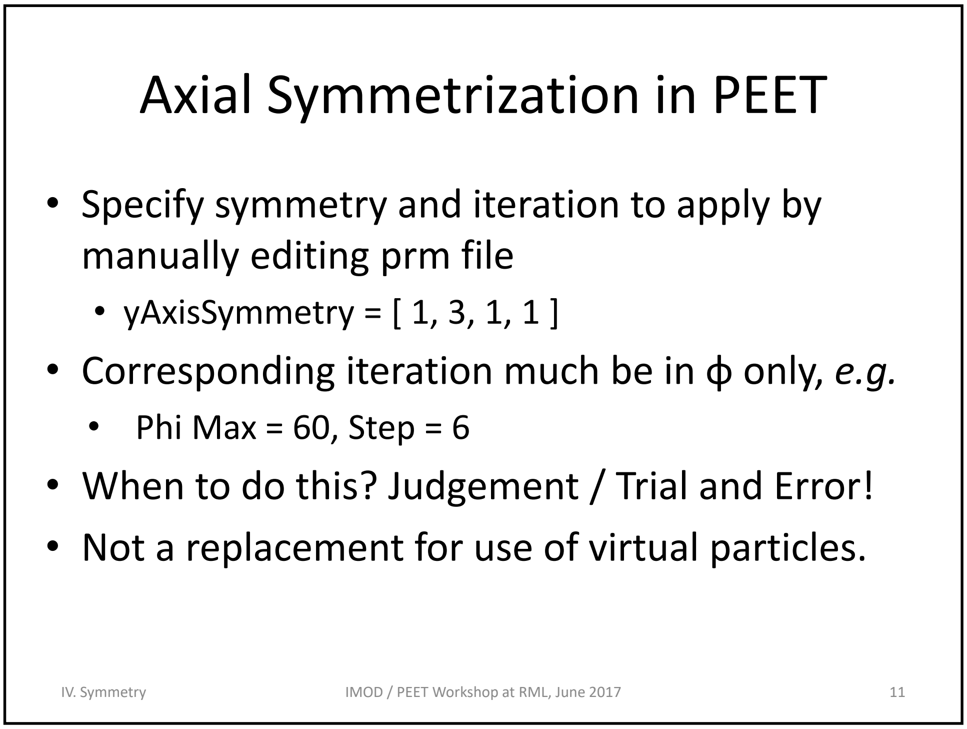
Two Paths To Symmetrization

- modifyMotiveList on output motive lists
  - Simpler, fewer steps, usually effective
- createAlignedModel, then modifyMotiveList
  - Separates position (model) and orientation (motive lists)
  - Also revises particle Y axes estimates if present
    - Effective when initial alignment is good, not always
  - Easiest path when changing voxel size



Axial Symmetrization – *e.g.* Spikes

- Goal: preserve axial symmetry even when axis is not yet accurately located
- Kuybeda *et al* (2013) JSB 181:116-127
- In some cases, symmetrized axial-only search can find ~correct rotation, even when axis is well- centered or oriented



Axial Symmetrization in PEET

- Specify symmetry and iteration to apply by manually editing prm file
  - yAxisSymmetry = [1, 3, 1, 1]
- Corresponding iteration much be in ϕ only, *e.g.*
  - Phi Max = 60, Step = 6
- When to do this? Judgement / Trial and Error!
- Not a replacement for use of virtual particles.

Questions?

THIS PAGE INTENTIONALLY LEFT BLANK

Symmetrization

First, we'll practice using modifyMotive list to symmetrize the very simple "pi" data from the first exercise.

1) `cd $WORKSHOP_HOME/PEET_Labs/Intro-1/PEET/run1`
This directory should already be populated with files from running the introductory lab. If not, you can copy the files from or pre-computed version: "`cp ../run1Done/* .`".

2) `3dmod pi_AvgVol_2P3.mrc`
Recall that this is an output average (with 3 particles from Iteration 2) from our previous alignment. Note that the "pi" is centered and oriented vertically. The vertical alignment is because we selected Align averages to have their Y axes vertical under Optional / Advanced Features during the initial alignment. The vertical alignment angles are not included in the final motive list, so we'll need to find them in **finish.log* and create a motive list containing them. As noted in the lecture, when PEET 1.12.0 is released this step will no longer be necessary, since a **_Vertical.csv* motive list will be created automatically whenever vertical alignment is enabled. Exit 3dmod.

3) `grep Slicer *finish.log`
You should see something like: **Averages rotated by Slicer (Z-Y-X) angles rotX: -0.03 rotY: -0.00 rotZ: 3.49 degrees**. (Your numbers may vary if you've run an alignment with your own model). You can also locate this information near the end of the **finish.log* without running grep, if you prefer. Let's apply these angles to the original, output motive list. Notice that by default modifyMotiveList expects Z-Y-X Slicer angles, but they must be entered in the order X, Y, Z.

4)
   ```
   modifyMotiveList pi_MOTL_Tom1_Iter3.csv \
   pi_vertical.csv “-0.03,0,3.49”
   ```
   where “\” should be followed immediately by **Enter**. We now have a new motive list, *pi_vertical.csv*, reflecting an alignment in which the individual particles (and, as a result, the average) have been oriented vertically. You can also apply additional rotations and shifts sequentially using modifyMotiveList. To illustrate, let’s choose to further rotate our average by 45° counterclockwise about the tomogram Z axis, and to shift it to the right by 5 voxels.

5)
   ```
   modifyMotiveList pi_vertical.csv pi_shifted.csv \
   “0,0,-45” “5,0,0”
   ```

   Next we’ll use this new motive list, *pi_shifted.csv*, to create a corresponding new average. It’s prudent to backup parameter files, motive lists, averages, etc. before modifying them. Create a backup directory by executing `mkdir  backup` followed by `cp  *  backup`. Depending on the size of your directory, you may want to only copy specific files to the *backup* directory. In this tutorial there is already a backup copy of everything under *../run1Done*.

6)
   ```
   cp pi_shifted.csv pi_MOTL_Tom1_Iter3.csv
   ```
   We are copying our modified motive list over the original generated during alignment, so it will be the one seen by Etomo and PEET.

7)
   ```
   etomo *.epe
   ```

8) Uncheck **Align averages to have their Y axes vertical** (on the **Run** tab under **Optional / Advanced Features**). This will prevent the new average from being automatically rotated back to vertical. Press **Remake averages**. Wait a few moments for the averages to complete. (Check the progress bar at the top of the Run tab).

9) Press **Open averages in 3dmod** and verify that all 3 averages have shifted and rotated as we’ve requested.

10) Exit Etomo and the 3dmod windows when finished. There are command-line versions of all these operations. "Remake averages" corresponds to "`averageAll pi.prm 2 averages`", for example. The keyword "averages" can be replaced with "reference" or "both", depending on the desired output. "Open averages in 3dmod" corresponds to the command "`3dmod -Z -V -E U pi_AvgVol*.mrc`".

SYMMETRIZING THE 15 PROTOFILAMENT MICROTUBULE

For the 15 PF microtubule, vertical alignment was used, so we'll again need to extract those angles from the **finish.log*. Additionally, we'll make sure the tube is well centered, and then we'll repeatedly apply 24° rotations around the Y axis combined with shifts of 16 nm / (15 * 0.906 nm / voxel) = 1.17734 voxels.

11) `cd ../../../MT/PEET/firstSearch/`

12) `grep Slicer *finish.log`
Note the resulting angles. We'll apply them with modifyMotiveList in step 22 using a shell script. Next, we'll check the XZ centering of the current average. (The tube is extended and periodic in Y, so we can ignore that dimension in this case).

13) `3dmod *AvgVol*.mrc`

14) Scroll to the final average with 243 particles, open a Slicer window (\) and rotate **-90** about **X**. As before, the Y axis now points into the screen and you're looking down on an XZ view.

15) Select **Model** mode and place a point (middle-click) near the center of the tube.

16) Press **Edit / Object / Type** and set **Sphere radius for points** to **12**.

17) In the Slicer window, set **Img** to **20** and **Mod** to **5**.

18) Right-click (repeatedly if necessary) to move the model point until you're happy with the centering of the green and yellow circles in the microtubule.

19) Note that the yellow circle (the current point) is now almost directly above of (*i.e.* +Z) the red cross (the center of the image where X,Y,Z in the 3dmod info window is 28, 28, 28).

20) To measure the actual displacement, zoom in the Slicer window to a **Zoom** of **16** so the yellow circle and the red cross are clearly separated, center the cursor over the red cross with the mouse, and then press **q**. Pixel coordinates (X, Y, and Z) of both the current point and the cursor as well as the total distance between them will be reported in the 3dmod info window. Take the difference between the 2 X coordinates to calculate the X shift required, and between the 2 Z coordinates to calculate the Z shift needed to properly center the particle. (Signs are best handled by remembering that a +X shift will move the average to the right, while a +Z shift will move it up toward the top of the page). In this case you'll see that we need to shift down by approximately 1 voxel in Z… *i.e.* a Z shift of -1.

21) Exit 3dmod without saving the changed model, `cd ../full15Fold`, and run `gedit reorient.sh`. This is the script that we'll use to generate virtual particles for symmetrization. Examine its contents and notice the use of the shift from step 20 and rotations determined in step 12 above. Exit gedit when finished.

22) (Optional) Execute this script by running `./reorient.sh`. The script will take a few minutes to run. Open a new shell window from your terminal and proceed with the following steps in the new window without waiting for it to complete.

23) `cd $WORKSHOP_HOME/PEET_Labs/MT/PEET/full15Fold`

24) `etomo *.epe`

Let's examine the settings used for the symmetrization alignment. On the Setup tab, notice that we've selected File names are templates and specified the initial motive list as *initMOTL01-15.csv*. This is convenient shorthand equivalent to entering 15 separate rows in the Volume Table, with motive lists *initMOTL01.csv*, *initMOTL02.csv*, *etc*. Templates may also be used for the other filenames in the Volume Table. In this example that the entries in each "row" all refer to our single, original volume.

We've selected particle 120 of volume 1 (the same as in the firstSearch alignment) as our initial Reference. Alternatively, we could have chosen the output reference file from the previous run as our starting reference; in this case, we would also have copied / renamed the corresponding binary wedge mask so PEET would be able to use it for missing wedge compensation at iteration 1. Missing wedge compensation doesn't really matter in this case, because the first iteration will be no-search, and we've told PEET to use all the available particles (15 * 243 = 3645) for the new reference created after this iteration. *I.e.* the first iteration simply creates a new reference based on our initial, approximate symmetrization of the previous alignment.

On the Run tab, we've selected Use Equal Numbers of Particles From All Tomograms both for average volumes and new references, since each of the tomograms are equivalent (and in fact identical!). After the no-search first iteration, we do 2 quick refinement iterations to improve the alignment. After each of these, we use approximately 2/3 of the available particles to generate references. This is typical, and allows for the possibility that some particles may be damaged or have been poorly aligned. Symmetrization takes considerable time, so we won't run it during the workshop; you're of course welcome to do so on your own. Instead, we've provided all the important output files for you.

25) Press **Open averages in 3dmod** and **Open references in 3dmod** and examine the results. Note that

   a. The initial reference (*full15Fold_Ref2.mrc*) looks good, indicating that we've applied the symmetry operations correctly.

   b. The mask is appropriately sized and positioned.

   c. The references improve monotonically at successive iterations. Remember that you can use the ZaP window's **4th D** left and right arrows to scroll through iterations. Reference 1 is simply a single particle, and is not actually used here, since iteration 1 is no-search.

   d. Averages with increasing numbers of particles look similar, with improving SNRs, indicating that damaged or misaligned particles are not a serious problem. Particles with higher cross-correlation coefficient are typically incorporated first, depending whether you've selected "Use equal numbers…" or "Randomized particle selection". When substantial numbers of damaged or misaligned particles are present, the best average may not be the one with the most particles.

26) Examine the other output files generated during the run. PEETCleanup has already been run, so many intermediate files have been deleted, and only output files deemed important remain. Exit all Etomo and 3dmod windows when finished.

SYMMETRIZING BINNED BPV

Symmetrization of BPV is similar in many respects. Vertical alignment was not used during the initial BPV alignment, so it’s simpler in that regard. Instead of applying helical / screw symmetry, we will need to bring each of the twelve 5-fold vertices to top center in a standard orientation and then apply 72° rotations around the Z axis.

27)
```
cd ../../../BPV_-3/PEET/firstSearch
```

28)
```
3dmod –S bpv_AvgVol_4P132.mrc myPentamers.mod
```

Next, we will check if the average is adequately centered in all 3 dimensions, or if it needs to be translated before symmetrization. As before, select **Model** mode, click on **Edit / Object / Type**, set **Object Type** to **Scattered** and increase **Sphere radius for points** to **14** and **line width** to **3**. In the Slicer window, adjust **Img** to **10** and **Mod** to **2**. As in the previous exercise, place a temporary model point to check centering; now however, we will check for possible shifts in all 3 dimensions rather than just X and Y. X and Y shifts can be checked in the unrotated (XY) slicer view, while X and Z shifts can be checked after rotating -90° around X. You should find that the average needs to be shifted approximately -1 voxel in Y. (Don’t worry about the exact results; anything within about 1 voxel is probably okay).

29) Restore the Slicer view to its original orientation (set all angles to **0**) and position (X=Y=Z=**27**) in the 3dmod info window.

30) Delete any temporary model points you may have added while checking centering.

31) Open an Isosurface view (**Shift+U** or **Image / Isosurface**). In the Model View window, select **Edit / Controls** and press **Link to top Slicer Angles**. The Isosurface and Slicer images will now rotate together. Zoom in the Model View with the mouse wheel until the virus nearly fills the image.

    In the following, we will be rotating the average in the Slicer and Isosurface views. There are several ways to do this:

    a) If your keyboard has a numeric keypad, you can rotate the particle around X using the 2 and 8 keys, around Y using 4 and 6 keys, and around Z using the 3 and 9 keys. For X and Y, it is helpful to think of the keys as pointing toward the compass direction in which you would like to rotate.
    b) Under the Model View window, you can choose Edit / Controls and then either click (and, optionally hold) on the X, Y, Z rotation arrows or type the desired angles directly in the corresponding text fields.
    c) In the Slicer window, you can drag the X, Y, and Z rotation sliders, or click on the red rotation arrows near the upper right of the window, or you can Shift+middle-click and drag.
    d) In Model View, you can middle-click and drag.

    In each case, you can monitor the orientation in the Model and Slicer Views until the desired orientation is obtained. In the following, you are free to use whichever of these methods you prefer, subject, of course, to the constraint that most laptops lack a numeric keypad, so method “a” may not be available. Note that when you wish to change focus to a particular window (*e.g.* the Slicer or Model View) without changing anything else, you can do so by left-clicking in that window’s title bar, rather than in the window’s display area.

32) Notice that a pentamer / 5-fold axis happens to be located just to the right and above the center of the starting view. Rotate the average using one of the above methods until this pentamer is centered in the Model View and facing you.

33) Turn the Isosurface **Threshold** down to **80**. At this setting, the capsomeres should look like isolated particles, and you'll more easily be able to see their relation to one another. Refine the previous rotation so that the centered, front pentamer is directly over the one behind it, and so that one of the vertices / capsomeres is pointing directly to 12 o'clock. We'll define this as our "standard" orientation, and will seek rotations which position each pentamer in such an orientation.

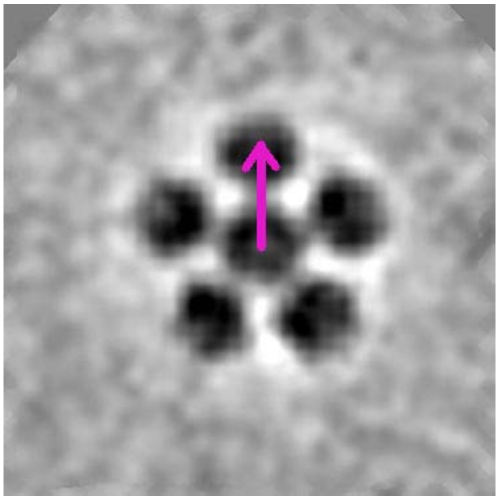
34) You're now looking directly down a 5-fold axis, in standard orientation, and have located 2 pentamers. Finding pentamers is typically easy, but enumerating all of them can get confusing unless you carefully mark the position and orientation of previously located ones as you go. We'll illustrate one method for keeping track of this information.

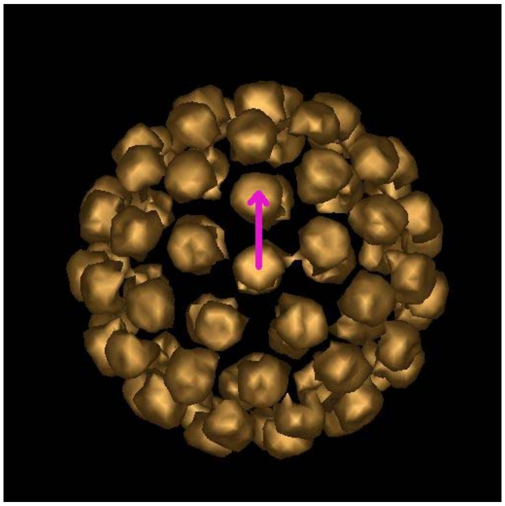
35) In the main 3dmod info window, press **Edit / Angles**. Arrange your screen so the Slicer, Model View, Slicer Angles, and 3dmodv Control windows are all visible.

36) A pentamer should already be at the front of both the Slicer and Isosurface views, but the Slicer window is probably not at the correct slice height, so the pentamer may not be visible in it yet. Select the Slicer window by left clicking on its title bar (***not*** in the image itself!), and the press **PgUp / PgDn** or adjust the **View axis** position slider until the Slicer images is centered on the central capsomere of the front pentamer, and the 5 surrounding capsomeres are visible, but less dense. Be sure that you always use the front / upper-most pentamer! Adjusting slice height manually in this fashion will always suffice to locate the front / uppermost pentamer in the Slicer view. However, another 3dmod feature can be used to speed up this process, as we'll illustrate next.

37) With the Slicer window selected, press **PgDn** 10-20 times so the pentamer is no longer visible, simulating the state we were in at the start of the previous step. The pentamer will no longer be visible in Slicer. In the Slicer window, enable **centering** by clicking on the third control icon. Verify that this icon changes from a box within a box to a dot in a box, indicating that centering mode is enabled. ***The following will only work correctly in centering mode!!!*** In the Model View, right click on the capsomere at the center of the front pentamer. This will instantly change the all 3 Slicer coordinates, including slice height, to match the point you clicked on in the Isosurface; the capsomere at the center of the pentamer should immediately become visible in the center of the Slicer window. Select the Slicer window by left clicking in its title bar, and refine the slice height as before until you are in the middle of the central capsomere with the 5 neighboring capsomeres are visible but less dense.

38) Regardless of which of the 2 previous methods you use, you should now adjust the orientation of the front pentamer, referring to both the Slicer and Model View windows. Rotate the pentamer until a line from the central capsomere to one of the surrounding vertices points toward 12 o'clock. Ensure that a vertex is oriented toward 12 o'clock and that the 5-fold axis is centered (as illustrated below) before proceeding. Next, we will add a model point to mark this pentamer and will save the angles used to bring it to this standard orientation.





In the Slicer window, add a model point by middle-clicking on the center of the central capsomere. Using 3dmod's **Edit / Object / Type** dialog, set **Sphere radius for points** to **3**, and verify in the Model View that the model point is located where you'd like. If not, you can move it in X, Y, or Z by right-clicking in the Slicer window.

39) Also in the Slicer window, press **New** to add a new row to the **3dmod Slicer Angles** table with the rotation required to bring this pentamer to the standard orientation.

40) Recall that there's another pentamer directly behind / underneath this one. Select the Model View, and rotate **180**° around X until this new pentamer is in front, directly over the green sphere marking the previous one.

41) Then rotate around Z until a vertex is pointed at 12 o'clock.

42) As before, locate the center of the new pentamer in the Slicer view either by right clicking in the Model View or by changing slice height. Refine the slice height and orientation as previously, place a new model point by middle-clicking in the Slicer window, and press **New** to store the corresponding angles. ***It's important that model points and stored angles remain in 1-to-1 correspondence!***

By rotating the isosurface, search for another pentamer and its partner on the opposite side of the virus. Add the corresponding 2 model points and sets of angles, as above. To complete the model, you would simply repeat this procedure until all 12 pentamers had been located. There's little additional to be learned by doing so, however, so we've created such a model for you.

43) Save your partial model (**File / Save)**, and open our predefined *pentamers.mod* using **File / Open Model.** The resulting table of angles contains the remaining information needed to symmetrize our starting alignment. You can either make note of these manually, or you can find them by looking in the listing produced by running `imodinfo –a pentamers.mod`.

44) (Optional) If you wish, explore the geometry of the 5 fold axes in an icosahedral virus by setting the Isosurface **Threshold** so only the green spheres are visible. Can you detect a regular pattern?

45) Close any 3dmod windows when finished.

46) Quite a few steps are involved in actually applying all the symmetry operations, so we've placed the necessary commands in a script, *../iscosahedral/reorient.sh*, for you. Run `cd ../icosahedral` followed by `gedit reorient.sh` and see if you understand the various steps. If not ask your instructor for help. Exit gedit without changing anything when finished.

The *reorient.sh* script itself is rather lengthy. The bash shell has built-in looping and integer arithmetic capabilities that allow for more concise scripting, if you're willing to take the time to learn the somewhat arcane syntax. I've provided an equivalent example in *reorientWithLoop.sh*. Examine this script with `gedit  reorientWithLoop.sh` now, exiting when finished. Good tutorials on bash shell scripting are available on the web. If you do choose to learn Linux / Unix shell scripting, bash is the Linux shell I'd recommend.

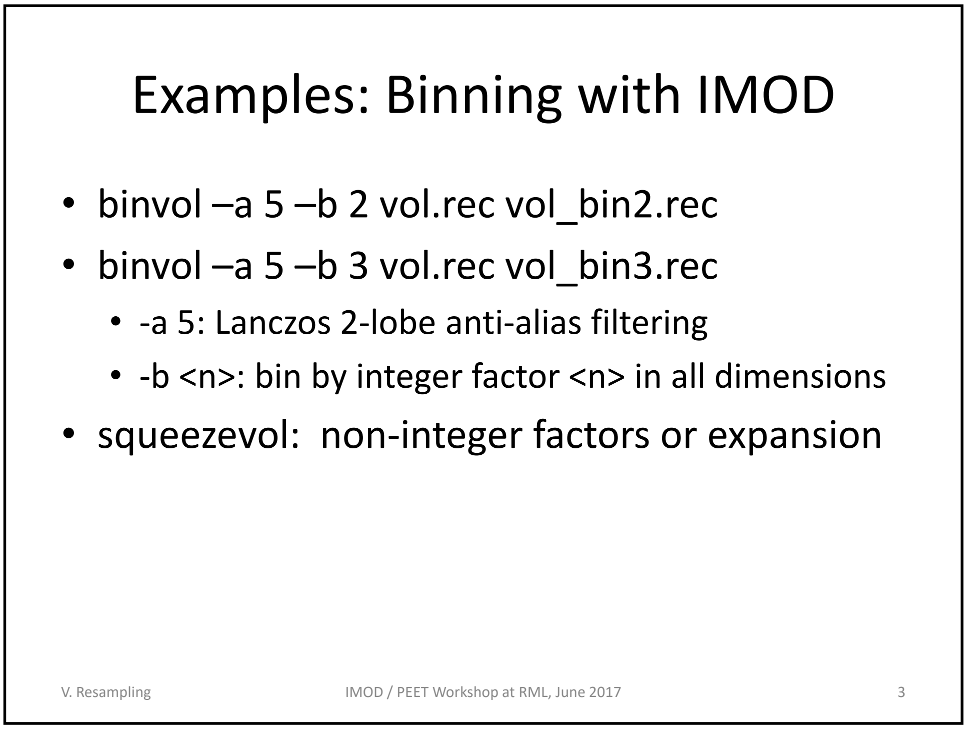
47) As before, we've already done the actual alignment run and removed most intermediate files. `etomo  *.epe` to examine the parameter file settings, output averages, and references. You may also wish to examine the **.prm* file or other text files of interest with gedit, or to open other volumes (**.mrc*) with 3dmod.

	June 29 (Thurs)
8:00	Security
	Questions
	Resampling
9:00	John
	Resampling Lab
10:00	Modeling Aids
	John
	Coffee Break
11:00	Modeling Aids Lab
	Semi-Automated Picking
12:00	John
	Lunch
1:00	Optional Tour of Rocky Mountain Labs: Ken Pekoc
	Semi-Automated Picking
2:00	
	Debugging
	John
3:00	Debugging
	Coffee Break
4:00	
	Verification
	John
	Fourier Shell Correlation
5:00	John
6:00	
	BBQ at Hiker's Cabin
	Lake Como
7:00	
8:00	
9:00	

Resampling: Changing Voxel Size

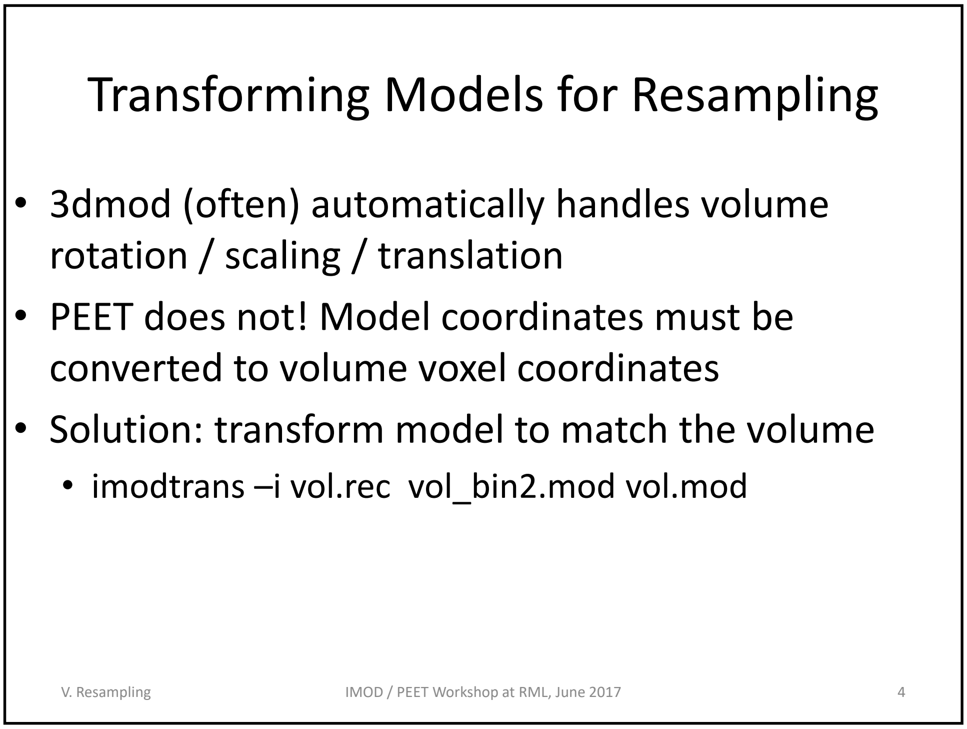
Why Resample?

- Initial modeling may be easier / more accurate after binning / down-sampling and perhaps NAD (or other) filtering
- Initial alignment / averaging is much faster with down-sampled data (time ~ voxels^3).
- Final alignment typically unbinned for best resolution



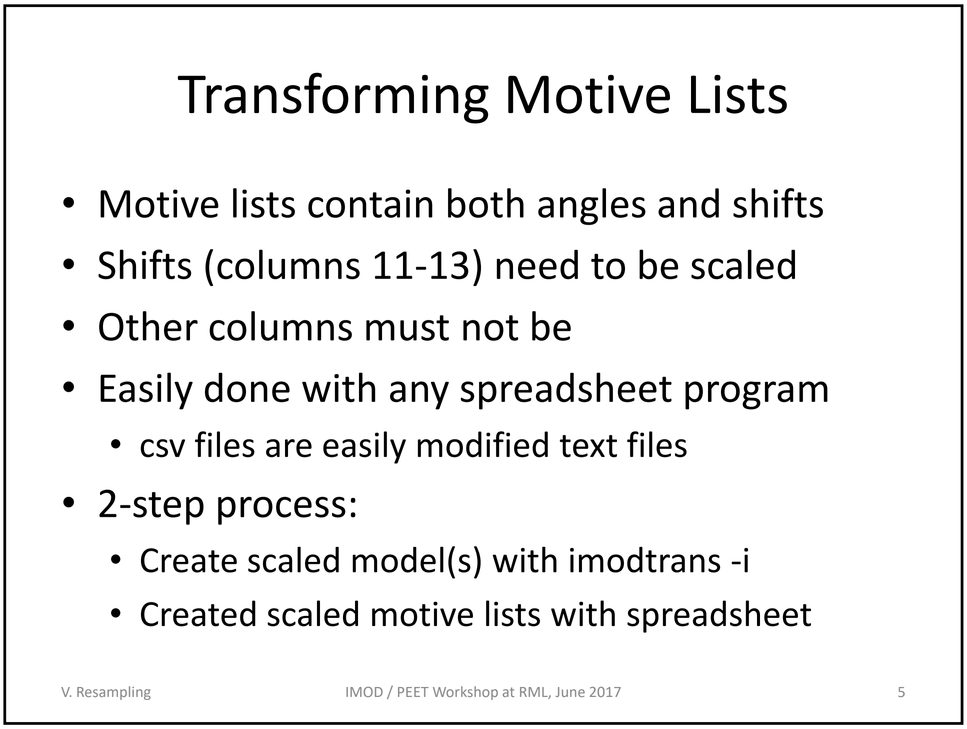
Examples: Binning with IMOD

- binvol –a 5 –b 2 vol.rec vol_bin2.rec
- binvol –a 5 –b 3 vol.rec vol_bin3.rec
  - -a 5: Lanczos 2-lobe anti-alias filtering
  - -b <n>: bin by integer factor <n> in all dimensions
- squeezevol: non-integer factors or expansion



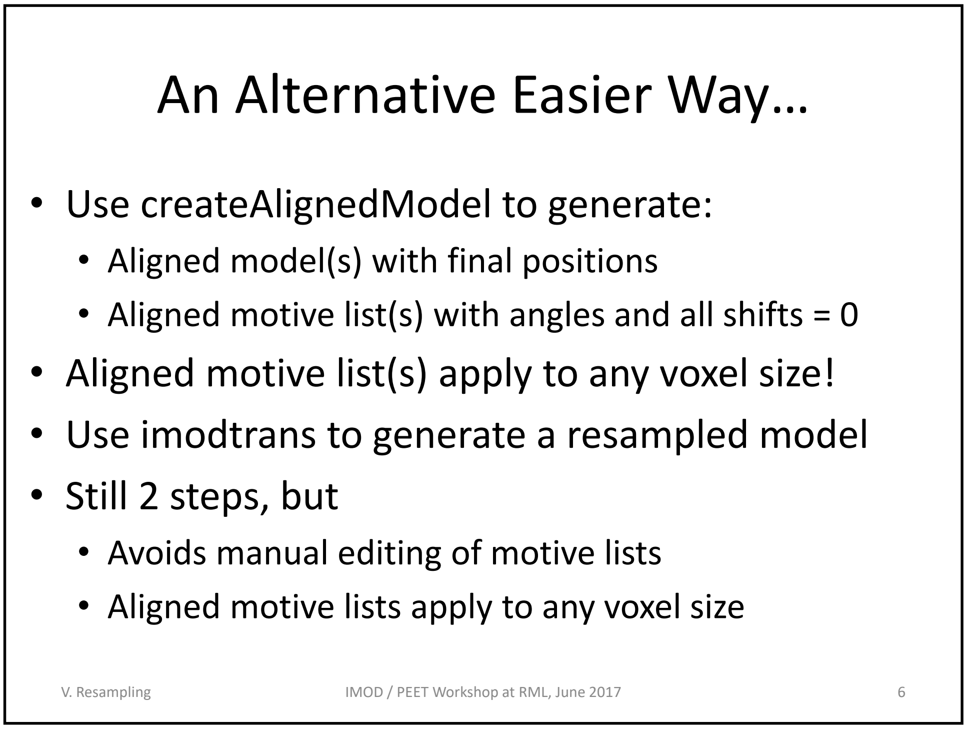
Transforming Models for Resampling

- 3dmod (often) automatically handles volume rotation / scaling / translation
- PEET does not! Model coordinates must be converted to volume voxel coordinates
- Solution: transform model to match the volume
  - imodtrans –i vol.rec vol_bin2.mod vol.mod



Transforming Motive Lists

- Motive lists contain both angles and shifts
- Shifts (columns 11-13) need to be scaled
- Other columns must not be
- Easily done with any spreadsheet program
  - csv files are easily modified text files
- 2-step process:
  - Create scaled model(s) with imodtrans -i
  - Created scaled motive lists with spreadsheet



An Alternative Easier Way…

- Use createAlignedModel to generate:
  - Aligned model(s) with final positions
  - Aligned motive list(s) with angles and all shifts = 0
- Aligned motive list(s) apply to any voxel size!
- Use imodtrans to generate a resampled model
- Still 2 steps, but
  - Avoids manual editing of motive lists
  - Aligned motive lists apply to any voxel size

createAlignedModel and rotAxes Files

- So far, particle Y axes have been implicit
- In some cases (*e.g.* using meshInit, stalkInit, spikeInit or user-specified Y axes), explicit rotAxes files will be present
- createAlignedModel also generates new rotAxes file(s) with revised estimate of axes
  - New estimates typically better if alignment is good
  - Use with caution if initial alignment is poor!

Questions?

THIS PAGE INTENTIONALLY LEFT BLANK

Resampling

PEET alignments are often performed in a multi-resolution fashion, starting with binned data, for example, and then proceeding to unbinned data for higher resolution. Let's explore how to use imodtrans to transform a PEET model created for use with one tomogram to apply to another tomogram with a different voxel size. PEET's requirements are more stringent than 3dmod in this regard. Just because a model displays correctly in 3dmod on a given volume does not guarantee that it will work correctly with PEET. In the BPV alignment exercise, we used a model created on a 2X binned, NAD-filtered volume. Let's generate a new version of this model, which will work with the original, unbinned tomogram in PEET.

1)
```
cd $WORKSHOP_HOME/PEET_Labs/BPV_-3
```

2)
```
3dmod bpv_bin2.rec bpv_bin2.rec.nad bpv_bin2.mod
```

3) Press **Auto** and check **Float** and **Subarea** on the 3dmod info window to automatically adjust the brightness and contrast as the images change. Toggle between the raw tomogram and the filtered (nad) tomogram using the **4th D** arrows on the ZaP window. While the model lines up properly on both volumes, the virus particles are difficult to see against the background noise in the unfiltered volume. That's why many people use a heavily NAD-filtered volume for initial modeling. Close 3dmod.

4) `imodinfo bpv_bin2.mod`

Notice the presence of the **Model to Image index cords**: section and, specifically, that SCALE is 15.2 for each of X, Y, and Z. Run `header bpv_bin2.rec` and you will see that this matches the voxel size in angstroms of the binned volume. Because this model was created using 3dmod on an identically binned volume, the appropriate information has been stored automatically in the model header. Even without this information, (*e.g.* if a model was created using point2model), PEET would function correctly as long as the model coordinates are entered in voxels.

5) `model2point -float bpv_bin2.mod bpv_bin2.txt`

This saves a text version of the particle locations for later comparison. The float option cause coordinates to be save as floating point, rather than rounding to the nearest integer.

6) `imodtrans -i bpv.rec bpv_bin2.mod bpv.mod`

This creates a new model, suitable for use with the original, unbinned volume. Note that if *bpv_bin2.mod* lacked the Model to Image Index cords information, we would first need to create a model containing that information, *e.g.* by running:

```
imodtrans -i bpv_bin2.rec bpv_bin2.mod temp.mod
mv temp.mod bpv_bin2.mod.mod
imodtrans -i bpv.rec bpv_bin2.mod bpv.mod
```

7) Run `imodinfo  bpv.mod` and note that the new SCALE reflects the correct 7.6 angstrom voxel size for the unbinned volume.

8) `model2point -float bpv.mod bpv.txt`

9) Run `head bpv.txt bpv_bin2.txt` and compare the coordinates of the first few particles. Notice that the coordinates for the original, unbinned volume are twice those of the 2X binned volumes, as you'd expect. This illustrates how to transform models from one scaling to another while always satisfying a key PEET requirement: Particle coordinates must reflect positions within the volume in voxels, with the origin at the lower left corner.

BPV ALIGNMENT WITH UNBINNED DATA

We've already examined a symmetrized alignment of BPV using 2X binned data. Now let's apply those results to generate a final, symmetrized average using the original, unbinned data. Because there will now be 60 entries in the volume table, this case is considerably more complex than the previous one, although the ideas are exactly the same. To avoid having to manually modify the shifts in 60 motive lists, we'll first use createAlignedModel to generate aligned models and motive lists. Then we'll transform the aligned models for use with the original, unbinned tomogram. Finally, we'll use the transformed models and the aligned motive lists to perform the final, unbinned alignment. Because a number of commands are required, I've created a script named *prepareForUnbinnedAlignment.sh* in the symmetrized, binned alignment directory to handle the necessary steps. Let's take a look at it.

10) `cd PEET/icosahedral`

11) `gedit prepareForUnbinnedAlignment.sh`

Read over this shell script and its comments. In addition to things we have discussed before, one new feature is illustrated. After generating the aligned motive lists, we'd like to make them visible in the directory we'll use for the final alignment under different names—ones that we can easily expand using templates. While this could also be done by renaming the files while copying or moving them, in this case, we've chosen to just make a "link" back to the original files. The net effect will be the same either way. NOTE: `ln <src> <dst>` creates a "hard link" and will work on any of Linux, OS X, or Windows with Cygwin. Symbolic links, created by `ln -s <src> <dst>,` are often preferred on Linux or OS X, but will not work reliably on Windows. Exit gedit.

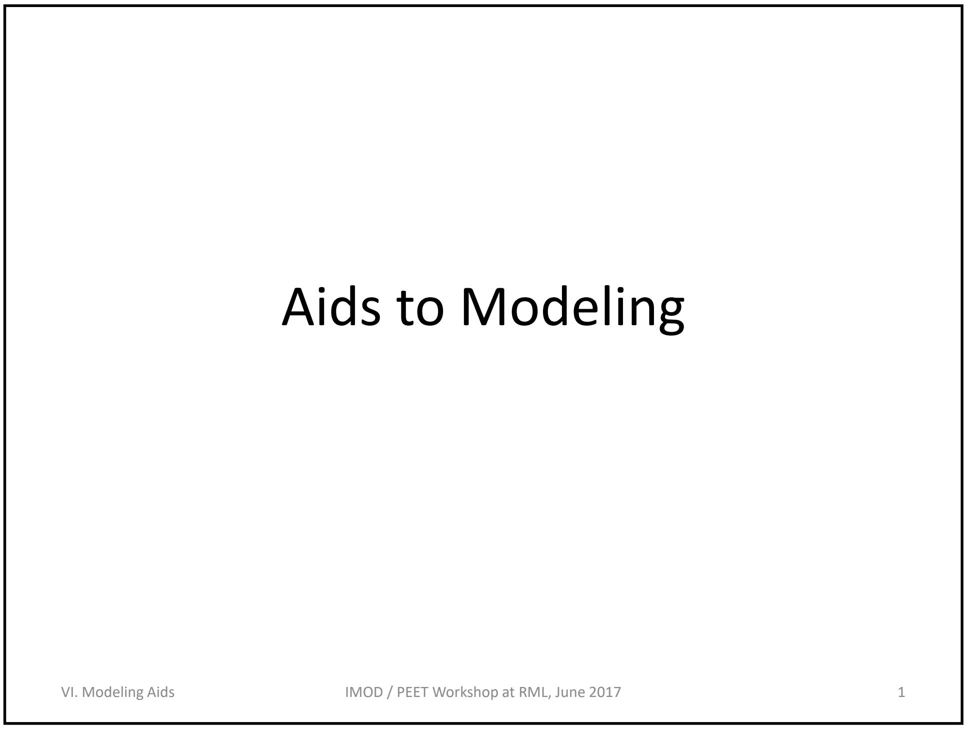
12)
```
./prepareForUnbinnedAlignment.sh 2>&1 \
tee prepareForUnbinnedAlignment.log
```

where "\" should be followed immediately by **Enter**. This will execute the commands in the *prepareForUnbinnedAlignment.sh* script. It will run rather quickly, completing in less than a minute.

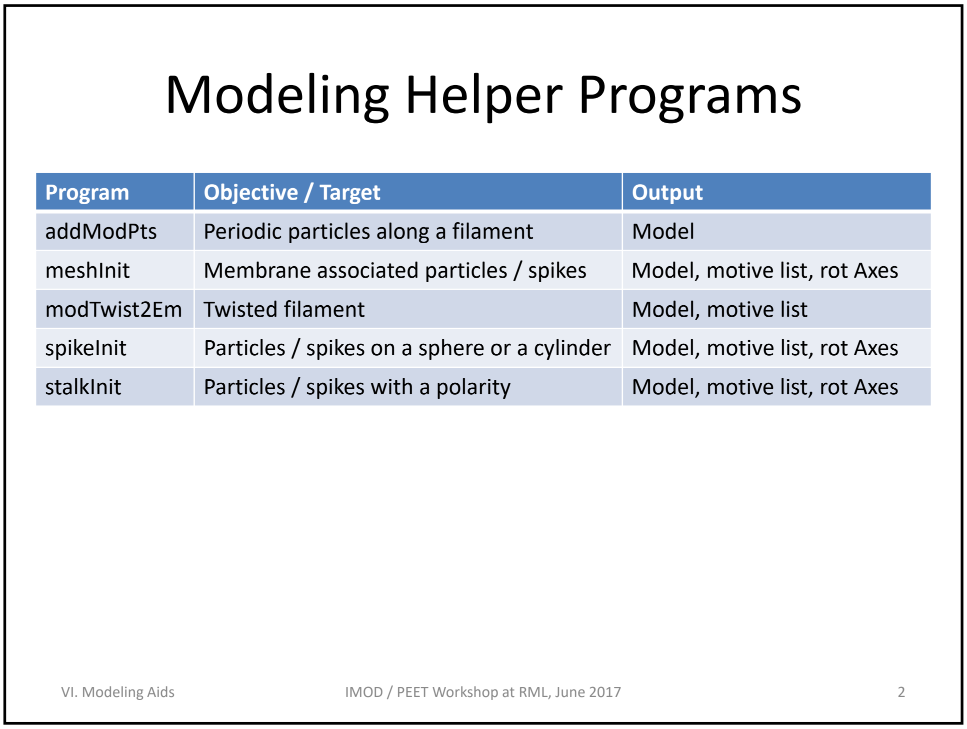
13) `cd ../unbinned`

14) `etomo *.epe`

As in previous exercises, examine the parameter settings, making sure you understand them. Then press **Open averages in 3dmod** to explore the resulting averages which we've precomputed for you. As a point of reference, this unbinned alignment required about 20 hours using 20-30 cores with 2.4 – 3.3 GHz clock rates. Close Etomo and any 3dmod windows.

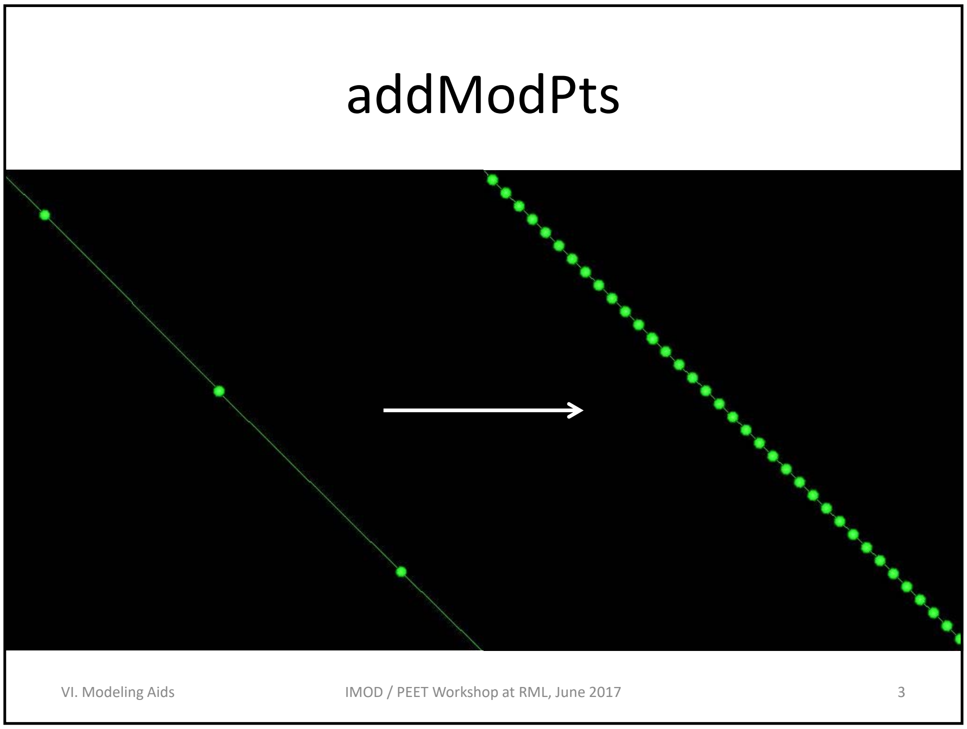


Aids to Modeling

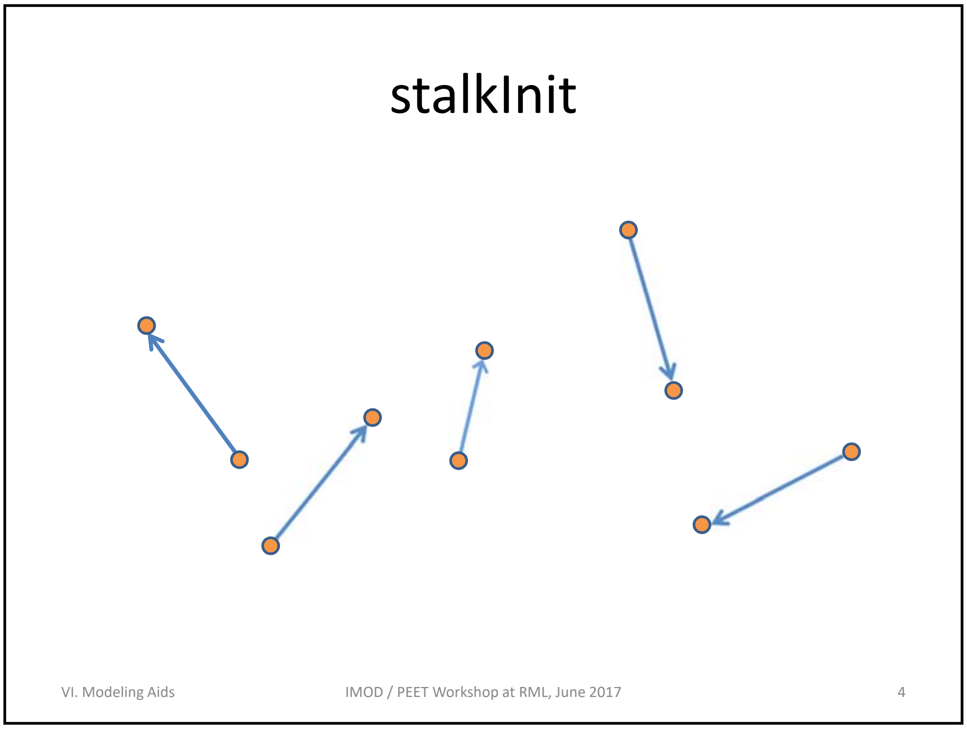


Modeling Helper Programs

Program	Objective / Target	Output
addModPts	Periodic particles along a filament	Model
meshInit	Membrane associated particles / spikes	Model, motive list, rot Axes
modTwist2Em	Twisted filament	Model, motive list
spikeInit	Particles / spikes on a sphere or a cylinder	Model, motive list, rot Axes
stalkInit	Particles / spikes with a polarity	Model, motive list, rot Axes

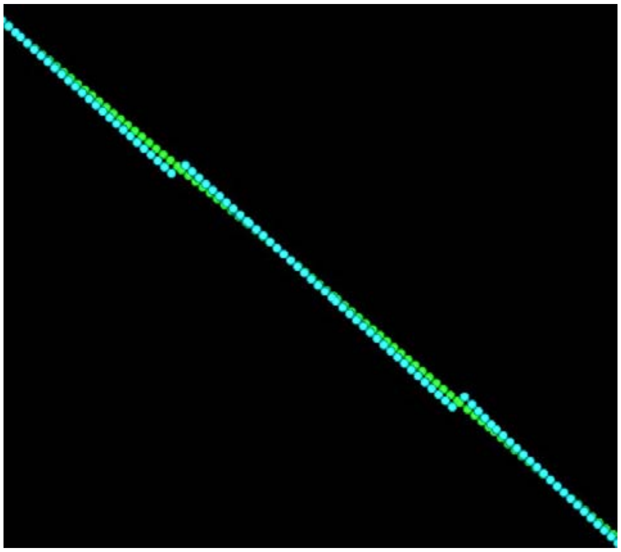


addModPts

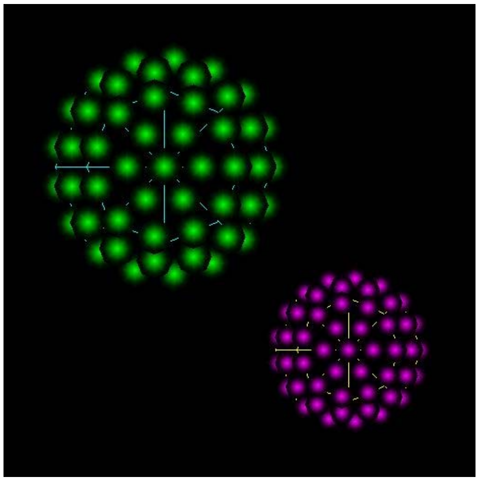


stalkInit

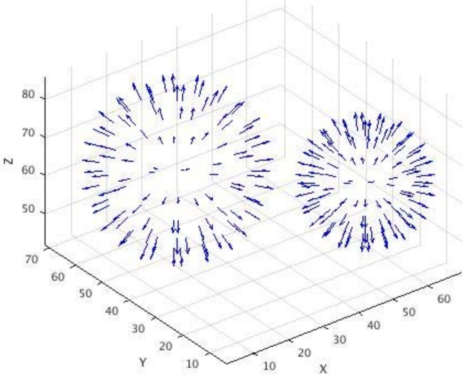
modTwist2EM



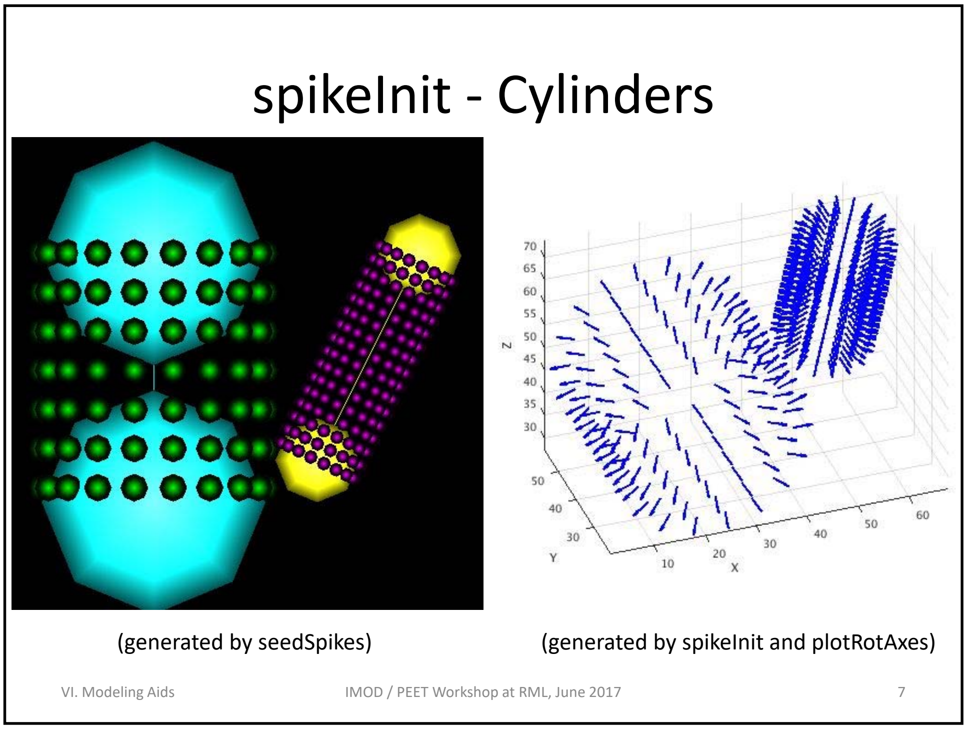
spikeInit - Spheres



(generated by seedSpikes)



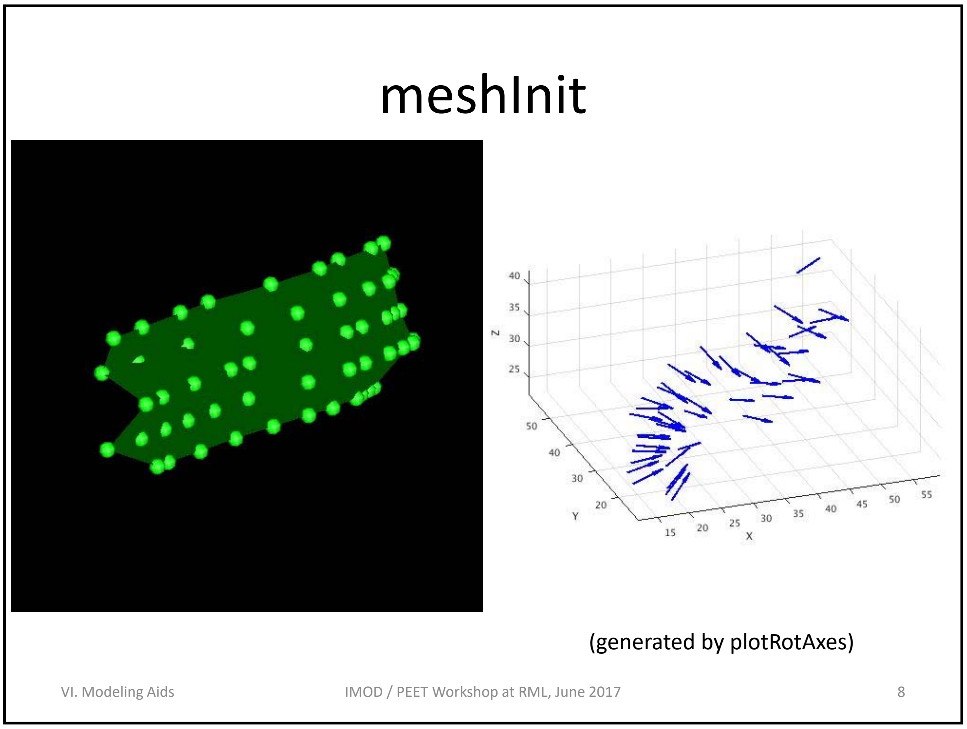
(generated by spikeInit and plotRotAxes)



spikeInit - Cylinders

(generated by seedSpikes)

(generated by spikeInit and plotRotAxes)



meshInit

(generated by plotRotAxes)

Questions?

THIS PAGE INTENTIONALLY LEFT BLANK

Modeling Aids

PEET provide several programs designed to ease the process of creating models for common cases: regularly spaced particles, spikes protruding from or embedded in membranes, spheres, or cylinders, and so on. In this exercise, we'll explore some of these programs.

SPHERES AND CYLINDERS

First, we'll consider seedSpikes to generate regularly spaced points with axes pointing radially inward or outward from the surface of one or more spheres or cylinders. Then we'll convert these models to a form suitable for use with PEET using spikeInit.

1) `cd $WORKSHOP_HOME/PEET_Labs/ModelingAids`

2) `3dmod spheres.mod`, switch to **Model** mode, and select **Object 1**. You'll see that this model contains 2 single-point objects, where Object 1 has a point size of 18 (**Edit / Point / Size**) and Object 2 has a point size of 12. Suppose we wanted to place candidate spikes pointing outward every 20° around these spheres.

3) `seedSpikes spheres.mod mySphereSeeds.mod sphere 20`

4) `3dmod -V mySphereSeeds.mod`
   You'll find you now have a model with 4 objects; Objects 1 and 3 contain the newly seeded particles, while Objects 2 and 4 contain the centers and radii of the spheres from *spheres.mod*. To make the display less confusing, select **Edit / Objects** from the Model View window, chose **Points** and set the **Sphere size** to **2** for **Object 1** and **1** for **Object 3**, respectively. From the 3dmod info window, select **File / Save Model,** and then exit all 3dmod windows when finished.

5) Cylindrical surfaces are handled in a similar fashion, but now 2-point open contours are required to define the radius, orientation, and height of the starting cylinder(s). `3dmod cylinders.mod` and toggle between Objects 1 and 2 in the 3dmod info window to see their definitions.

6)
```
seedSpikes cylinders.mod myCylinderSeeds.mod cylinder 20
```

7)
```
3dmod -V myCylinderSeeds.mod
```

As before, adjust the point sizes for better visibility. After examining the resulting model, exit all 3dmod windows, saving any changes if desired. If you wish, explore the man page for seedSpikes to see additional program options. Models generated by seedSpikes are not directly suitable for use in PEET alignments; instead, they're designed to be used as input to another accessory program, spikeInit, which will generate the final models, initial motive lists, and rotation axes files.

8)
```
spikeInit mySphereSeeds.mod sphere
```

Note the names of the output model, initial motive list, and rotation axes. Check the spikeInit man page for additional options, if you wish.

9)
```
3dmod -V mySphereSeeds_sphere_spikes.mod
```

All the spike model points have been moved to Object 1 (potentially in several contours) as required for PEET alignments.

10)
```
plotRotAxes mySphereSeeds_sphere_spikes.mod \
mySphereSeeds_sphere_RotAxes.csv 5 1
```

where "\" should be followed immediately by **Enter**. This creates a "quiver plot", with "5" and "1" specifying the length and line thickness of the arrows in the quiver. Try zooming, panning, and rotating the plot with some of the controls at the top of the figure window. Close the figure and 3dmod windows when finished.

11) `spikeInit myCylinderSeeds.mod cylinder`

Examine the results as in steps 9 and 10 above.

```
3dmod -V myCylinderSeeds_cylinder_spikes.mod
```

```
plotRotAxes myCylinderSeeds_cylinder_spikes.mod \
myCylinderSeeds_cylinder_RotAxes.csv 5 1
```

where “\” should be followed immediately by **Enter**.

Close all the various windows when finished

IRREGULARLY SHAPED SURFACES

Spikes or particles of interest often lie on an irregularly shaped membrane or surface not well approximated by a sphere or a cylinder. The program meshInit can be used in place of spikeInit in these cases to generate spike orientations perpendicular to the membrane. Membranes are first modeled as an IMOD mesh. Meshing is covered in the 3dmod documentation. We will not repeat those instructions here, but will mention that meshes can be created either by choosing specific vertex locations (e.g. if you are modeling individual, visible spikes / particles) or by dragging with the middle mouse button to place points at regular intervals along an open contour defining the shape of the membrane in a particular slice (e.g. if you wish to seed spikes with approximately constant spacing). Spacing in the latter mode of operation is set by 3dmod’s Edit / Model / Header / Drawing resolution. Typically spacing along each contour should be approximately the same as the interval between contours. Either open or closed contours can be used depending on the structure you wish to model.

12) `3dmod -V mesh.mod`

In this model, I’ve drawn 5 open contours indicating the shape of a hypothetical membrane in 5 slices, and then meshed the model. Exit 3dmod after examining the model.

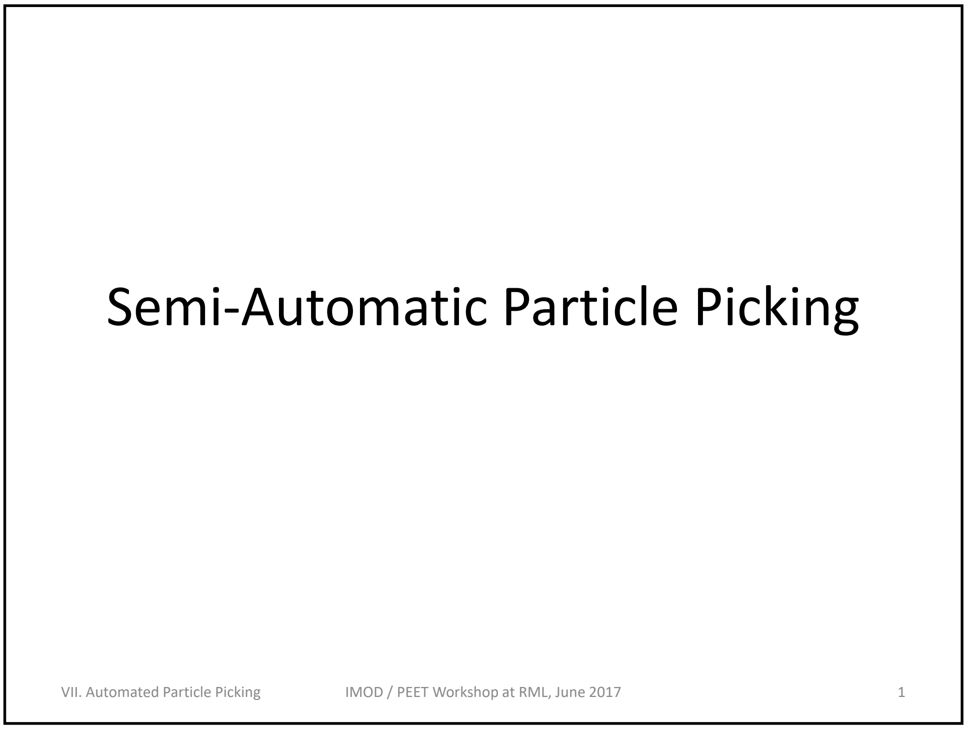
13) `meshInit mesh.mod`

As above, note the names of the generated output files. You may wish to examine the meshInit man page for additional options.

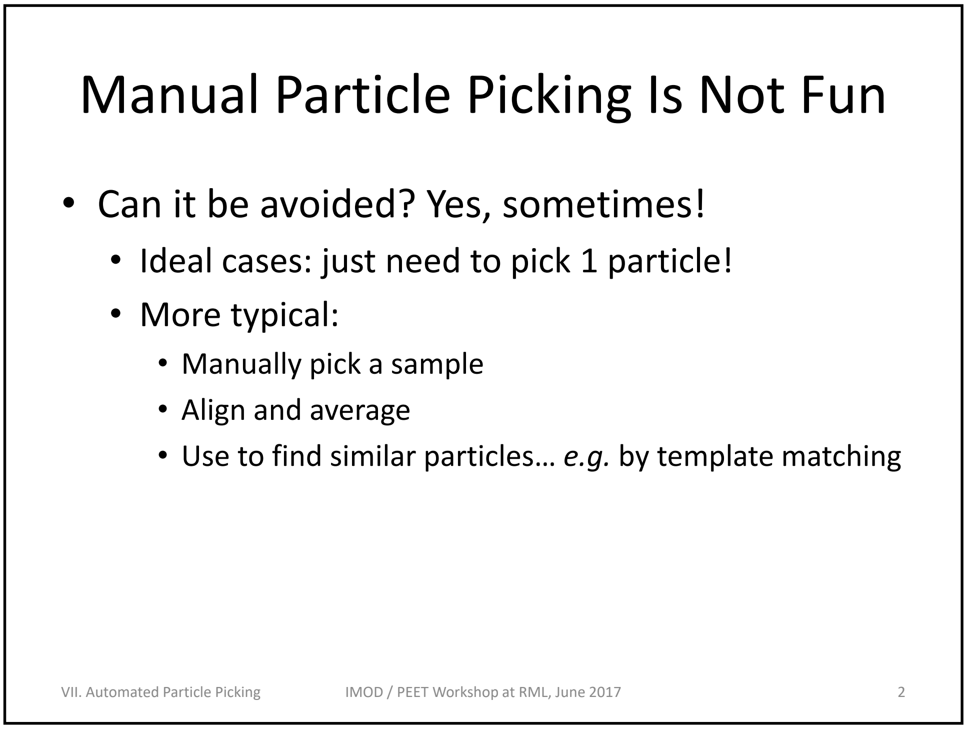
14)
```
3dmodv mesh_mesh.mod
```
and examine the resulting model. Exit 3dmodv when finished.

15)
```
plotRotAxes mesh_mesh.mod mesh_mesh_RotAxes.csv 5 1
```

16) Feel free to repeat steps 13-15 if you like, experimenting with meshInit's **offset** and **flgBothSides** options.

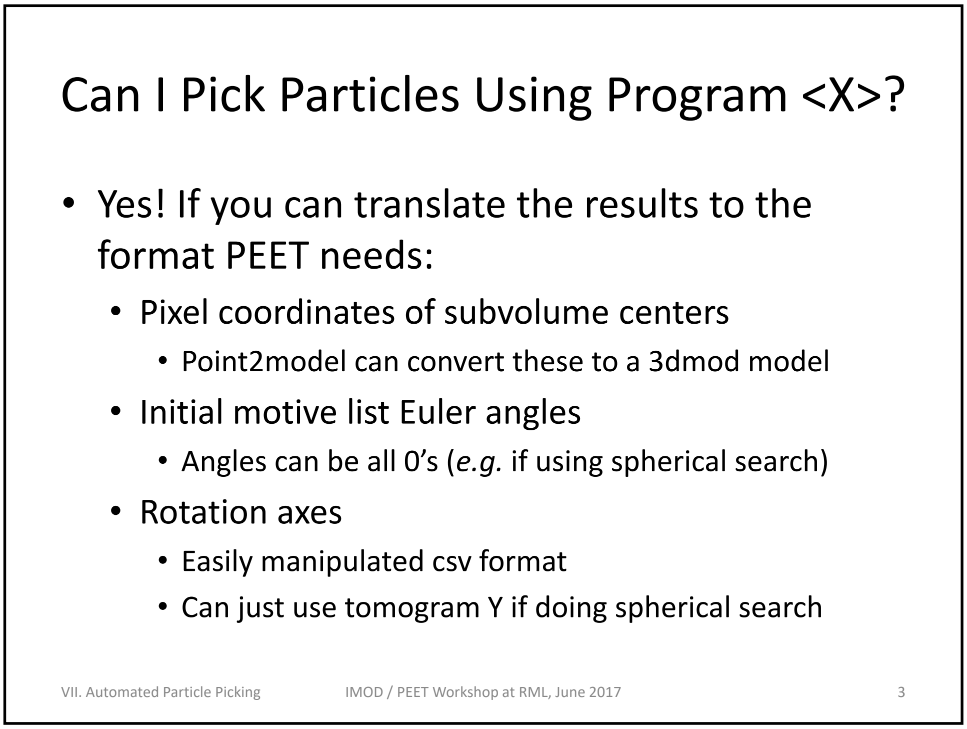


Semi-Automatic Particle Picking



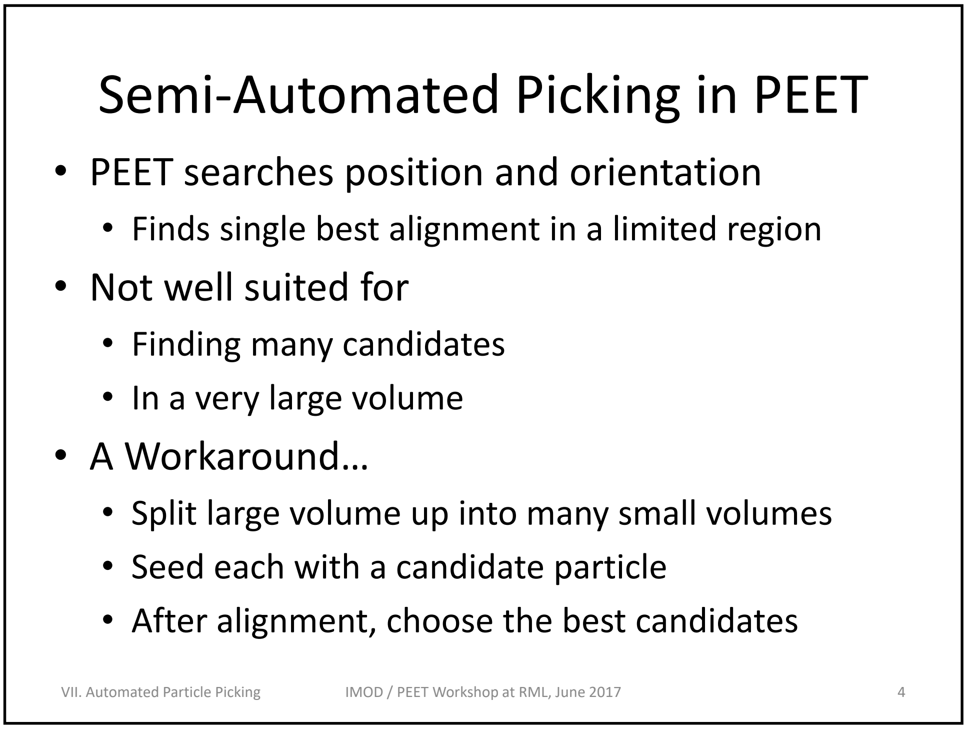
Manual Particle Picking Is Not Fun

- Can it be avoided? Yes, sometimes!
  - Ideal cases: just need to pick 1 particle!
  - More typical:
    - Manually pick a sample
    - Align and average
    - Use to find similar particles… *e.g.* by template matching



Can I Pick Particles Using Program <X>?

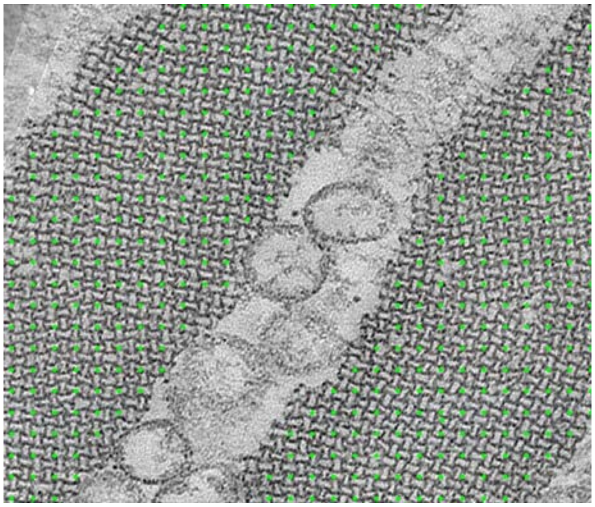
- Yes! If you can translate the results to the format PEET needs:
  - Pixel coordinates of subvolume centers
    - Point2model can convert these to a 3dmod model
  - Initial motive list Euler angles
    - Angles can be all 0’s (*e.g.* if using spherical search)
  - Rotation axes
    - Easily manipulated csv format
    - Can just use tomogram Y if doing spherical search



Semi-Automated Picking in PEET

- PEET searches position and orientation
  - Finds single best alignment in a limited region
- Not well suited for
  - Finding many candidates
  - In a very large volume
- A Workaround…
  - Split large volume up into many small volumes
  - Seed each with a candidate particle
  - After alignment, choose the best candidates

A Case Where 1 Point Suffices!

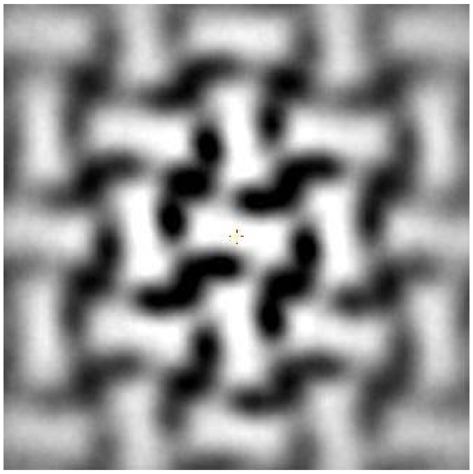


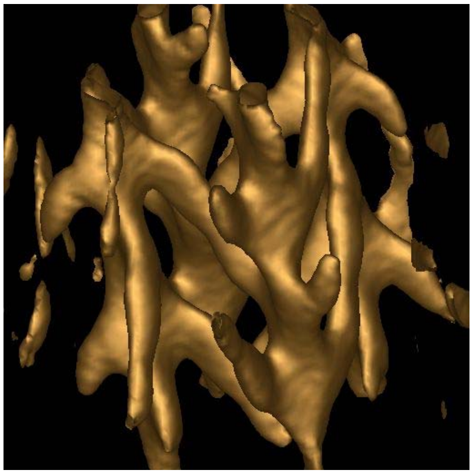
~30 nm Slice of Sonic Muscle Z-line (with Pradeep Luther)

Sonic Muscle Picking Strategy

- Seed points with gridInit
- Drawing tools eraser to clean up initial seeds
- 1 manually picked point as a starting reference
- Align and average
- Threshold by cross-correlation
    - *E.g.* use selectClassID and createAlignedModel
- Manually remove bad points – not needed here
- Symmetrize

Sonic Muscle Results





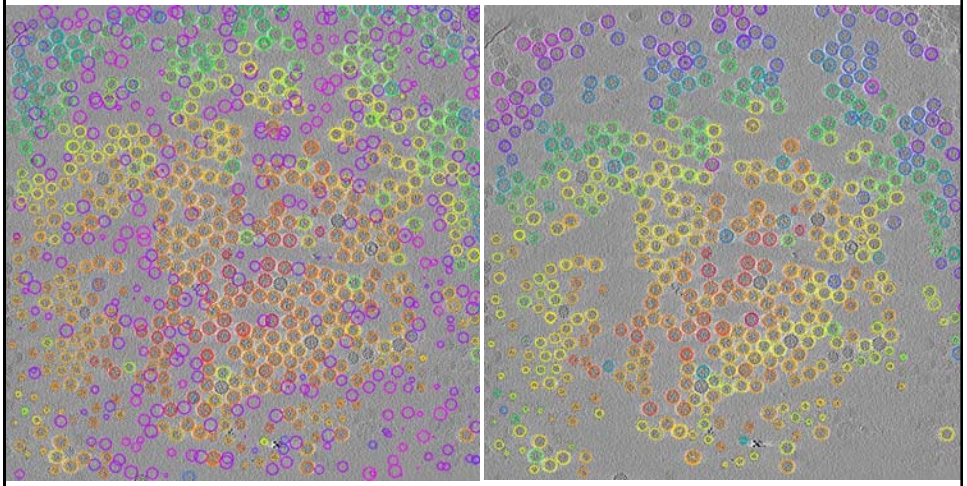
More Typically…

- Align and average several hundred particles
- Use the resulting average as a template to find additional particles (often in other volumes)
- Template matching in PEET
  - Select “No reference refinement” in Etomo
  - “flgNoReferenceRefinement = 1” in prm file
- Beware of Reference Bias
  - Bin or low-pass filter the template
  - Final locations should look as if manually picked

Semi-automated Particle Picking

- Widely applicable
  - 2D crystals
  - Isolated particles (will explore BPV in exercise)
  - Spikes / fusion proteins
  - Membrane associated proteins
- PEET tools / programs are fairly flexible
- Strategy / details will vary with application

Isolated Particles: Binned BPV



Before Thresholding

After Thresholding

Pseudo-colors (inserted by createAlignedModel) reflect cross-correlation coefficient

Questions?

Semi-automated particle picking for BPV

In this exercise, we'll explore semi-automated particle picking using the binned BPV data which we've averaged and symmetrized in previous exercises. For simplicity, we'll use the initial, unsymmetrized average to pick particles in the same tomogram from which the initial average was created. The results are fairly typical of what you're likely to experience when applying template matching.

1) `cd $WORKSHOP_HOME/PEET_Labs/BPV_-3/PEET/templateMatching`

2) `3dmod ../../bpv_bin2.rec`

   Using the ZaP or Slicer windows, re-examine this volume. Recall that it contains around 130 BPV particles at varying Z heights. We will use another modeling aid, gridInit, to prepare for template matching. Typically, you will want to choose parameters that place between 1 and 5 times as many candidate particles as there are real ones. In this case, we will end up with about 3X and we'll spread the candidate particles over 2 layers. Exit 3dmod.

3) `gridInit ../../bpv_bin2.rec 65 65 35:65:110`

   Here, we've placed candidates at 65 voxel intervals in each of the 3 dimensions, leading to 450 particles. In the case of Z, I've explicitly specified start:step:end to avoid one layer being placed close to the edge of the volume. The above command puts points on Z=36 and Z=101. The default would be points at 65 and 130, and this volume is only 136 slices thick.

4) `3dmod ../../bpv_bin2.rec bpv_bin2_grid.mod`

   Explore the model created by gridInit in the ZaP and Model view windows. Exit 3dmod when finished.

5) `etomo *.epe`

   Notice in particular the following settings.

   a) On the Setup tab, we've specified the final reference volume generated by our previous firstSearch alignment.
   b) We've specified initial Uniform random rotations of the particles to minimize missing wedge artifacts, since we'll be using a limited angular search range.
   c) On the Run tab, we've specified template matching by selecting No reference refinement under Optional / Advanced Features. We've also selected Strict search limit checking.
   d) Since candidate particles are spaced every 65 voxels, we need an initial search distance of no less than ±32.5; I've chosen 38.
   e) Finally, because we're intentionally searching far enough to allow neighboring particles to possibly converge to the same location and orientation, I've turned on duplicate removal on the Run tab. To prevent partially overlapping particles, I have chosen an unusually loose duplicate shift tolerance of 20 voxels and disabled the angular criterion by setting it to 360°.

   The actual alignment takes a few hours using 20-30 cpus. As usual, we have completed the run and provided the output for you. Next we'll examine these results and decide where to set a cross-correlation threshold for which of the candidate particles to accept as genuine.

6) Press **Open averages in 3dmod** and examine the variation in the resulting averages with number of particles. The initial average with 100 particles looks reasonable. Subsequent averages aren't bad, but become increasingly noisy. This is expected, since there are only approximately 130 real particles in this tomogram. Exit Etomo and answer **Yes** to end all 3dmod programs.

7) Another way to characterize the results is to look at a histogram of the cross-correlation scores. The output motive list is in a simple .csv format readable by almost any spreadsheet, graphics, or scientific programming package. Cross-correlation scores are in column 1, so creating a histogram using Excel or LibreOffice Calc is straightforward. In this case, I've already created such a histogram for you using MATLAB. Examine it by browsing to and double-clicking on scoreHistogram.pdf. You'll see a bimodal distribution, with a smaller, upper peak of good scores above 0.35 – 0.4. Hopefully, these constitute mostly real virus particles. Close the histogram when finished.

8)
```
createAlignedModel *.prm
```

9)
```
3dmod ../../bpv_bin2.rec bpv_bin2_grid_Tom1_Iter3.mod
```

10) Open **Edit / Object / Type** and set **Sphere radius for points** to **19**, press **Edit / Scale bar**, check **color ramp**, and set a **minimum scale bar length** of **50** pixels. Don't close the scale bar dialog or the scale bar will disappear; you can minimize it, however, if your screen feels cluttered. In addition to placing particle model points at their aligned positions, createAlignedModel stores the cross-correlation score with each point, and can display that score as a pseudo-color. Page up and down using the ZaP or Slicer windows, and you'll see that higher cross-correlation purple and red points typically look good, while blue, green, and yellow points are often questionable. To see the numeric value of the cross-correlation score for a particular point, enter **Model** mode, left-click on the point of interest, and then press **Edit / Fine Grain**. The cross-correlation score for the selected point will be displayed at the bottom of the Fine Grain dialog under **General Value**. Try selecting a few points with different colors to see their scores. Based on this examination and the previous results visualized in the histogram, I chose to accept as genuine particles with cross-correlation scores of 0.35 or better. Save your model. Next, we'll see how to create model, motive list, and RotAxes files containing only the selected particles.

Column 20 of motive lists is used in PEET for a class ID number. In this case, duplicate removal is enabled, so particles removed as duplicates have been assigned to class -9999, while all other particles are in class 0. We'll assign all particles not flagged as duplicates and with scores of 0.35 or larger to class 1. Since we need to refer to the existing class assignment to check for duplicates, we'll do this initially in a temporary column (**V**) within a spreadsheet program.

11) First, we'll backup the original motive list

```
cp bpv_MOTL_Tom1_Iter4.csv bpv_MOTL_Tom1_Iter4.csv.orig
```

12) `oocalc bpv_MOTL_Tom1_Iter4.csv`

    Make sure **from row** is set to **1**, only **Comma** is checked under **Separated by**, and press **OK**.

13) Select cell **V2** and enter **=IF(AND(T2=0, A2>=0.35),1,0)**. Press **Enter**.

14) Select cell **V2** again and press **Edit / copy** or **Ctrl-C**.

15) Press **Ctrl-Shift-End** to highlight **V2:V451**, and then press **Ctrl-V** or select **Edit / Paste**). Press **Yes** when asked if it is okay to overwrite cells already containing data.

    This will copy the formula from **V2** to the remaining cells in this column, adjusting row and column numbers as needed. Column **V** should now have 1's marking those particles we wish to include. Verify that this is the case.

16) **V2:V451** should still be selected. Press **Ctrl-C** to copy the entire column.

17) Right-click in cell **T2**, and select **Paste Special**.

18) In the **Paste Special** dialog, under **Selection** uncheck everything except **Numbers**. Press **OK** and then **Yes**. This will replace the contents of column **T** with only the values from column **V**. It's important to paste only values and not formulae at this step; if we pasted formulae, row numbers would get adjusted relative to their new location, and the result would no longer be correct.

19) We no longer need column **V**. Left click on the label **V** at the top of this column to select it, and then press **Delete** to erase its contents.

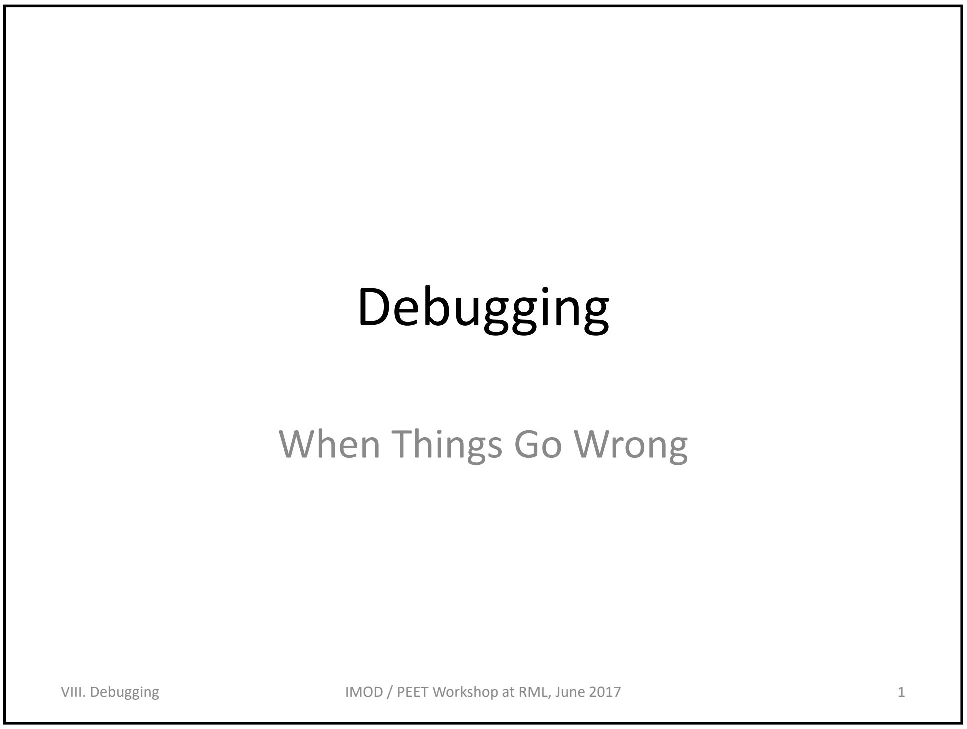
20) Save the file; if asked, select **Use Text CSV Format**. Exit oocalc when finished.

21) Next we need to tell PEET (specifically, those PEET programs which honor the directive we're about to use) that we only wish to process particles in class 1. This can be done in either of 2, equivalent ways:
    a. Run `gedit *.prm`, add a line at the bottom **selectClassID = 1**, save the file, and exit gedit. *NOTE*: because etomo reads the prm file on startup and writes it automatically on exit (and at other times), you must never manually edit a prm file while it is also open in etomo.
    b. Run `etomo *.epe`, under **Run** tab set **Average only members of classes** to **1**, and exit etomo.

22) `createAlignedModel *.prm`
This generates model, motive list, and RotAxes files containing only the selected particles. In 3dmod select to **File / Reload Model** to see the newly created model on *bpv_bin2.rec* and see if you are satisfied with the selection of particles. Often, minor manual editing of automatically picked particles is desirable. In this case, there are a few reasonable looking particles which have not been included and which we might want to add manually.

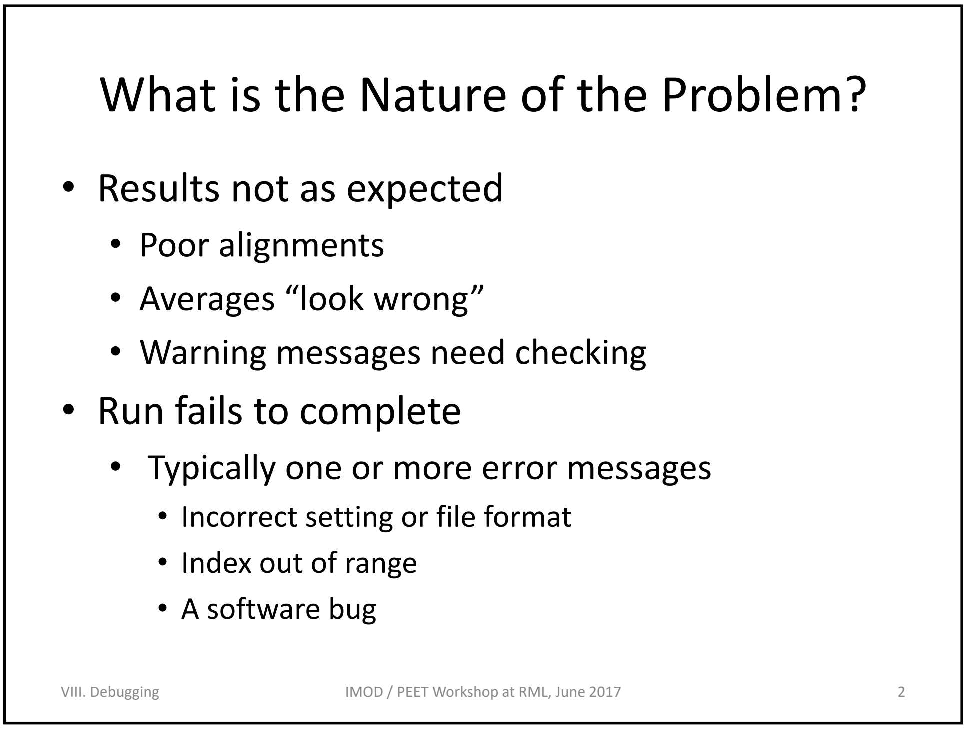
Programs that honor selectClassID include averageAll, createAlignedModel, the classification, and FSC / SSNR programs. Related settings includeList and excludeList allow specifying particles by number rather than class.

After using selectClassID, includeList, or excludeList it is a good idea to delete or comment out that setting once you've finished with it. `gedit *.prm` and put a # before the **selectClassID = [1]**. This will tell PEET to ignore that line. Alternatively, you can run Etomo, set **Average only members of class** to **blank**, and exit Etomo. Confusing and seemingly mysterious results can arise if you've specified a restricted (possibly no longer even existing) subset of particles and then forgotten about it!



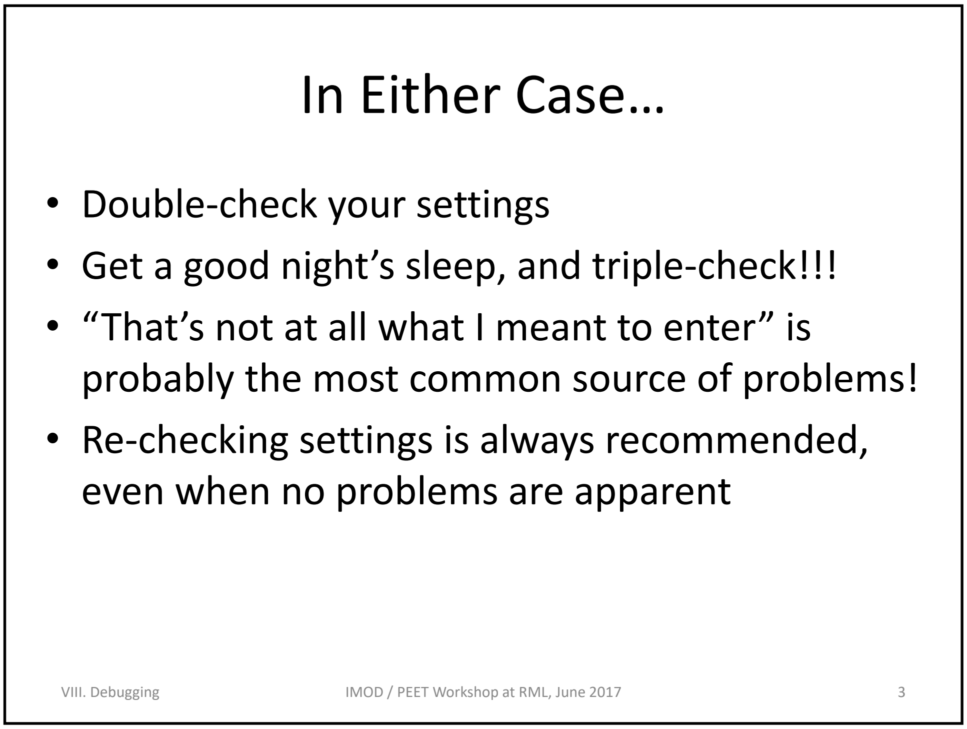
Debugging

When Things Go Wrong



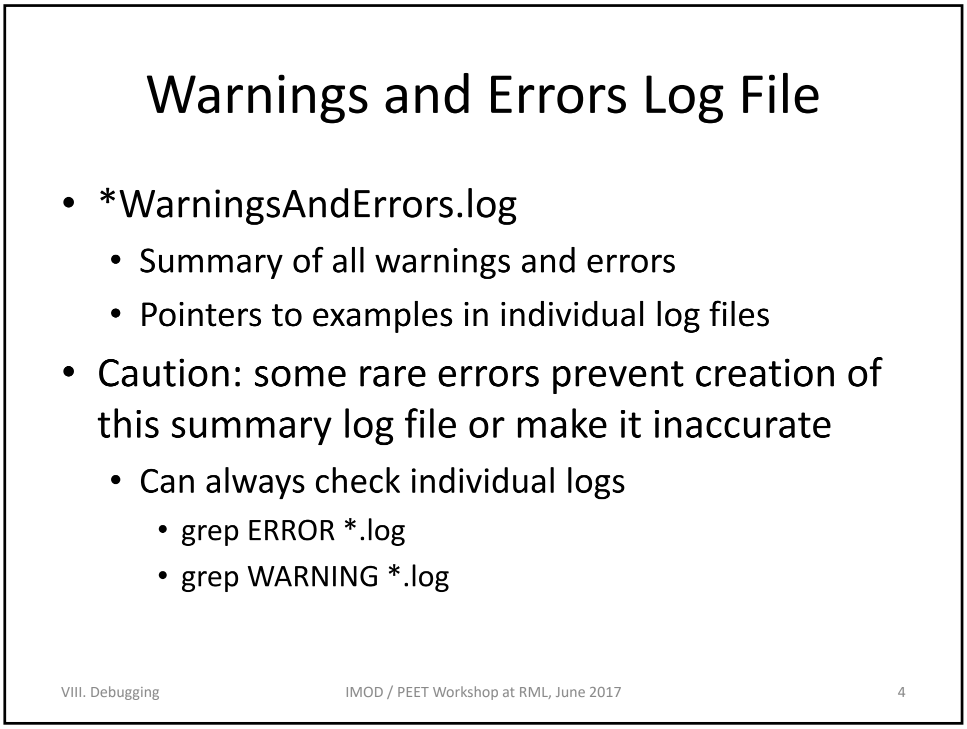
What is the Nature of the Problem?

- Results not as expected
  - Poor alignments
  - Averages “look wrong”
  - Warning messages need checking
- Run fails to complete
  - Typically one or more error messages
    - Incorrect setting or file format
    - Index out of range
    - A software bug



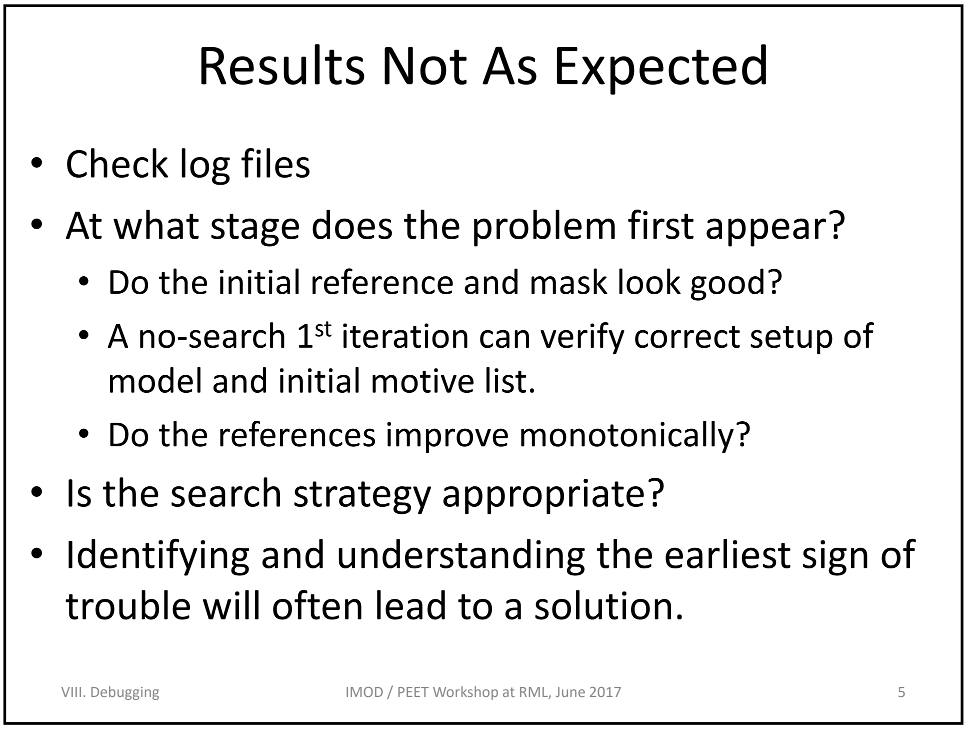
In Either Case…

- Double-check your settings
- Get a good night’s sleep, and triple-check!!!
- “That’s not at all what I meant to enter” is probably the most common source of problems!
- Re-checking settings is always recommended, even when no problems are apparent



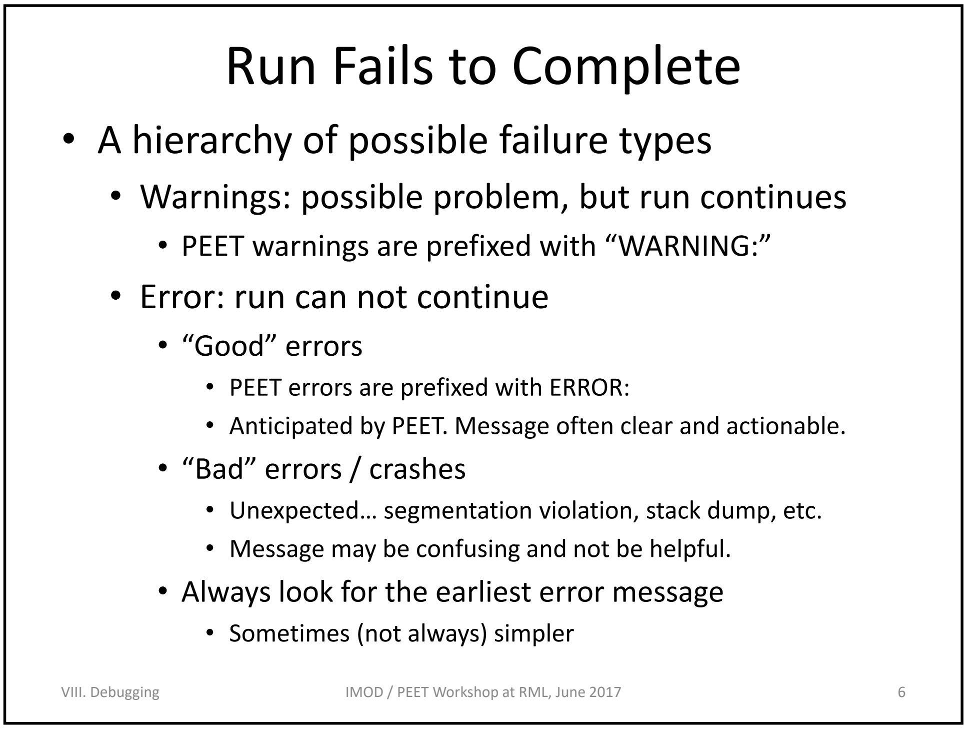
Warnings and Errors Log File

- *WarningsAndErrors.log
  - Summary of all warnings and errors
  - Pointers to examples in individual log files
- Caution: some rare errors prevent creation of this summary log file or make it inaccurate
  - Can always check individual logs
    - grep ERROR *.log
    - grep WARNING *.log



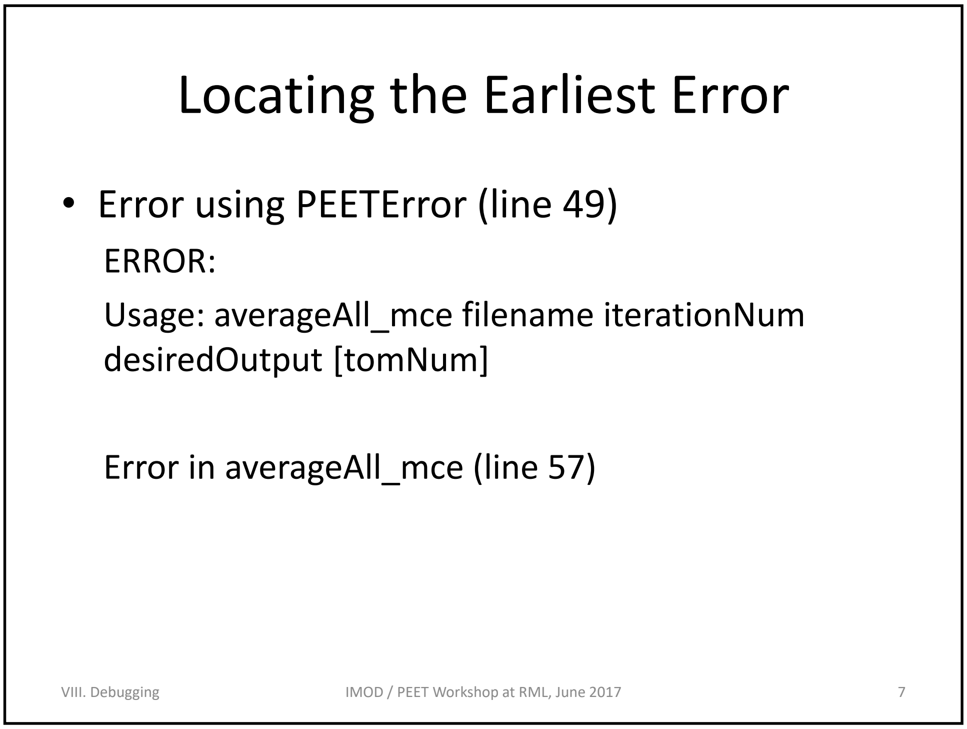
Results Not As Expected

- Check log files
- At what stage does the problem first appear?
  - Do the initial reference and mask look good?
  - A no-search 1st iteration can verify correct setup of model and initial motive list.
  - Do the references improve monotonically?
- Is the search strategy appropriate?
- Identifying and understanding the earliest sign of trouble will often lead to a solution.



Run Fails to Complete

- A hierarchy of possible failure types
  - Warnings: possible problem, but run continues
    - PEET warnings are prefixed with “WARNING:”
  - Error: run can not continue
    - “Good” errors
      - PEET errors are prefixed with ERROR:
      - Anticipated by PEET. Message often clear and actionable.
    - “Bad” errors / crashes
      - Unexpected… segmentation violation, stack dump, etc.
      - Message may be confusing and not be helpful.
    - Always look for the earliest error message
      - Sometimes (not always) simpler

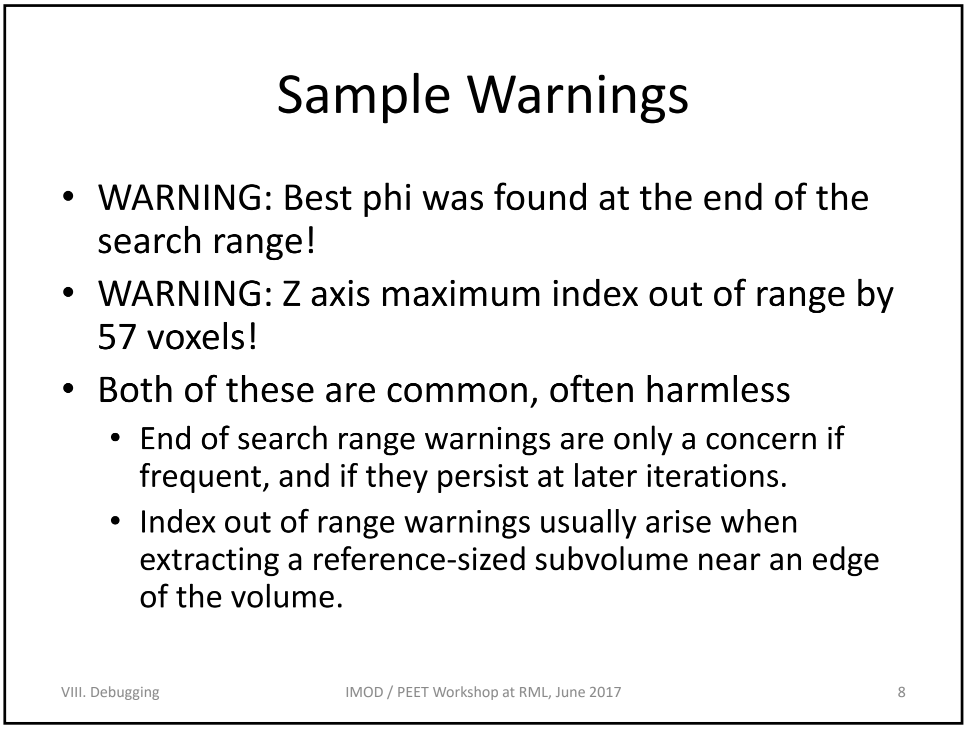


Locating the Earliest Error

- Error using PEETError (line 49)

```
ERROR:
Usage: averageAll_mce filename iterationNum
desiredOutput [tomNum]


Error in averageAll_mce (line 57)
```



Sample Warnings

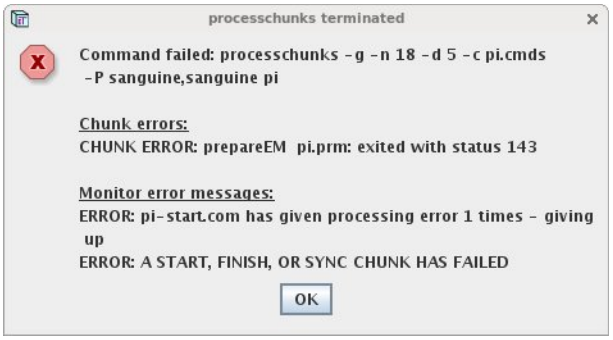
- WARNING: Best phi was found at the end of the search range!
- WARNING: Z axis maximum index out of range by 57 voxels!
- Both of these are common, often harmless
  - End of search range warnings are only a concern if frequent, and if they persist at later iterations.
  - Index out of range warnings usually arise when extracting a reference-sized subvolume near an edge of the volume.

Sample "Good" Errors

- Usage messages
  - ERROR:

    Usage: averageAll_mce filename iterationNum desiredOutput [tomNum]
- ERROR: Volumes must have the same voxel size!
  - A volume has incorrect or missing voxel size in header
  - Or you're trying to combine incompatible volumes…
- ERROR: Motive list size does not match number of model particles!
  - Often caused by incorrect model or MOTL name

Examples of "Bad" Errors

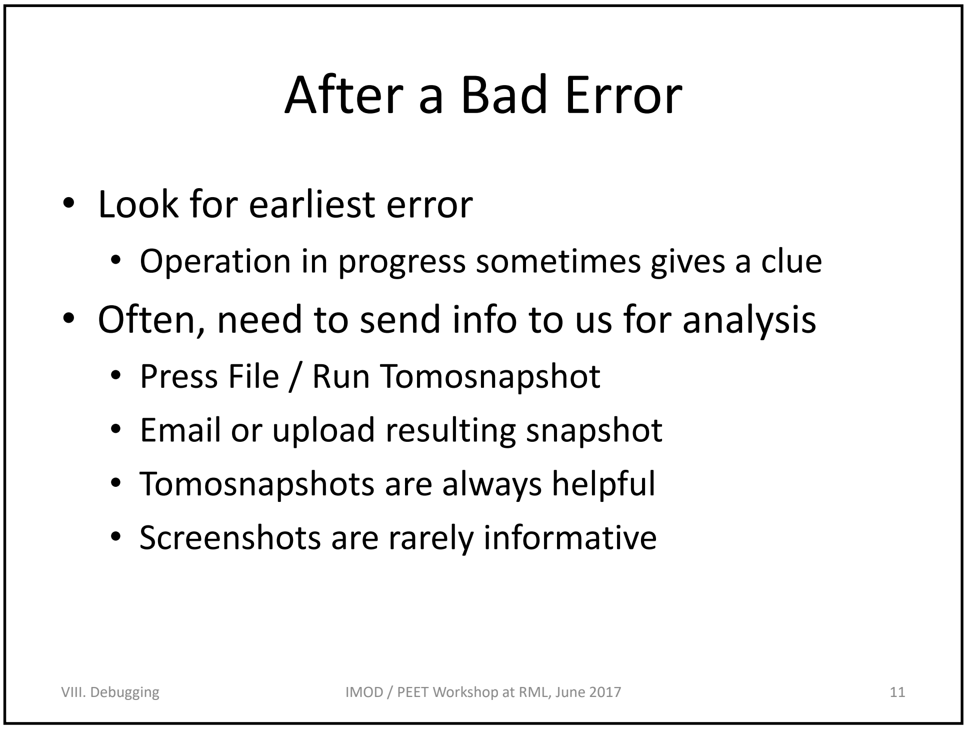


Command failed: processchunks -g -n 18 -d 5 -c pi.cmds -P sanguine,sanguine pi

Chunk errors:
CHUNK ERROR: prepareEM pi.prm: exited with status 143

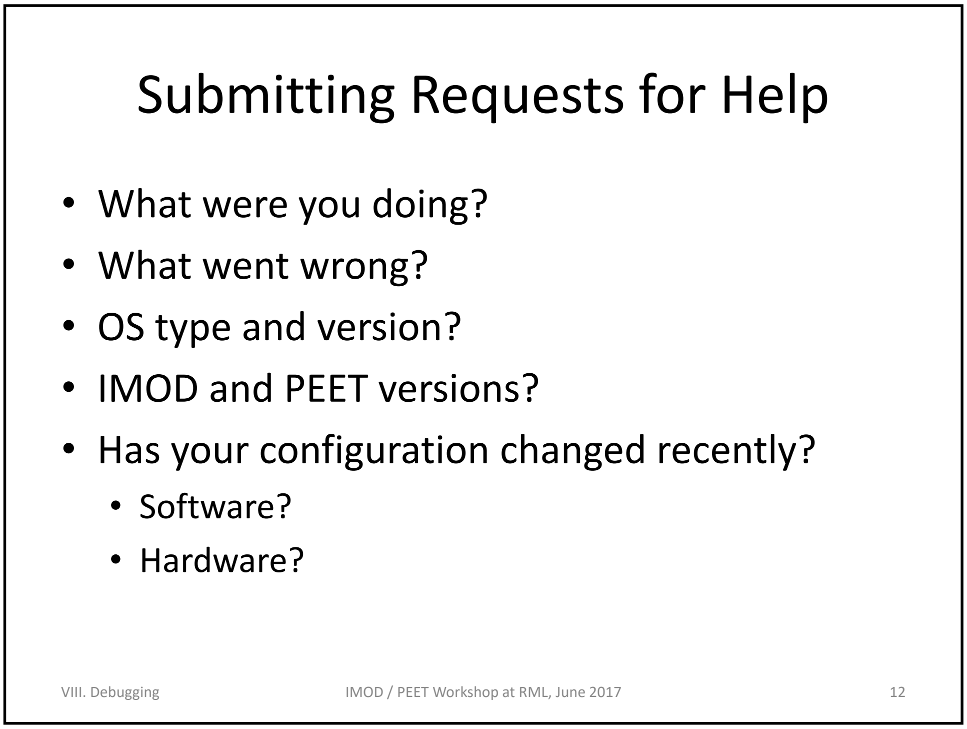
Monitor error messages:
ERROR: pi-start.com has given processing error 1 times - giving up
ERROR: A START, FINISH, OR SYNC CHUNK HAS FAILED

- Index out of range
- Segmentation violation / core dumped



After a Bad Error

- Look for earliest error
  - Operation in progress sometimes gives a clue
- Often, need to send info to us for analysis
  - Press File / Run Tomosnapshot
  - Email or upload resulting snapshot
  - Tomosnapshots are always helpful
  - Screenshots are rarely informative



Submitting Requests for Help

- What were you doing?
- What went wrong?
- OS type and version?
- IMOD and PEET versions?
- Has your configuration changed recently?
  - Software?
  - Hardware?

Questions?

THIS PAGE INTENTIONALLY LEFT BLANK

Debugging

Debugging tends to be highly problem specific. If you get an error message that is not self-explanatory, the general approach is to find the earliest indication of trouble, and to try to figure out what operation was in progress and what has gone wrong. In this exercise, we'll look at few examples of errors that cause PEET to fail in the earliest stages of processing.

1) `cd $WORKSHOP_HOME/PEET_Labs/Intro-1/PEET/badSettings`

2) `etomo *.epe`
   Etomo will display a dialog stating that it is unable to parse the parameter file, *pi.prm*. Additionally, it will state that the problem appears to be a missing "[" delimiter and that the problem occurred while attempting to parse line 96.

3) Close the dialog by pressing **No** and then exit the resulting Etomo window.

4) `gedit pi.prm`

5) In the upper right, select the **Go to line** function in the menu of your version of Gedit and enter "**96**" and press **return**. Notice that line 96 currently reads "lstThresholds = 1:1:-3]". As the error message stated, there appears to be a missing "[" delimiter. Change the line to read **lstThresholds = [1:1:-3]**, Press **Save** and exit gedit.

Errors that occur very early in processing and particularly syntax errors that prevent parsing the parameter file can often prevent generation of helpful error messages. When Etomo is unable to start and no clear message is given, it is often helpful to try to manually run `prmParser *.prm` and examine the resulting output; in this case, the error message was reasonably descriptive, and we were able to locate the problem, so this is not necessary. If Etomo runs, but an unclear error message results shortly after pressing Run, it can be helpful to examine the *prmParser.log*, **start.log*, and **WarningsAndErrors.log* files for clues as to the source of the problem.

6) `etomo *.epe`
   Now Etomo starts successfully, but gives a prominent warning in red that one or more specified files cannot be found.

7) Press **Fix Incorrect Paths**
   A file chooser dialog will open; you'll see that it's looking for the initial motive list file named *initMOTL.csv*, and is currently expecting this file to be in the PEET directory one level up. It's actually in the *badSettings* project directory.

8) In the resulting file chooser, double-click *badSettings*, and then *initMOTL.csv*. In this case, this is the only file Etomo was having trouble locating, so the red error text in Fix Incorrect Paths dialog will automatically disappear. If there were additional missing files, they would be shown in successive file choosers.

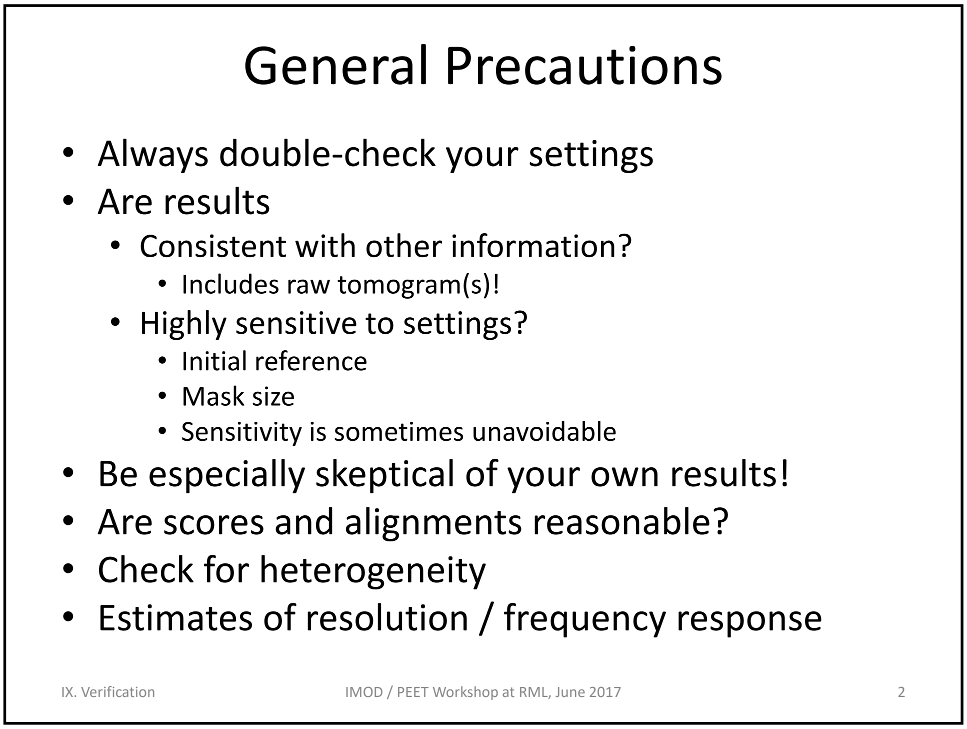
9) Go to the **Run** tab and press **Run** at the bottom of the page
   Now we get a "Chunk Error" saying "ERROR: prepareEM pi.prm: exited with status 255". Chunk errors and exit status are often not helpful. Press **OK** to exit the error dialogue. At least we know that the problem occurred during prepareEM, which, in this case, is the final step in the *pi-start.com* chunk; specifically, this is where the initial motive list for the first iteration is generated. (Motive lists were stored in a different format and called EM files in early versions of PEET).

When an unhelpful error message is given, often a good next step is to try to rerun the first failing step manually. *I.e.* in this case, we would run `prepareEM pi.prm` and examine the output. `gedit pi_WarningsAndErrors.log` followed by `gedit pi-start.log` to see a better readout of the error. In this case the chunk error message and the log files *pi_WarningsAndErrors.log* and *pi-start.log* all give a more explicit, more helpful error message: "ERROR: List of number of particles to average can not be empty!"

10) In Etomo, go to the **Run** tab and check the setting for the number of particles to average. You'll see that the End value has been mistyped as -3 rather than 3. *I.e.* we're asking to average a list of particles numbers from 1 to -3 in steps of +1. MATLAB evaluates that as a null or empty vector, which explains this error message. Change the **End value for Number of Particles to Average** to **3** and press **Run** again. Now the project should run to completion.

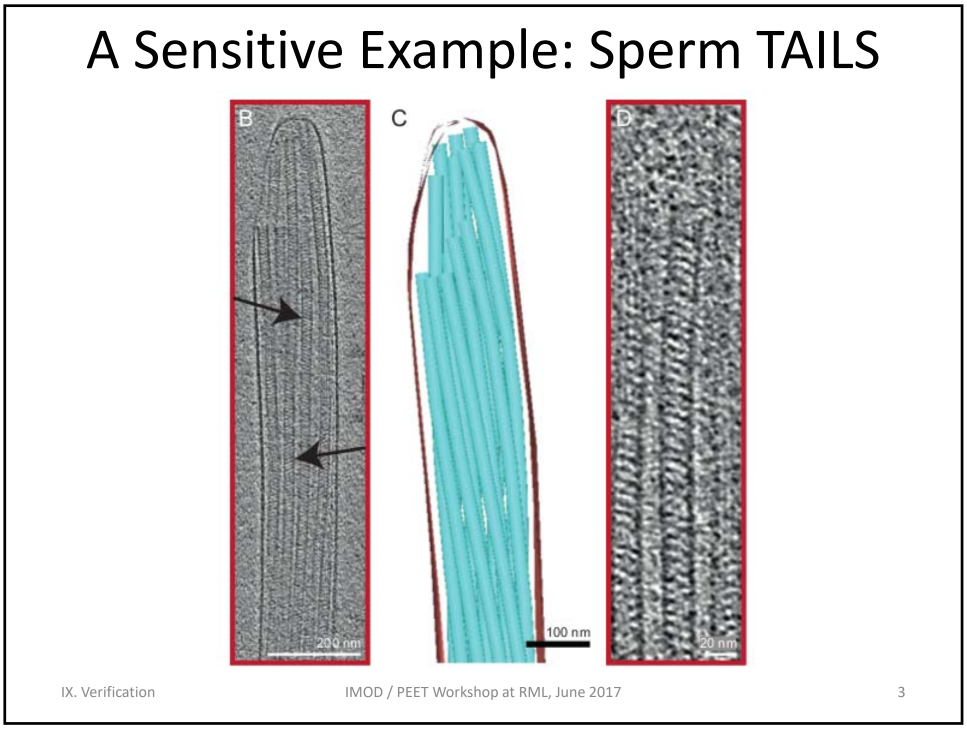
11) Exit Etomo and run `PEETCleanup *.prm` followed by `rm *~` to remove unwanted files.

Verifying Results



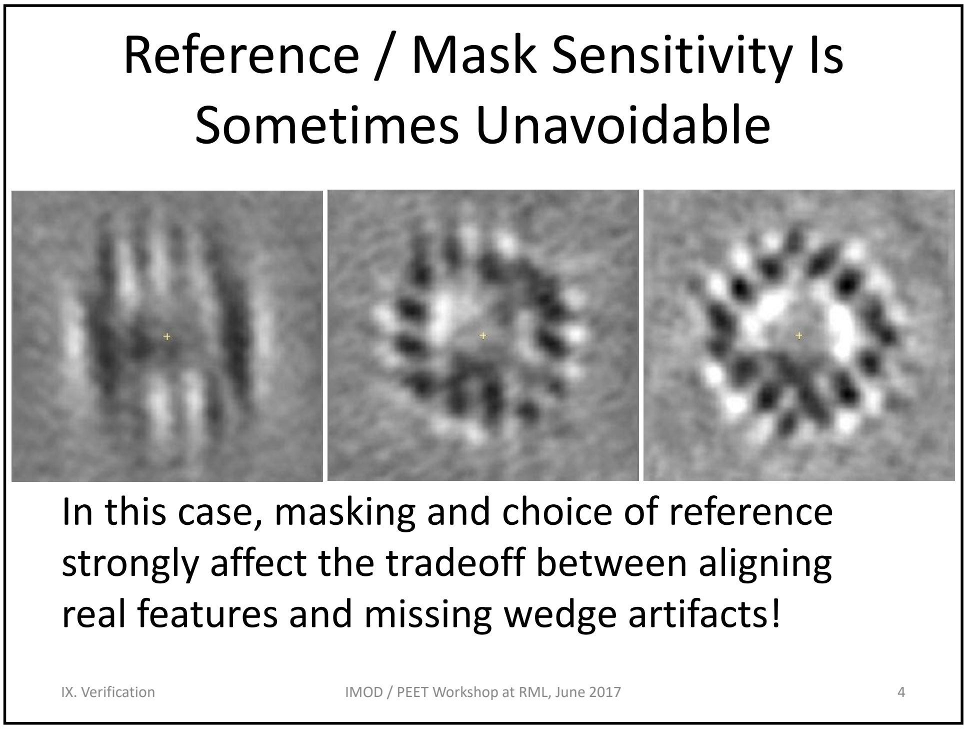
General Precautions

- Always double-check your settings
- Are results
  - Consistent with other information?
    - Includes raw tomogram(s)!
  - Highly sensitive to settings?
    - Initial reference
    - Mask size
    - Sensitivity is sometimes unavoidable
- Be especially skeptical of your own results!
- Are scores and alignments reasonable?
- Check for heterogeneity
- Estimates of resolution / frequency response



A Sensitive Example: Sperm TAILS

IX. Verification IMOD / PEET Workshop at RML, June 2017 3

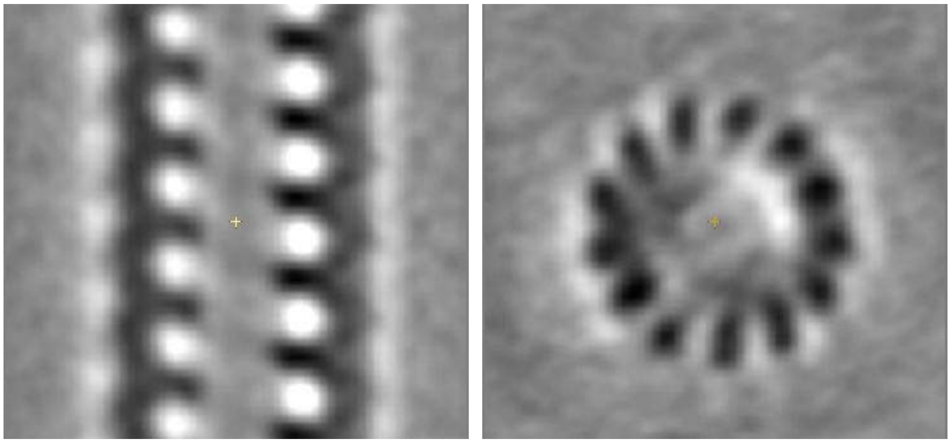


Reference / Mask Sensitivity Is Sometimes Unavoidable

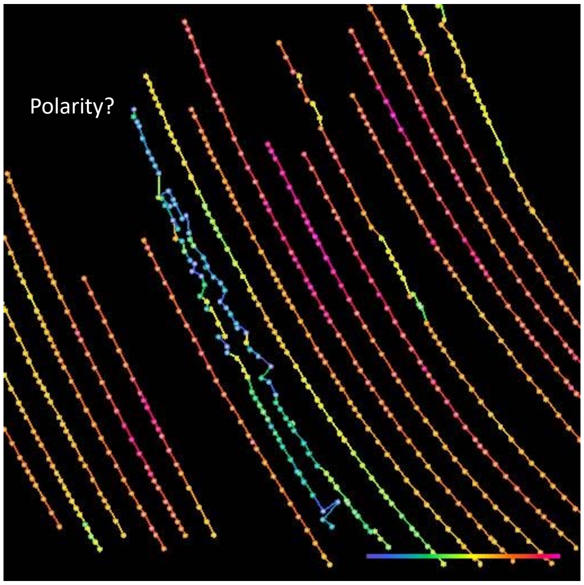
In this case, masking and choice of reference strongly affect the tradeoff between aligning real features and missing wedge artifacts!

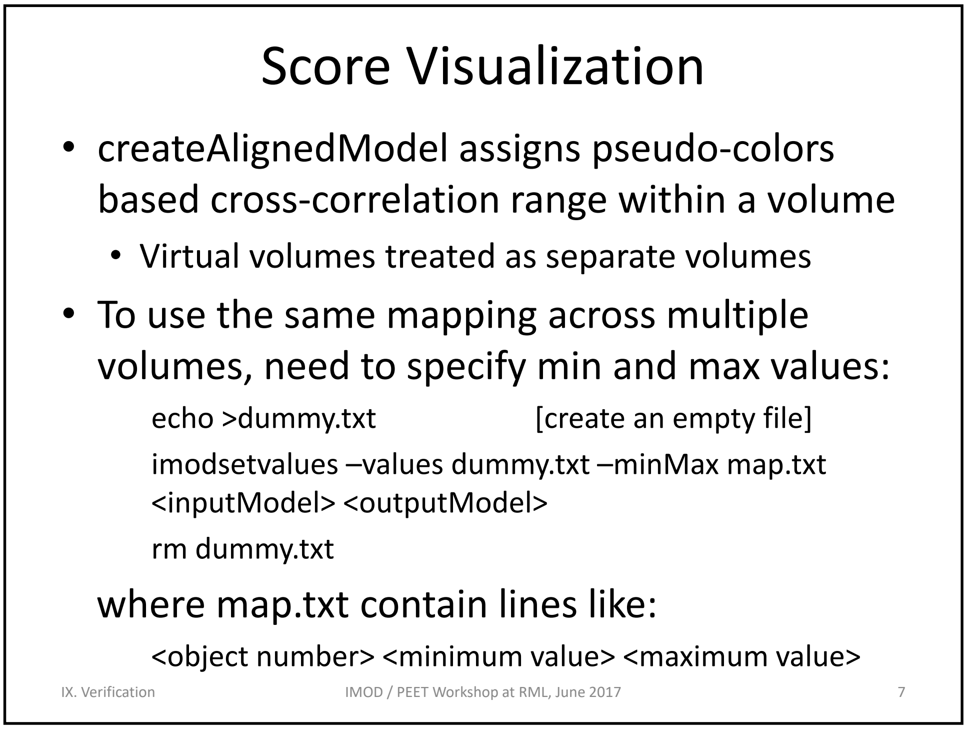
IX. Verification IMOD / PEET Workshop at RML, June 2017 4

TAILS Final Average



Checking Alignment and Scores





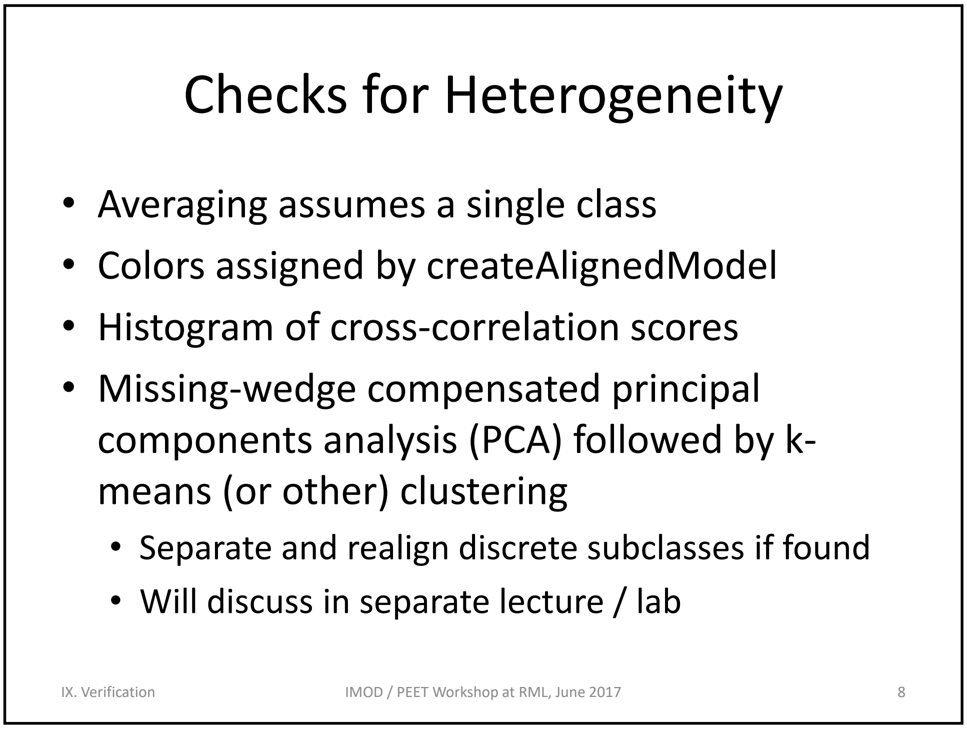
Score Visualization

- createAlignedModel assigns pseudo-colors based cross-correlation range within a volume
  - Virtual volumes treated as separate volumes
- To use the same mapping across multiple volumes, need to specify min and max values:

```
echo >dummy.txt                [create an empty file]
imodsetvalues –values dummy.txt –minMax map.txt
<inputModel> <outputModel>
rm dummy.txt
```

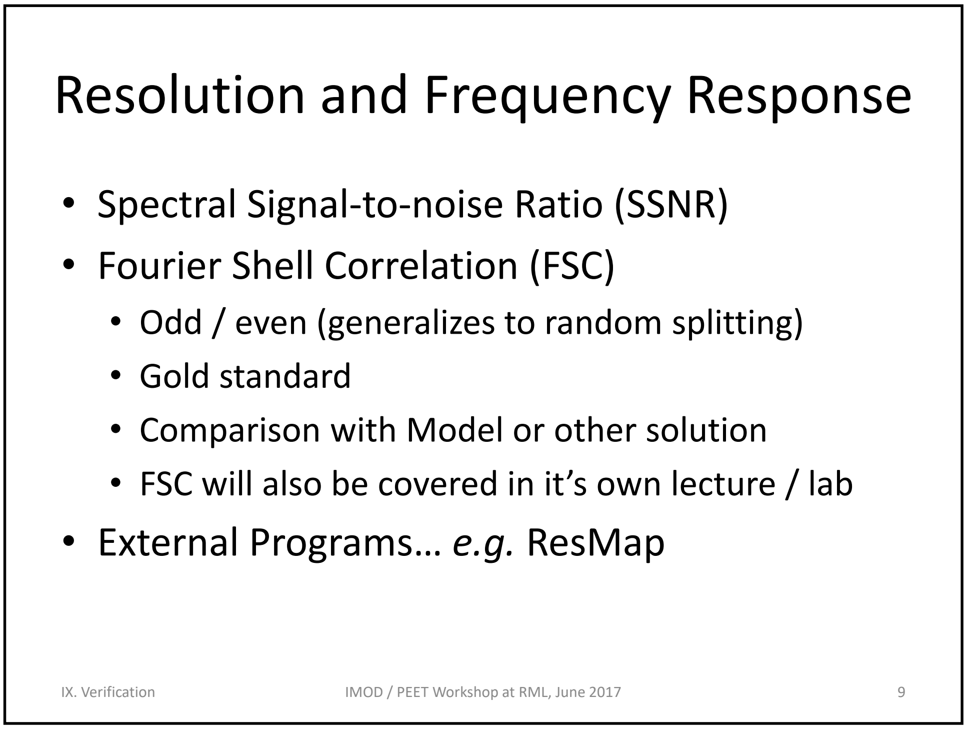
where map.txt contain lines like:

```
<object number> <minimum value> <maximum value>
```



Checks for Heterogeneity

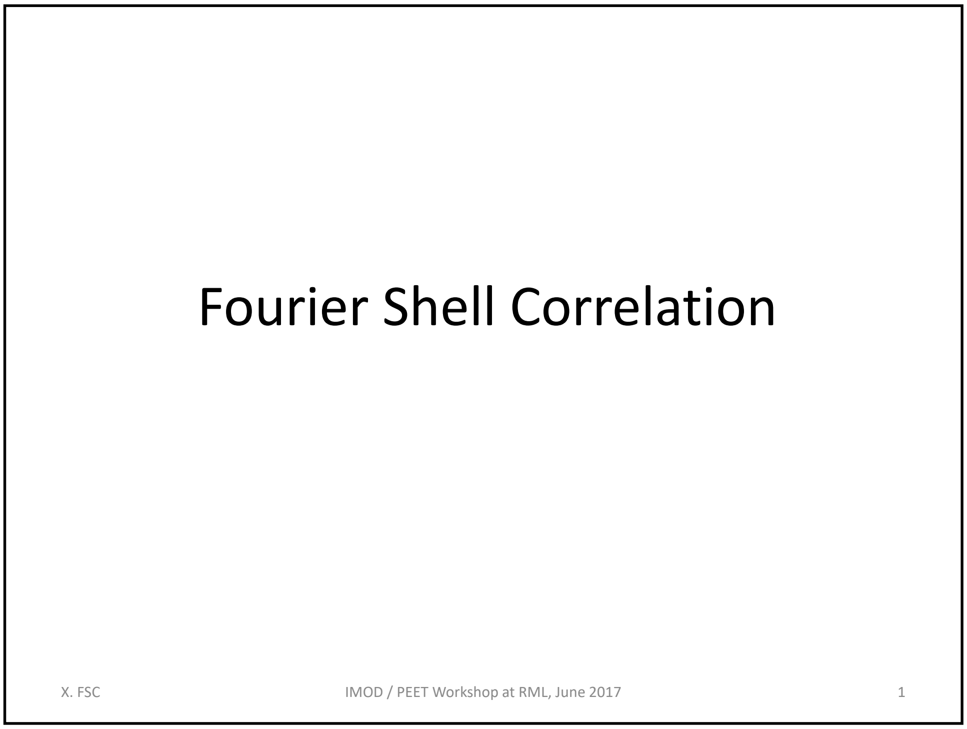
- Averaging assumes a single class
- Colors assigned by createAlignedModel
- Histogram of cross-correlation scores
- Missing-wedge compensated principal components analysis (PCA) followed by k-means (or other) clustering
  - Separate and realign discrete subclasses if found
  - Will discuss in separate lecture / lab



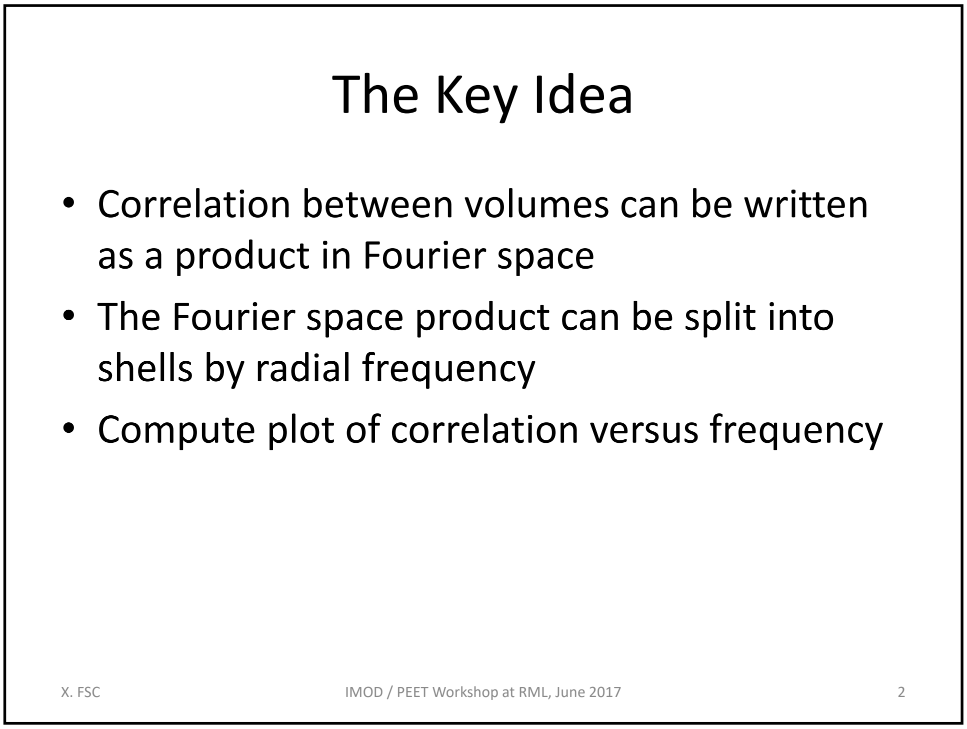
Resolution and Frequency Response

- Spectral Signal-to-noise Ratio (SSNR)
- Fourier Shell Correlation (FSC)
  - Odd / even (generalizes to random splitting)
  - Gold standard
  - Comparison with Model or other solution
  - FSC will also be covered in it’s own lecture / lab
- External Programs… *e.g.* ResMap

Questions?



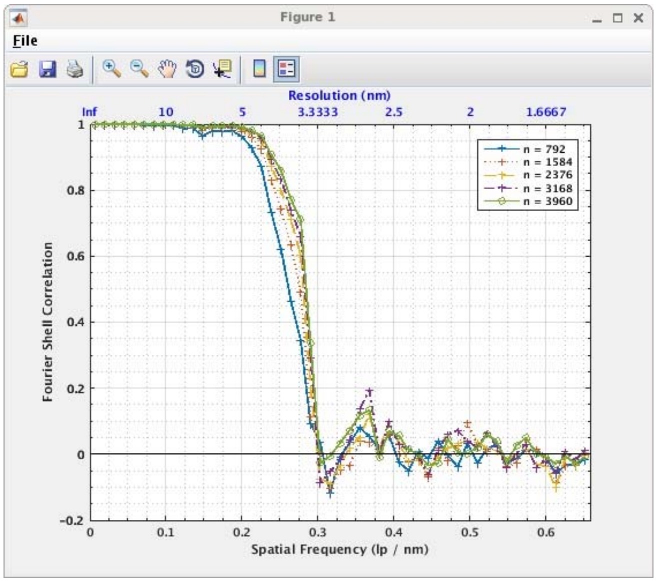
Fourier Shell Correlation



The Key Idea

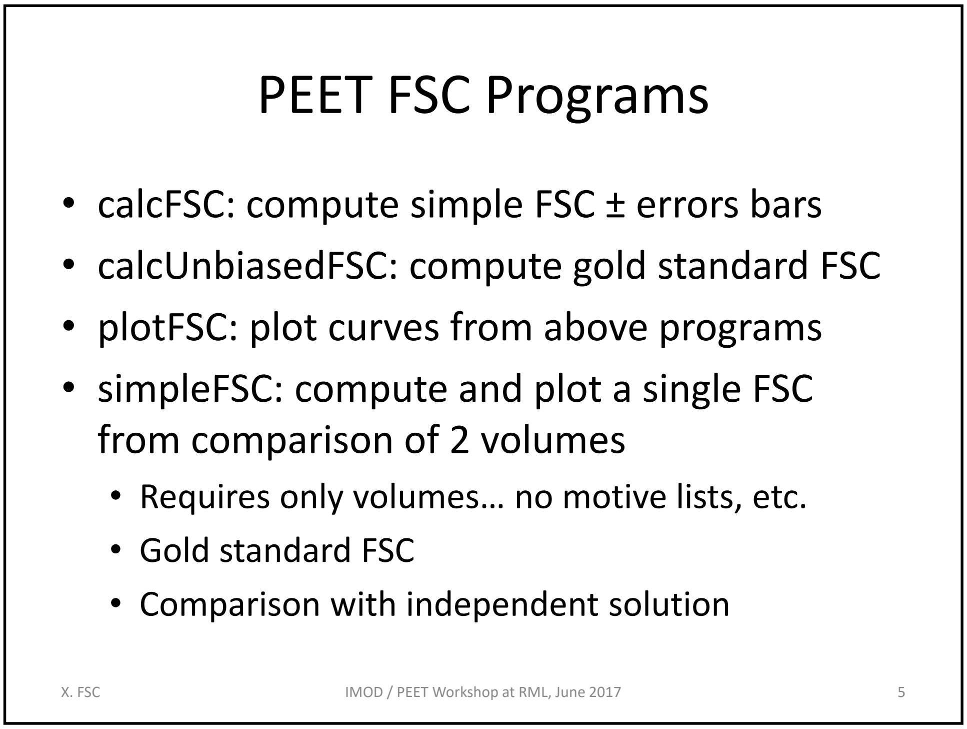
- Correlation between volumes can be written as a product in Fourier space
- The Fourier space product can be split into shells by radial frequency
- Compute plot of correlation versus frequency

Sample FSC Curve



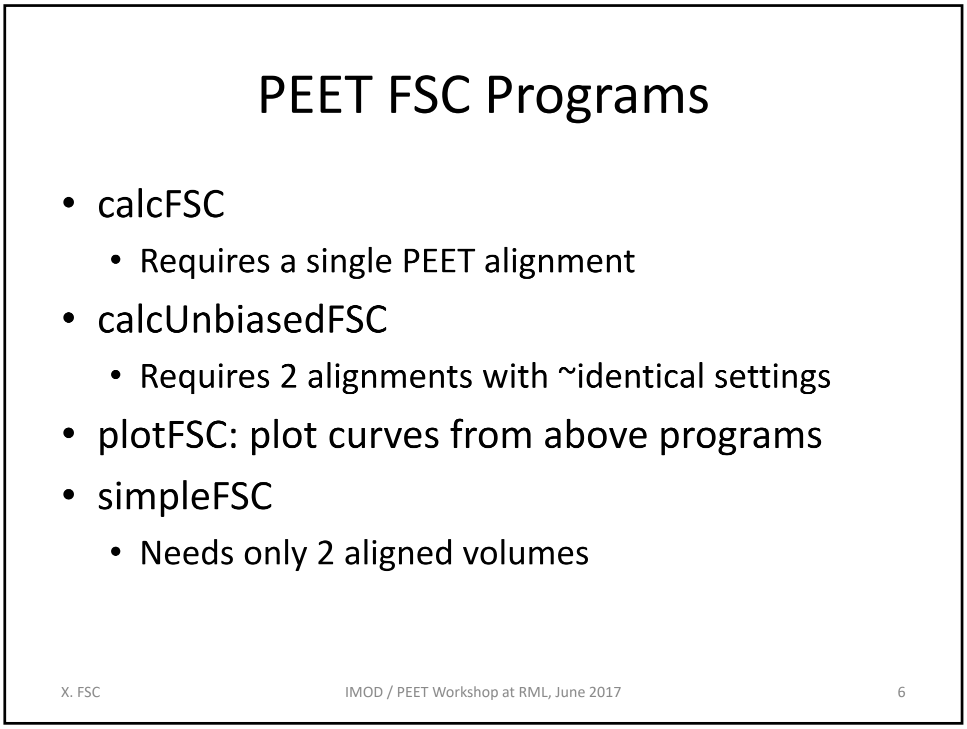
Types of FSC

- Odd / even
  - Simplest
  - Random sampling generalization allows error bars
  - Subject to bias: half datasets are not independent
- “Gold standard”
  - 2 independently aligned half datasets
  - Reduces possibility of bias
- Comparison with unrelated solution



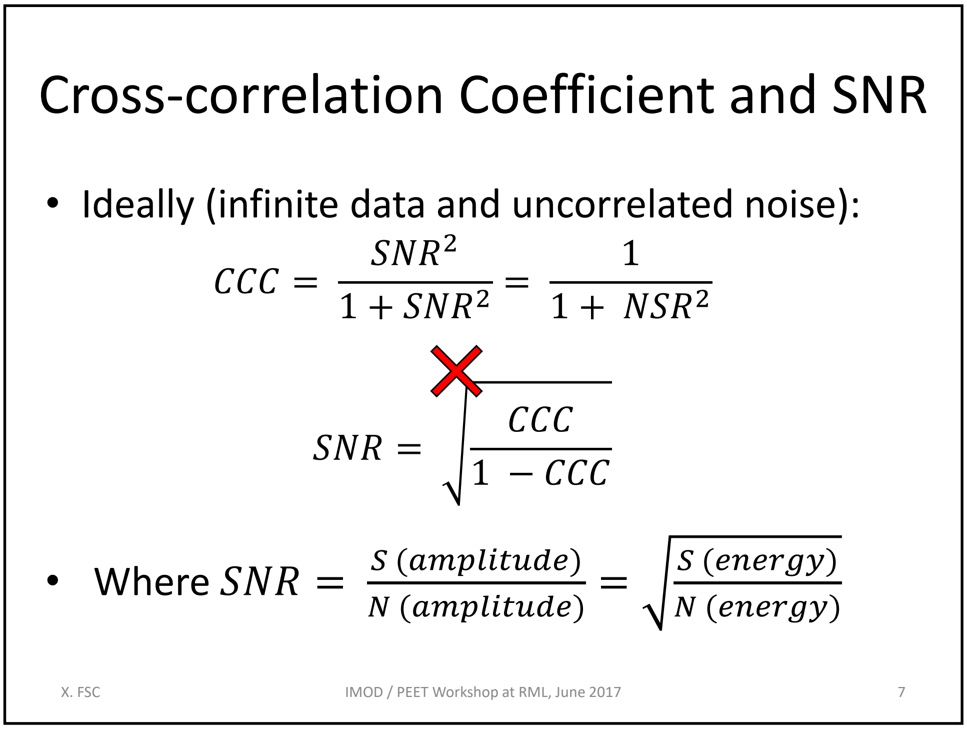
PEET FSC Programs

- calcFSC: compute simple FSC ± errors bars
- calcUnbiasedFSC: compute gold standard FSC
- plotFSC: plot curves from above programs
- simpleFSC: compute and plot a single FSC from comparison of 2 volumes
  - Requires only volumes… no motive lists, etc.
  - Gold standard FSC
  - Comparison with independent solution



PEET FSC Programs

- calcFSC
  - Requires a single PEET alignment
- calcUnbiasedFSC
  - Requires 2 alignments with ~identical settings
- plotFSC: plot curves from above programs
- simpleFSC
  - Needs only 2 aligned volumes



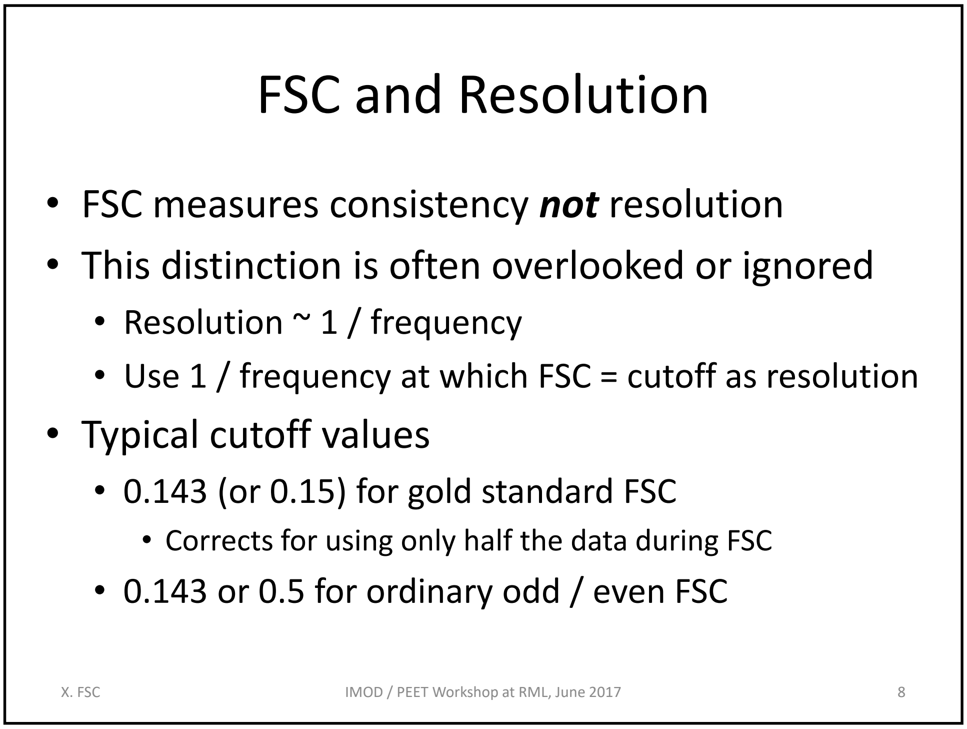
Cross-correlation Coefficient and SNR

- Ideally (infinite data and uncorrelated noise):

$$CCC = \frac{SNR^2}{1 + SNR^2} = \frac{1}{1 + \ NSR^2}$$

$$SNR = \cancel{\sqrt{\frac{CCC}{1 \ - CCC}}}$$

- Where $SNR = \dfrac{S\ (amplitude)}{N\ (amplitude)} = \sqrt{\dfrac{S\ (energy)}{N\ (energy)}}$



FSC and Resolution

- FSC measures consistency ***not*** resolution
- This distinction is often overlooked or ignored
  - Resolution ~ 1 / frequency
  - Use 1 / frequency at which FSC = cutoff as resolution
- Typical cutoff values
  - 0.143 (or 0.15) for gold standard FSC
    - Corrects for using only half the data during FSC
  - 0.143 or 0.5 for ordinary odd / even FSC

Questions?

THIS PAGE INTENTIONALLY LEFT BLANK

	June 30 (Fri)
8:00	Security
	Questions
	Fourier Shell Correlation
9:00	
	Clustering
10:00	John
	Coffee Break
	Clustering
11:00	
	Combining Data
	John
12:00	Lunch
1:00	
	Visualization with Demo First
	Cindi
2:00	
	Spikes
3:00	John
	Built in Extra Time if needed
4:00	
5:00	Final Remarks
6:00	
7:00	
8:00	
9:00	

THIS PAGE INTENTIONALLY LEFT BLANK

Fourier Shell Correlation

In this exercise, we'll explore the various options in PEET for creating and examining FSC curves.

CALCFSC / PLOTFSC

1) `cd $WORKSHOP_HOME/PEET_Labs/MT/PEET/firstSearch`

2) `calcFSC *.prm`

   The program should complete within a minute or 2. Wait for it to finish before proceeding. calcFSC generates an "odd / even" FSC from a previously completed PEET alignment. Outputs from calcFSC (and the other FSC programs to be discussed below) will be written to a number of **.txt* files. These can be useful if you wish to use a different plotting program or to analyze the data further.

3) plotFSC is one of the rare PEET commands that does not require any mandatory arguments. (PEETHelp is another). As a result, if you want to see what optional arguments are available, you'll need to refer to the man page. Do this now with `man plotFSC`

4) Run `plotFSC`. By default, both a linear Spatial Frequency axis at the bottom and a non-linear Resolution (1 / Spatial Frequency) axis at the top will be shown. Both will be adjusted automatically as you pan, zoom, resize, or maximize the plot. Try using the **Magnifying Glass** and **Hand** icons from the toolbar. Since the default graph is linear in spatial frequency, interpolation to find frequency / resolution is best done using the frequency axis in this mode.

Practice this by finding the frequency at which the FSC is 0.5. Zoom and pan until the points on either side of **0.5** in the green curve are clearly visible. Select the **data cursor** (the 3d icon from the right in the toolbar, which looks like a curve with a visible "+" and a note above it). Click on the points on either side of 0.5 with this tool, and you'll see that the corresponding FSC values are 0.5204 at a frequency of 0.2441 lp/nm and 0.3437 at 0.2653 lp/nm. Interpolating linearly, we see that an FSC of 0.5 would occur at 0.2465 lp/nm, corresponding to a resolution of 4.1 nm. Close the plot window when finished.

5) `calcFSC *.prm 3`

The optional "3" means that the calculation will be repeated 3 times with different pseudo-random assignment of particles to the 2 subsets at each repetition. This gives a measure of repeatability and will allow plotting error bars on the FSC curves. Naturally, the calculation will take 3 times as long. Note that 3 is the minimum number of repeats which permits computation of error bars. Typically, you should use 5 or more repeats for reliable estimation. Wait for the computation to complete.

6) `plotFSC`

Zoom in and you should see the error bars corresponding to 95% confidence intervals for the various points. Keep in mind that these error bars indicate only the variability due to sampling errors nad not absolute accuracy. Close the plot window when finished.

7)
```
plotFSC 1 0
```

The first argument says to label the plot as Figure 1 (must be an integer), and the 2nd argument says to plot versus resolution rather than frequency. Error bars are suppressed by default when plotting versus resolution, since they tend to be scaled inappropriately. There is an optional argument, which will allow you to manually turn them on if you insist, but you probably won't like the results! Notice that you can read off resolution at a given FSC value directly in this mode; don't expect the values to exactly match those found with a linear frequency axis. Close the plot window. For information on additional options, consult the plotFSC man page. calcSSNR is somewhat similar to calcFSC but computes Spectral signal-to-noise ratio rather than Fourier Shell Correlation.

CALCUNBIASEDFSC

Next, we'll explore how to compute a "gold standard" FSC, which eliminates one common source of bias, namely that all the data used by calcFSC had previously been aligned jointly against a common reference.

8)
```
cd ../../../BPV_-3/PEET/odd
```

To compute a gold standard FSC, you first split your particles into an even number (typically 2) of equal-sized subsets, which must be aligned separately using independent references. Normally, you would split the initial model into 2 (or more) new models to do this. In this example, I've taken a shortcut. We've already aligned the complete data set using random rotations for the initial motive list. I've taken the initial motive list (containing just the random initial rotations) from that run (*../firstSearch/bpv_MOTL_Tom1_Iter1.csv*), copied it to this directory as *initMOTL.csv*, and modified it by inserting alternating class numbers 1 and 2 in column 20. Examine *initMOTL.csv* (using cat, gedit, head, or less) to see these class assignments.

9) `etomo *.epe`

   Examine the settings for this project, and you'll see they're identical to those we previously used for the initial BPV firstSearch run except that under Advanced / Optional Features I've set Average only members of class to 1, and I've approximately halved the number of particles to be used for references and the final average. Recall that class 1 contains the odd numbered particles; since these are the only particles allowed to enter averages or new references, we'll effectively be using only the odd-numbered half of the data. All of the particles will still be aligned, and represented in motive lists, so this isn't the most efficient way to do things; this data set is small and runs quickly, however, so this isn't an issue. More typically, you would use createAlignedModel twice with Average only members of class set to split the data set into halves. I've specified an initial reference from the odd half of the data. The run has already been done and intermediate files deleted. Press **Open averages in 3dmod** to see what the final average from this half set looks like.

10) `cd ../even`, run `etomo *.epe`, and again press **Open averages in 3dmod** to view the corresponding settings and average from the even half set. Notice that the 2 averages are in different orientations since they were aligned to different starting references. We'll need to take care of that next. Close all the 3dmod and Etomo windows.

11) `cd ../alignOddAndEven`

12) `etomo *.epe`

13) Press **Run**

    The run will only take a few minutes to complete. While it's running, examine the various settings. We're going to align the average from the even half-set against that from the odd half-set. I've specified no reference refinement, so only tomogram 2 (the even average) should show non-zero translations and rotations in the output motive list. After the alignment completes, check that this is true by running `cat bpv_MOTL_Tom*Iter4.csv`.

For this alignment, we need a model with a single point in the center of the volume to use with each of the 2 averages. Since the averages are 52^3, I created a file, center.txt, containing "26 26 25.5" and then ran `point2model center.txt center.mod` to generate such a model. In the Etomo parameter settings, notice that I specified the number of tilt axes as "2 or more" so I could supply the binary wedge masks automatically generated by PEET during the odd and even alignment runs. In this case, there is little remaining missing data in Fourier space, so it would also have been reasonable to simply disable missing wedge compensation. I selected Save individual aligned particles in the Etomo settings, so we will generate mrc volumes corresponding to the odd and even aligned half-set averages for verification.

You may wonder why we're using the odd average as the reference / template rather than the larger reference generated at the end of the original odd alignment. That indeed would seem more appropriate; in practice, however, doing so typically results in small rotations translations to volume 1 in addition to the larger ones for volume 2. When the run completes, exit Etomo.

14) `3dmod aligned*.mrc`

Verify that the alignment appears to have worked correctly, bringing the two averages to a common orientation. Exit 3dmod when finished.

15) `cat bpv_MOTL_Tom2_Iter4.csv`

Make note of the corresponding Euler angles in columns **17**, **19**, and **18**, and translations in columns **11-13**. You'll see that the Euler angles are 113.413°, 34.918°, and -106.582°, and the translations are -1.35, 1.8, and 0.75. Next, we'll incorporate these rotations into the output motive list for the odd half-set, after backing up the original motive list first.

16) `cd ../even`

17) `cp bpv_MOTL_Tom1_Iter5.csv bpv_MOTL_Tom1_Iter5.csv.bkp`

As usual, we backup the original motive list before modifying it.

18)
```
modifyMotiveList bpv_MOTL_Tom1_Iter5.csv \
bpv_MOTL_Tom1_Iter5.csv.modified \
"113.413,34.918,-106.582" "-1.35,1.8,0.75" 0 1
```

Here, and subsequently, the “\” escape characters should be followed immediately by **Enter.** The optional “0 1” arguments say that we’re specifying the transform we wish to apply (rather than its inverse), and that we’re specifying Euler rather than Slicer angles. Note that in this case, we’re directly applying the angles found for volume 2 (the even average) to volume 1. Alternatively, we could equally well have applied the inverse of this transform to volume 2. It’s easy to get confused about whether an inverse transform is needed or not. When in doubt, do a no-search alignment with Save individual aligned particles to verify that things have gone as expected.

19)
```
cp bpv_MOTL_Tom1_Iter5.csv.modified \
bpv_MOTL_Tom1_Iter5.csv
```

(Once again, “\” should be followed immediately by **Enter**).

20)
```
cd ..
```

21) Run `calcUnbiasedFSC odd/*.prm even/*.prm`, wait for the calculation to complete, and then run `plotFSC` to display the gold standard FSC curves. Leave this figure window open for now.

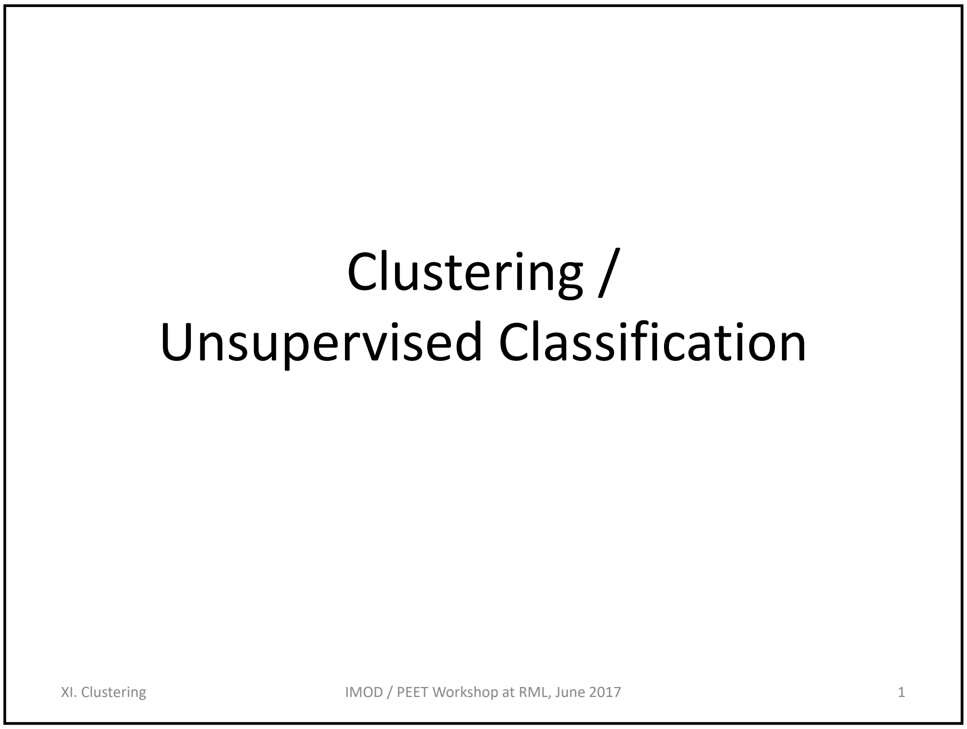
SIMPLEFSC

PEET also has a simpleFSC program, which will compute and plot an FSC curve from two corresponding volumes. For example, we could generate the gold standard FSC corresponding to using all available particles in step 20 using this program.

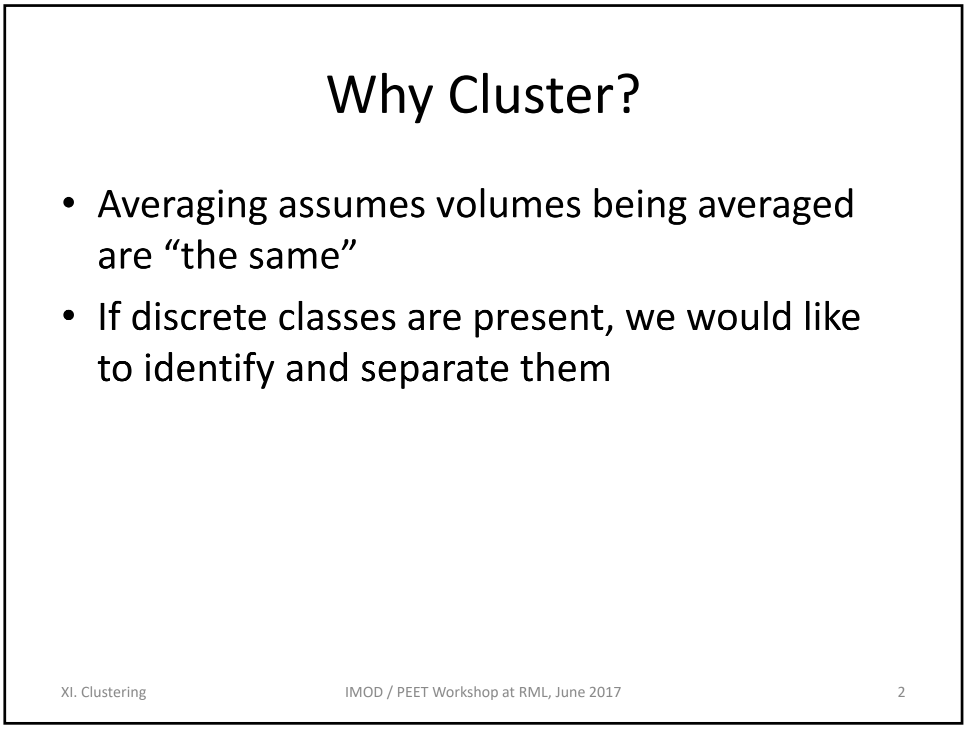
22)
```
cd alignOddAndEven
```

23)
```
simpleFSC aligned_tom1_P0001.mrc aligned_tom2_P0001.mrc
```

Notice that the results are identical to the green curve generated in step 20. simpleFSC is applicable whenever you wish to compare 2 volumes with common position and alignment, and is not restricted to computation of gold standard FSCs. For example, it can be used to compute the FSC between a subvolume average and a PDB model. Unlike calcFSC and calcUnbiasedFSC, it requires only the 2 volumes and no other information from a PEET alignment. Close both plot windows.

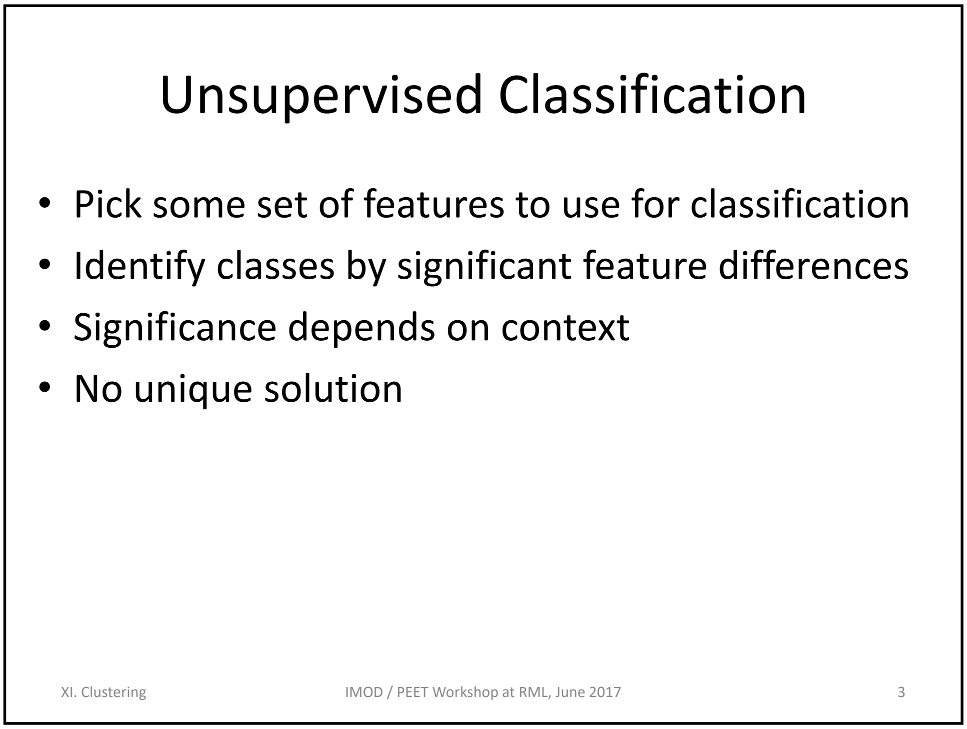


Clustering / Unsupervised Classification



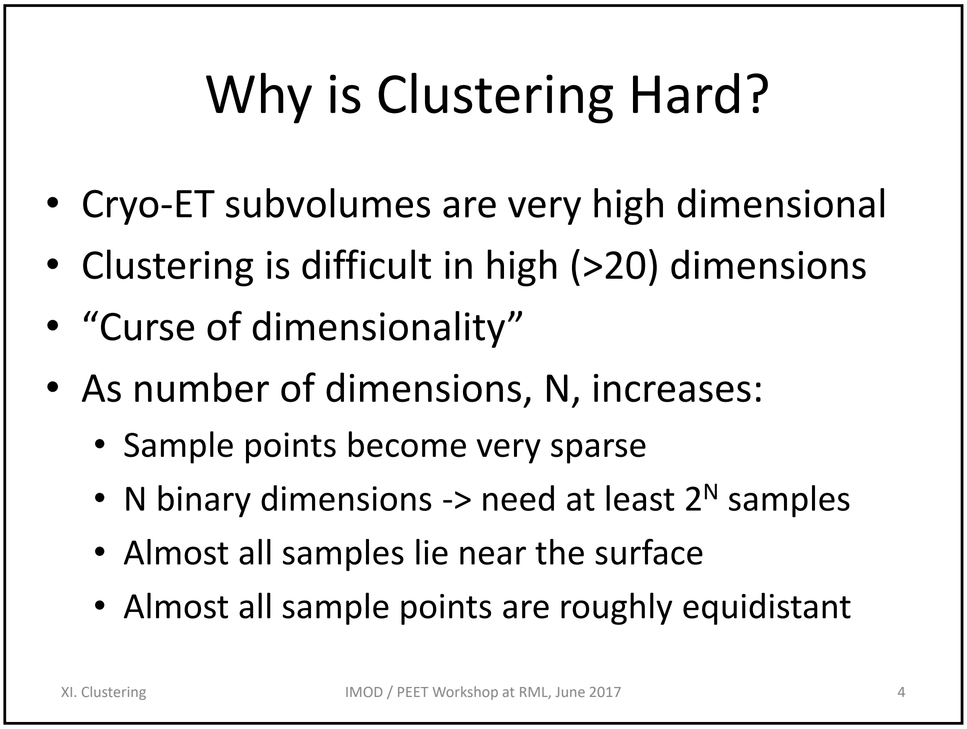
Why Cluster?

- Averaging assumes volumes being averaged are “the same”
- If discrete classes are present, we would like to identify and separate them



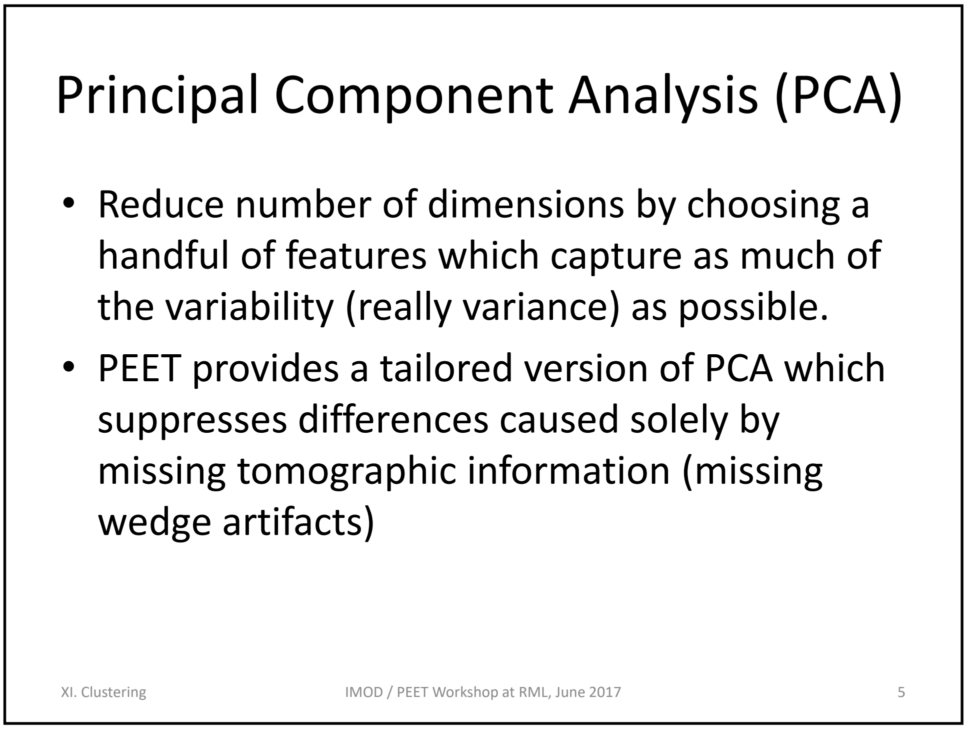
Unsupervised Classification

- Pick some set of features to use for classification
- Identify classes by significant feature differences
- Significance depends on context
- No unique solution



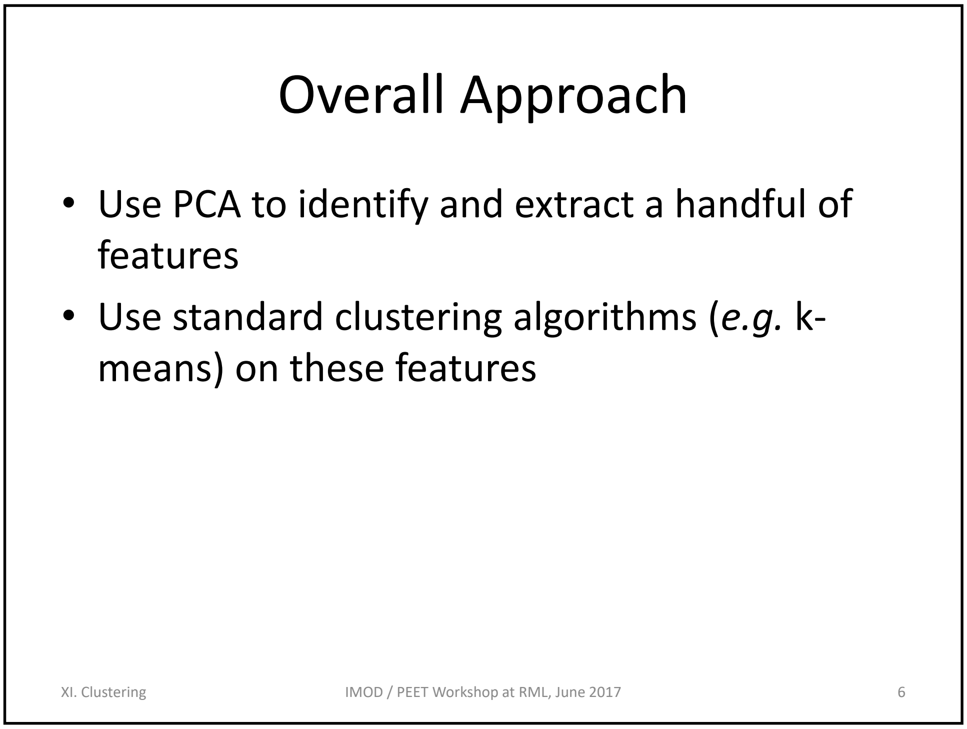
Why is Clustering Hard?

- Cryo-ET subvolumes are very high dimensional
- Clustering is difficult in high (>20) dimensions
- “Curse of dimensionality”
- As number of dimensions, N, increases:
  - Sample points become very sparse
  - N binary dimensions -> need at least 2^N samples
  - Almost all samples lie near the surface
  - Almost all sample points are roughly equidistant



Principal Component Analysis (PCA)

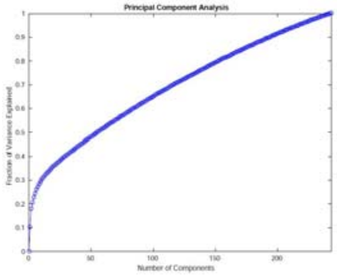
- Reduce number of dimensions by choosing a handful of features which capture as much of the variability (really variance) as possible.
- PEET provides a tailored version of PCA which suppresses differences caused solely by missing tomographic information (missing wedge artifacts)

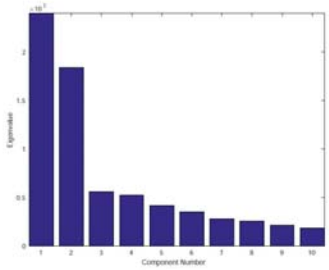


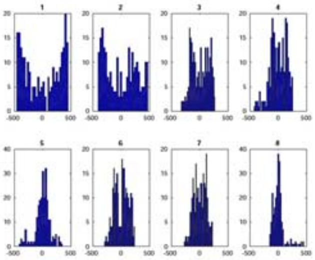
Overall Approach

- Use PCA to identify and extract a handful of features
- Use standard clustering algorithms (*e.g.* k-means) on these features

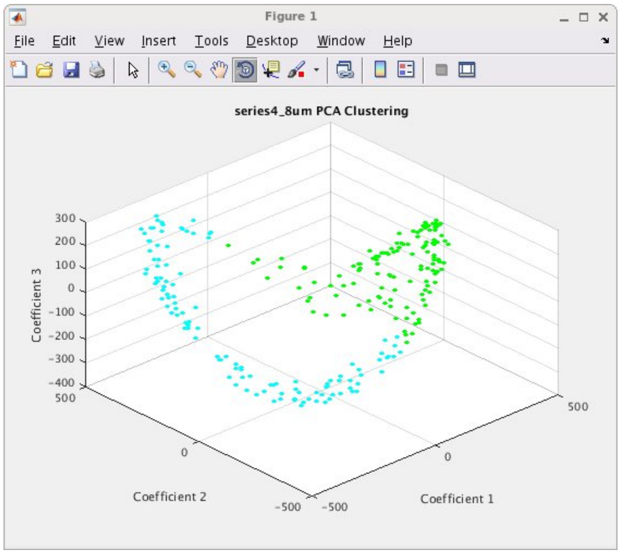
Sample PEET PCA Output







MT 1st Search Clustering

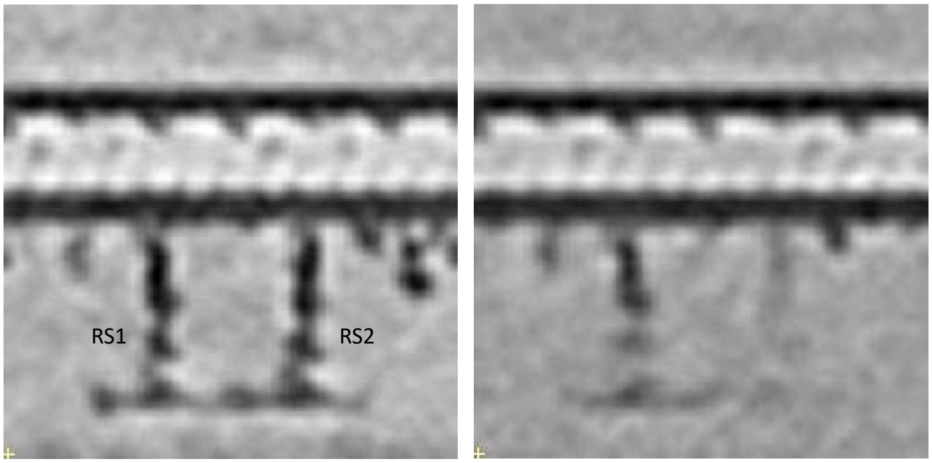


MT 1st Search Clustering Interpretation

- Statistically significant but probably not due to real sample heterogeneity
- Missing wedge artifacts affect alignment which affects clustering
- Variation with axial orientation

Chlamydomonas Axonemes

(from D. Nicastro and T. Heuser)



pWT

6E6

Chlamydomonas Axonemes

- RS2 density greatly reduced in 6E6 mutant
- Is this due to flexibility or heterogeneity?
  - Maybe all mutants have RS2, but configuration is variable
  - Maybe some mutants completely lack RS2
- We will explore this case in the lab exercise

Questions?

THIS PAGE INTENTIONALLY LEFT BLANK

Unsupervised Classification / Clustering

15 PROTOFILAMENT MICROTUBULE

PEET uses a version of Principal Component Analysis (PCA) adapted to minimize missing wedge artifacts followed by k-means clustering to detect and manage heterogeneity in averaged subvolumes. First let's examine the initial 15 protofilament microtubule alignment described in an earlier exercise. Recall that in this example we used axial randomization to suppress missing wedge artifacts; some missing wedge artifacts will still be present, however.

1) `cd $WORKSHOP_HOME/PEET_Labs/MT/PEET/firstSearch`

2) `pca *.prm 2 243 series4_8um_AvgVol_2P243.mrc`

   Principal component analysis is cpu and memory intensive. In this case, however, we're using binned data and only 243 particles, so the analysis will complete fairly quickly. Based on the resulting plots, it looks like the first 4 or 6 principal components will be useful as features for clustering. We'll go ahead and try using the first 6. Typically, you would try several combinations. In this case, the results are insensitive to the specific features chosen. Close the plot windows when finished examining them. The pca program automatically saves pdf copies of these plots for you.

3)

```
clusterPca *.prm pca243_series4_8um.mat 2 1:6
```

Generally, I recommend starting with a small number of features and clusters and working your way up as needed. Some users routinely use the first 20 features, although I consider this risky and prone to overestimating heterogeneity. Notice from the output in the terminal window that both AIC and BIC indicate significant improvement in scores as a result of clustering. Roughly speaking, and under assumptions which are seldom fully satisfied, the likelihood of getting a score improvement *S* by random chance is approximately *exp(-S/2).*

4) Use the rotate tool to view the cluster plot from different perspectives. While we have clustered in a 6-dimensional space, things have been projected down to the 3-dimensional space of the first 3 features (principal components) for visualization. You'll see that the data look somewhat like the outline of a saddle or a potato chip. While the AIC and BIC suggest that the clustering is highly non-random, this figure strongly suggest that we are looking at continuous variation of a single parameter rather than discrete clusters. Notice that we could arbitrarily rotate the boundary between the 2 classes around this saddle / potato chip and still get highly significant results. As discussed in the lecture, axial position around this shape likely corresponds to axial orientation in the original tomogram. This is a case where, despite our efforts to suppress them, we've wound up clustering according to residual missing wedge artifacts. It's advisable to always be aware of this possibility. Missing wedge artifacts are almost always present in the tomograms and are frequently dominant. There is no indication that the subvolumes themselves are actually heterogeneous in this instance. Close the plot window when finished.

MUTANT CHLAMYDOMONAS RS2

Recall from the lecture that the density of Radial Spoke 2 (RS2) was greatly attenuated in averages from the 6E6 mutant strain compared to wild type. Let’s analyze this case further to see whether this strong but incomplete reduction in density is due to conformational flexibility or because the majority of structures completely lack RS2.

5)
```
cd ../../../Axoneme/
```

6)
```
3dmod PEET/pWT/pWT_AvgVol_3P159.mrc \
PEET/6E6/6E6_AvgVol_3P162.mrc
```

where “\” should be followed immediately by **Enter**. Compare the two averages and notice that RS2 is almost completely missing in the case of 6E6, as previously described. Exit 3dmod when finished. Because this is a large, somewhat variable structure, we’ll restrict pca’s attention to RS2 by using a binary mask. An appropriate mask has already been created by Tom Heuser, and is in *RS2MaskLong.mrc*. Let’s apply it to the average so we can visualize the results and verify that it selects RS2.

7)
```
applyBinaryMask RS2MaskLong.mrc \
PEET/pWT/pWT_AvgVol_3P159.mrc masked_pWT.mrc
```

8)
```
3dmod masked_pWT.mrc
```

9) In the 3dmod info window, adjust the **Black** and **White** sliders to **200** and **255**, respectively, and page up and down to verify that the masked region includes most of RS2 and little else. Exit 3dmod when finished.

10)
```
cd PEET/6E6
```

The alignment has already been run and intermediate files removed.

11) `gedit 6E6.prm`, scroll to the end of the file, and uncomment (**remove the "#"**) from the line #**pcaFnParticleMask = '../../RS2MaskLong.mrc'**. Normally, you would have to add this line manually. This tells the pca program to use this mask. See the pca man page for additional details. See the imodmop man page for one way to create such a binary mask. Save the modified prm file and exit gedit.

12)
```
pca *.prm 3 162 6E6_AvgVol_3P162.mrc
```
Examine the figures after the program completes. Notice that the first principal component is significantly larger than any other, and that the first 6 principal components explain about 20% of the variance. We will use components 1-6 as features for clustering. Close all the figure windows.

13)
```
clusterPca *.prm pca162_6E6.mat 2 1:6
```
Here, we've asked clusterPca to split the data into 2 classes using features 1 through 6. The AIC and BIC scores indicate significant results. Notice that the majority of the points are in class 1, which is reasonably compact. Class 2 is spread out and probably contains several additional subclasses. Because it contains so few particles, however, we won't try to further subdivide it. Close the figure window.

14) After backing up the original, we copy the motive list created by clustering into place.
```
cp 6E6_MOTL_Tom1_Iter4.csv 6E6_MOTL_Tom1_Iter4.csv.orig
cp pca_6E6_MOTL_Tom1_Iter4.csv 6E6_MOTL_Tom1_Iter4.csv
```
This version will have the class numbers assigned during clustering in column 20 of the motive list. (Verify this if you like).

15)
```
etomo *.epe
```

16) On the **Run** tab, set **Average only members of classes** to **1** and press **Remake averages**. Wait for averaging to finish.

17) `mv 6E6_AvgVol_3P128.mrc class1_AvgVol_3P128.mrc`

Notice that the number of particles in the class averages matches the number reported by clusterPca. It's good practice to rename class averages to avoid confusing with the original averages. In fact, you should also take care that a class average does not accidentally overwrite a prior average… *i.e.* if they happen to contain the same number of particles. When saving results from multiple attempts at clustering, I recommend moving the class averages and other results you wish to preserve… *e.g.* figures, motive lists with class labels, etc… to a subdirectory. In this case, for example, I might create a subdirectory named "*2ClassesFeatures1-6*".

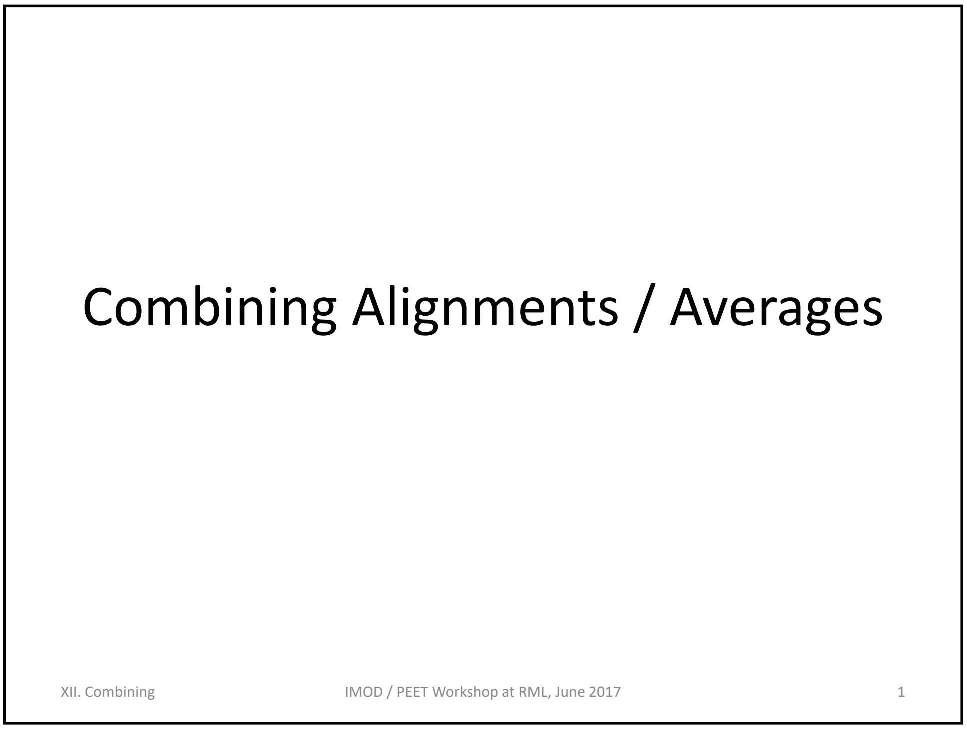
18) On the **Run** tab, set **Average only members of classes** to **2** and press **Remake averages**. Once again, wait for averaging to finish.

19) `mv 6E6_AvgVol_3P034.mrc class2_AvgVol_3P034.mrc`

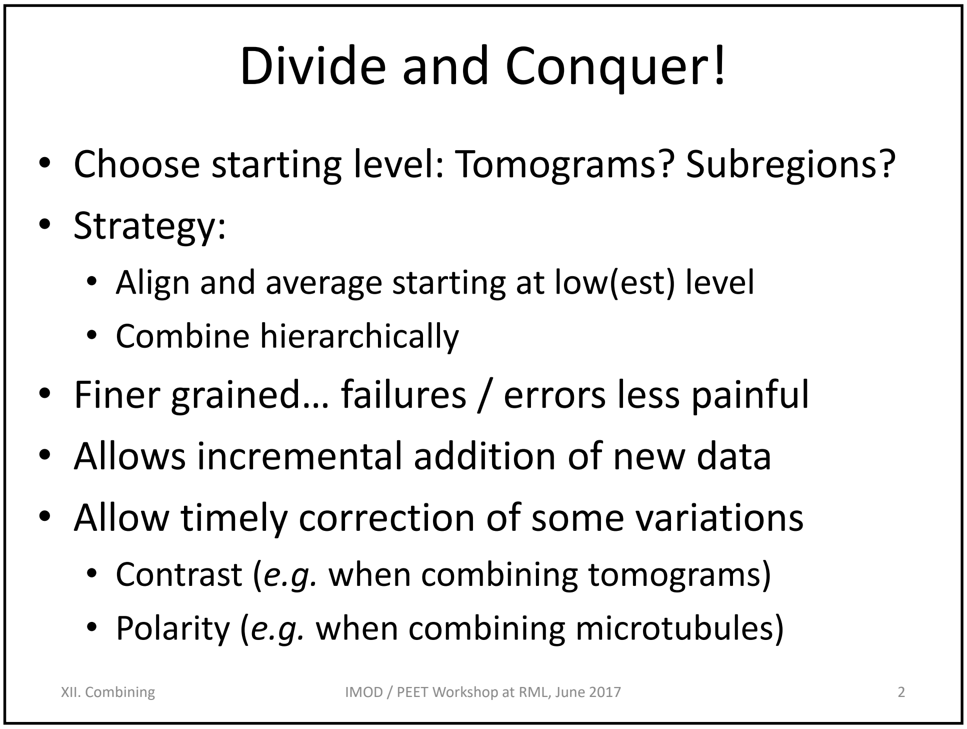
20) Clear the entry for **Average only members of classes** and exit Etomo. The prm file will be automatically saved on exit. The Average only members of classes setting, which corresponds to selectClassID in the prm file, is an example of a parameter which can lead to very confusing results if accidentally left in an unintended state!

21) `3dmod class*.mrc`

Notice that class 1, contains most of the particles and appears to be comprised of particles in which RS2 is completely absent. The minority class 2, on the other hand, consist of particles which all seem to contain RS2 in its normal configuration as it appears at full or near-full intensity. As mentioned above, class 2 may consist of particles with varying conformations. Because there are so few particles in this class, we will not attempt to analyze this heterogeneity further here. Exit 3dmod when finished.



Combining Alignments / Averages



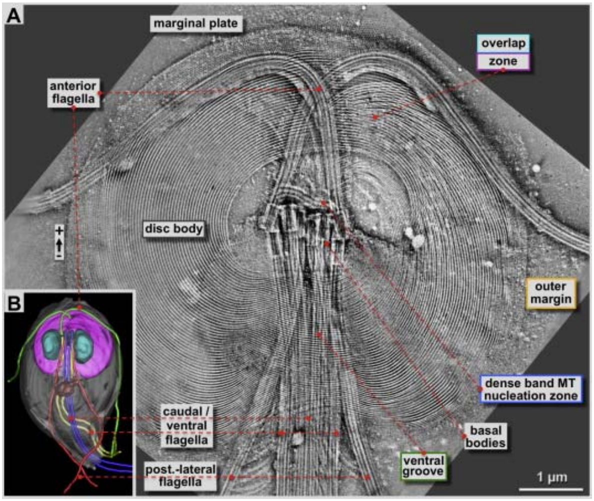
Divide and Conquer!

- Choose starting level: Tomograms? Subregions?
- Strategy:
  - Align and average starting at low(est) level
  - Combine hierarchically
- Finer grained… failures / errors less painful
- Allows incremental addition of new data
- Allow timely correction of some variations
  - Contrast (*e.g.* when combining tomograms)
  - Polarity (*e.g.* when combining microtubules)

Few or No New Tools Needed!

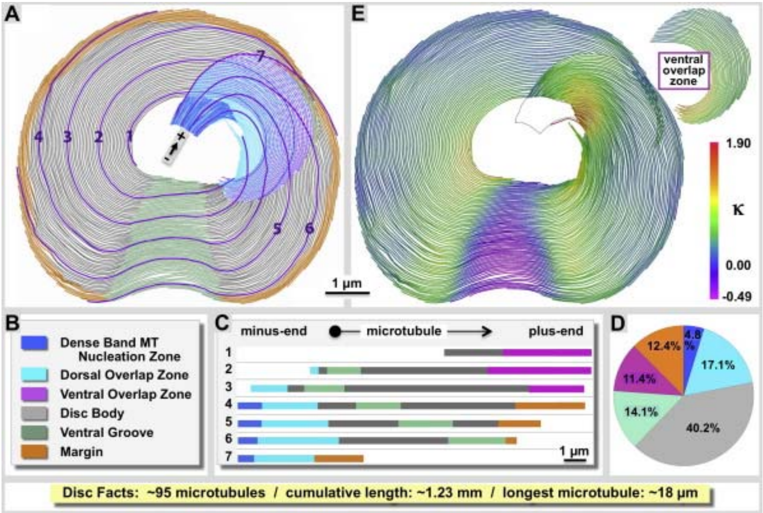
- Simple searches to find common alignment
  - *E.g.* as in gold standard FSC exercise
  - Can extend to multiple averages at once
- modifyMotiveList
- createAlignedModel
- Symmetrization is a type of combining
- May need IMOD to adjust tomogram contrast

Giardia Ventral Disk Example



Brown et al (2015) JSB **94**:38-48

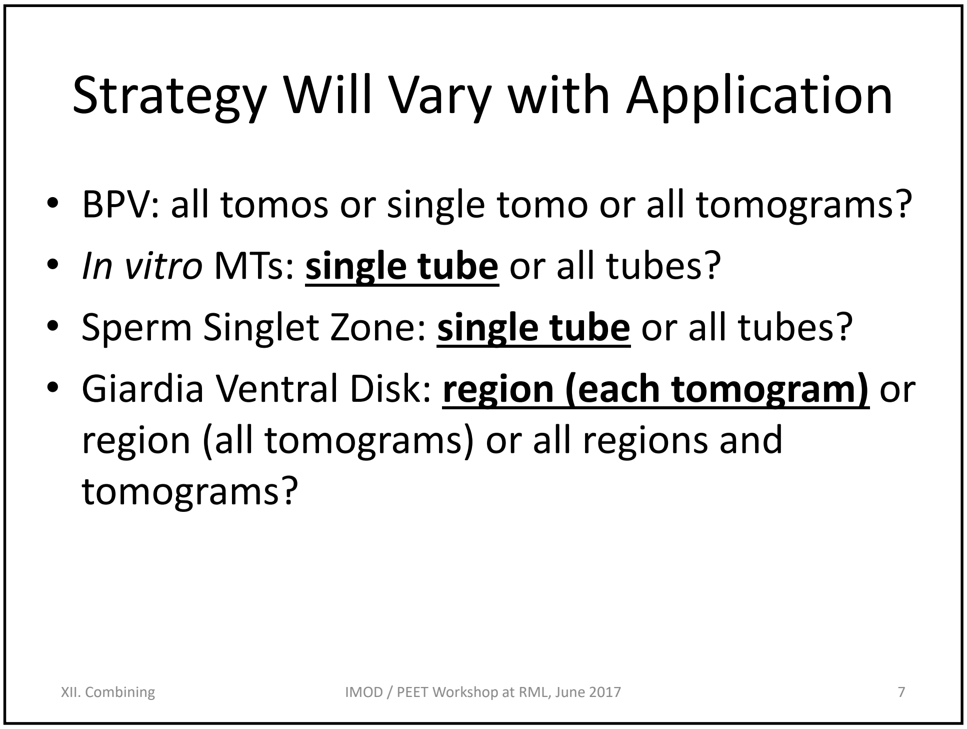
Giardia Ventral Disk Example



Brown et al (2015) JSB **94**:38-48

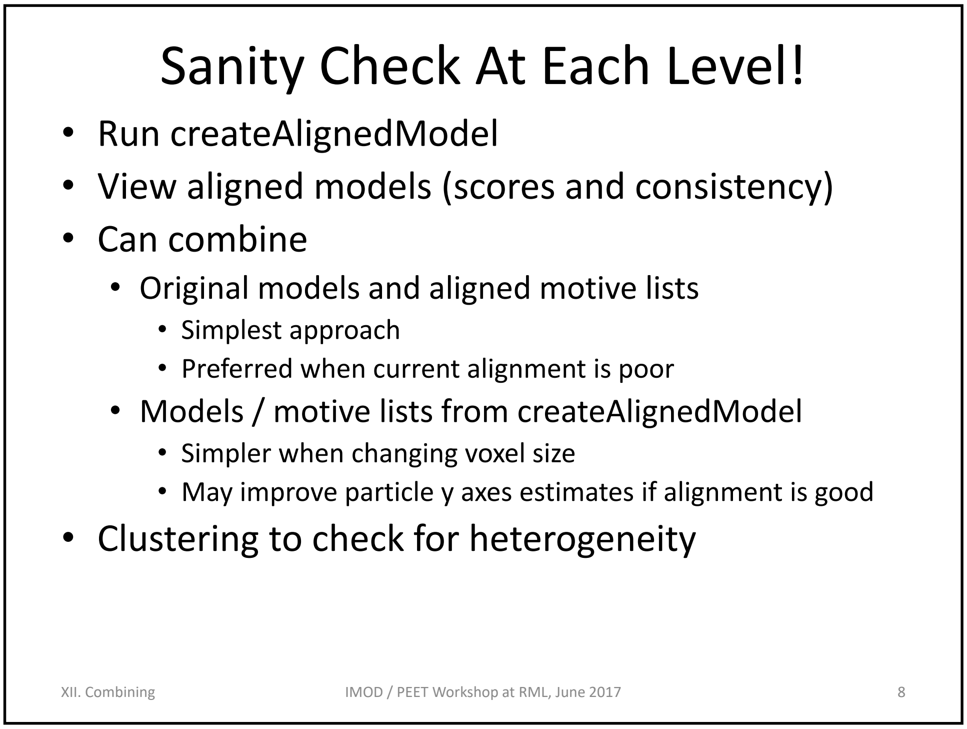
Alignment Hierarchy

- Overall structure (for difference map)
  - Region (multiple tomograms)
    - Region (Individual tomogram)
      - Individual microtubule
        - Region (Individual microtubule)
- Check / correct consistency at “natural” level
  - Contrast (tomogram)
  - Voxel size (tomogram)
  - Imaging conditions (tomogram)
  - Polarity (microtubule)
  - Protofilament number (microtubule)



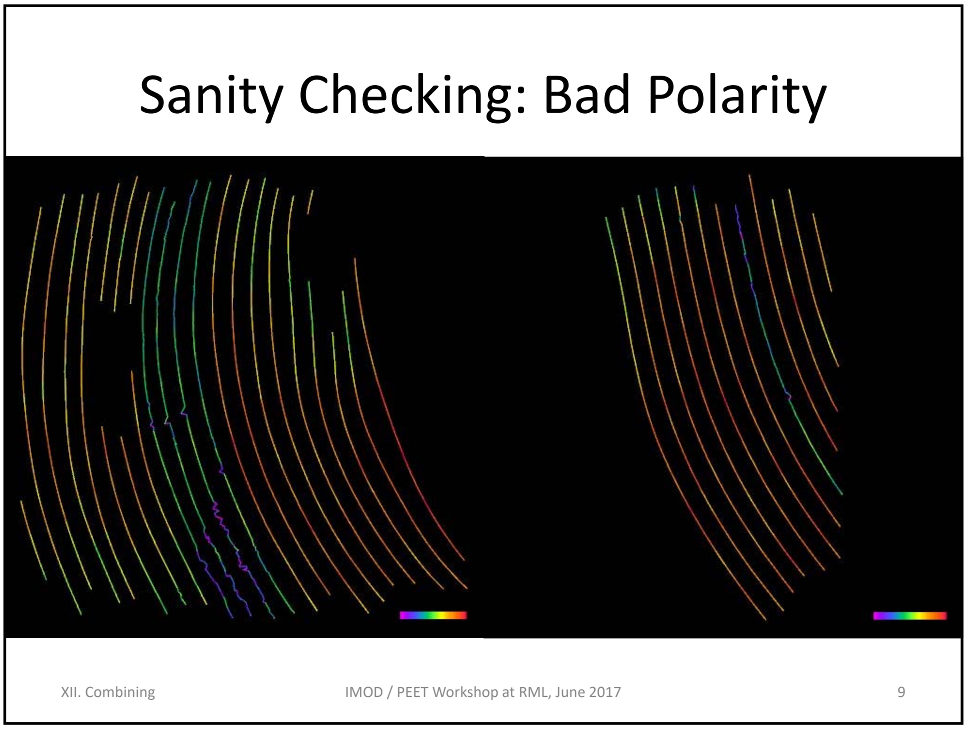
Strategy Will Vary with Application

- BPV: all tomos or single tomo or all tomograms?
- *In vitro* MTs: **single tube** or all tubes?
- Sperm Singlet Zone: **single tube** or all tubes?
- Giardia Ventral Disk: **region (each tomogram)** or region (all tomograms) or all regions and tomograms?

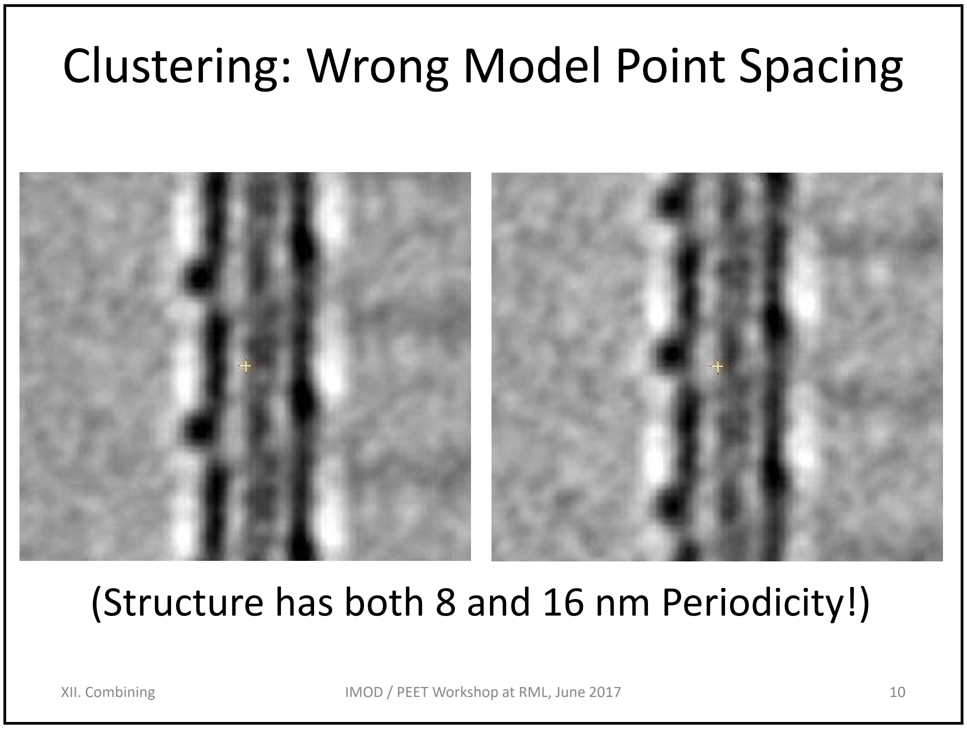


Sanity Check At Each Level!

- Run createAlignedModel
- View aligned models (scores and consistency)
- Can combine
  - Original models and aligned motive lists
    - Simplest approach
    - Preferred when current alignment is poor
  - Models / motive lists from createAlignedModel
    - Simpler when changing voxel size
    - May improve particle y axes estimates if alignment is good
- Clustering to check for heterogeneity



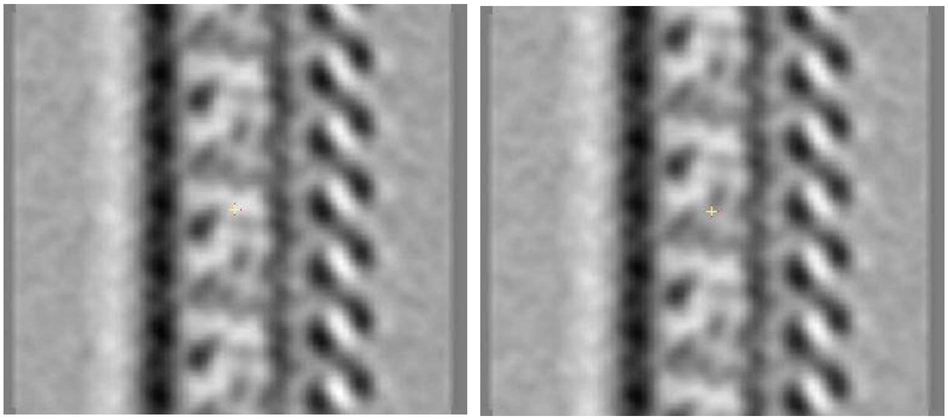
Sanity Checking: Bad Polarity



Clustering: Wrong Model Point Spacing

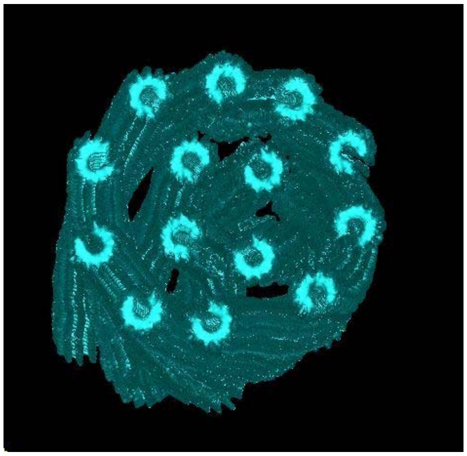
(Structure has both 8 and 16 nm Periodicity!)

Clustering: Wrong Model Point Spacing

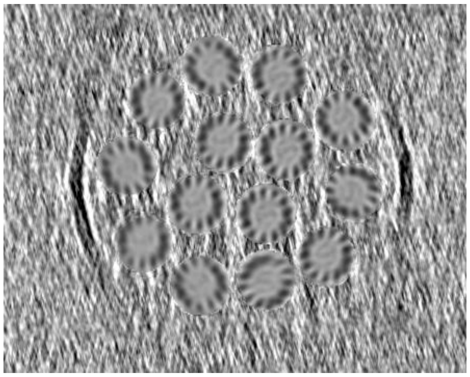


(Structure has both 8 and 16 nm Periodicity!)

Visualization: cloneModel / cloneVolume



Created with cloneModel



Created with cloneVolume

Questions?

XII. Combining IMOD / PEET Workshop at RML, June 2017 13

Visualization

We've used createAlignedModel fairly extensively… for sanity checking, changing voxel size, and symmetrization. It's also used in the same manner when combining results from multiple alignments. However, we have not yet made use of the "Summary" csv file(s) it generates. Here we'll use these file(s) for visualization in conjunction with IMOD programs clonevolume and clonemodel. This can be very helpful in analyzing and understanding your average as well as in preparing figures for publication.

USING CLONEVOLUME

1)
```
cd $WORKSHOP_HOME/PEET_Labs/MT/PEET/full15Fold
```

2)
```
createAlignedModel *.prm
```
Notice from the output that a "Summary" file (*series4-8um_PtsAdded_Tom*_Iter3_Summary.csv*) is generated for each volume.

3)
```
clonevolume -into ../../series4-8um-cor.rec \
-at series4-8um_PtsAdded_Tom1_Iter3_Summary.csv \
full15Fold_AvgVol_3P2528.mrc clonedVolume.mrc
```
where "\" should be followed immediately by **Enter**. The summary file is passed to clonevolume following the "-at" option. It gives the positions and orientations at which each aligned subvolume should be placed. The "-into" option is followed by the name of the volume into which we will be cloning. The final 2 arguments are the name of an mrc file containing the averaged subvolume to be cloned and the desired name for the output file. Several other options are available. Consult the clonevolume man page for more details.

4)
```
3dmod ../../series4-8um-cor.rec clonedVolume.mrc
```
Compare the original and cloned volume at various slice heights and, if you wish, from various perspectives. Exit 3dmod when finished.

Cloning can also be used to generate models containing correctly oriented, aligned isosurfaces representing a subvolume average. This is particularly useful for complex structures containing multiple types of components. We will illustrate a simple, single component case, but the concepts are the same. For complex structures, you would first generate separate models for each component and then join them with IMOD's joinmodel.

5)
```
3dmod full15Fold_AvgVol_3P2528.mrc
```

6) Press **Shift+U** or **Image / Isosurface** to display an isosurface in the Model View.

7) In the resulting Isosurface View dialog
   a. uncheck **View Bounding Box**
   b. If desired, adjust the Isosurface **Threshold** slider to get the desired appearance.
   c. Press **Save Object**

8) In the main 3dmod window, press **File / Save Model As**, save the model as *isosurface.mod* and exit 3dmod.

9)
```
clonemodel -x 550,650 -y 1000,1100 -z 0,100 \
-at series4-8um_PtsAdded_Tom1_Iter3_Summary.csv \
isosurface.mod clonedModel.mod
```
(where each \ is immediately followed by **Enter**). Unlike clonevolume, clonemodel always clones into a new model, so there is no "-into" option. The resulting models can be easily combined using IMOD's joinmodel program. We've specified a limited region of the volume to be cloned with the x, y, and z options. Cloning the entire volume is quite possible, but the resulting isosurface models can have so many facets that they're extremely slow to display and rotate, even with reasonably high-performance graphics cards.

10) Use `3dmodv  clonedModel.mod` to examine the results. Exit 3dmodv when finished.

COLORING SURFACES AND PREPARING STILL IMAGES

In this portion, we will be painting an isosurface to distinguish a single pentamer and its neighbors from others on the BPV particle. We will use clonemodel to generate a model with a few pentamers highlighted. We will then load the model on the tomogram for visualization.

11)
```
cd ../../../BPV_-3/PEET/firstSearch
3dmod bpv_AvgVol_4P132.mrc pentamers.mod
```
Open an isosurface (**Shift+U**) and turn off **View bounding box**. Open Slicer (\), and turn on **centering** (box within a box icon). From the Model View window, select **Edit / Controls** and check **Link to top Slicer angles**. From the 3dmod info window go to **Model** mode and select **Edit / Object / Type**. Change **Sphere radius for points** to **4**.

12) Rotate your virus particle so that a pentamer is centered in the Model View. From the Isosurface window, select **Paint Obj: 1**. This will now color the portion of the isosurface surrounded by the model spheres to green, the current color of Object 1. To change the color of Object 1, go to the Model View **Edit / Objects** and use the slider at the top to go to **Object 1**. Click on **Line Color** in the menu below and use the **Red**, **Green**, and **Blue** sliders to change the center of the pentamer to any color you want. From the 3dmod info window choose **File / Save Model As** and save it with a model name of *pentamers-color.mod*.

13) From the 3dmod info window, select **Edit / Fine Grain**. Now, make a new contour (hotkey **N**) and make sure you are in **Contour 2**. In the Model View window, right-click on a subunit adjacent to the central, colored one and the Slicer window should now move to that position. Place a new model point (middle-click) in the Slicer window and then right-click in Slicer until you have the point centered over this subunit in the isosurface. The selected subunit should now be the same color as the center of the pentamer. To change the color which will be used for this and the other subunits surrounding the central one, go to the **Fine Grain** dialogue and select **Contour** in the **Edit** box. Then, under **Line Color**, press **Set** and change the **Red**, **Green**, and **Blue** sliders to a different color. Press **Done** on the color changer when finished.

    Repeat this process of right-clicking in the Model View on other subunits that surround the center of the pentamer and adding a well-centered point in Slicer. You should end up with the subunit at the center of the pentamer one color and the surrounding neighbors a different color. The 5 surrounding subunits should be in Contour 2, while the central one should be in Contour 1.

14) In the Isosurface window, uncheck **View user model**. Now you can see the painted model without interference from the spheres. Adjust the **Threshold** of the isosurface to your liking and press **Save Object**. Close the Isosurface window. In Model View, you should now see a 'blue' BPV particle with your painted pentamers and the model points as spheres. To turn off the spheres, at the top of the **Objects** window, make sure only **2** is checked. Then change the color of Object 2 by selecting **Object 2** with the slider and go to **Line Color**. Adjust the **Red**, **Green**, and **Blue** sliders to a color you like.. Save your model from the 3dmod info window by selecting **File / Save Model**. Close all 3dmod windows.

15) `createAlignedModel *.prm`

16)

```
clonemodel -x 473,549 -y 339,420 -z 33,83 -at \
bpv_bin2_Tom1_Iter4_Summary.csv pentamers-color.mod \
clonebpv.mod
```

17)

```
3dmod ../../bpv_bin2.rec.nad clonebpv.mod
```

Isosurfaces—even those saved as an object—are not displayed in the ZaP or Slicer windows, so you need to visualize them in the Model View window with hotkey **v**. You should see 2 BPV particles with the painted pentamer in various orientations. As noted before, too many isosurfaces will cause your computer to run very slowly and possibly even hang or crash. From Model View, open **Edit / Controls**. Reset **X**, **Y**, and **Z** angles to **zero**.

PREPARING IMAGES FOR PUBLICATION

Images captured from IMOD windows are limited by screen resolution. We will demonstrate how to create higher resolution images suitable for publication as well as smaller images suitable for movie-making.

18) Minimize the Model View window. When preparing still images for publication, it is best to save your images in their own directory. From the 3dmod info window, select **File / Set Snap Dir**. In the resulting file chooser, make a new folder called *Stills* and press **Choose**.

19) Open **Edit / Scale Bar** and uncheck **Color ramp**. Notice at the bottom of the scale bar dialogue that the scale bar is in pixels. This model lacks pixel size information. The tomogram has the appropriate pixel size information (1.52 nm). Go to **Edit / Model / Header** and press **Set Pixel Size from Image**. Then press **Done**. Save your model with the hotkey **s**. The Scale bar dialogue now shows the length of the bar in nm for both the ZaP and Model View windows.

20) In the ZaP window, zoom to **2**, resize the window as desired, and center a group of particles so that you have a nice image such as you might want to use for publication. Notice the scale bar value changes dynamically as you zoom in and out. Left-click anywhere in the image. You should see a yellow cross where you clicked. To turn off the cross, press **Shift+T**. To save a low-resolution image suitable for presentations, select the ZaP window and press **Ctrl+S**. This saves a tif image (*zap000.tif*) in the *Stills* directory. Confirmation of this will be displayed in the 3dmod info window. Similarly, Shift+S saves a jpeg image. I often rename the file immediately to something more meaningful and often include the scale bar size in the file name.

```
mv Stills/zap000.tif Stills/BPV-1_50nm.tif
```

For this step and subsequent steps, substitute the correct value for your own scale bar if it's something other than 50 nm.

21) Next, we will create larger images suitable for publications using montaging. In the 3dmod info window, select **File / Movie/Montage**. This is the 3dmod Movie window. Check **Montage** by **2** in **Zap montage snapshots** near the bottom of the dialog box. Select the ZaP window and press **Ctrl+S**. You will see 4 images flash by as a seamless montaged image is created. Rename the file with:

```
mv Stills/zap000.tif Stills/BPV-2_50nm.tif
```
.

22) Restore the Model View window and press **Ctrl+S**. A message will print in the terminal window that your image (*modv0000.tif*) was saved in the *Stills* directory. Rename using

```
mv Stills/modv0000.tif Stills/model-1_50nm.tif
```
.

Model View images will save in the same directory (*Stills*) you already chose for the ZaP images; you can move or save them to a different directory if you so choose.

23) From the Model View window, select **File / Movie/Montage**. This is called the 3dmodv Movie window. Notice that this window is different than the 3dmod Movie window accessed from the 3dmod info window. Near the bottom of the window, change to **Montage** and press **Make**. Again, you will see a series of 4 images flash on the screen as the montage is created. Rename the file with

```
mv Stills/modv0000.tif Stills/model-2_50nm.tif
```

24) To see the size of the files, type `ls -l -h Stills/*.tif`. The –h option says to list file sizes in human readable format. Notice how the first images are considerably smaller than the second images. The first files are appropriate for presentations and the larger files are appropriate for scaling to generate high quality images for publication. You can also make smaller files by using a different format such as png, especially for model images.

MAKING A SERIES OF IMAGES FOR A MOVIE

25) Now you are going to save a set of files for a simple movie showing different slices through the tomogram. Select **File / Set Snap Dir** and create a folder named *ZaPs*. Go to the ZaP window and determine where the main layer of the BPV particle starts and stops (~ slices **36-88**). Resize the ZaP window so that it is about 5x5 inches on the screen. From the **3dmod Movie** window, change the **Start** and **End** sliders to the numbers you just determined. Make sure that **None** is selected under **Snapshot** and uncheck **Montage**. You can middle- or right-click in the ZaP window to preview what your movie will look like. Once you are satisfied, change the **Snapshot** from None to **JPEG**. Middle-click in the ZaP window. You will see the file names *zap000-zapnnn.jpg* displayed in the 3dmod info window. When all the images have been saved, be sure to **change Snapshot** back to **None** or you may accidentally write more files.

26) Switch to the Model View window and change back from **Montage** to **Movie** in the **3dmodv Movie** window. Switch the **Save as** selection from TIFFs to **PNGs**. Go to **File / Set Snap Dir** and make a new folder called *model*.

27) For more complicated movies, the Movie Sequence window lets you save information about a series of movie segments, and record them all in sequence. This is particularly useful if there is any chance that you might need to remake the movie later. A movie segment is just the continuous set of operations that happen when you press Make in the 3dmodv Movie window. Press the **Sequence** button to open this window. We will now make a movie in the Model View window where we slice down through the tomogram image without out the isosurfaces present, back up, then down again, this time with the isosurfaces visible. We will then remove the image and rotate the isosurfaces a full 360° around the Y axis.

    Go to **Edit / Image** and check **View Z image** and press **Use 3dmod Black / White**. In the **Controls** window, be sure that all of the angles are **0**. Go to **Edit / Objects** and turn off all objects by unchecking them. In the **Image** window change the **Z** slider to **81**. On the **3dmodv Movie** window, press **Set Start**. Now change the **Z** slider to **31** and press **Set End**. Change the # **of movie frames** to **25**. Press **Make**. In the **Movie Sequence** window, press **Add After** to save this as the first segment. In the **Movie Sequence** table, **Label** this as **Movie Down (81-31)**.

    Now, we need to reverse this process by pressing **Set Start**. Then change the **Z** slider to **81** and press **Set End**. Press **Make** followed by **Add After**. **Label** as **Movie Up**.

    Let’s turn on the isosurfaces. In the **Objects** window, check **Objects 2 and 5**, then press **Set Start**. Move the **Z** slider to **31** and press **Set End**. Press **Make** and **Add After**. **Label** as **Movie Down with isosurface**.

Uncheck **View Z image** and press **Full 360 Y**. Change # **of movie frames** to **60**. Press **Make** and **Add After**. **Label** as **Full 360 Y** (this should happen automatically).

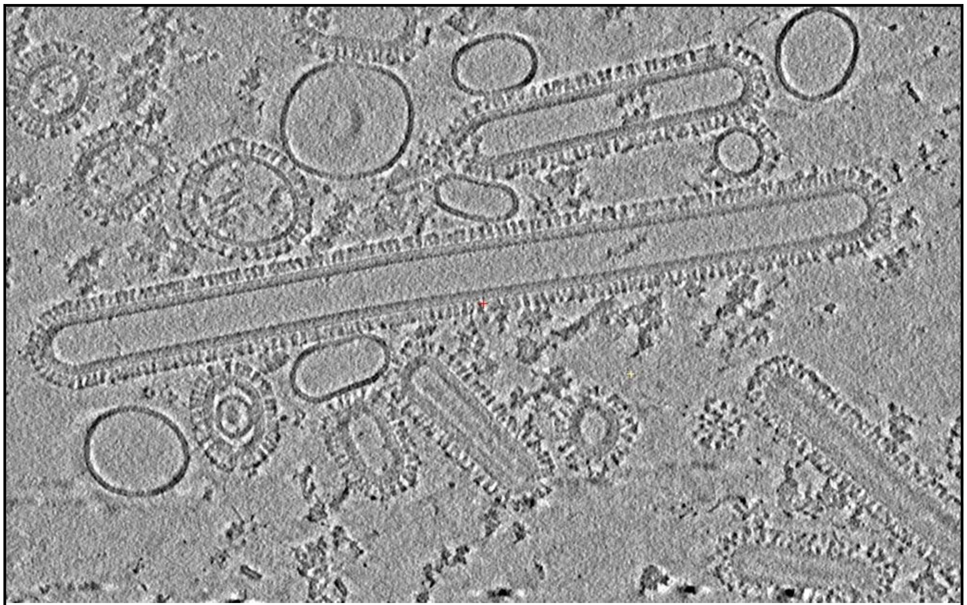
28) Notice that in most cases you need to define the end of the previous segment as your new start by pressing Set Start; this was not necessary in the case of Full 360 Y, which takes care of this automatically. Additionally, meaningful labels will help you remember what you did.

29) From the **Movie Sequence** window, press **Run All**. This is a preview of your movie sequence in order. You can change # **Frames** for any of the segments now, and then press **Run All** to see how the appearance changes. Naturally, the more frames you have, the larger the movie. Once you are pleased with your movie sequence, from the **3dmodv Movie** window, check the **Write files** box. This time, when you press **Run All**, the files will be written to the *model* folder. This image series can now be loaded into your favorite movie making software.

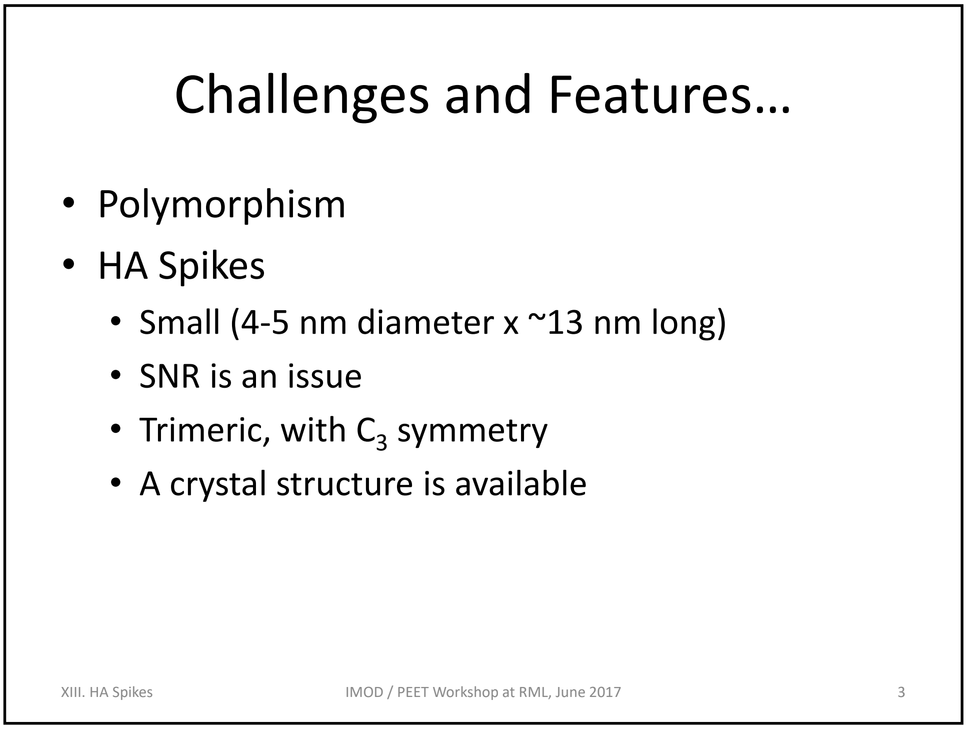
30) To save the sequence, press the **Save** button in the **Movie Sequence** window with the file name *ModelSequence.txt*. You can reload this file later, modify if desired, and remake the movie. Close all 3dmod windows and save your model when asked.

Influenza Hemagglutinin Spikes

A Very Challenging Example… Still in Progress!
(Data and Results Courtesy of Petr Chlanda)

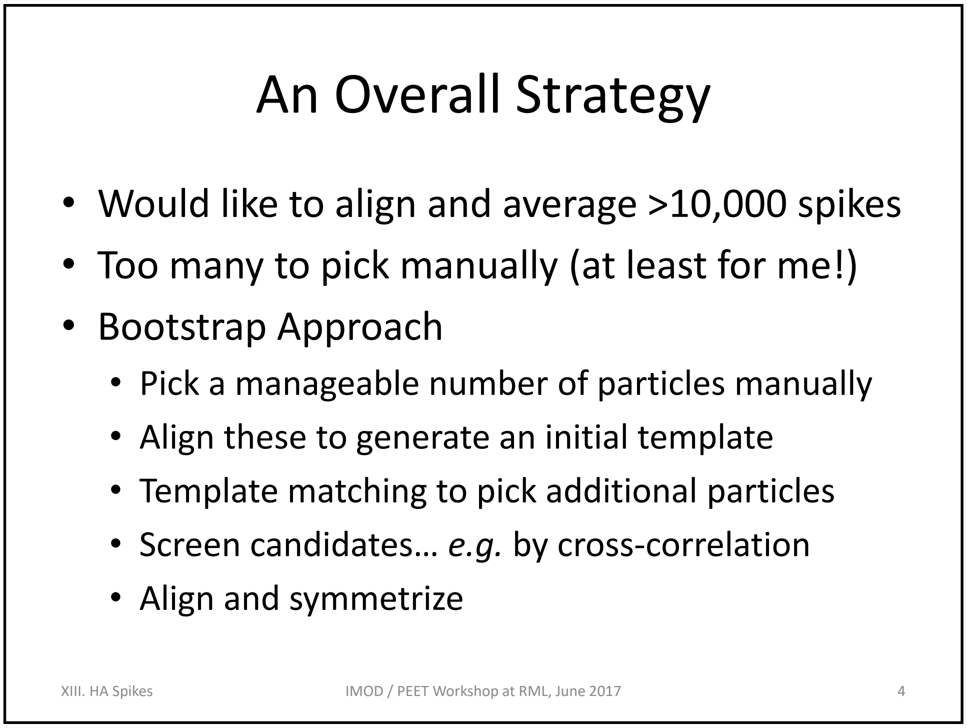


Sample Data (binned 2X, NAD filtered, cropped)



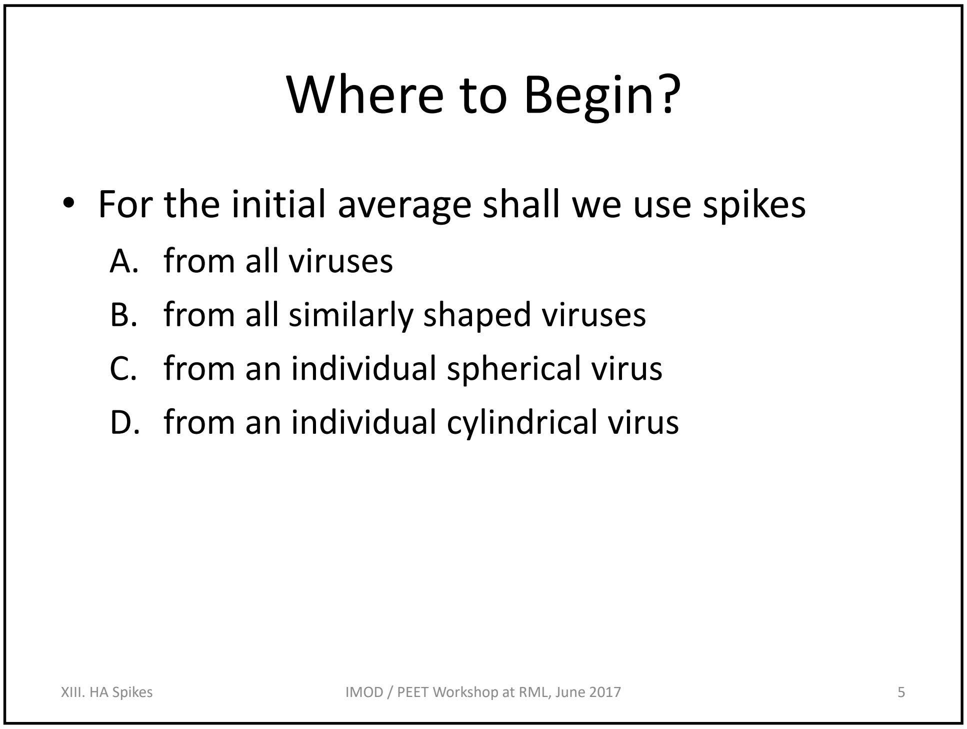
Challenges and Features…

- Polymorphism
- HA Spikes
  - Small (4-5 nm diameter x ~13 nm long)
  - SNR is an issue
  - Trimeric, with C_3 symmetry
  - A crystal structure is available



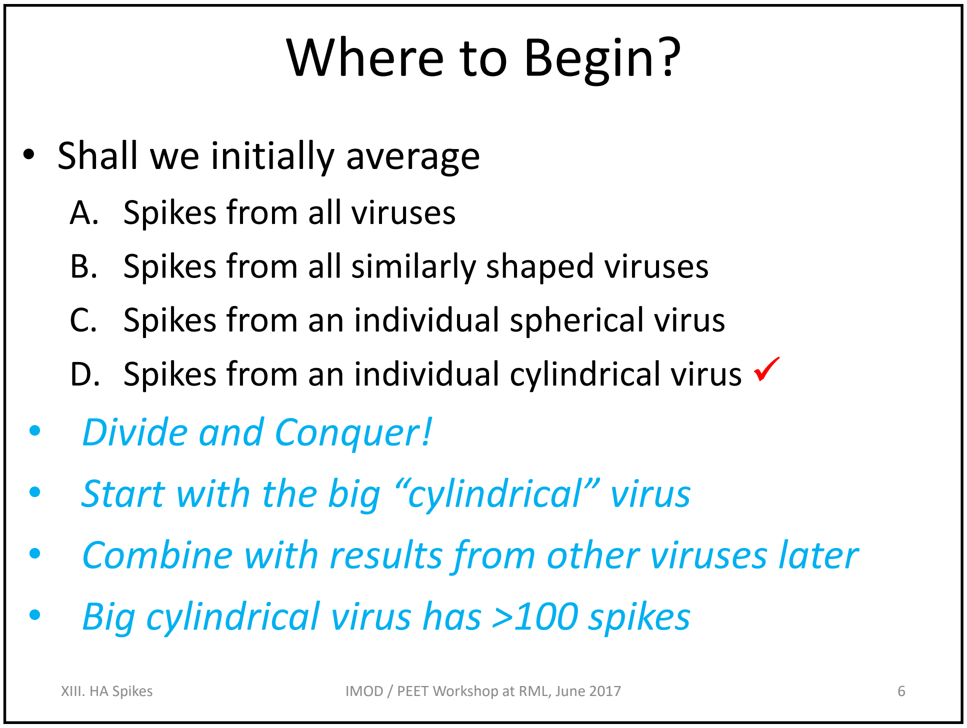
An Overall Strategy

- Would like to align and average >10,000 spikes
- Too many to pick manually (at least for me!)
- Bootstrap Approach
  - Pick a manageable number of particles manually
  - Align these to generate an initial template
  - Template matching to pick additional particles
  - Screen candidates… *e.g.* by cross-correlation
  - Align and symmetrize



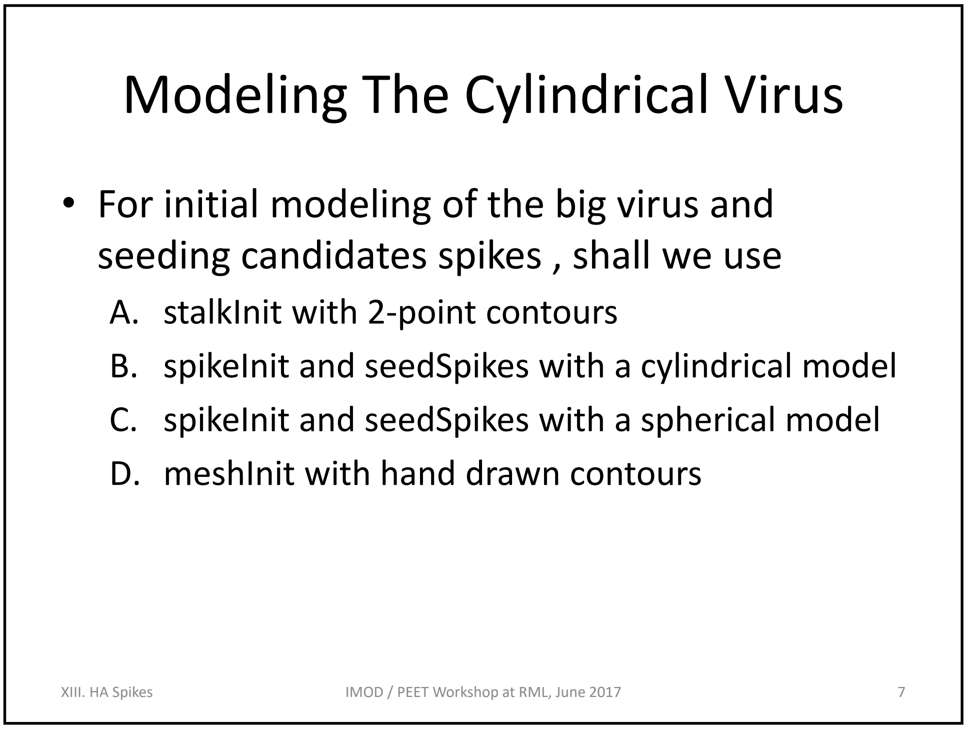
Where to Begin?

- For the initial average shall we use spikes
  - A. from all viruses
  - B. from all similarly shaped viruses
  - C. from an individual spherical virus
  - D. from an individual cylindrical virus



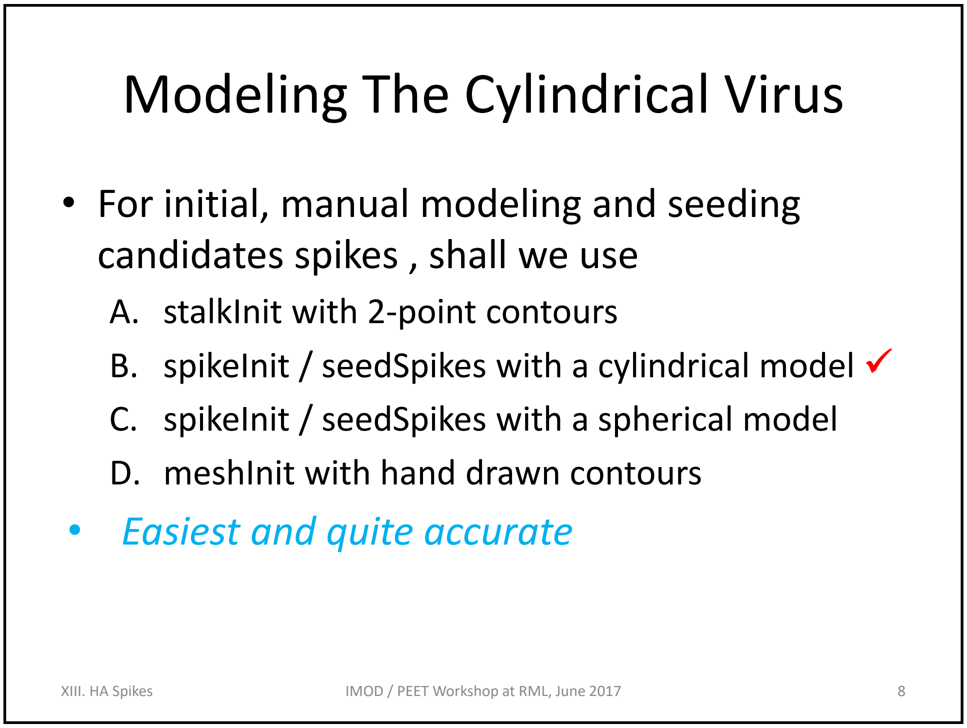
Where to Begin?

- Shall we initially average
  - A. Spikes from all viruses
  - B. Spikes from all similarly shaped viruses
  - C. Spikes from an individual spherical virus
  - D. Spikes from an individual cylindrical virus ✓
- *Divide and Conquer!*
- *Start with the big "cylindrical" virus*
- *Combine with results from other viruses later*
- *Big cylindrical virus has >100 spikes*



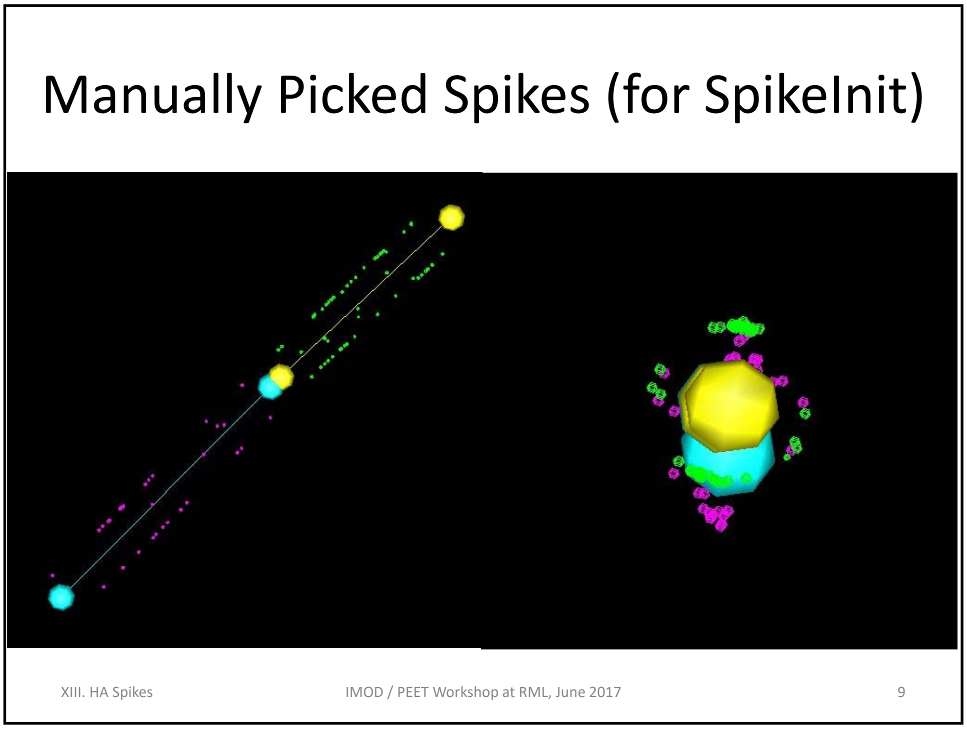
Modeling The Cylindrical Virus

- For initial modeling of the big virus and seeding candidates spikes , shall we use
  A. stalkInit with 2-point contours
  B. spikeInit and seedSpikes with a cylindrical model
  C. spikeInit and seedSpikes with a spherical model
  D. meshInit with hand drawn contours

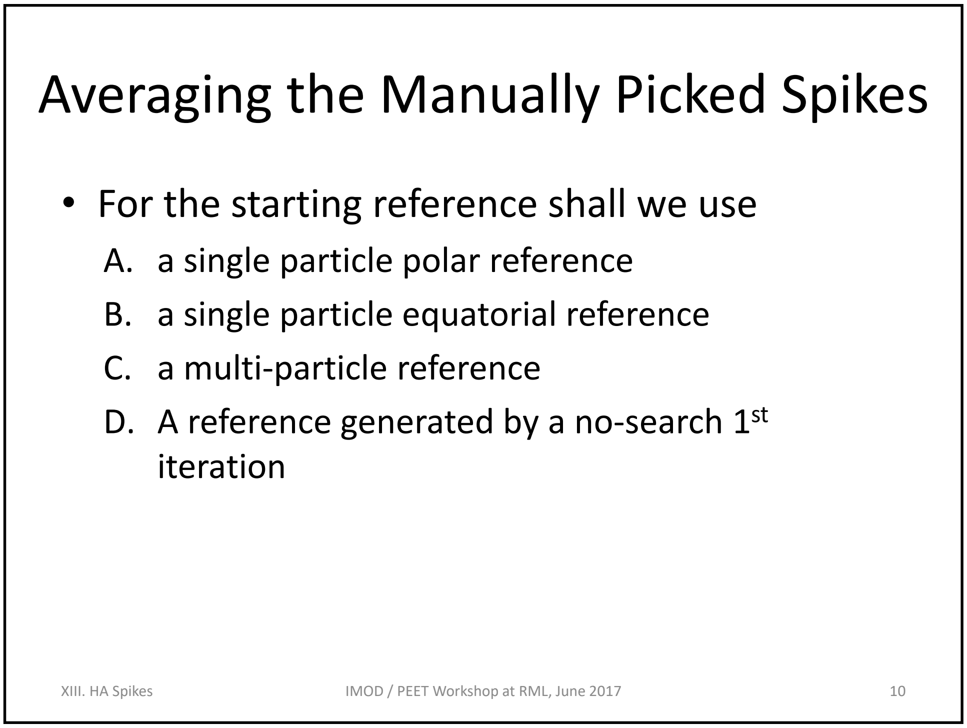


Modeling The Cylindrical Virus

- For initial, manual modeling and seeding candidates spikes , shall we use
  A. stalkInit with 2-point contours
  B. spikeInit / seedSpikes with a cylindrical model ✓
  C. spikeInit / seedSpikes with a spherical model
  D. meshInit with hand drawn contours
- *Easiest and quite accurate*



Manually Picked Spikes (for SpikeInit)



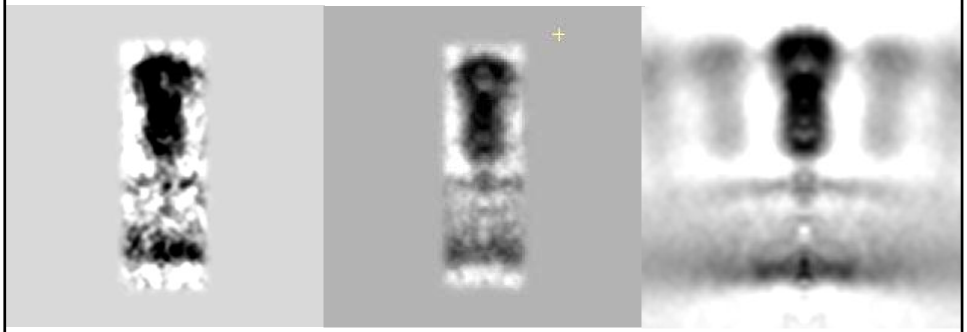
Averaging the Manually Picked Spikes

- For the starting reference shall we use
  A. a single particle polar reference
  B. a single particle equatorial reference
  C. a multi-particle reference
  D. A reference generated by a no-search 1st iteration

Averaging the Manually Picked Spikes

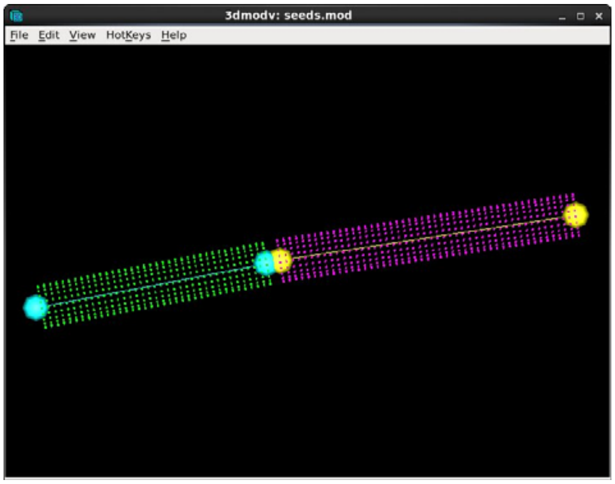
- For the starting reference shall we use
  A. a single particle polar reference ✓
  B. a single particle equatorial reference
  C. a multi-particle reference
  D. A reference generated by a no-search 1st iteration
- *A polar reference seems more likely to preserve the C_3 rotational symmetry*

Results From Manually Picked Spikes (For Template Matching)



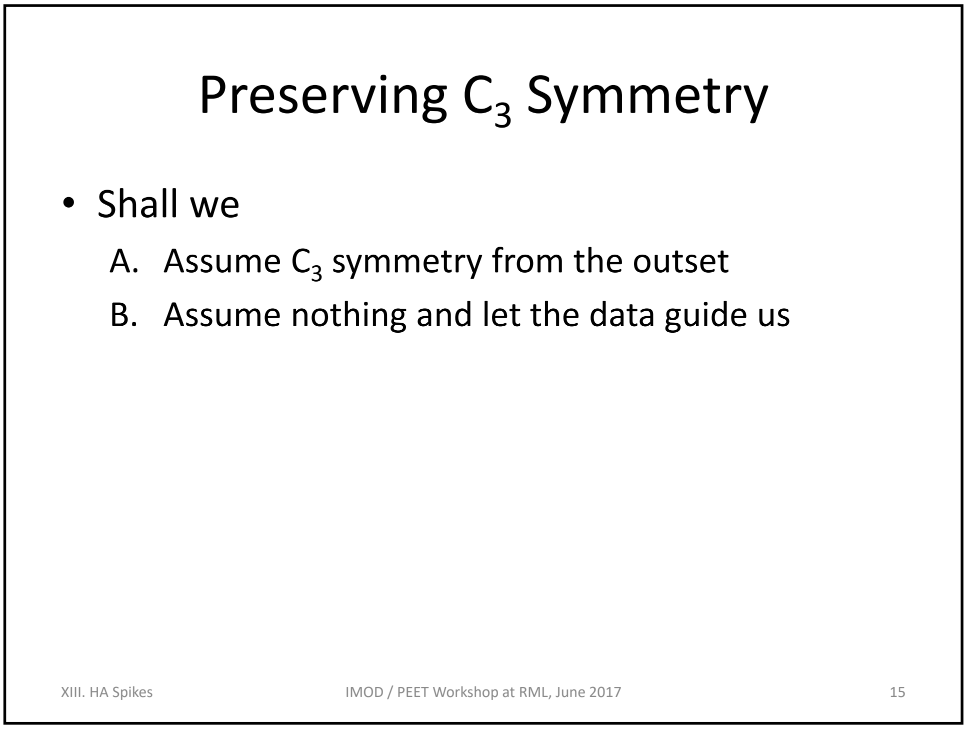
Original C_3 Symmetric C_{180} Symmetric

Seeding Candidate Spikes



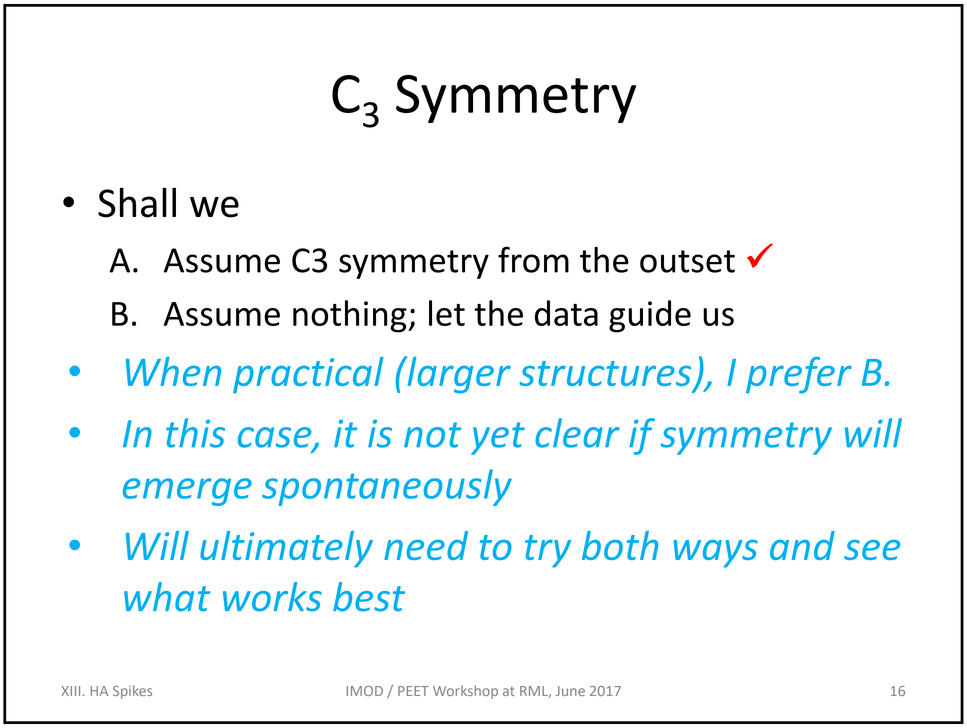
How Densely to Seed Candidates?

- More candidates with closer spacing
  - Smaller search distances
  - Discard more candidates afterwards
- Fewer candidates with larger spacing
  - Larger search distances
  - Don't need to discard as many candidates
- Not clear *a priori* which will work better
- Goals: minimize reference bias & maximize speed



Preserving C_3 Symmetry

- Shall we
  - A. Assume C_3 symmetry from the outset
  - B. Assume nothing and let the data guide us



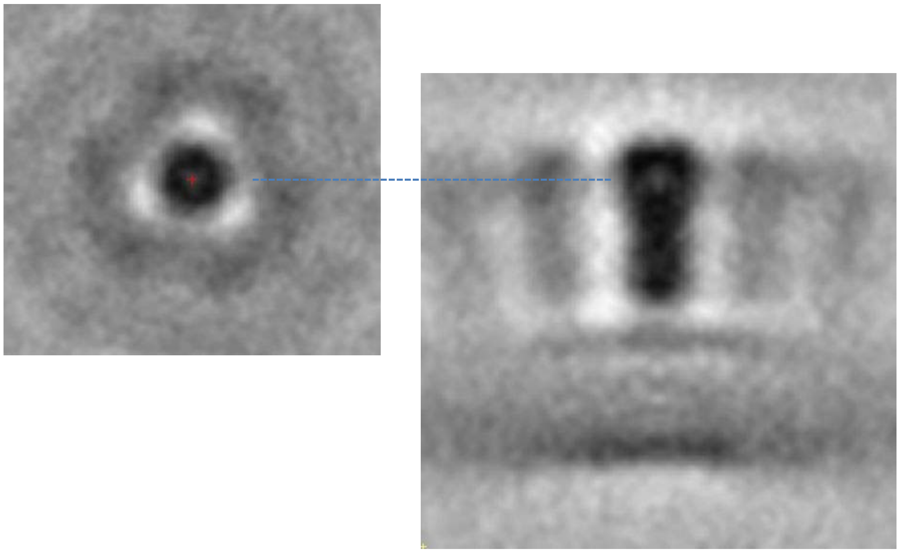
C_3 Symmetry

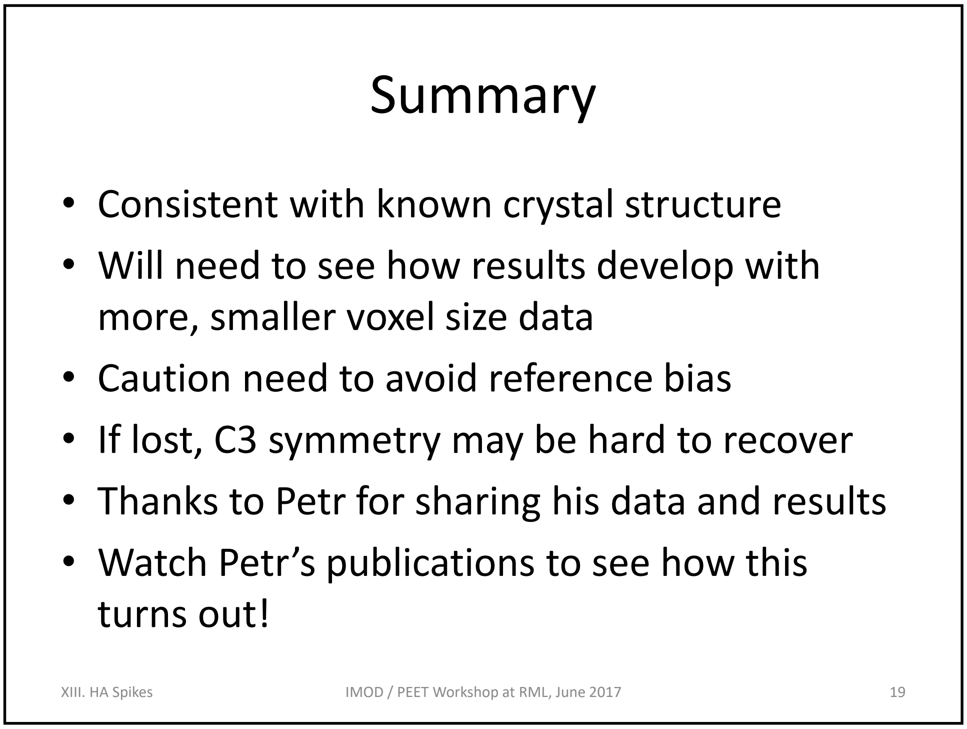
- Shall we
  - A. Assume C3 symmetry from the outset ✓
  - B. Assume nothing; let the data guide us
- *When practical (larger structures), I prefer B.*
- *In this case, it is not yet clear if symmetry will emerge spontaneously*
- *Will ultimately need to try both ways and see what works best*

C_3 Symmetry

- A hybrid approach
  - Use axial symmetry iterations early on
  - Impose C_3 symmetry with virtual particles only at later stages
- Still don’t have enough data to know which approach is best for this case

Current Averages





Summary

- Consistent with known crystal structure
- Will need to see how results develop with more, smaller voxel size data
- Caution need to avoid reference bias
- If lost, C3 symmetry may be hard to recover
- Thanks to Petr for sharing his data and results
- Watch Petr’s publications to see how this turns out!

Questions?

THIS PAGE INTENTIONALLY LEFT BLANK