



THE UNIVERSITY *of* TEXAS

**SCHOOL OF HEALTH INFORMATION
SCIENCES AT HOUSTON**

Virus Capsids and Icosahedral Reconstruction

For students of HI 6001-125

“Computational Structural Biology”

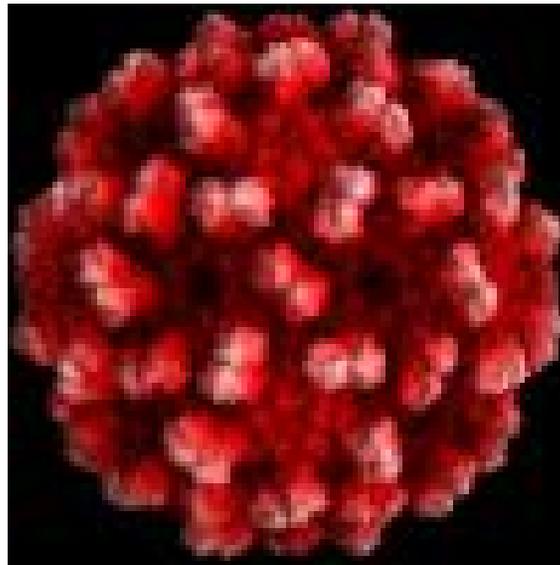
Willy Wriggers, Ph.D.

<http://biomachina.org/courses/structures/10.html>

Overview and Biological Relevance

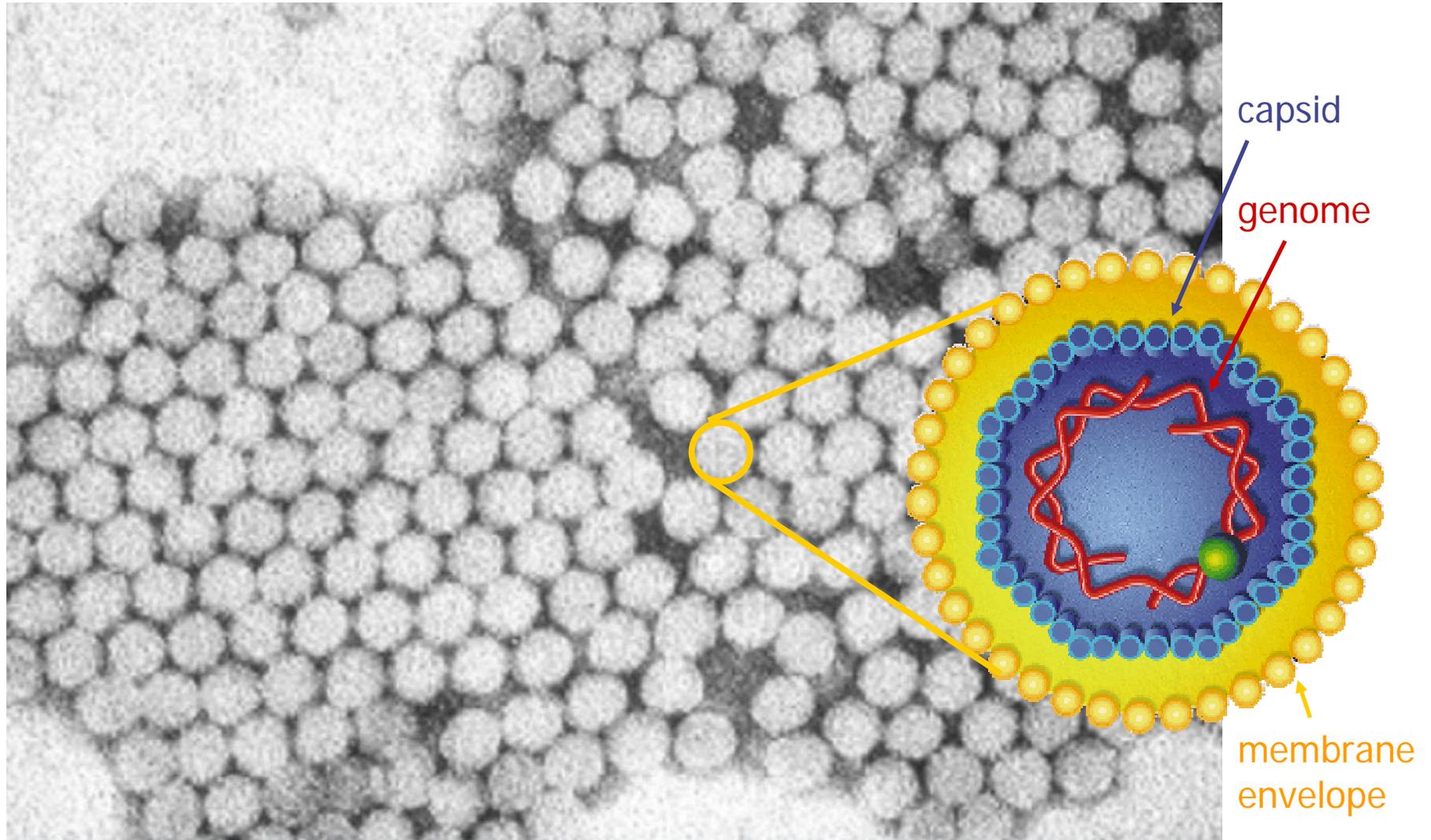
Virus Structure

- crystallization of a virus was first reported in the 1930s.
- first atomic resolution structure of a virus was 1978, tomato bushy stunt virus (Stephen Harrison):



Example

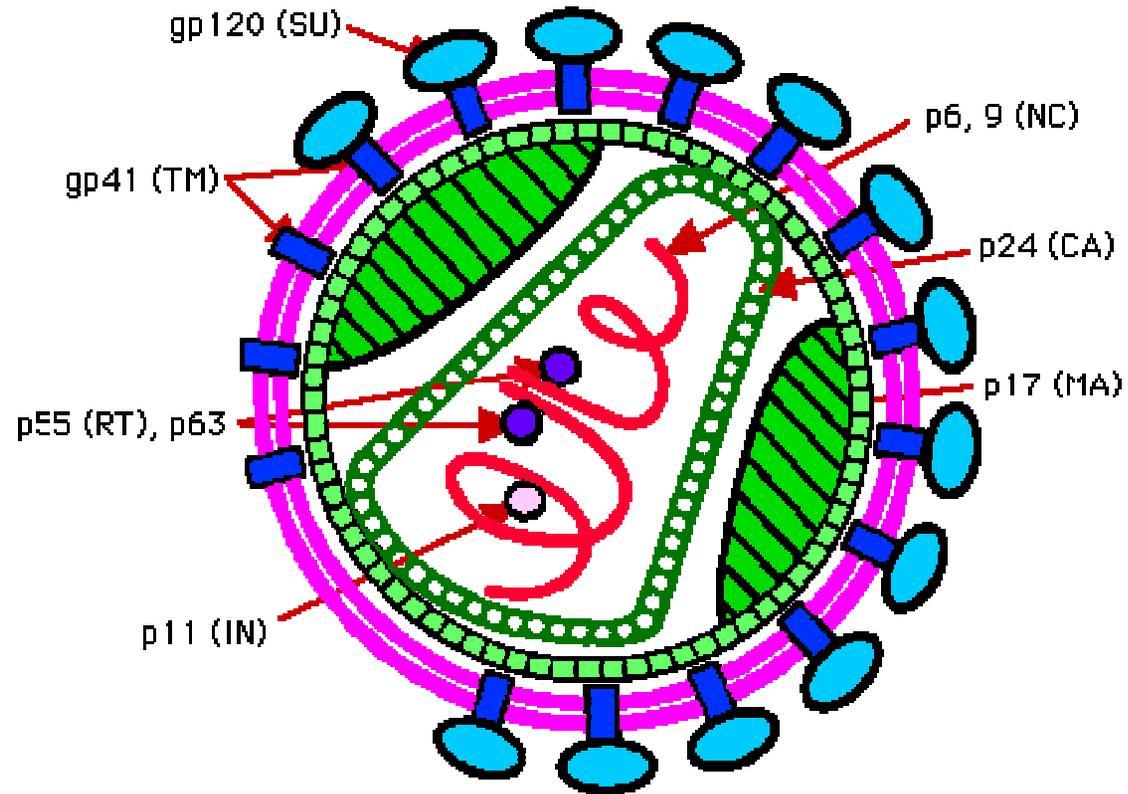
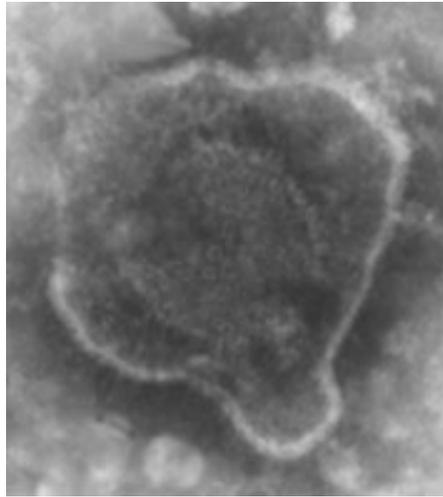
Hepatitis A virus.



Role of Virus Capsids

- Function of the outer shell (**capsid**) of a virus particle is to protect the fragile nucleic acid genome from:
 - **Physical damage** - Shearing by mechanical forces.
 - **Chemical damage**- UV irradiation (from sunlight) leading to chemical modification.
 - **Enzymatic damage** - Nucleases derived from dead or leaky cells or deliberately secreted by vertebrates as defence against infection.
- Protein subunits in a virus capsid are **multiply redundant**, i.e. present in many copies per particle. Damage to one subunit may render that subunit non-functional, but does not destroy the infectivity of the whole particle.

Membrane Envelopes



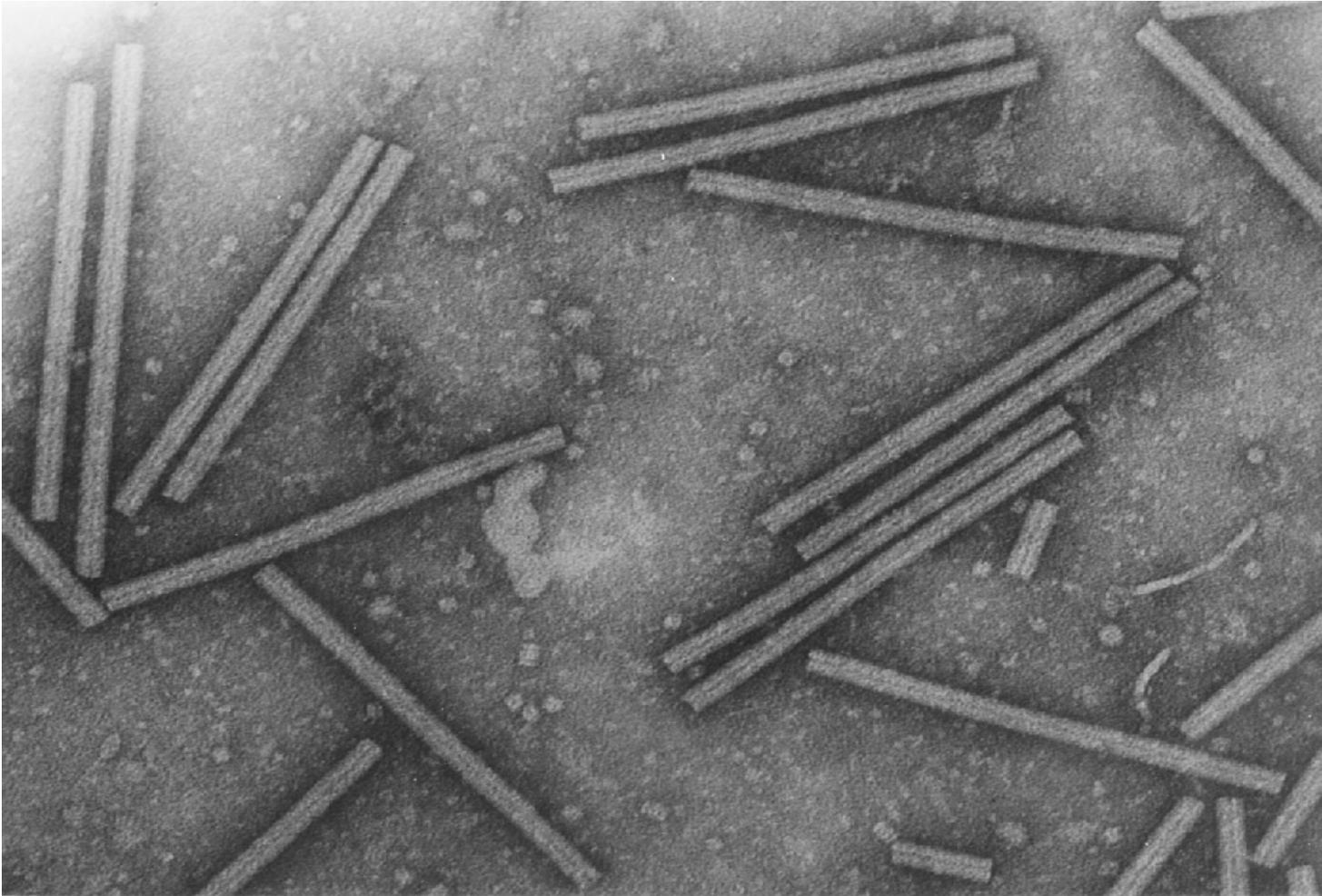
Membrane envelopes acquired from a cellular structure during release. Membranes are modified by proteins. Matrix proteins are found inside the envelope. Glycoproteins traverse the envelope.

Infection

- The outer surface of the virus is responsible for **recognition of the host cell**. Initially, this takes the form of binding of a specific **virus-attachment protein** to a **cellular receptor molecule**. The capsid also has a role to play in initiating infection by delivering the genome from its protective shell in a form in which it can interact with the host cell.
- To form an infectious particle, a virus must overcome two fundamental problems:
 1. assemble the particle utilizing only the information available from the components which make up the particle itself (capsid + genome).
 2. Form regular geometric shapes, even though the proteins from which they are made are irregularly shaped.

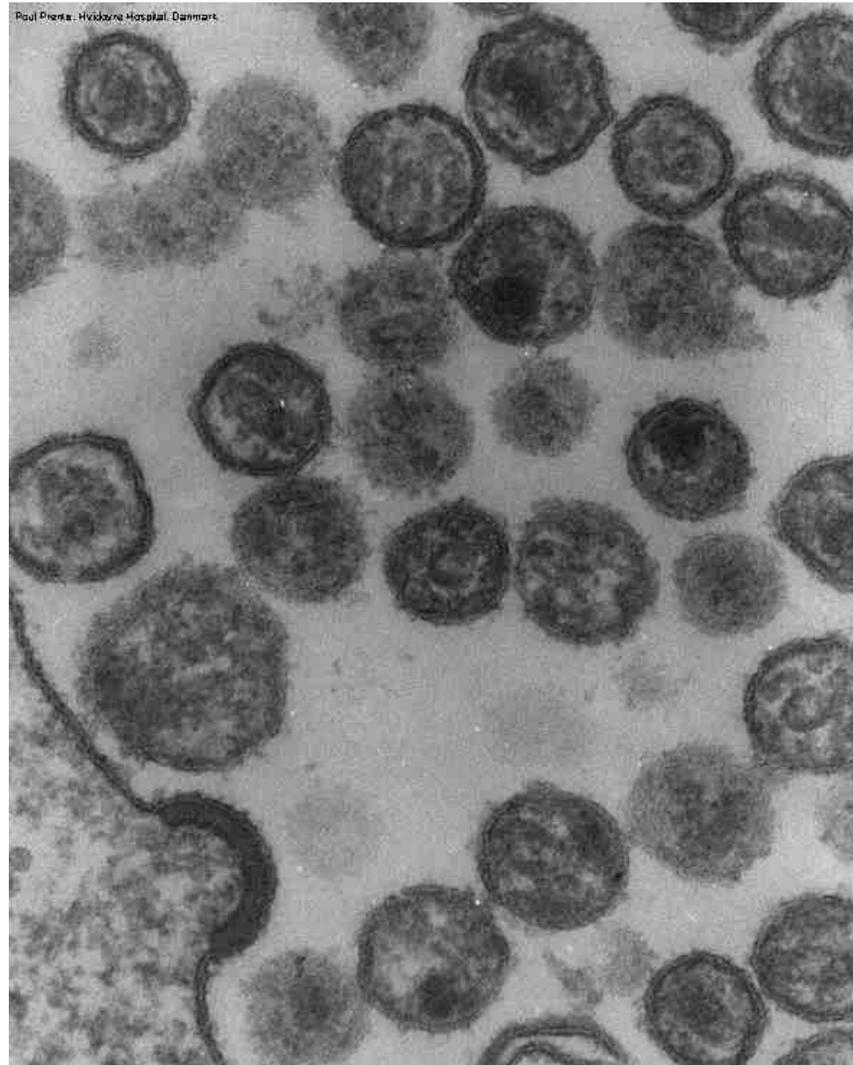
Virus Shapes

Tobacco mosaic virus (helical).



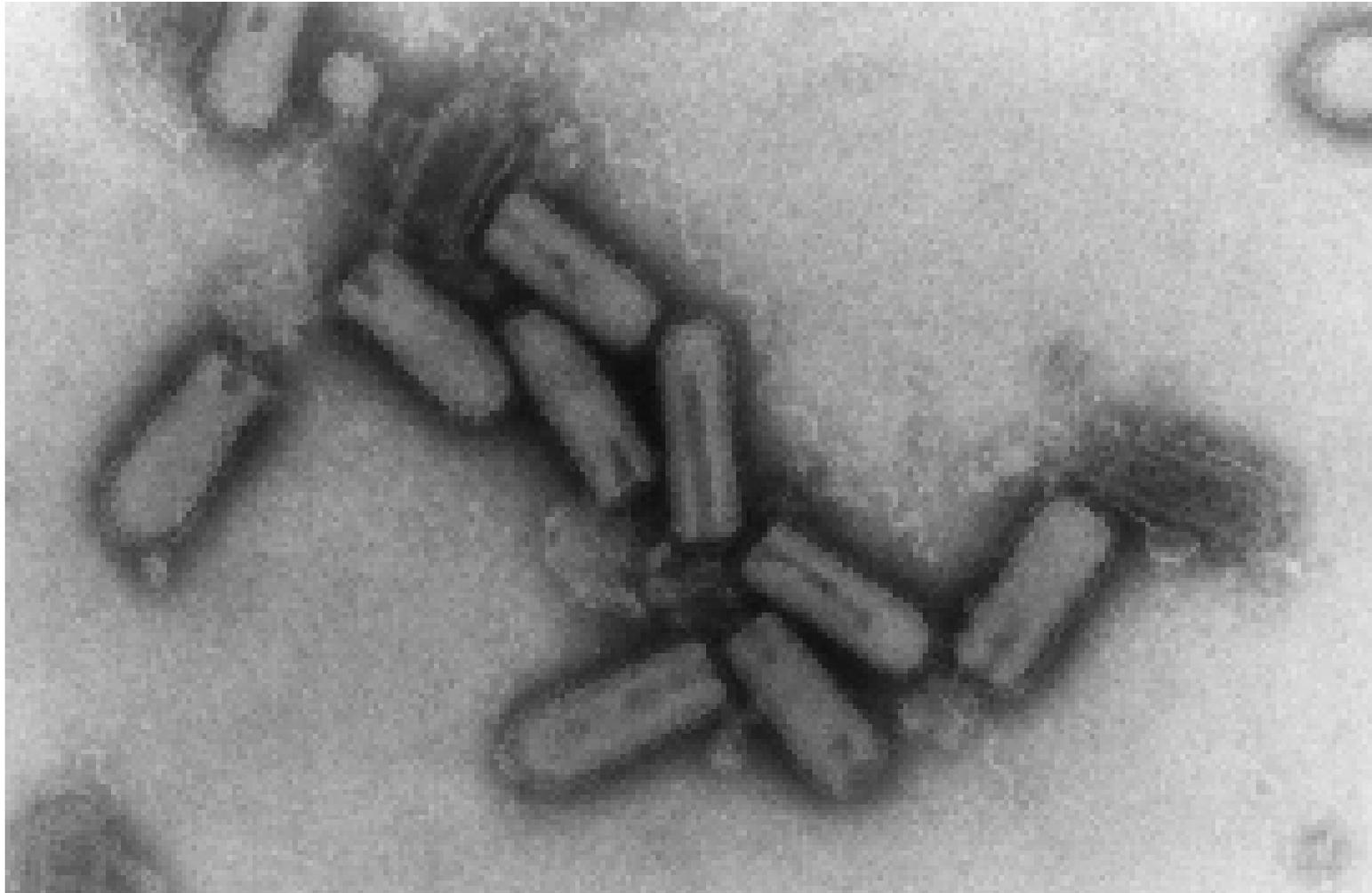
Virus Shapes

HIV (complex
globular,
enveloped)



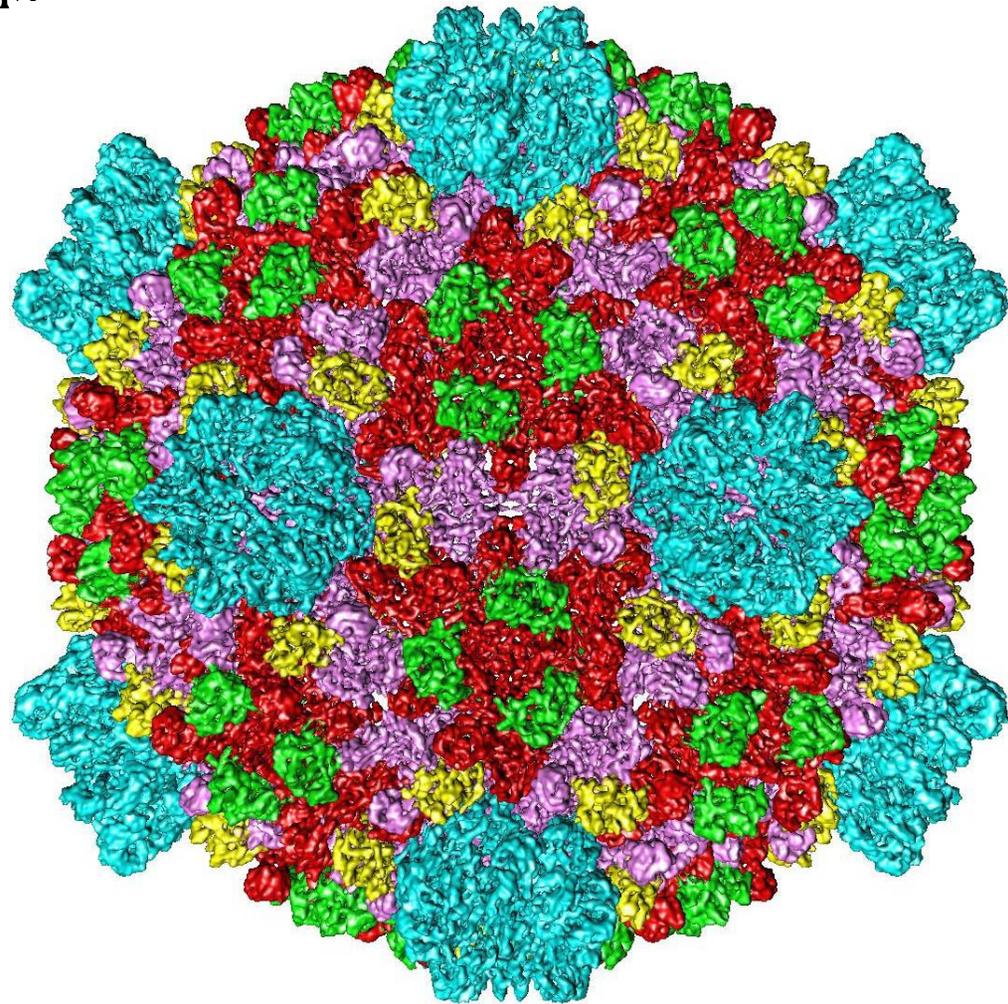
Virus Shapes

Vesicular stomatitis virus: bullet shaped



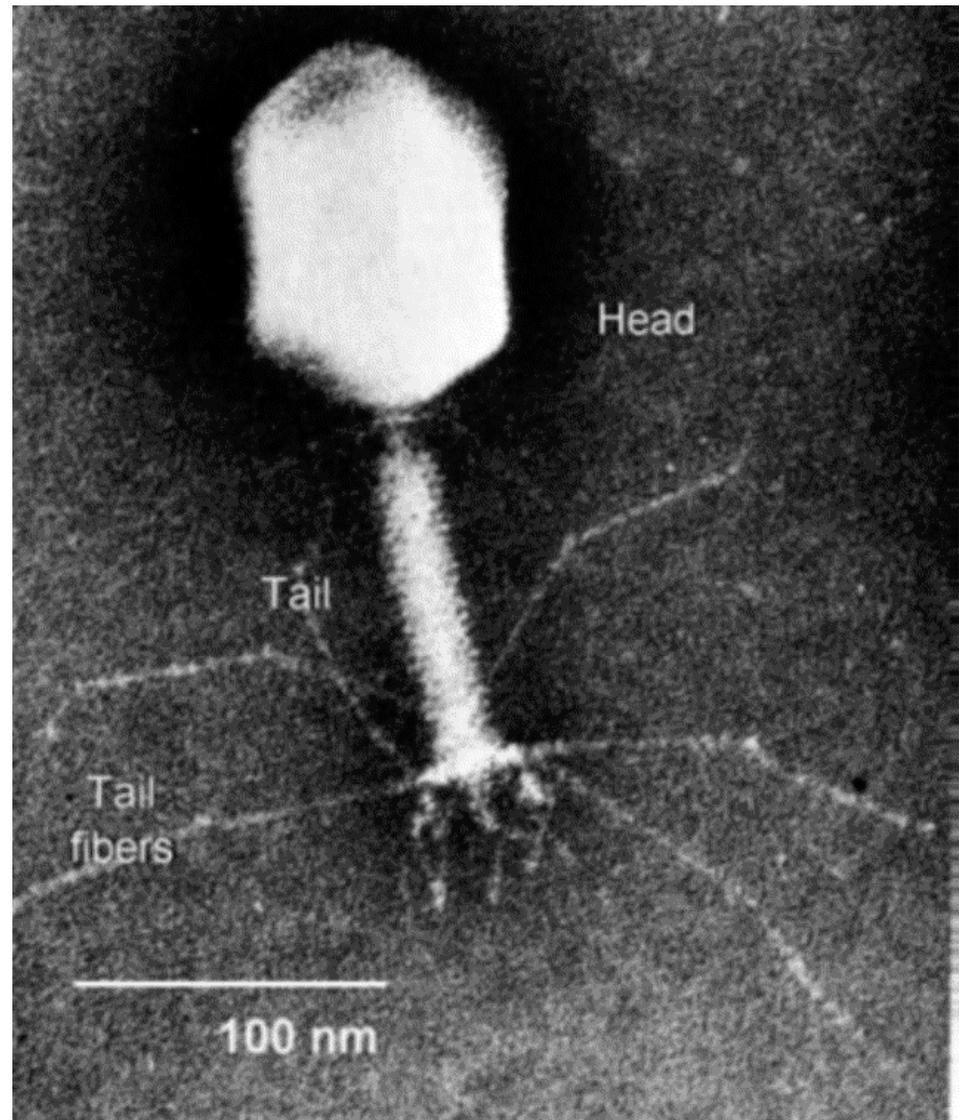
Virus Shapes

Cytoplasmic polyhedrosis
virus (icosahedral)



Virus Shapes

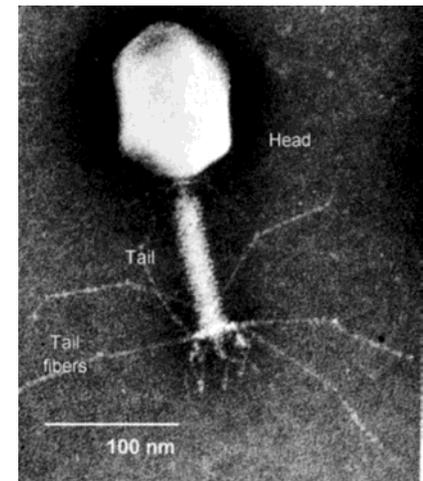
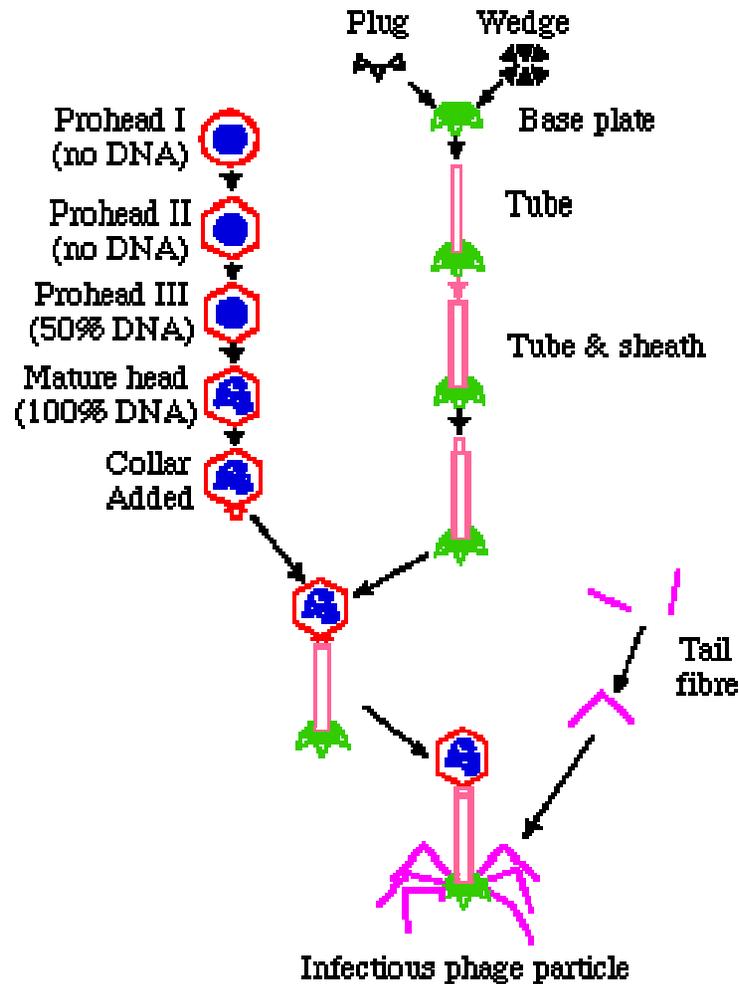
Bacteriophage T4
(icosahedral and helical)



Virus Shapes

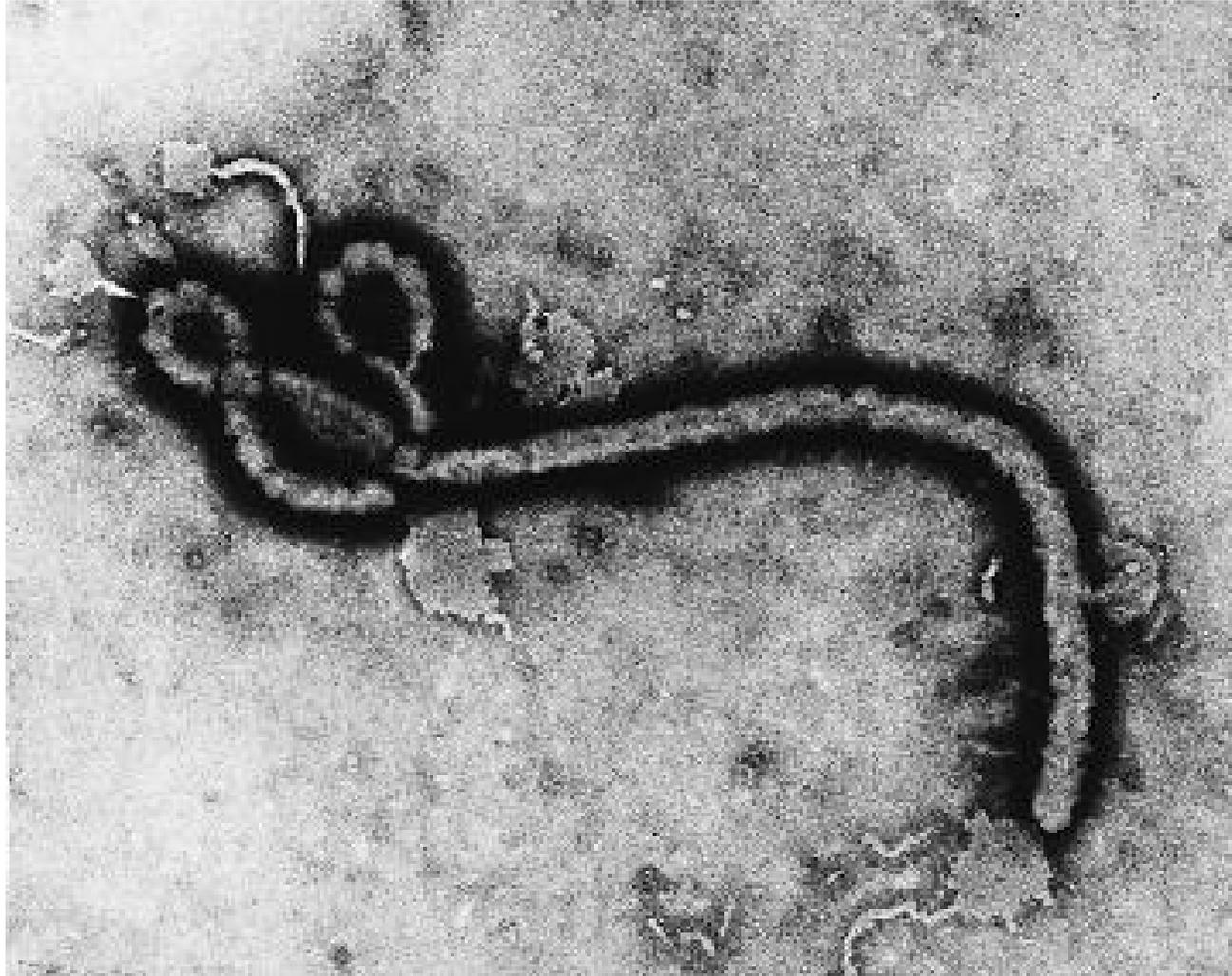
Bacteriophage T4

Head consists of an icosahedral shell attached via a collar to a helical tail. At the end of the tail is a plate which functions in attachment to the bacterial host. In addition thin protein fibres are attached to the plate, again involved in binding to host.



Virus Shapes

Ebola (irregular)

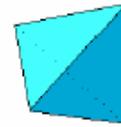


Icosahedral Symmetry

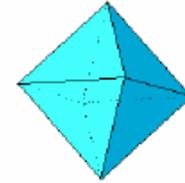
The Five Platonic Solids

From **equilateral triangles** you can make:

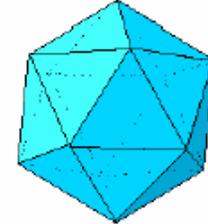
with 3 faces at each vertex, a **tetrahedron**



with 4 faces at each vertex, an **octahedron**

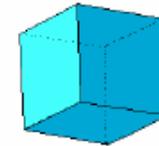


with 5 faces at each vertex, an **icosahedron**



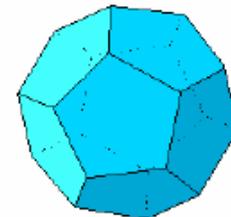
From **squares** you can make:

with 3 faces at each vertex, a **cube**

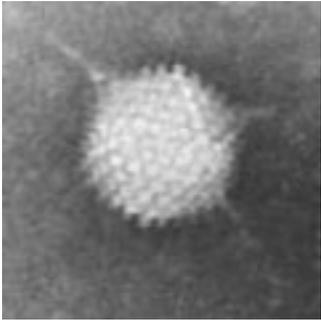


From **pentagons** you can make:

with 3 faces at each vertex, a **dodecahedron**



Virus Structure: Icosahedra

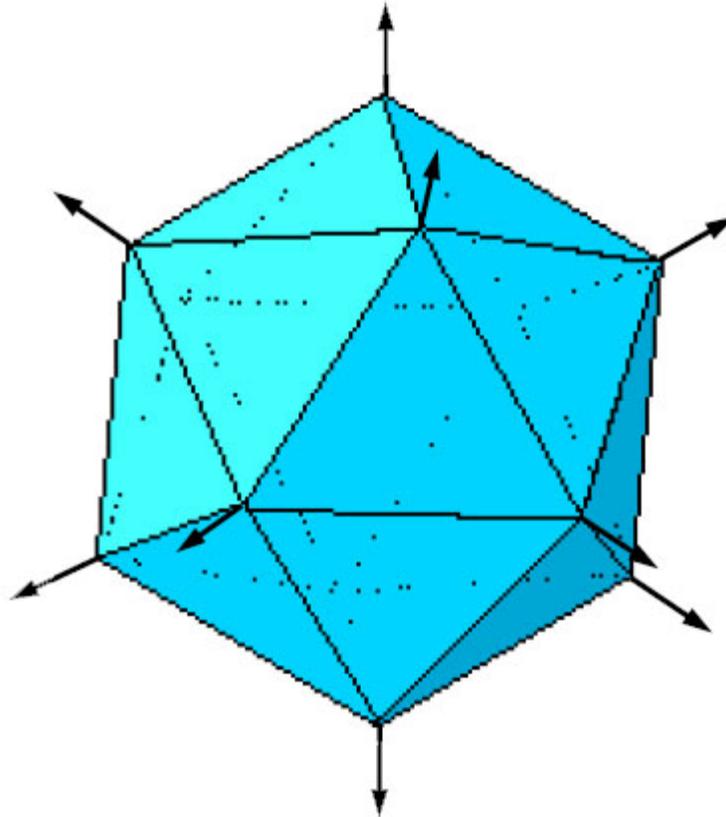


- A common way of building a virus capsid is to arrange protein subunits in the form of a hollow quasi-spherical structure, enclosing the genome within.

Crick & Watson (1956), after seeing electron micrographs, were the first to suggest that virus capsids are composed of numerous identical protein sub-units arranged either in helical or icosahedral symmetry.

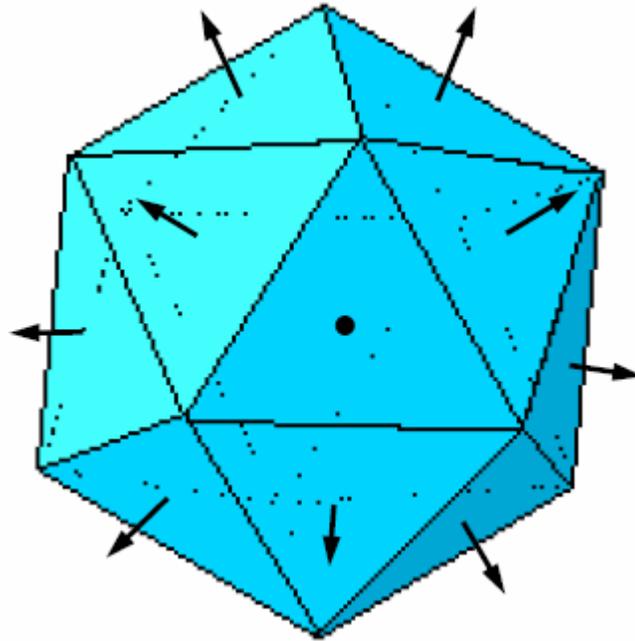
In order to construct a capsid from repeated subunits, a virus must 'know the rules' which dictate how these are arranged. For an icosahedron, the rules are based on 2-3-5 rotational symmetry.

The Icosahedron



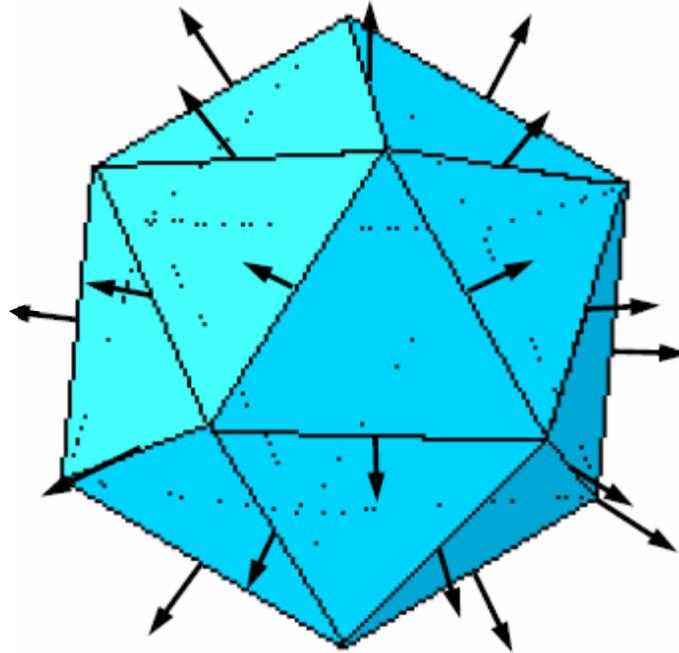
No of vertices : 12 (5-fold symmetry)

The Icosahedron



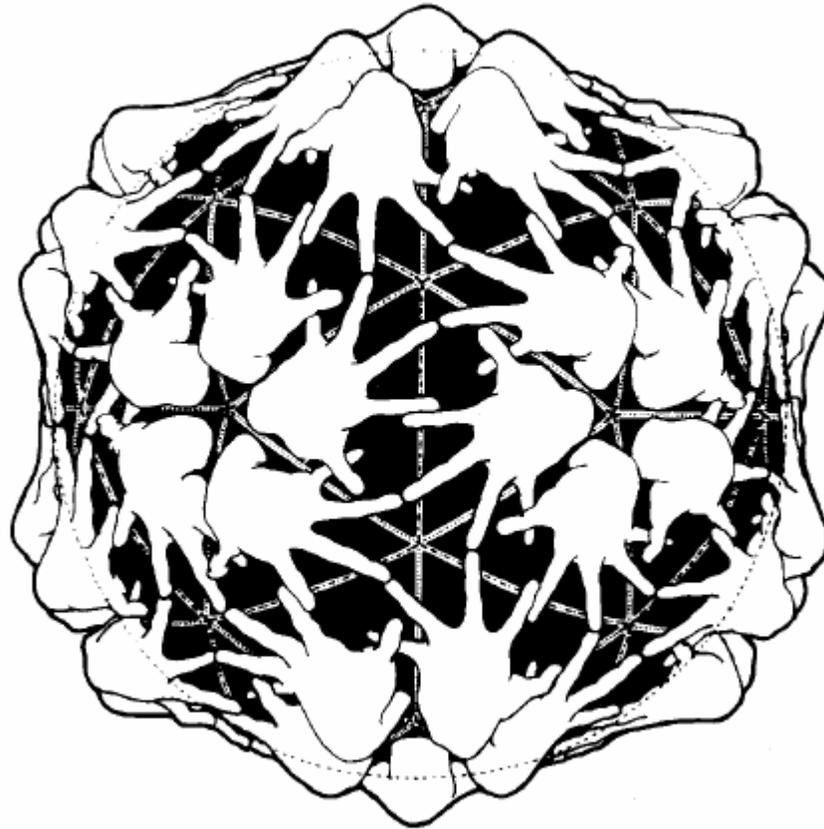
No of faces : 20 (3-fold symmetry)

The Icosahedron

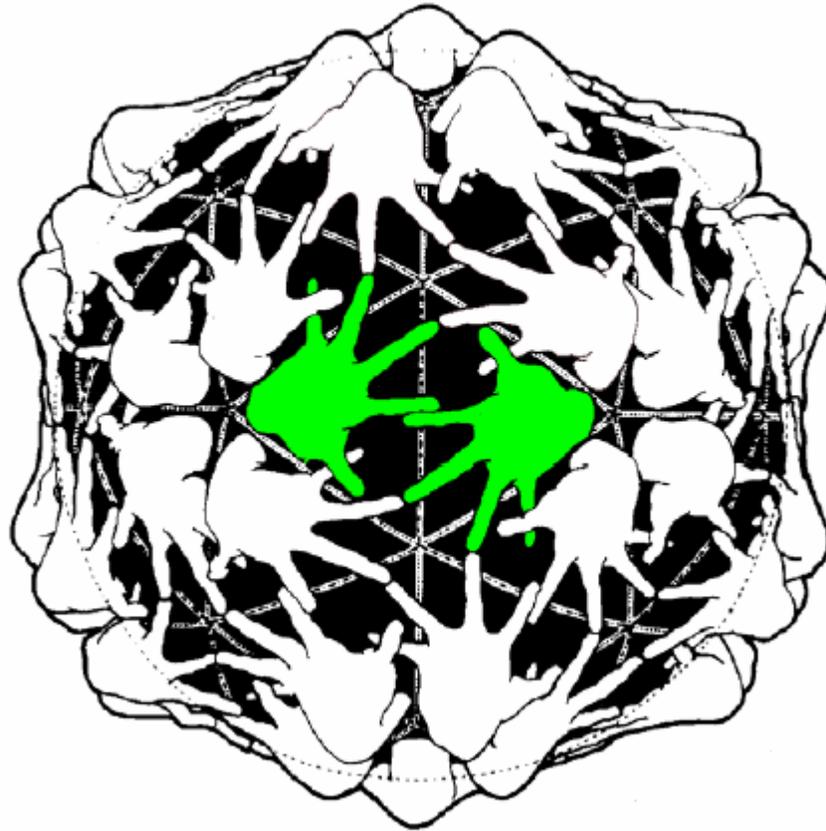


No of edges : 30 (2-fold symmetry)

532 Point Group Symmetry



532 Point Group Symmetry



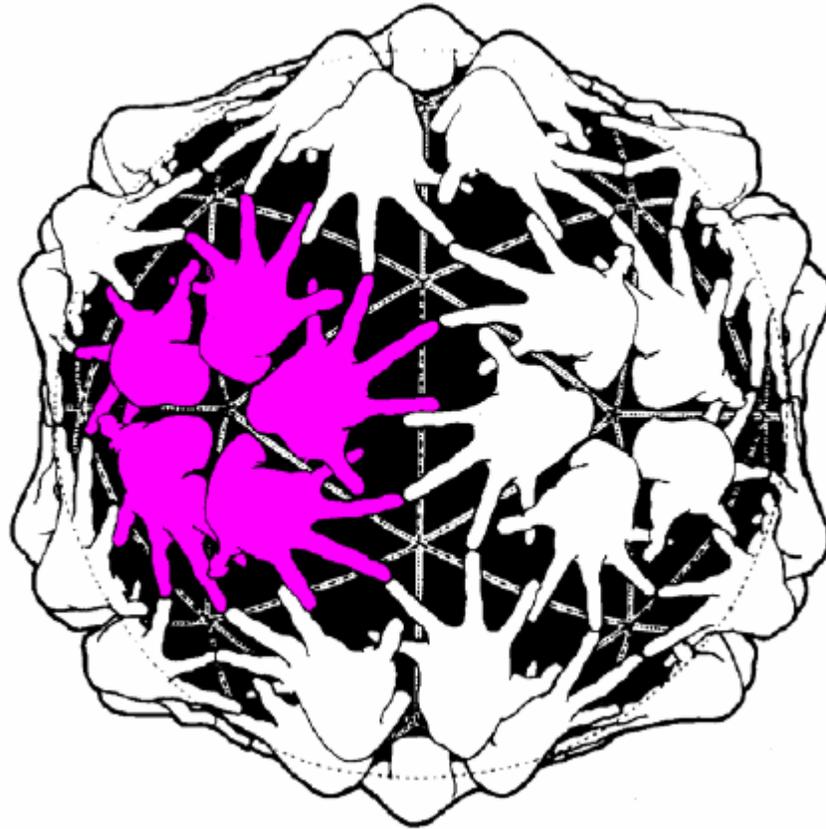
30 dimers

532 Point Group Symmetry



20 trimers

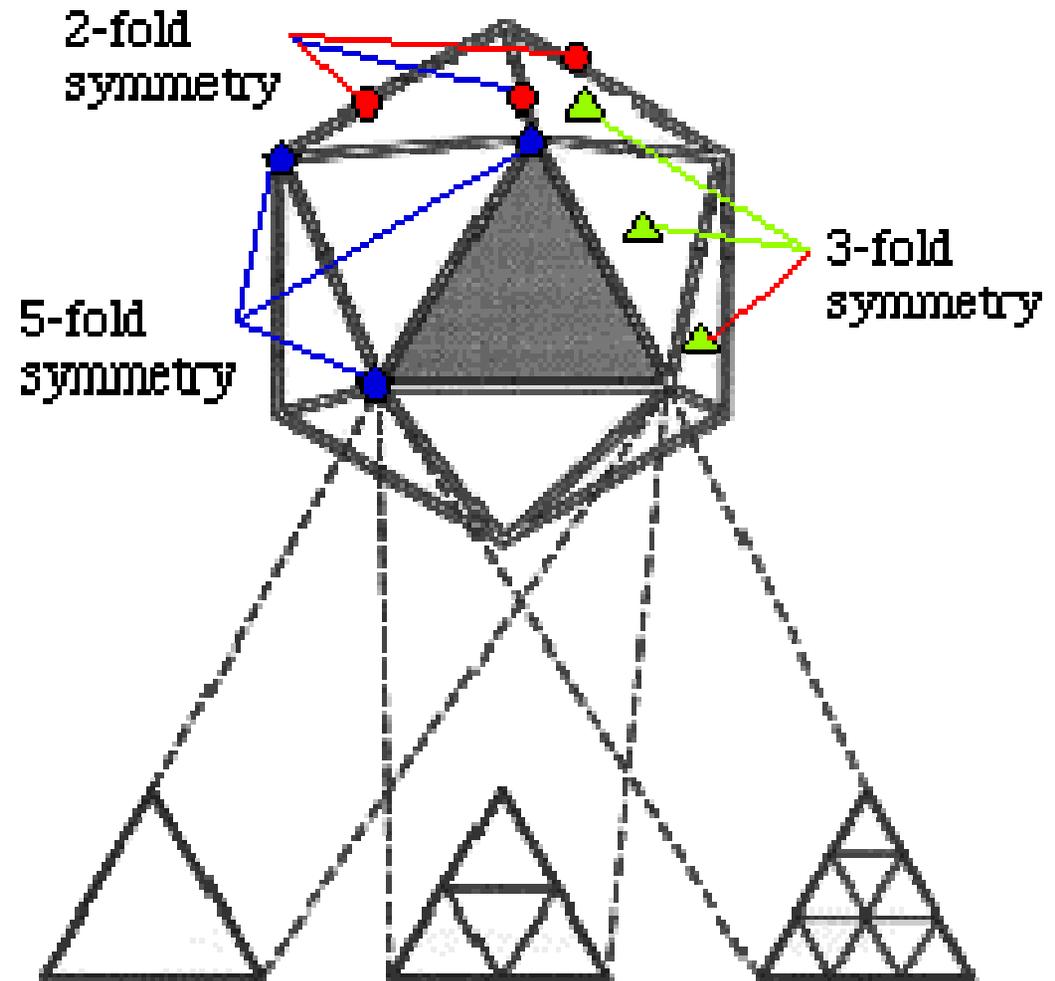
532 Point Group Symmetry



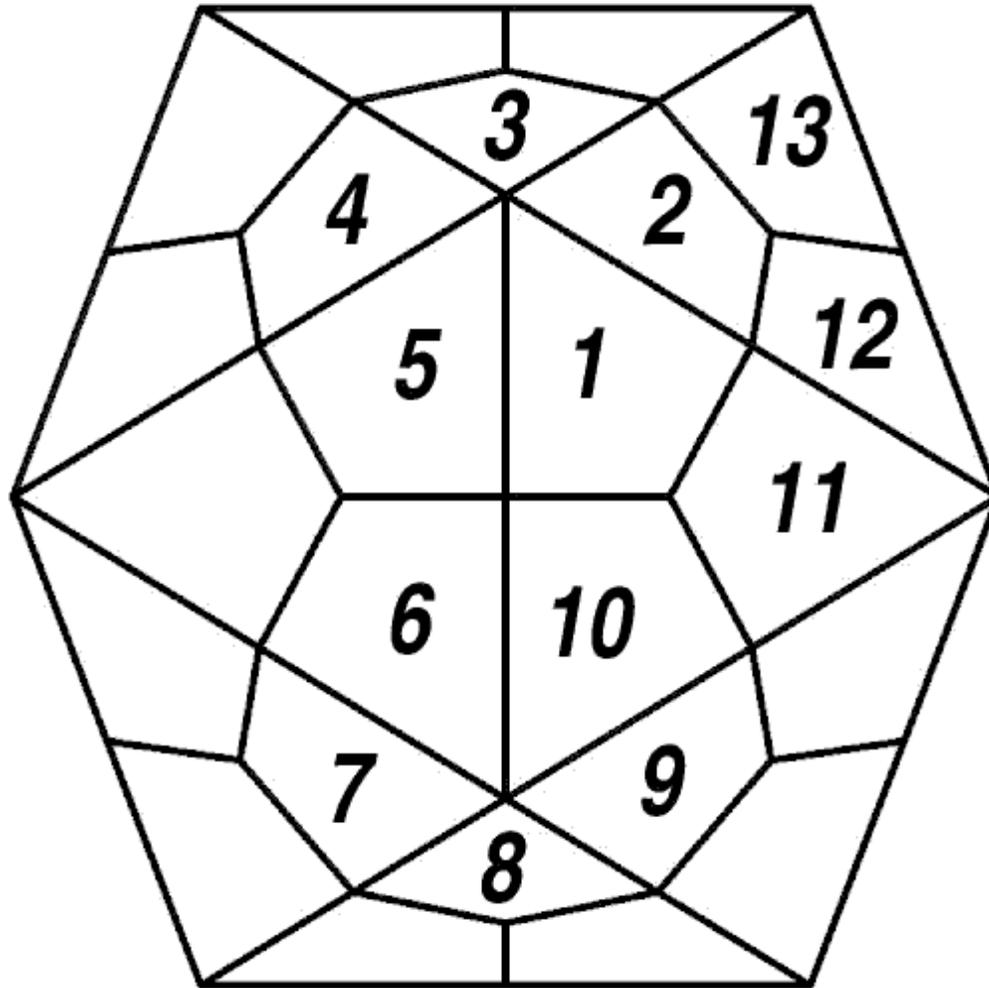
12 pentamers

Virus Structure: Icosahedra

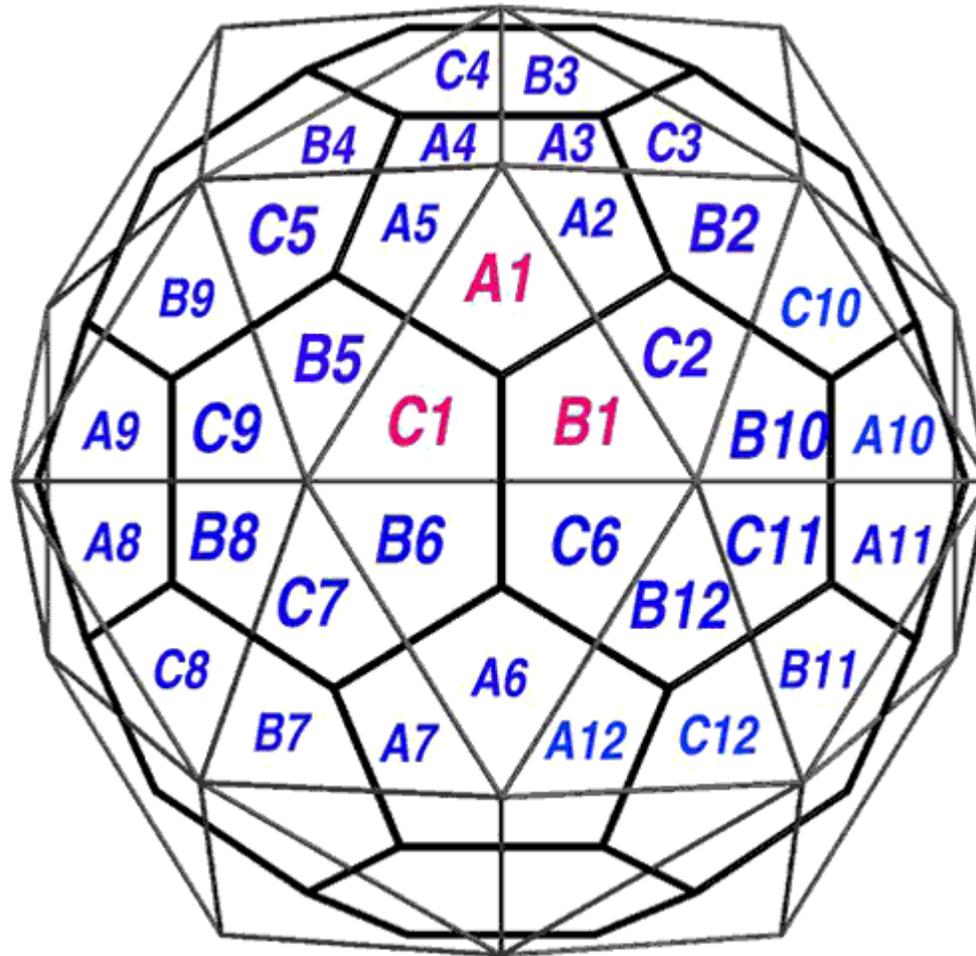
- 20 equilateral triangles arranged into a sphere.
- **bacteriophage ØX174.** 60 identical subunits form a capsid. 3 protein subunits per triangular face (T=1). This is the simplest case; most viruses have more subunits per face (higher T number).



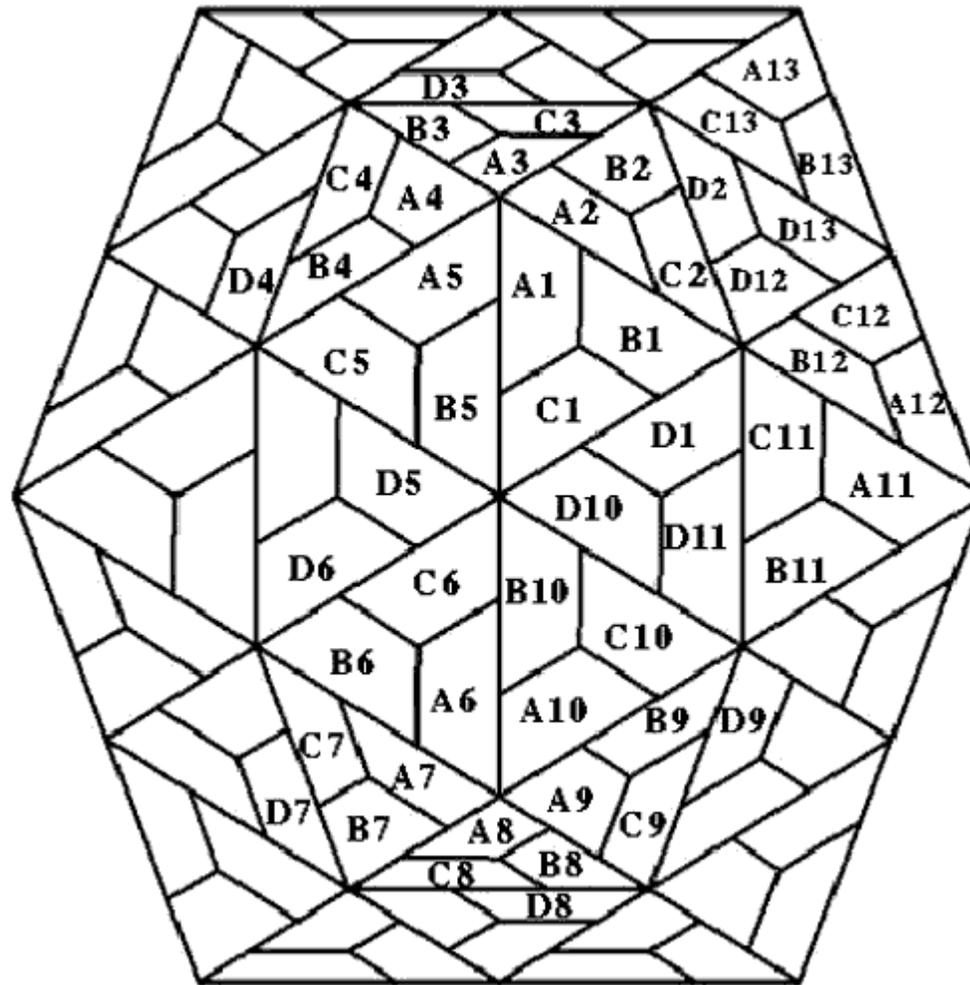
T=1 Triangulation



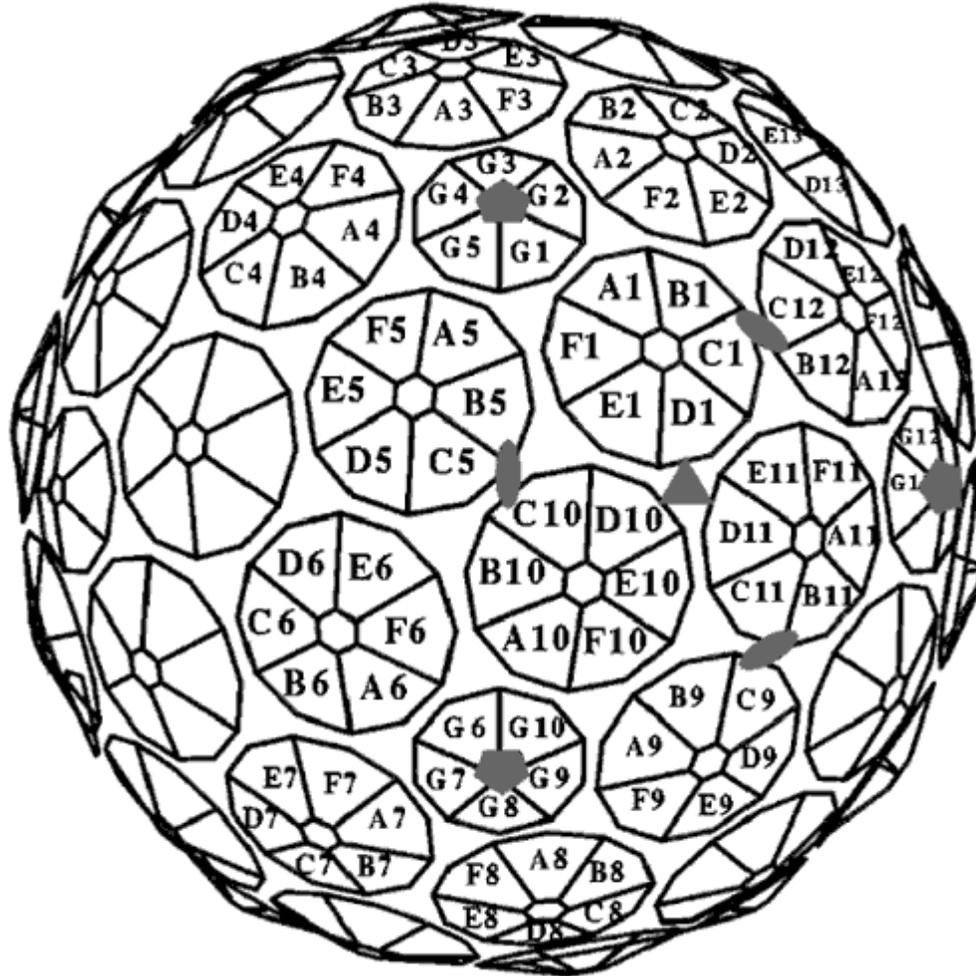
T=3 Triangulation



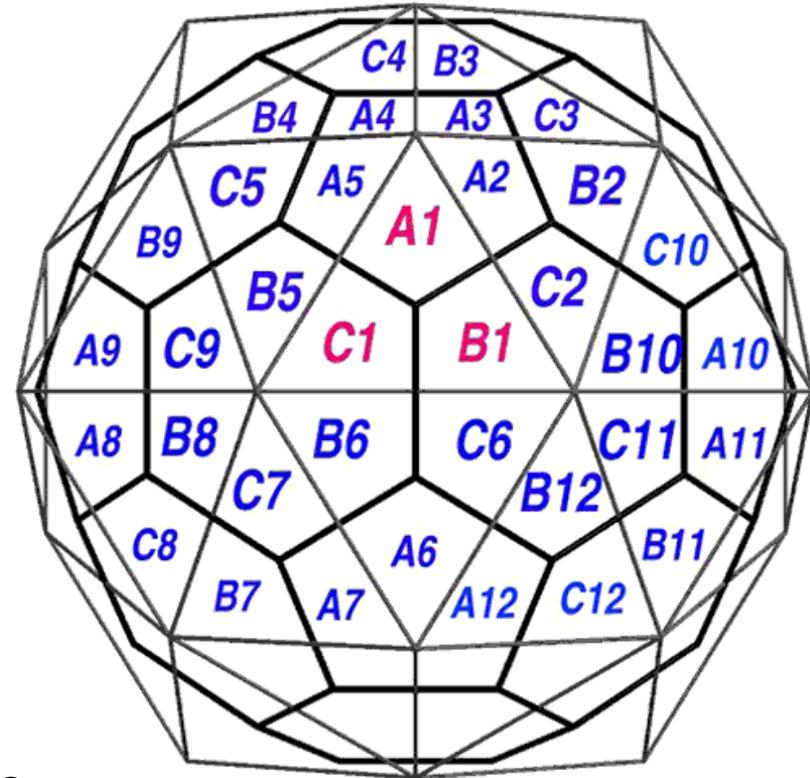
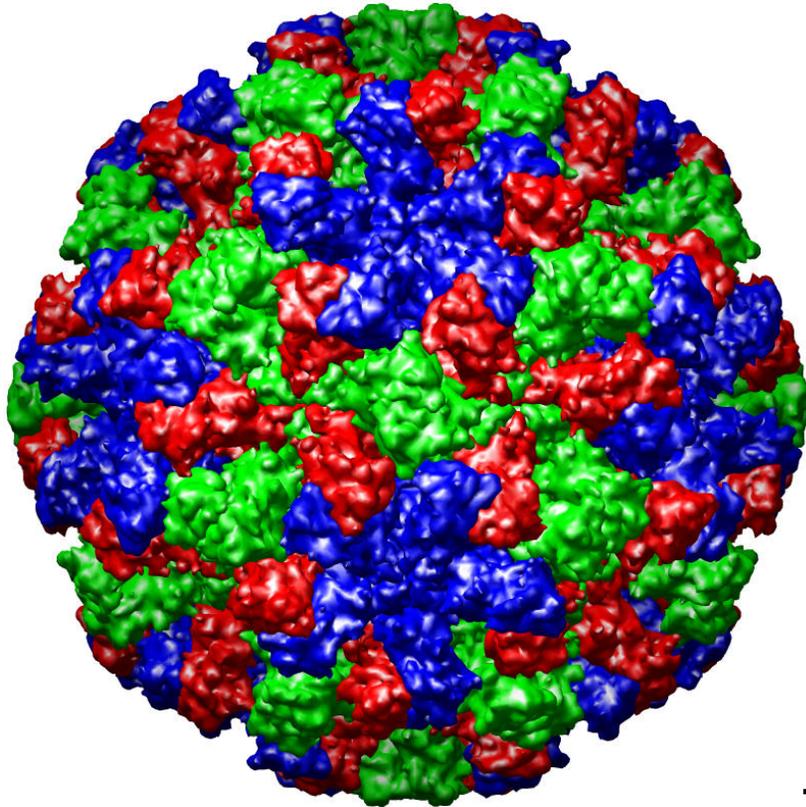
T=4 Triangulation



T=7 Triangulation

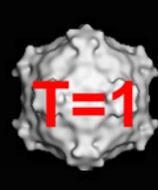
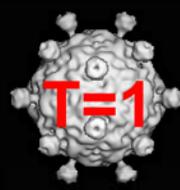
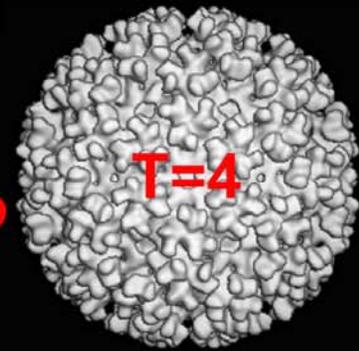
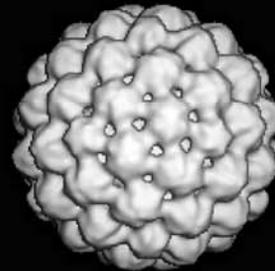
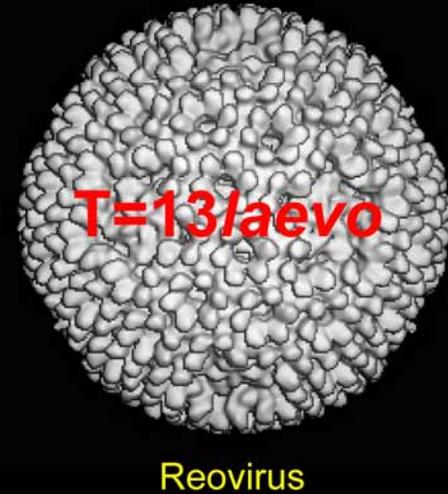
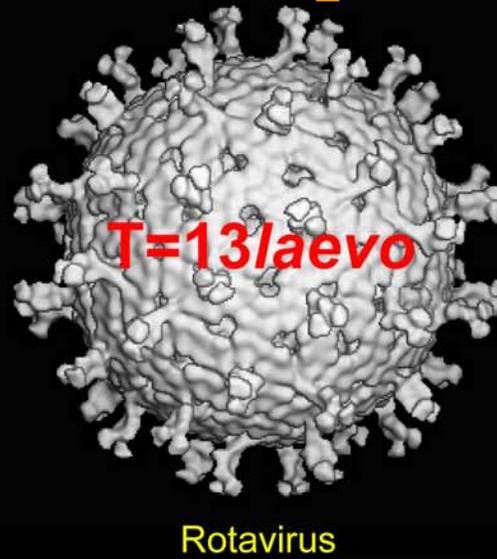
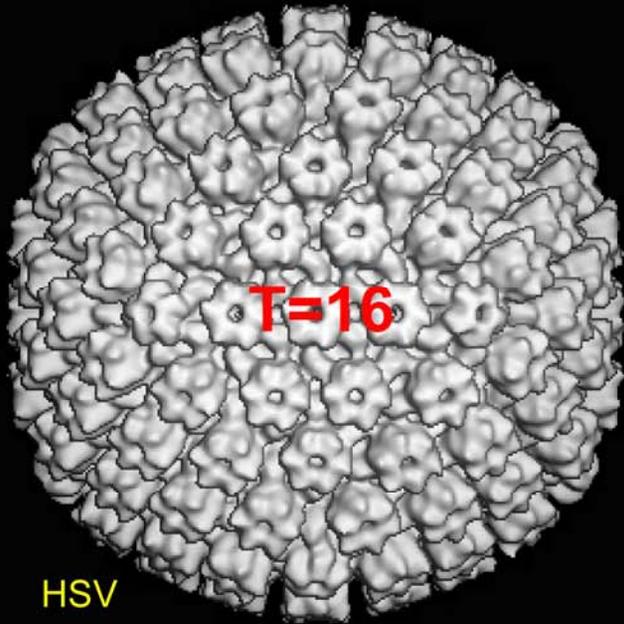


Example: Norwalk Virus Structure



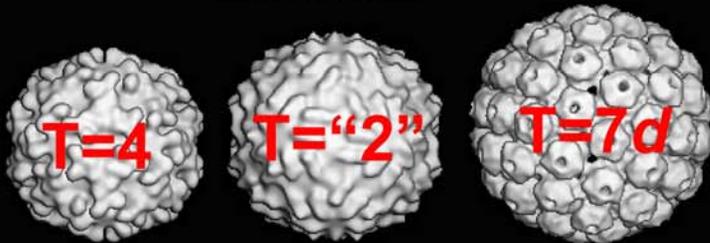
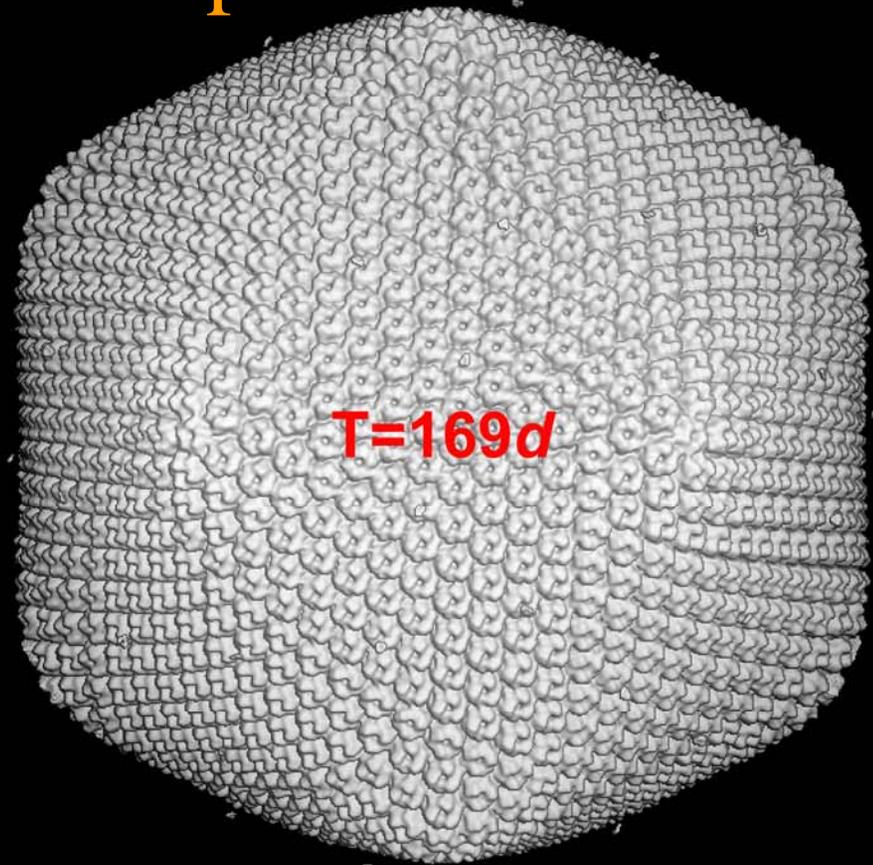
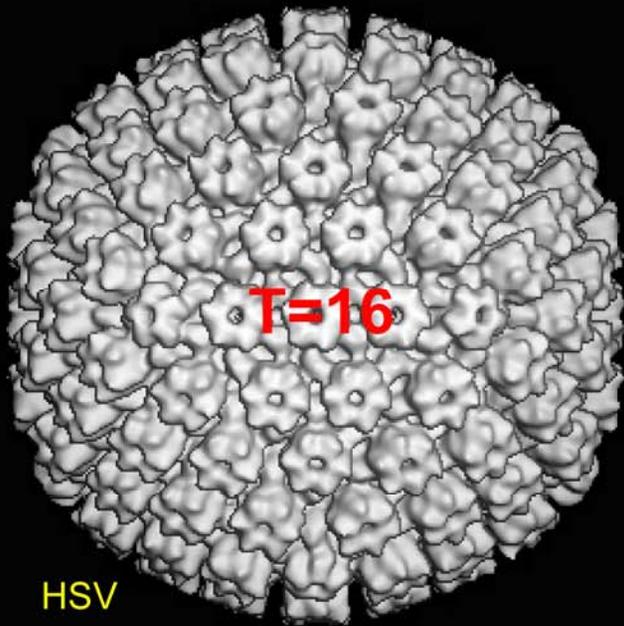
T=3

More Examples



500 Å

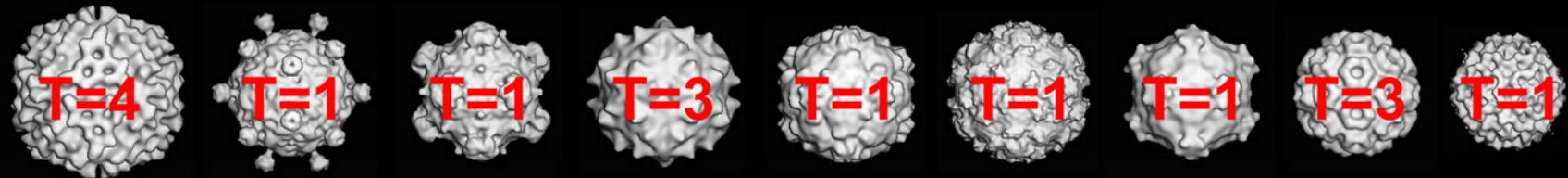
More Examples



NoV

LA-1

Polyoma



NβV

SpV-4

φX174

FHV

HRV-14

Polio

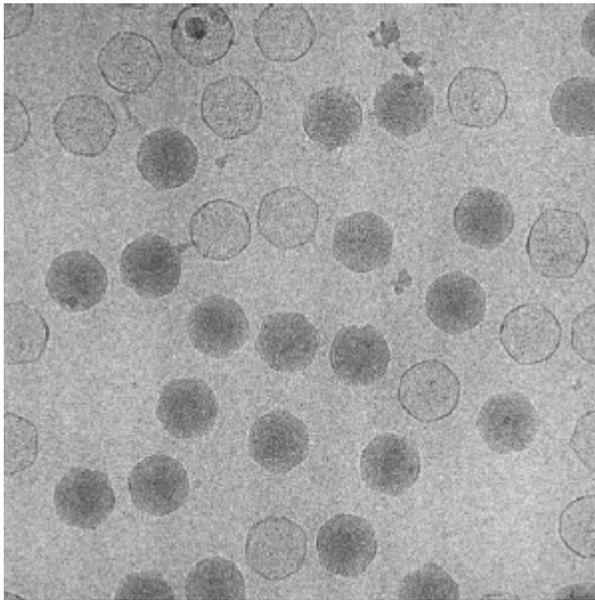
CPMV

CCMV

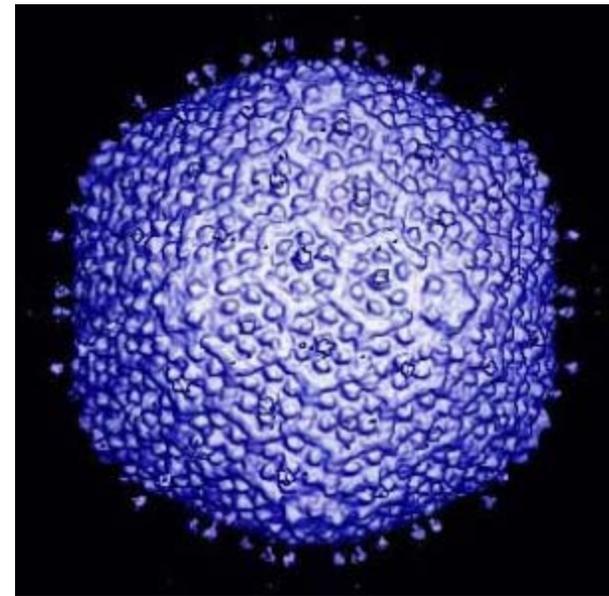
B19

500 Å

3D Reconstruction



2D



3D

Why CryoEM?

- Well suited for large macromolecules
- Resolution limit near 5Å
- Sample is frozen in vitreous ice and imaged at liquid nitrogen temperatures
- Imaging thousands of individual particle randomly orientated on a thin substrate
- Computer reconstructions to generate 3D structure

CryoEM

Sample : ~2-3 μl at 1-5 mg/ml

Specimen support: holey carbon film (1-2 μm)

Microscope: 200-300 keV with FEG

Defocus range: 1-3 μm underfocus

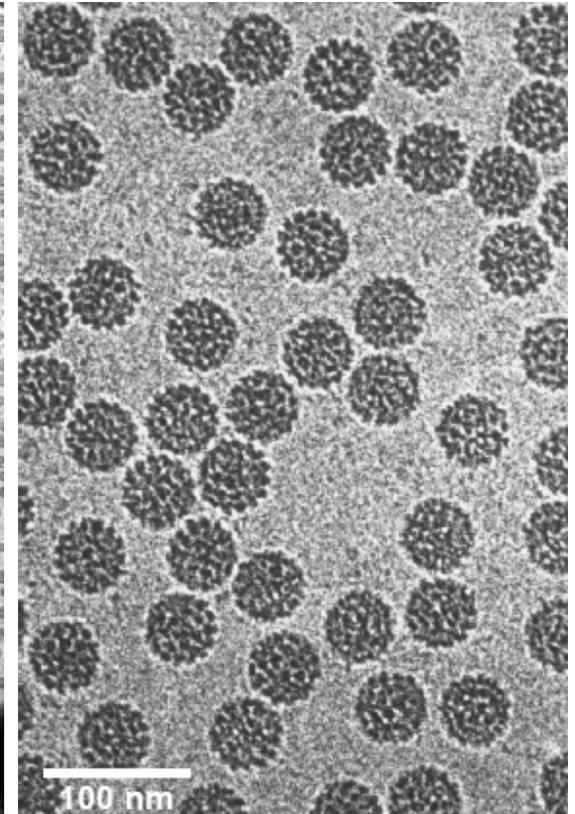
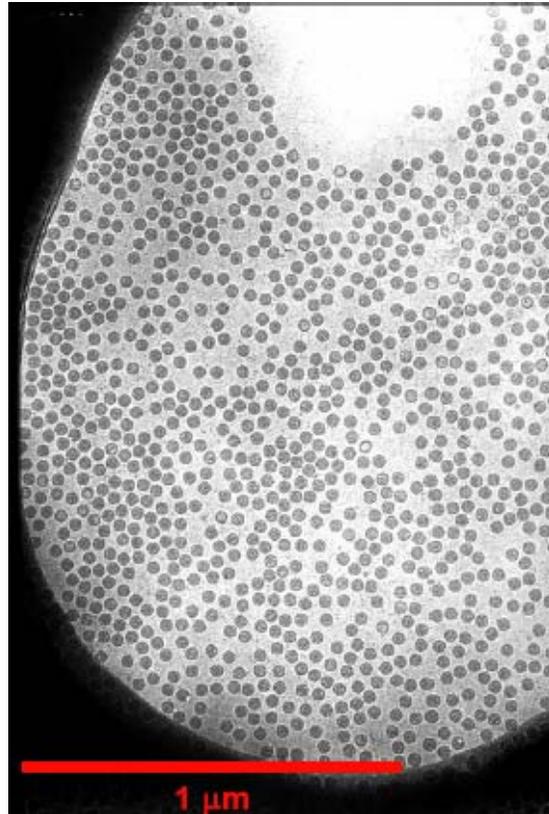
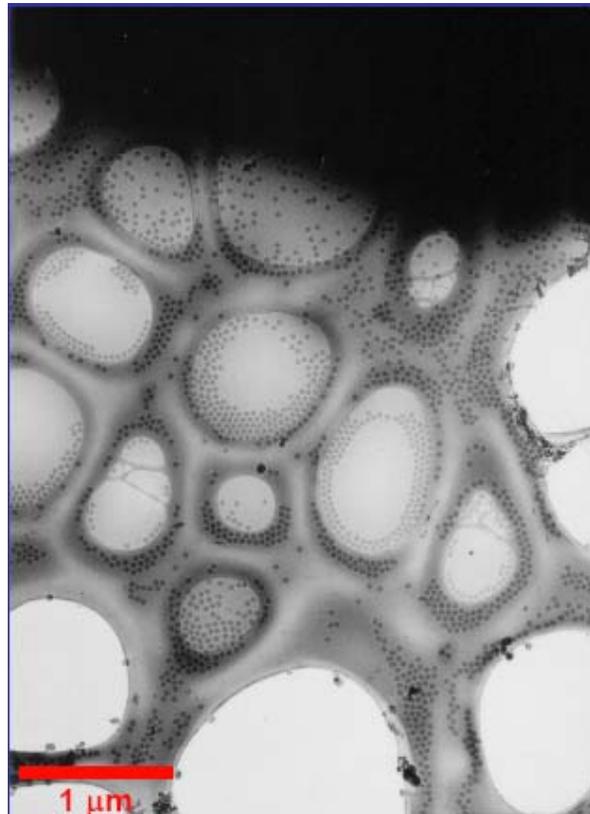
Dose: 10-20 $\text{e}^-/\text{\AA}^2$

Film: SO-163 (12 min, full strength)

Micrographs: 25-100

Particles: 10^3 - 10^4

Target resolution: 12 - 6 \AA



Basic Assumptions

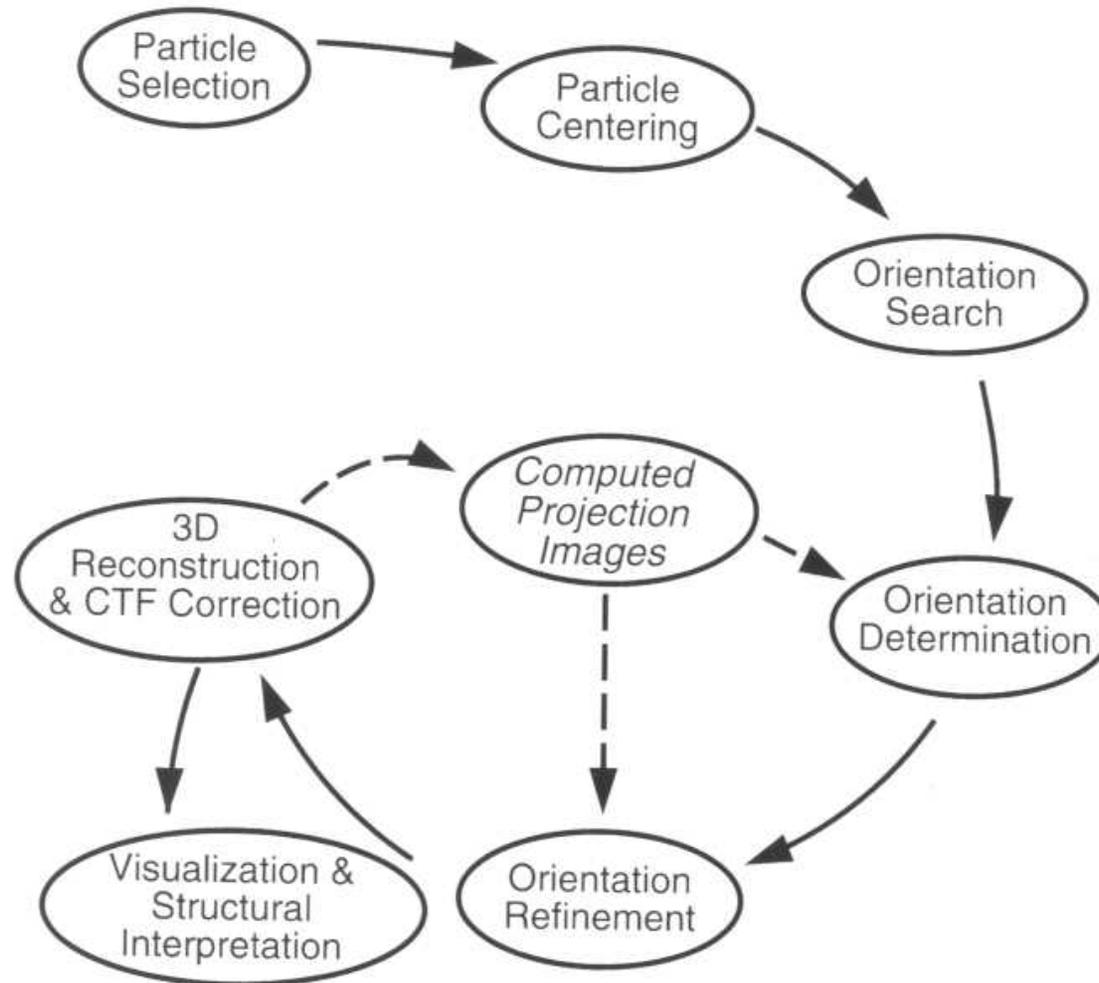
Specimen consists of stable particles with ‘identical’ structures
(else averaging is invalid)

Programs test for and *assume* presence of icosahedral (532)
symmetry

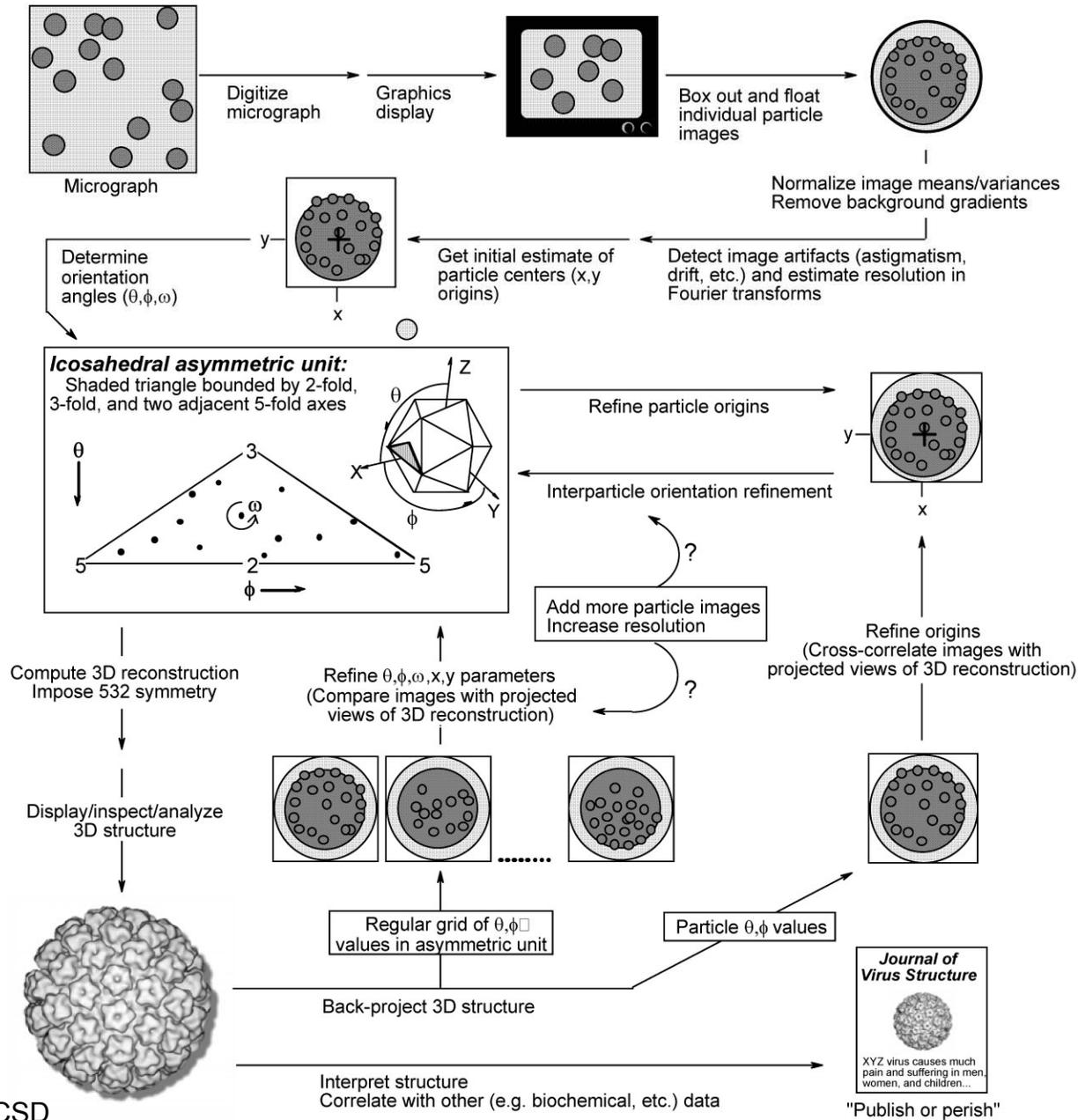
T (triangulation) symmetry is not incorporated into or
enforced by the 3D reconstruction algorithms

Hence, T symmetry emerges as a result of a properly performed
3D reconstruction analysis

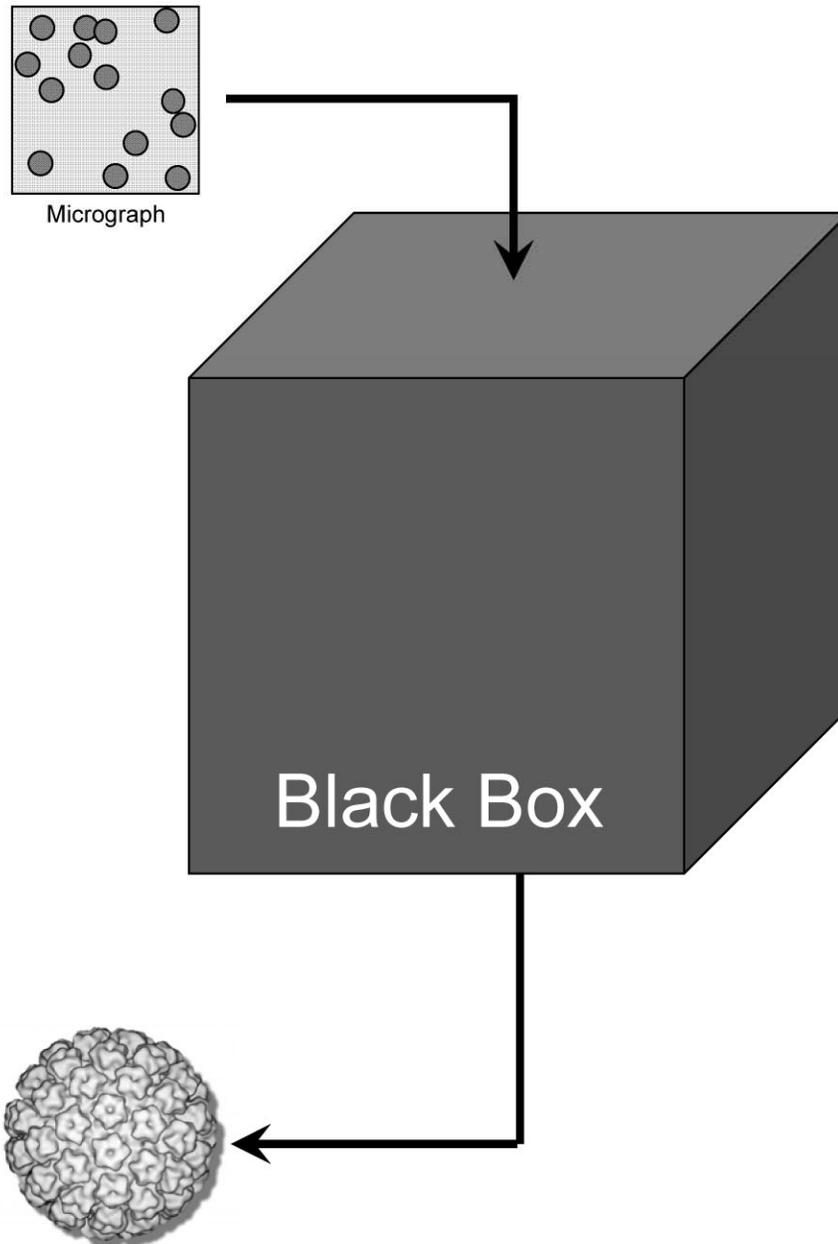
Overview of Reconstruction Scheme



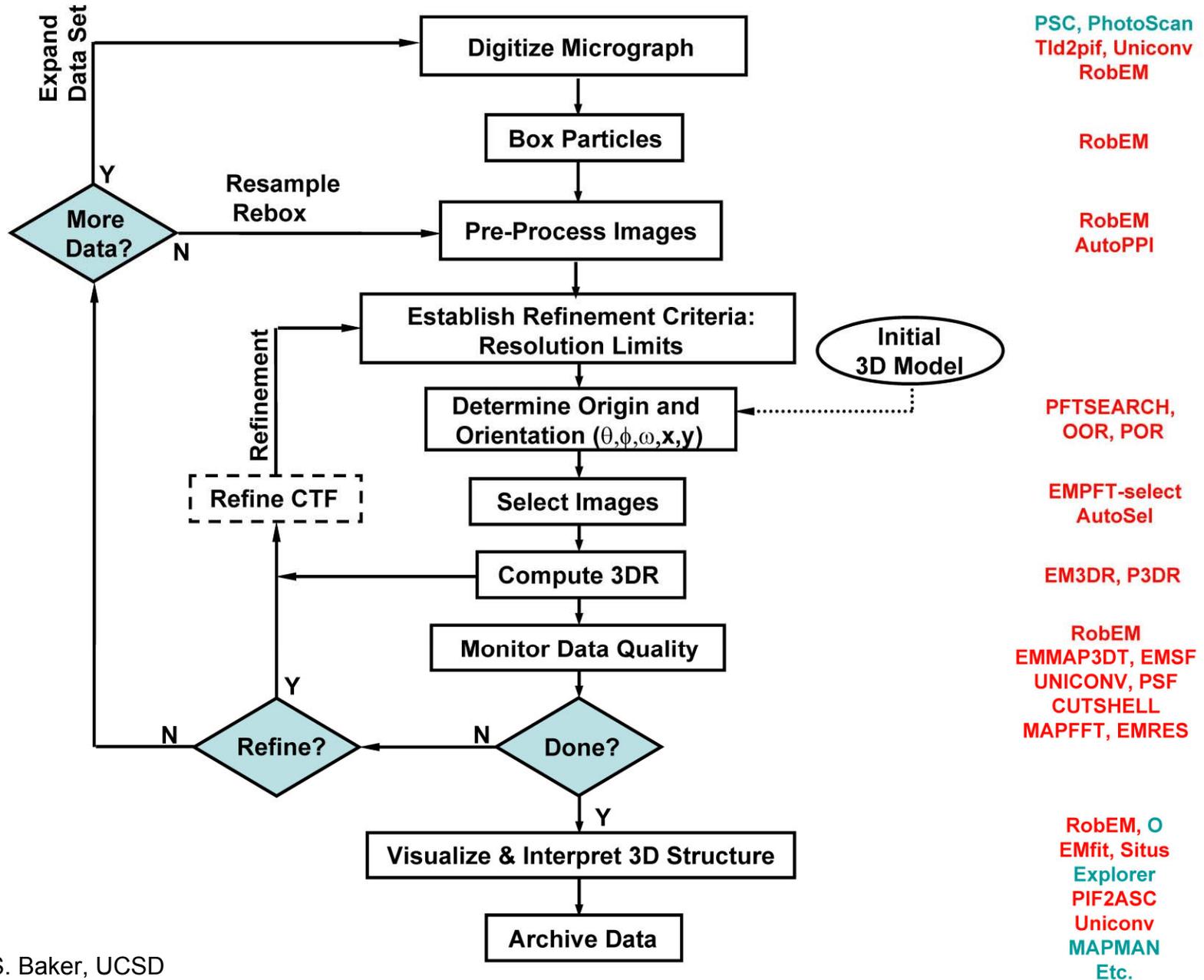
Icosahedral Particle Reconstruction Scheme



Icosahedral Particle Reconstruction Scheme



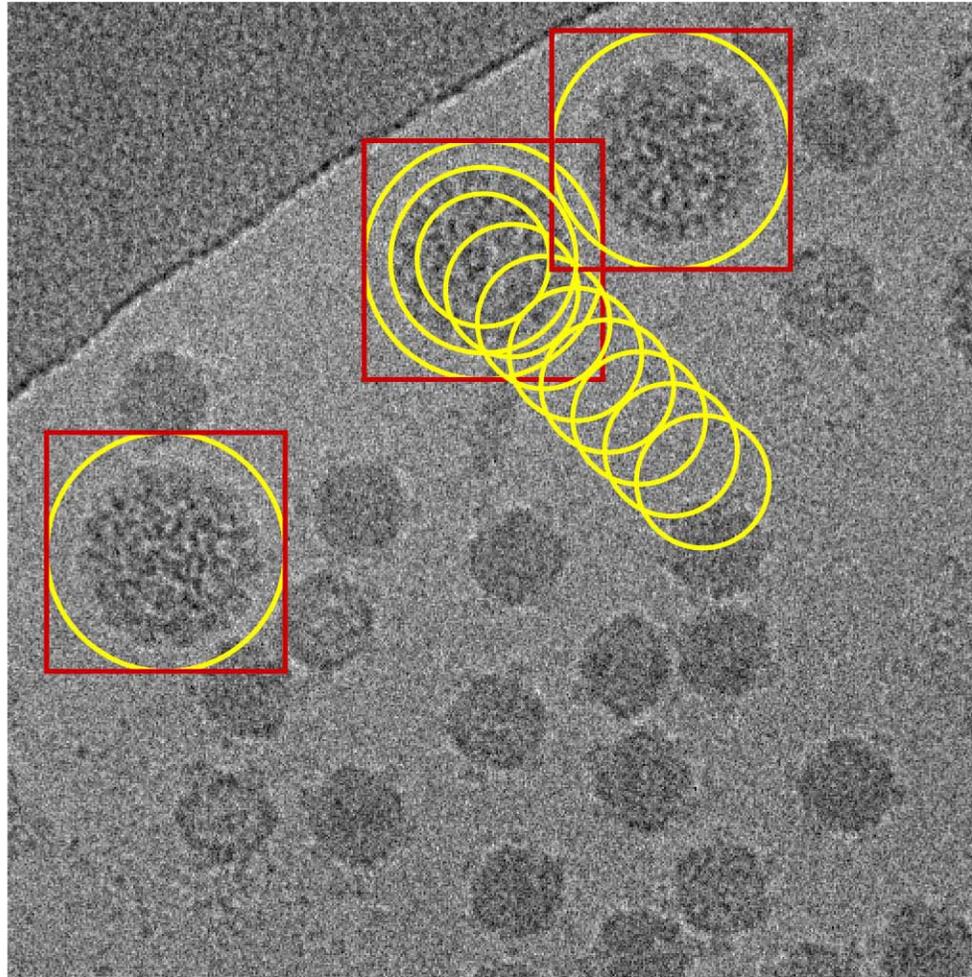
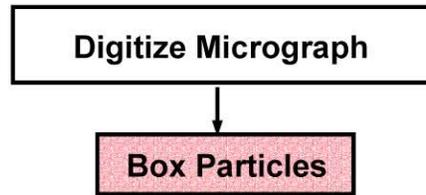
Icosahedral Particle Reconstruction Scheme



Icosahedral Particle Reconstruction Scheme

Digitize Micrograph

Icosahedral Particle Reconstruction Scheme

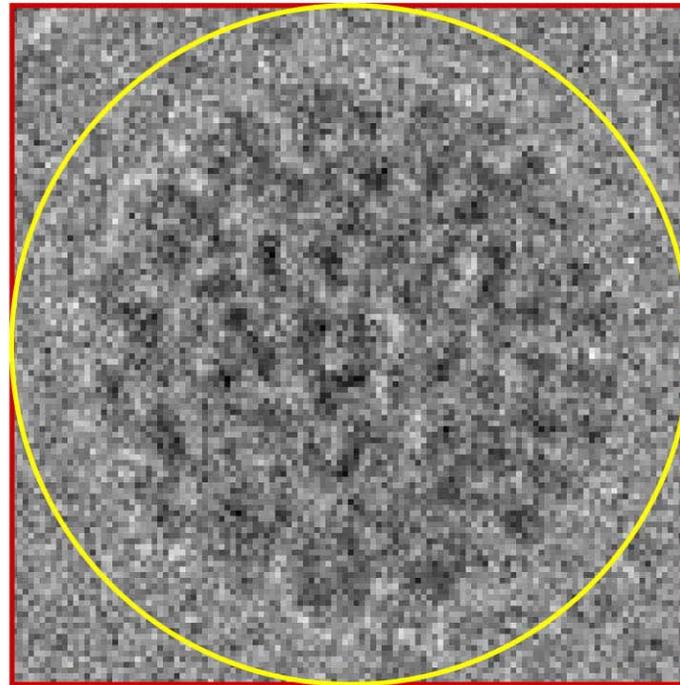


Icosahedral Particle Reconstruction Scheme

Digitize Micrograph

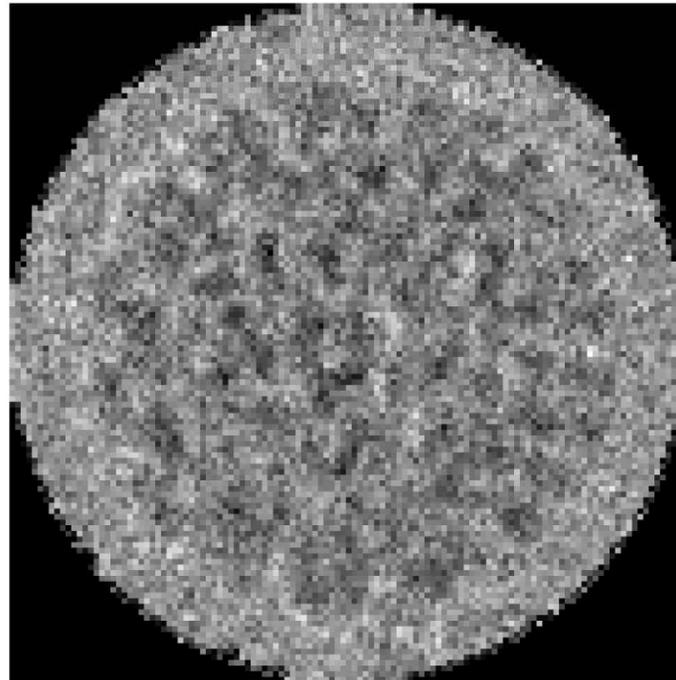
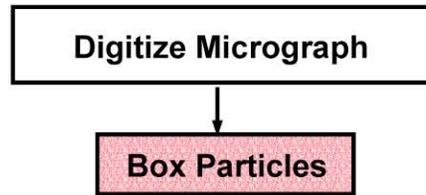


Box Particles



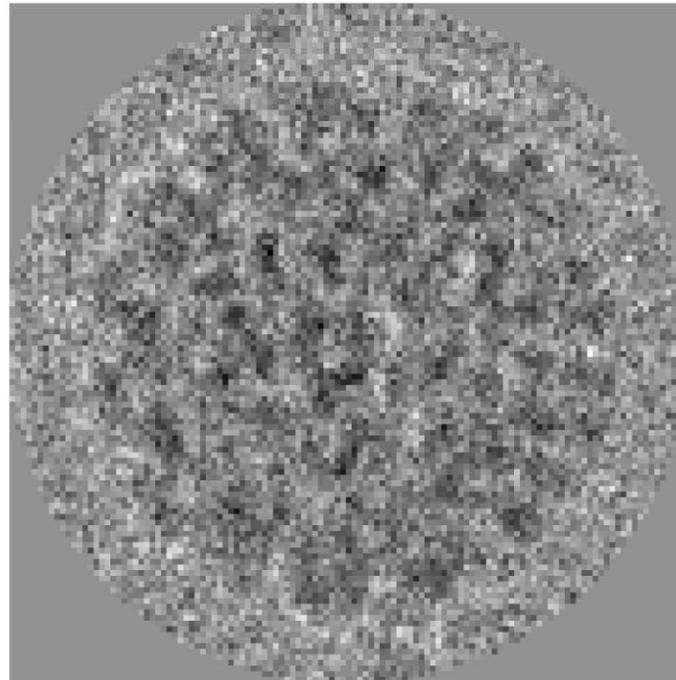
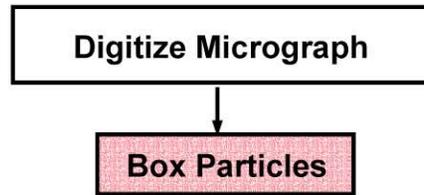
Extracted

Icosahedral Particle Reconstruction Scheme



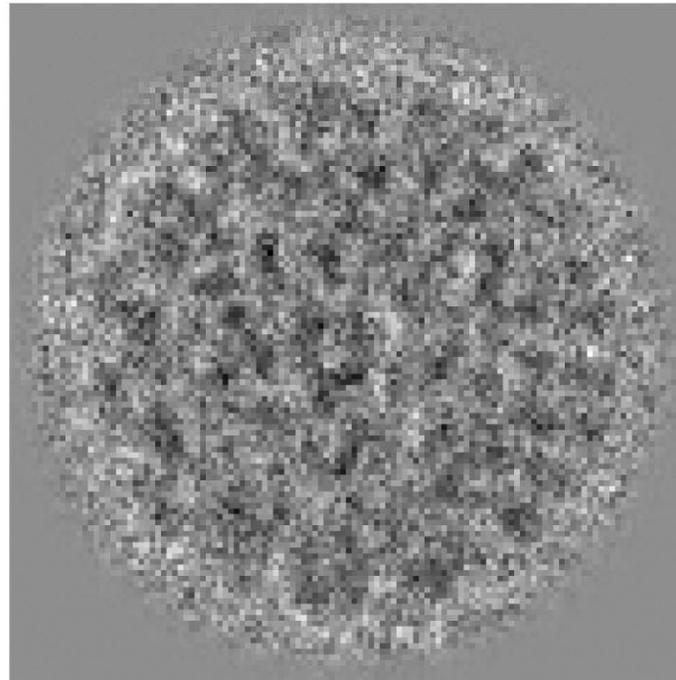
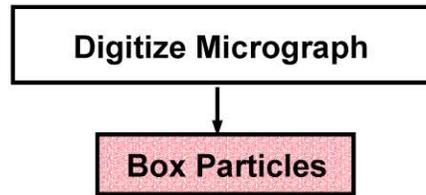
Masked

Icosahedral Particle Reconstruction Scheme



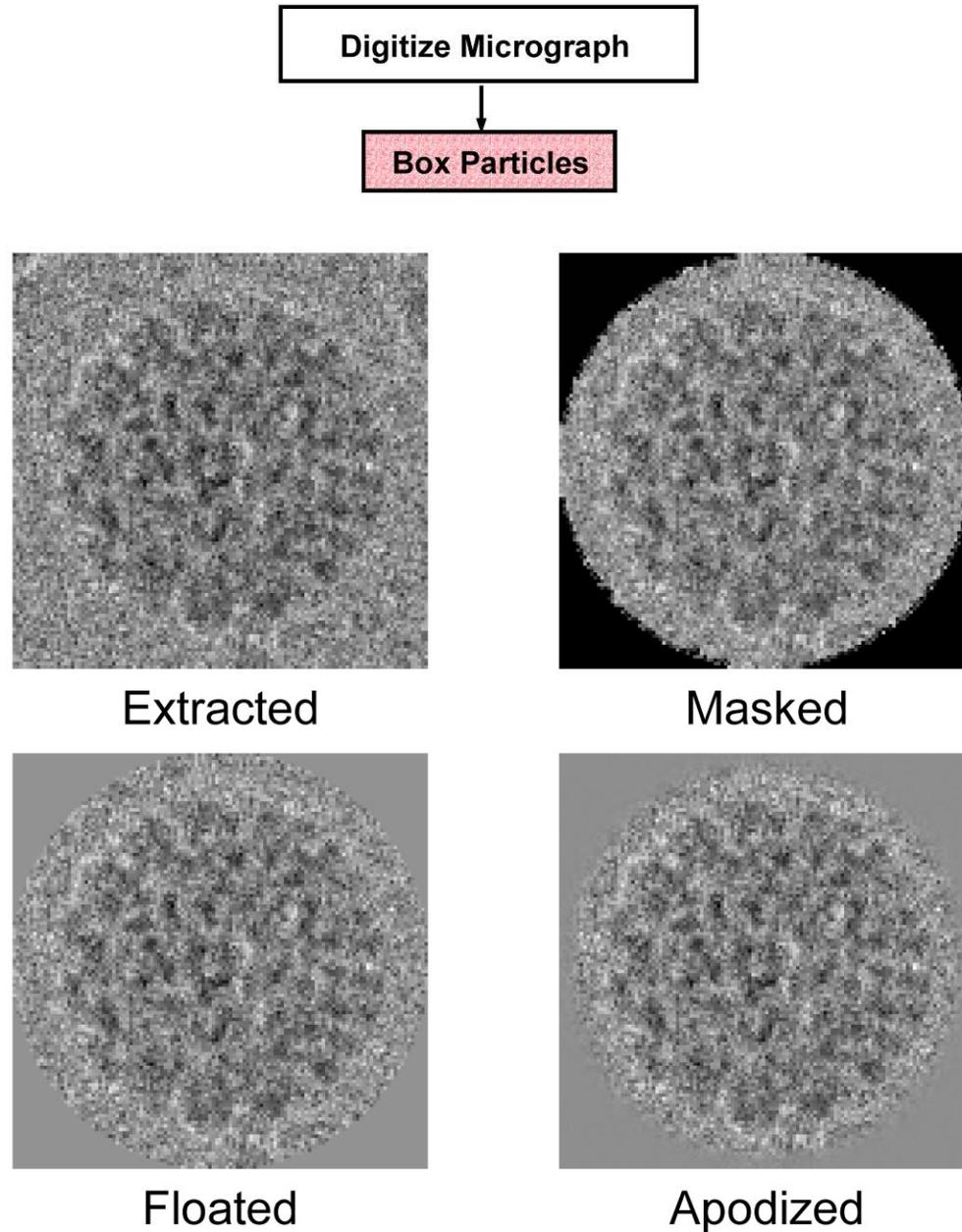
Floated

Icosahedral Particle Reconstruction Scheme

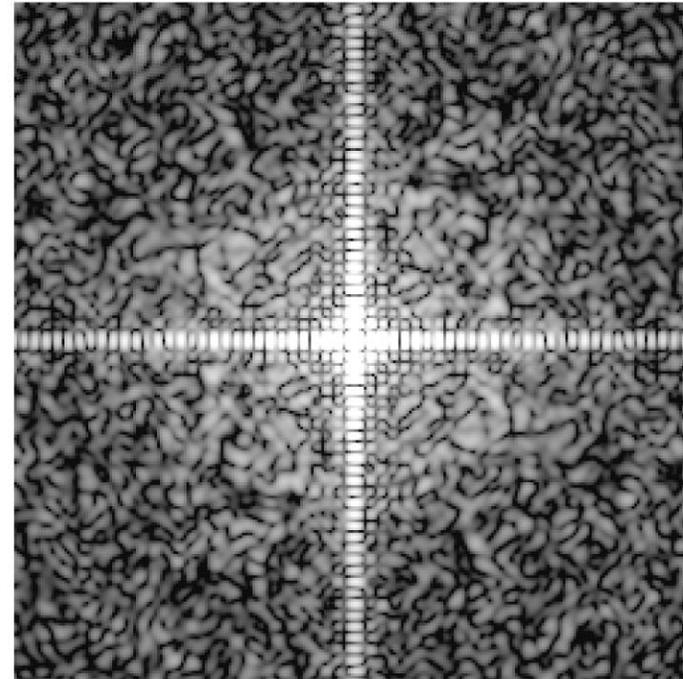
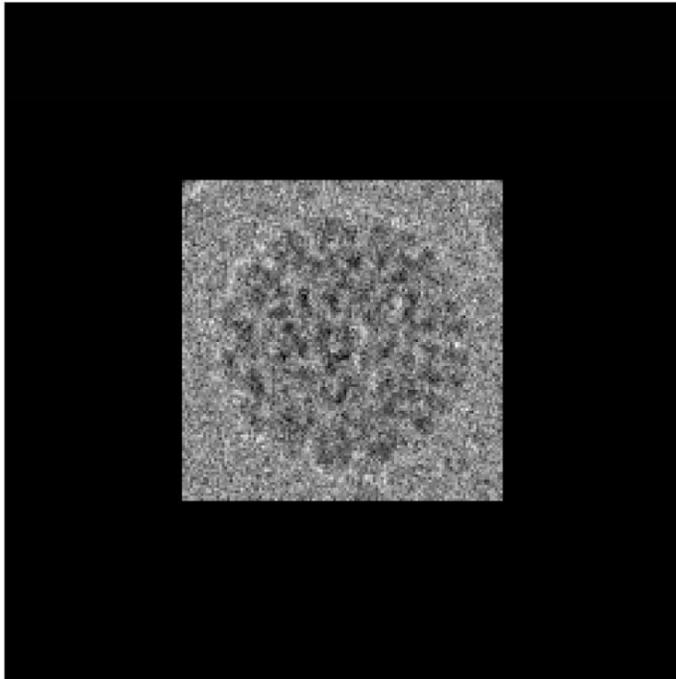
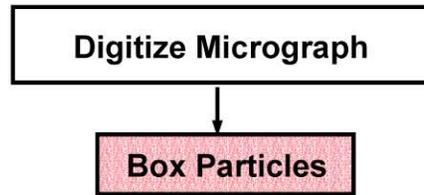


Apodized

Icosahedral Particle Reconstruction Scheme

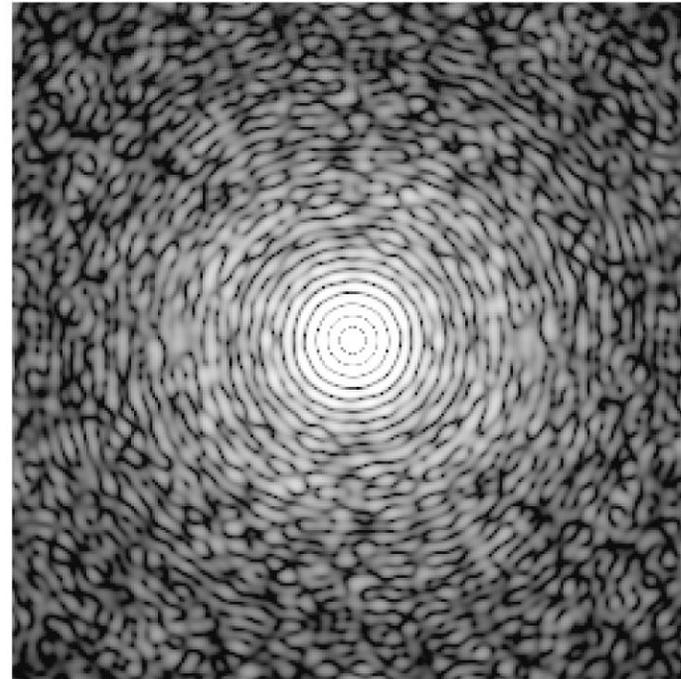
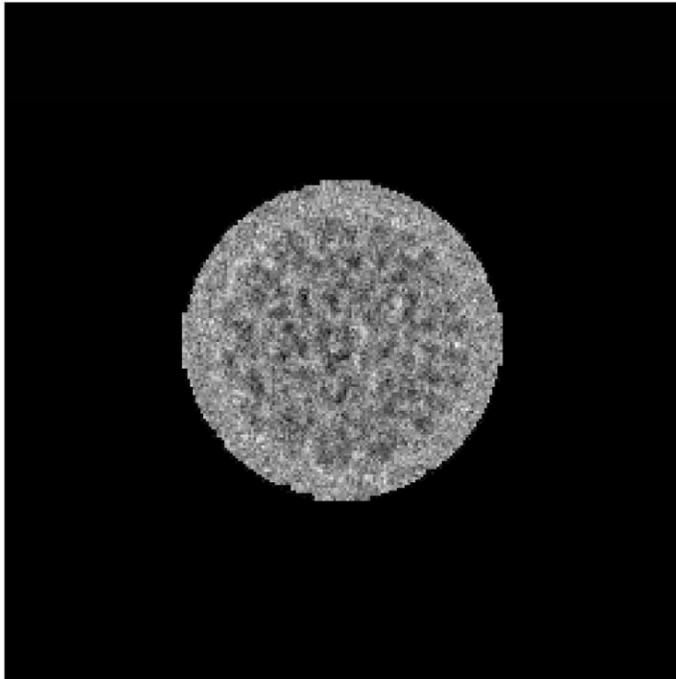
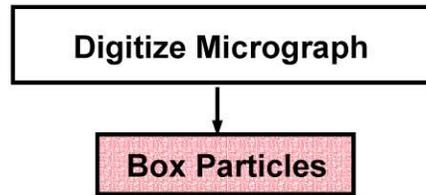


Icosahedral Particle Reconstruction Scheme



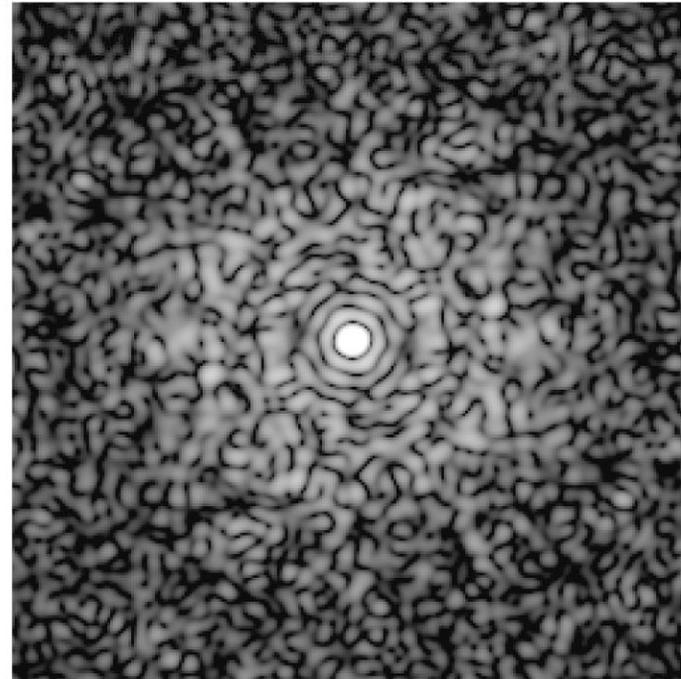
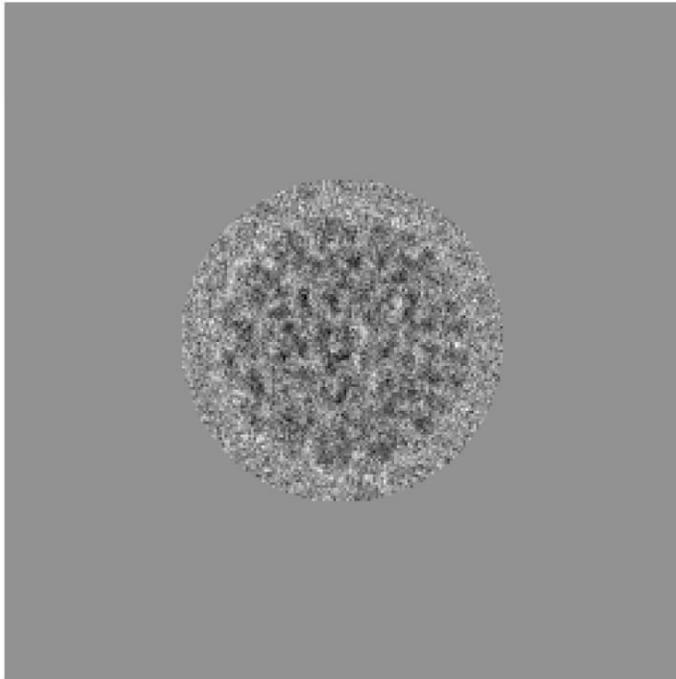
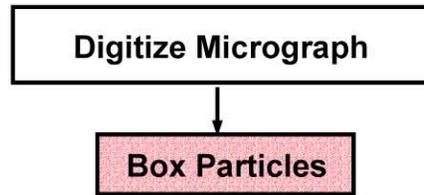
Square mask; unfloated

Icosahedral Particle Reconstruction Scheme



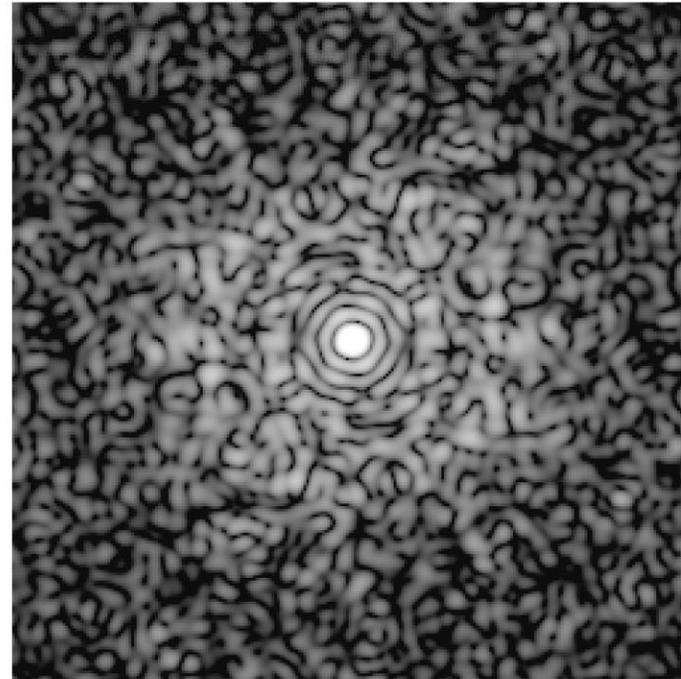
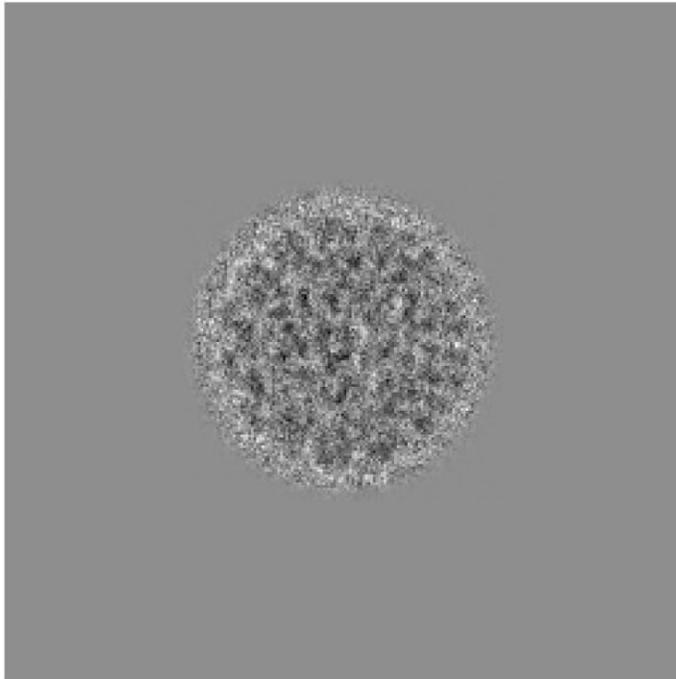
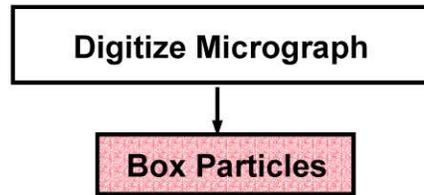
Circular mask; unfloated

Icosahedral Particle Reconstruction Scheme



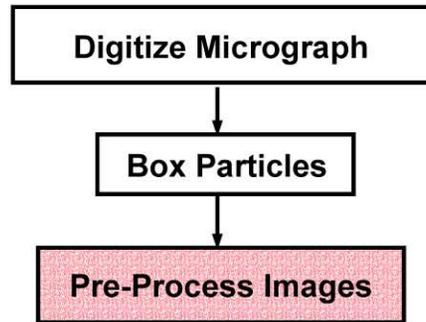
Circular mask; floated

Icosahedral Particle Reconstruction Scheme

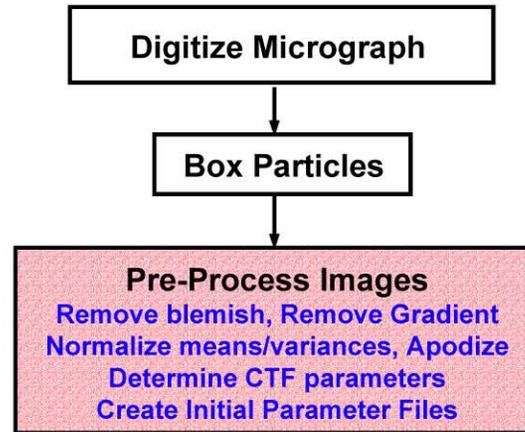


Circular mask; floated & apodized

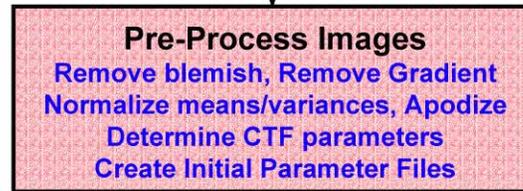
Icosahedral Particle Reconstruction Scheme



Icosahedral Particle Reconstruction Scheme

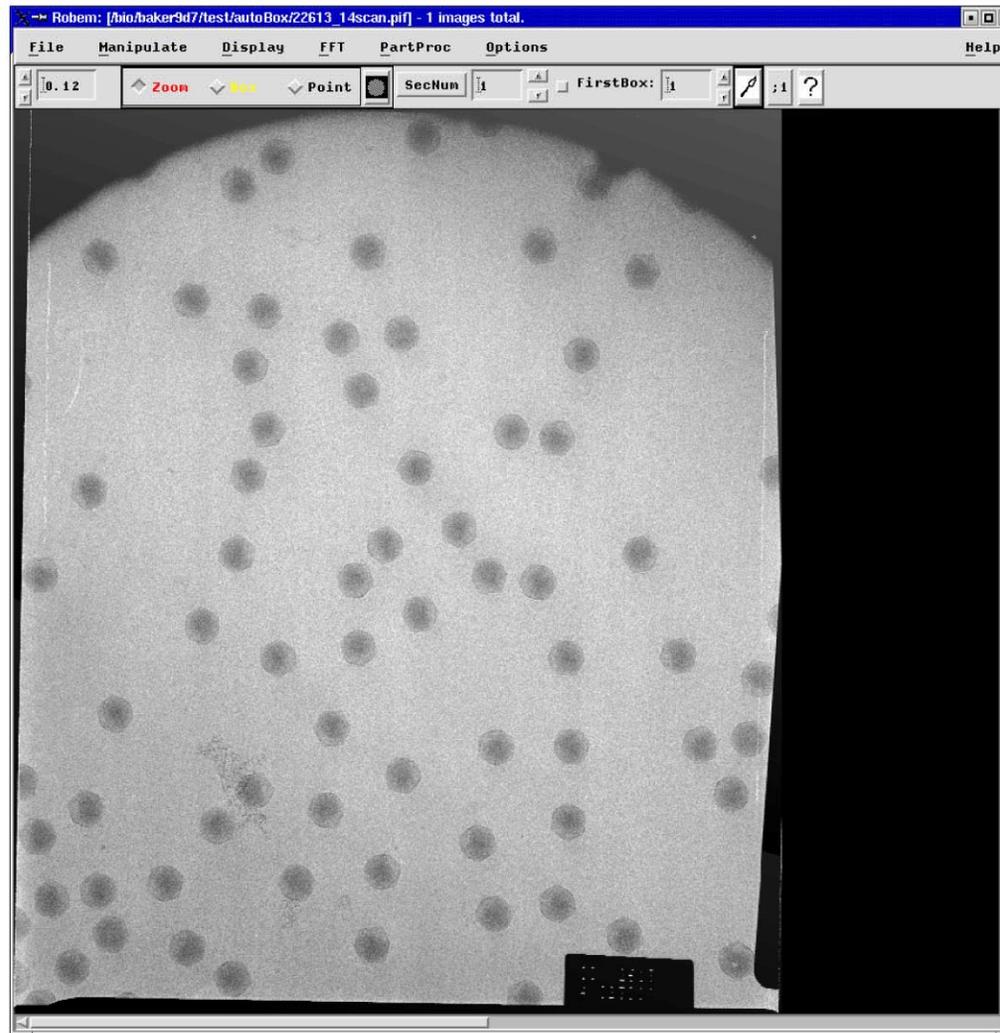


Icosahedral Particle Reconstruction Scheme



Icosahedral Particle Reconstruction Scheme

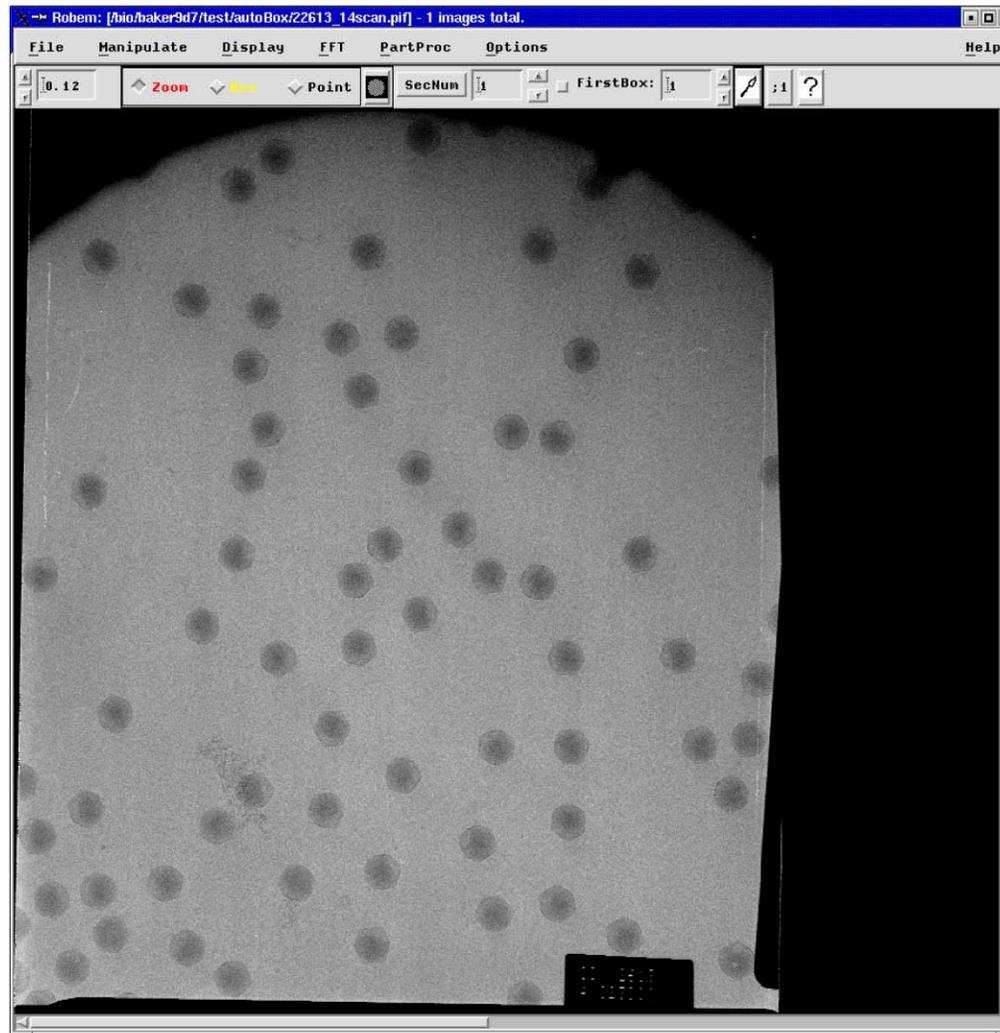
↓
Pre-Process images
Remove blemish, Remove Gradient
Normalize means/variances, Apodize
Determine CTF parameters
Create Initial Parameter Files



Gradient
removed

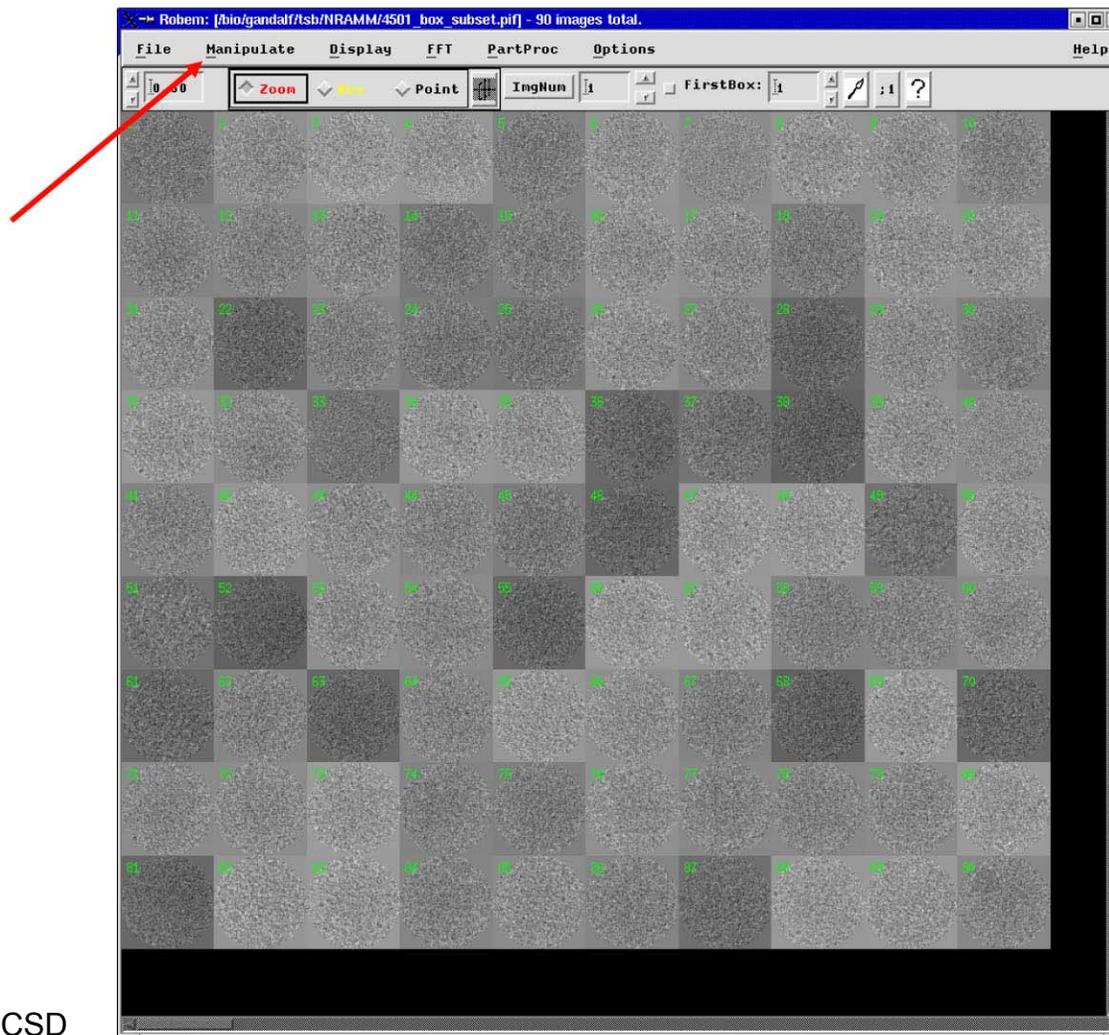
Icosahedral Particle Reconstruction Scheme

↓
Pre-Process images
Remove blemish, Remove Gradient
Normalize means/variances, Apodize
Determine CTF parameters
Create Initial Parameter Files



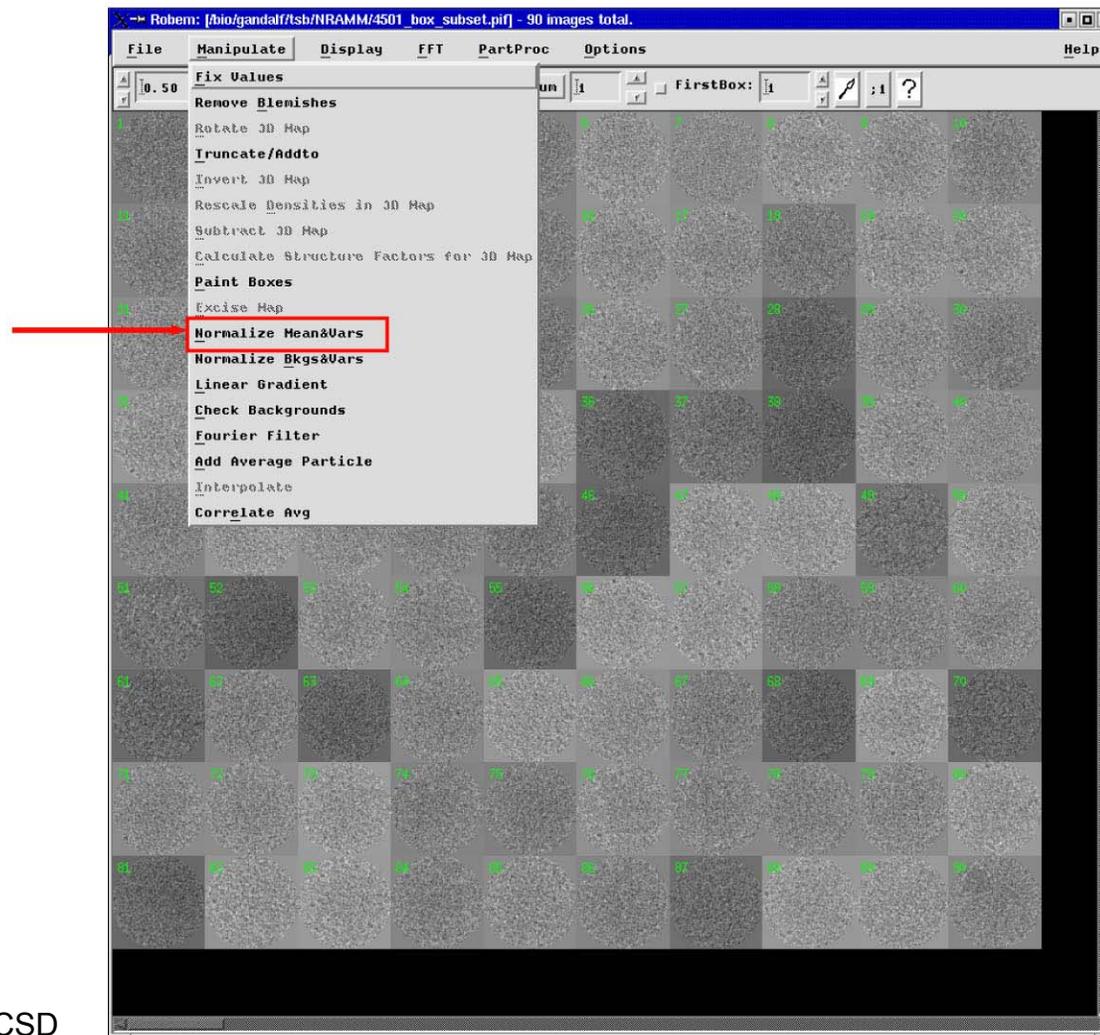
Icosahedral Particle Reconstruction Scheme

↓
Pre-Process images
Remove bias, Remove Gradient
Normalize means/variances, Apodize
Determine CTF parameters
Create Initial Parameter Files

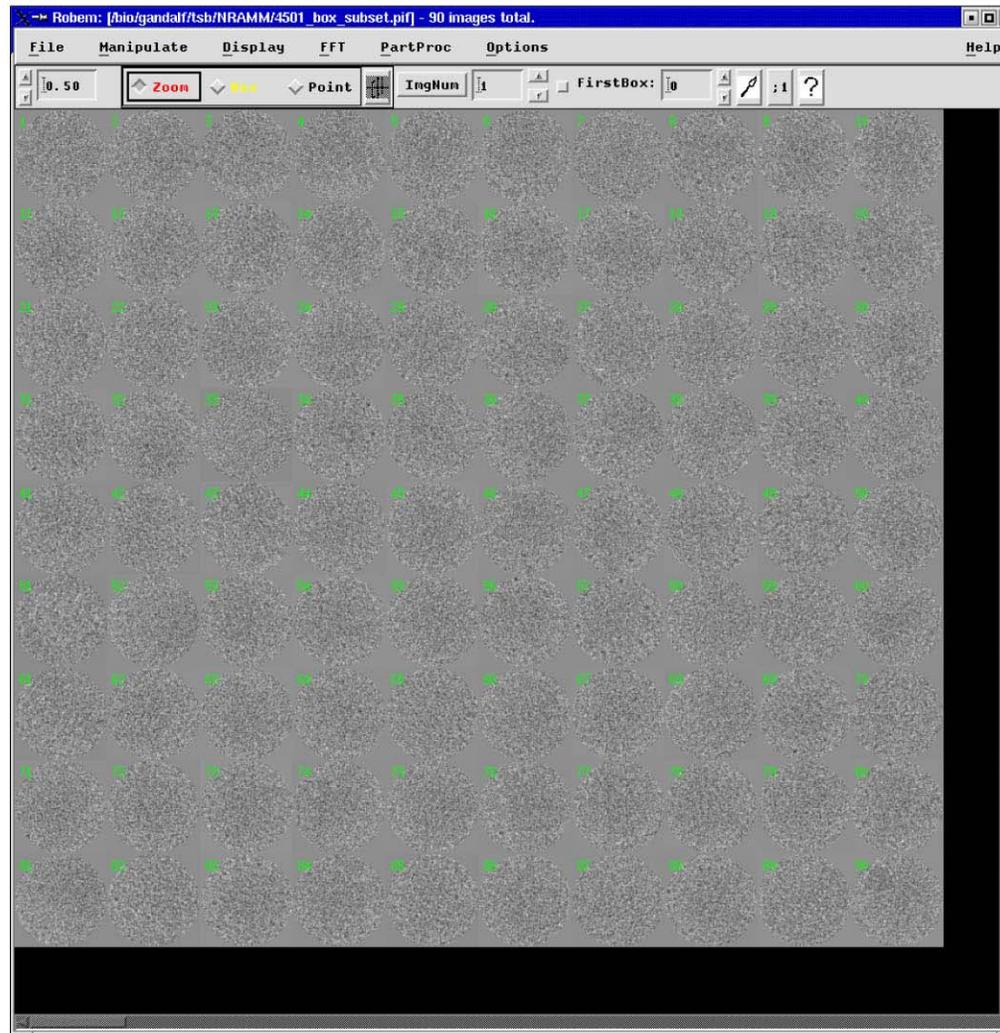
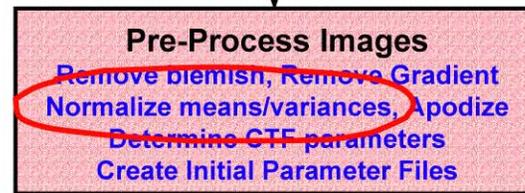


Icosahedral Particle Reconstruction Scheme

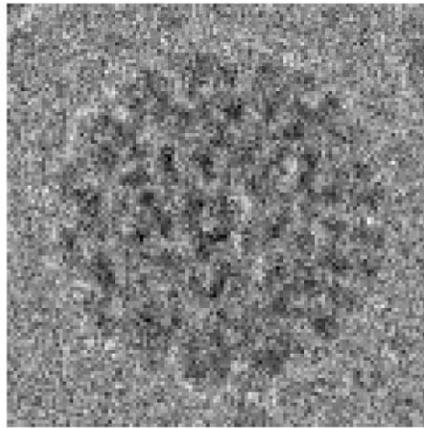
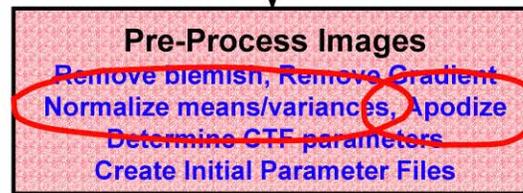
Pre-Process Images
Remove blenish, Remove Gradient
Normalize means/variances, Apodize
Determine CTF parameters
Create Initial Parameter Files



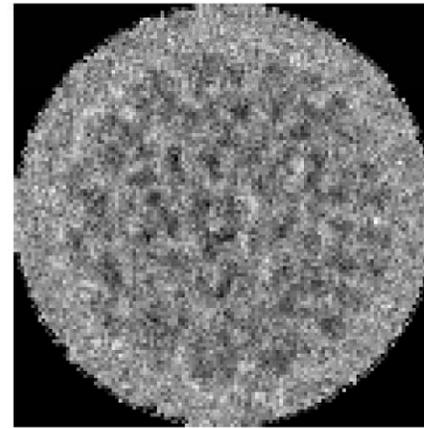
Icosahedral Particle Reconstruction Scheme



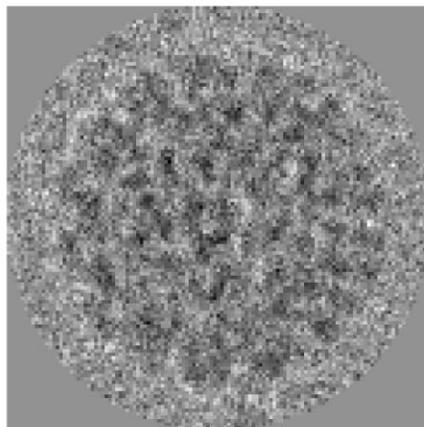
Icosahedral Particle Reconstruction Scheme



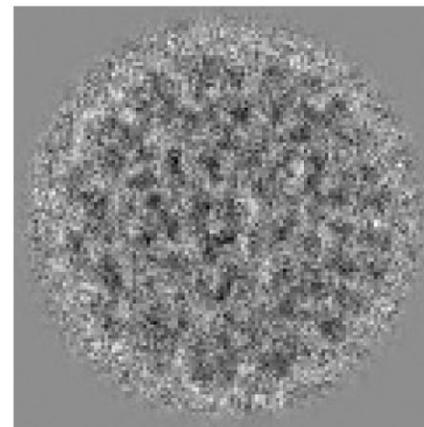
Extracted



Masked



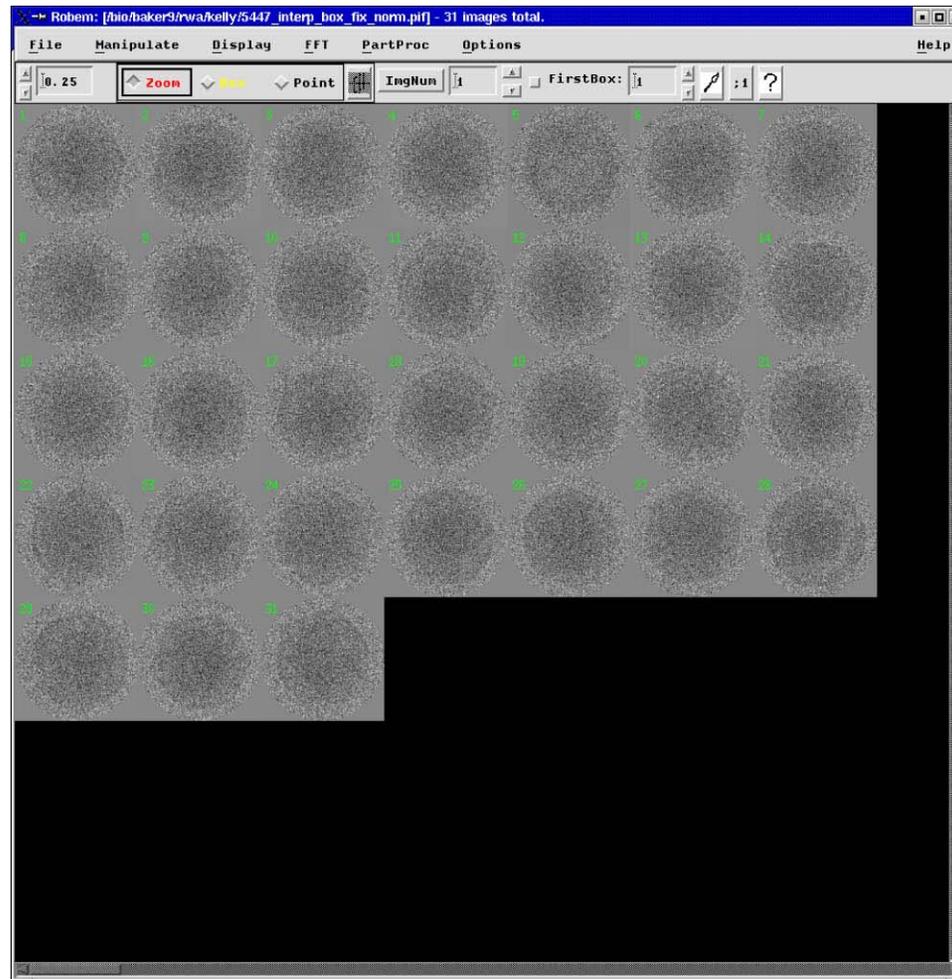
Floated



Apodized

Icosahedral Particle Reconstruction Scheme

↓
Pre-Process Images
Remove blemish, Remove Gradient
Normalize means/variances, Apodize
Determine CTF parameters
Create Initial Parameter Files



Icosahedral Particle Reconstruction Scheme

Pre-Process Images
Remove blemish, Remove Gradient
Normalize means/variances, Apodize
Determine CTF parameters
Create Initial Parameter Files

FFT - CTF Estimation

Linear
Log
Non-linear

1024 x 1024
512 x 512
256 x 256
128 x 128

Intensity
Recalc FFT

Lock Scrolls

Min 7.61
Max 8.52
TFac 0
Range Min: 3.16 Max: 12.69

Back Transform
Pointer Detail
Generate Default Param File
Defocus Refinement

Overlay Intensity:
Zoom
Disp 1D CA
Contour 1D CircAvg

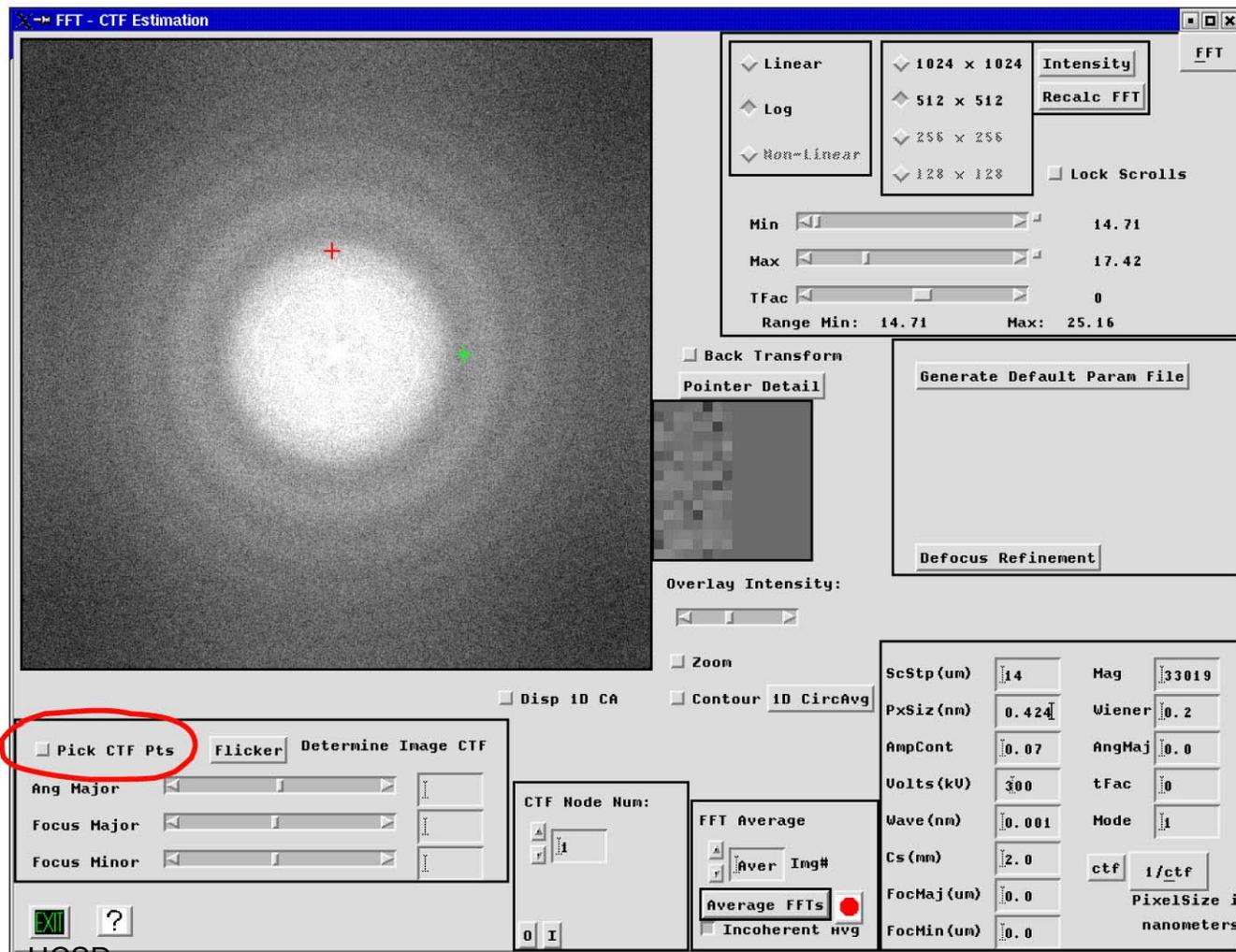
Pick CTF Pts Flicker Determine Image CTF
Ang Major
Focus Major
Focus Minor

CTF Node Num:
Average FFTs
Average FFTs
Incoherent hvv

ScStp (um) 14 Mag 33019
PxSiz (nm) 0.424 Wiener 0.2
AmpCont 0.07 AngMaj 0.0
Volts (kV) 300 tFac 0
Wave (nm) 0.001 Mode 1
Cs (nm) 2.0 ctf 1/ctf
FocMaj (um) 0.0 PixelSize i
FocMin (um) 0.0 nanometers

Icosahedral Particle Reconstruction Scheme

Pre-Process Images
Remove blemish, Remove Gradient
Normalize means/variances, Apodize
Determine CTF parameters
Create Initial Parameter Files



Icosahedral Particle Reconstruction Scheme

Pre-Process Images
Remove blemish, Remove Gradient
Normalize means/variances, Apodize
Determine CTF parameters
Create Initial Parameter Files

The screenshot displays the 'FFT - CTF Estimation' software interface. The main window shows a power spectrum with concentric rings and a red circle highlighting the central region. The interface includes various control panels for image processing, CTF estimation, and parameter adjustment. A red circle highlights the 'Defocus Refinement' button in the lower right section.

Linear (selected) | 1024 x 1024 | Intensity | FFT
Log | 512 x 512 | Recalc FFT
Non-linear | 256 x 256 | Lock Scrolls
| 128 x 128

Min: 14.71 | Max: 17.42 | TFac: 0 | Range Min: 14.71 | Max: 25.16

Back Transform | Pointer Detail | Generate Default Param File

Defocus Refinement

Overlay Intensity: | Zoom | Disp 1D CA | Contour 1D CircAvg

Pick CTF Pts | Flicker | Determine Image CTF

Ang Major: 359.4 | Focus Major: 2.16 | Focus Minor: 3.31

CTF Node Num: 1

FFT Average | Aver | Ing# | Average FFTs | Incoherent | Hvg

ScStp (um): 14 | Mag: 33019 | PxSiz (nm): 0.424 | Wiener: 0.2 | AmpCont: 0.07 | AngMaj: 359.4 | Volts (kV): 300 | tFac: 0 | Wave (nm): 0.001 | Mode: 1 | Cs (nm): 2.0 | ctf: 1/ctf | FocMaj (um): 2.16 | FocMin (um): 3.31 | PixelSize i nanometers

Icosahedral Particle Reconstruction Scheme

Pre-Process Images
Remove blemish, Remove Gradient
Normalize means/variances, Apodize
Determine CTF parameters
Create Initial Parameter Files

The screenshot shows a software interface with two main windows: "FFT - CTF Estimation" and "Automatic Defocus Refinement".

FFT - CTF Estimation window:

- Left panel: Radio buttons for "Linear", "Log", and "Non-linear".
- Right panel: "Min", "Max", "TFac", and "Range" sliders.
- Center: A large FFT image with concentric rings and a red circle overlaid on the center.
- Bottom: "Back Transf" checkbox, "Pointer Detail" window (showing a zoomed-in detail of the center), "Overlay Intensity" slider, "Zoom" checkbox, "Disp 1D CA" checkbox, and "Contour 1D Ci" checkbox.
- Bottom-left: "Pick CTF Pts" checkbox, "Flicker" button, "Determine Image CTF" button, and input fields for "Ang Major" (359.4), "Focus Major" (2.16), and "Focus Minor" (3.31).
- Bottom-center: "CTF Node Num:" input field (1), "FFT Average" section with "Aver" and "Ing#" buttons, "Average FFTs" checkbox (checked), and "Incoherent hvg" checkbox.
- Bottom-right: "AmpCont" (0.07), "Volts (kV)" (300), "Wave (nm)" (0.001), "Cs (nm)" (2.0), "FocMaj (um)" (2.16), "FocMin (um)" (3.31), "AngMaj" (359.4), "tFac" (0), "Mode" (1), "ctf" (1/ctf), and "PixelSize i nanometers" label.

Automatic Defocus Refinement window:

- Left: Input fields for a0 through a4, all set to 0.
- Right: Input fields for b0 through b4, all set to 0.
- Center: A circular arrow icon.
- Below: "Highest Resolution (Angs):" input field (8.480000), "Estimated Defocus (um):" input field, and "Chi squared:" input field.
- Bottom: "Select different plots:" section with checkboxes for "Circular Average" (checked), "S", "S - B", "ctf^2", "D", and "B". "Zoom FFT image" checkbox is also present.
- Bottom-right: "Change Overlay Intensity to see CTF curves." text.
- Bottom: "Estimate the Defocus *1" button (circled in red), "Update FFT screen" button, and "EXIT" and "?" buttons.

Icosahedral Particle Reconstruction Scheme

Pre-Process Images
Remove blemish, Remove Gradient
Normalize means/variances, Apodize
Determine CTF parameters
Create Initial Parameter Files

The screenshot shows a software interface with two main windows: "FFT - CTF Estimation" and "Automatic Defocus Refinement".

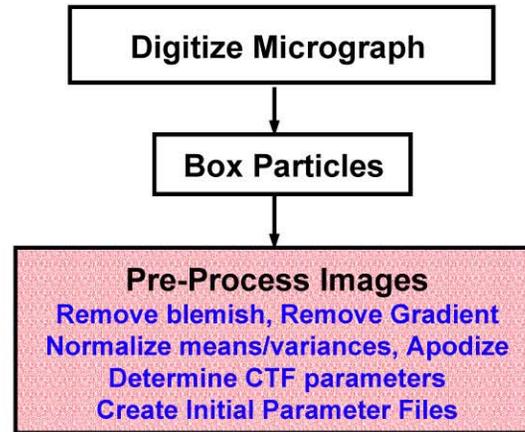
FFT - CTF Estimation window:

- Left panel: Radio buttons for "Linear", "Log", and "Non-linear".
- Right panel: "Min", "Max", "TFac", and "Range" sliders.
- Center: A 2D FFT image with concentric rings and a red circle overlay.
- Bottom: "Back Transf", "Pointer Detail" (with a zoomed-in inset), "Overlay Intensity" slider, "Zoom", "Disp 1D CA", and "Contour 1D Ci" checkboxes.
- Bottom-left: "Pick CTF Pts" checkbox, "Flicker" button, "Determine Image CTF" button, and input fields for "Ang Major" (359.4), "Focus Major" (2.204), and "Focus Minor" (2.204).
- Bottom-center: "CTF Node Num:" input field (value 1), "Average FFTs" button, "Average FFTs" checkbox (checked), and "Incoherent hv" checkbox.
- Bottom-right: "AmpCont" (0.07), "Volts (kV)" (300), "Wave (nm)" (0.001), "Cs (nm)" (2.0), "FocMaj (um)" (2.20), "FocMin (um)" (2.20), "AngMaj" (359.4), "tFac" (0), "Mode" (1), "ctf" (1/ctf), and "PixelSize i nanometers" field.

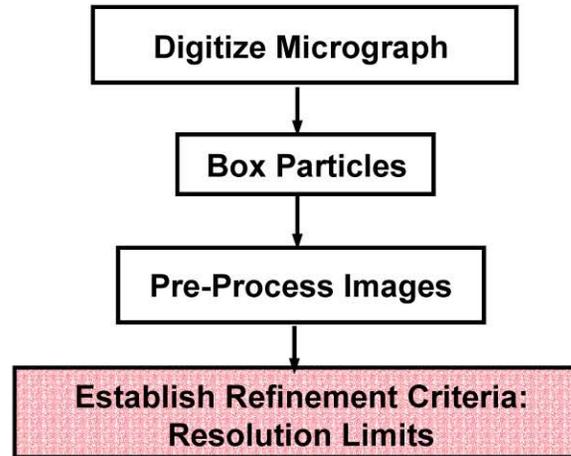
Automatic Defocus Refinement window:

- Parameters: a0: 1.739071E+00, a1: -6.500993E-01, a2: 3.357709E-05, a3: -1.871776E-01, a4: 3.767466E-10; b0: 2.172054E-01, b1: 2.077333E-03, b2: -4.360161E-01, b3: 2.043404E-07, b4: -3.037543E-11.
- Buttons: "Circular Average", "S", "S - B", "ctf^2", "B", "Zoom FFT image", "Estimate the Defocus *1" (circled in red), "Update FFT screen".
- Output: "Highest Resolution (Angs):" 8.480000, "Estimated Defocus (um):" 2.204652, "Chi squared:" 0.111960.
- Text: "Select different plots:", "Change Overlay Intensity to see CTF curves."

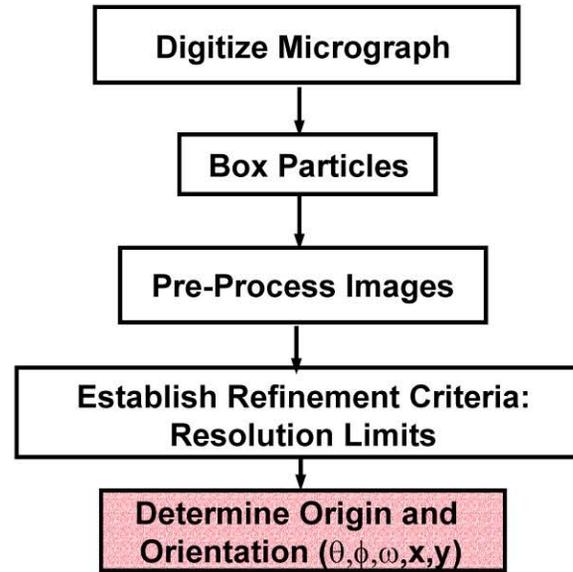
Icosahedral Particle Reconstruction Scheme



Icosahedral Particle Reconstruction Scheme

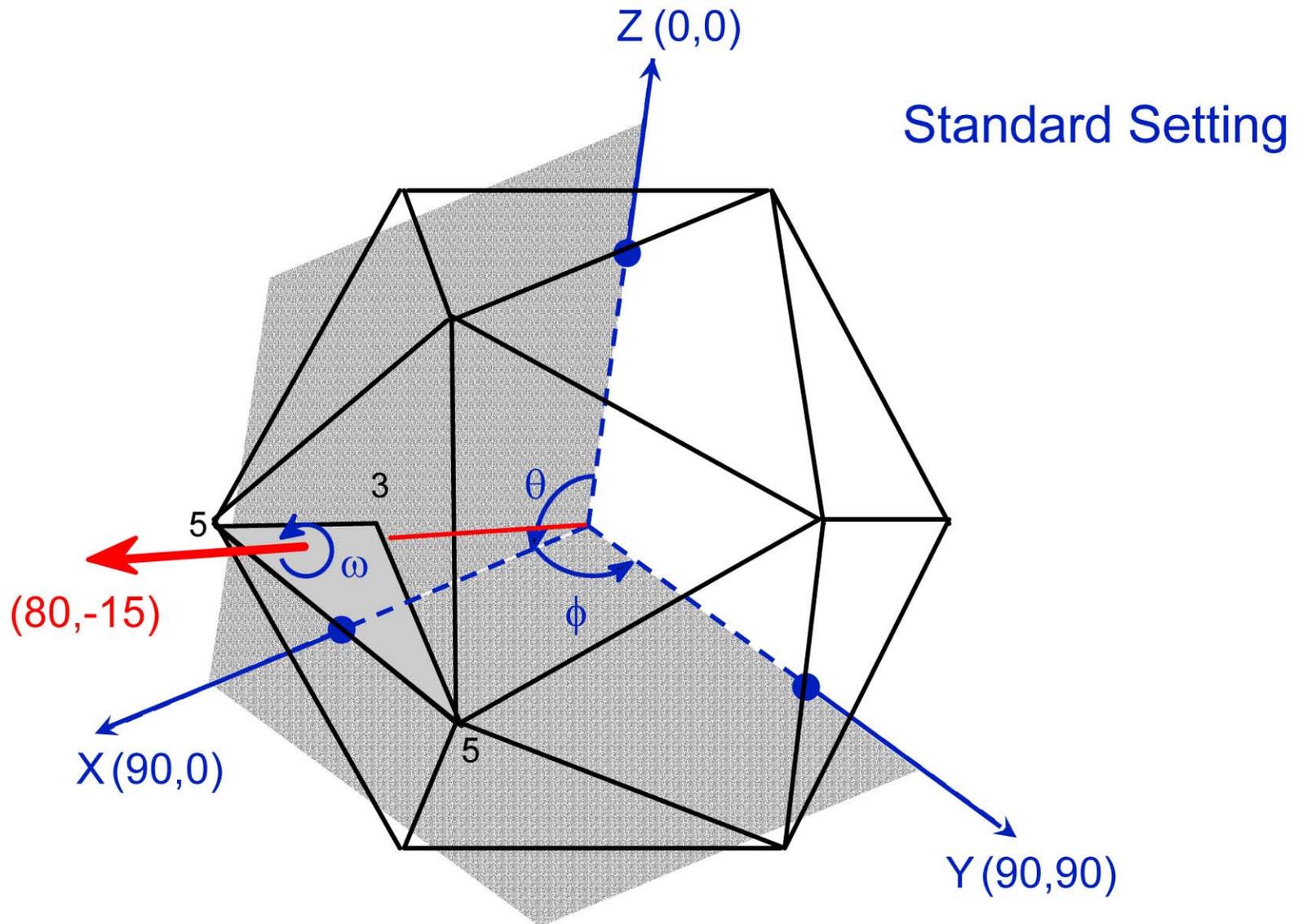


Icosahedral Particle Reconstruction Scheme

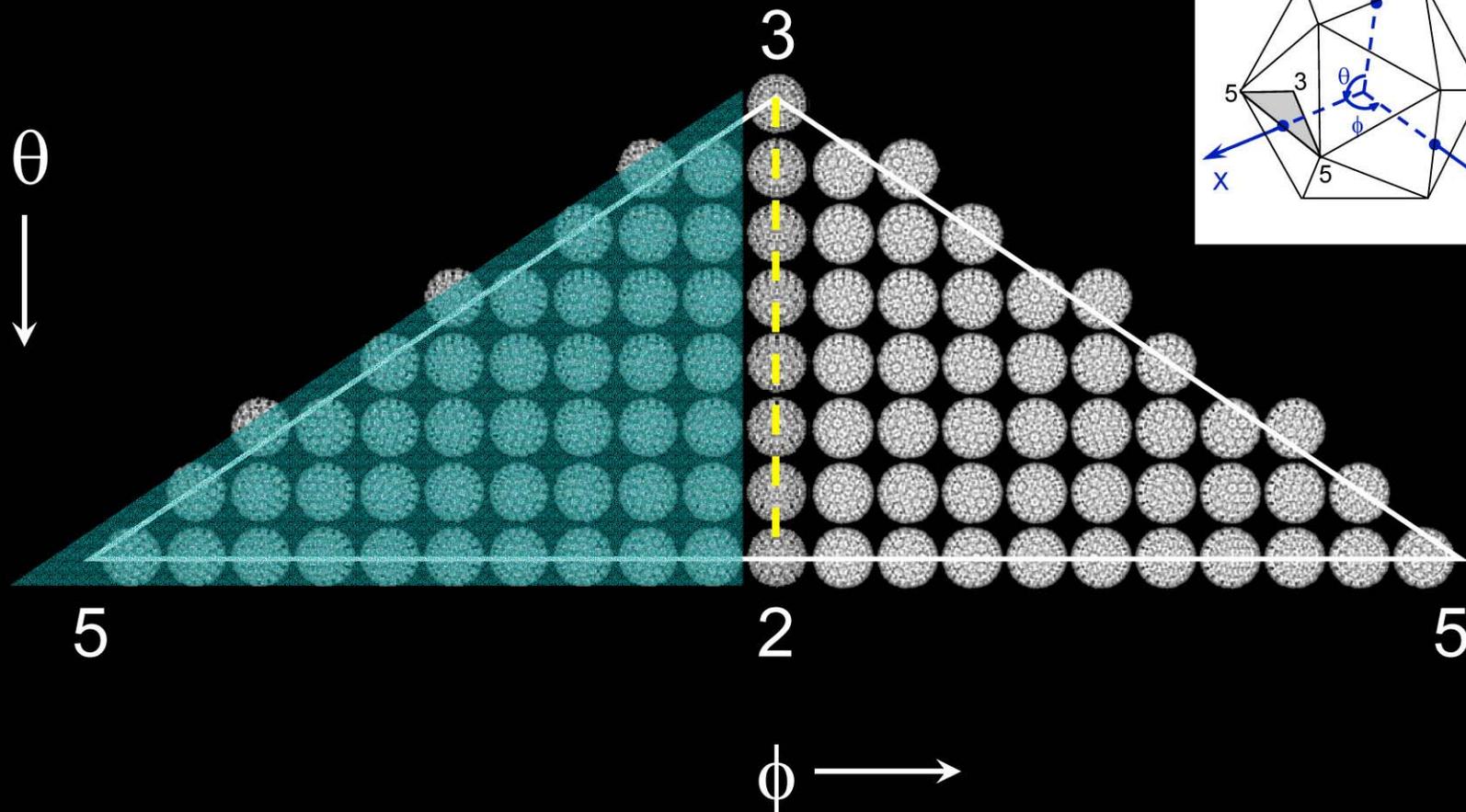


Determining Origin and Orientation

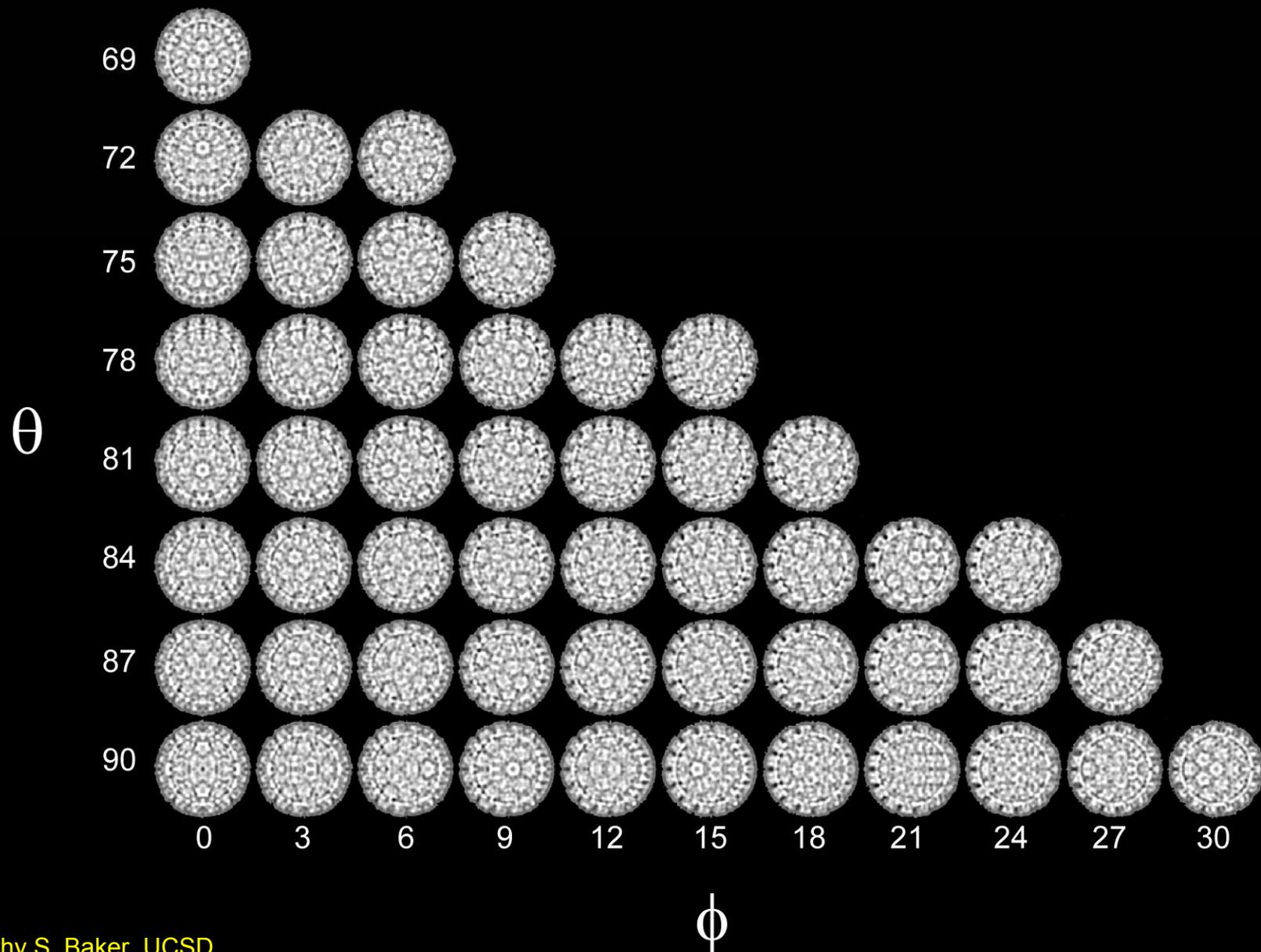
Convention of Coordinate System



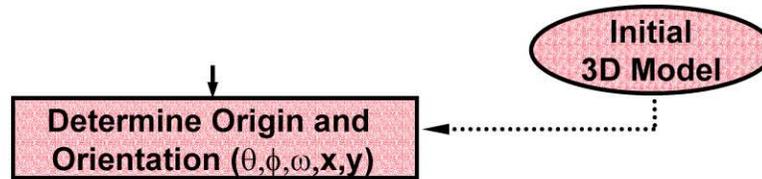
BPV Projections: Icosahedral ASU



BPV Projections: $\frac{1}{2}$ Icosahedral ASU



Icosahedral Particle Reconstruction Scheme



How do we determine the ($\theta, \phi, \omega, x, y$) parameters?

Two methods:

1. Common lines

New or unknown structure

2. Model-based (template) matching

General features of structure are known or a crude model can be generated

Icosahedral Particle Reconstruction Scheme

↓
Determine Origin and
Orientation ($\theta, \phi, \omega, x, y$)

Common Lines

The ‘gospel’ according to Tony Crowther (*Phil. Trans. R. Soc. Lond. B.*(1971) **261:221-230**)

“[Common lines] arise as follows:”

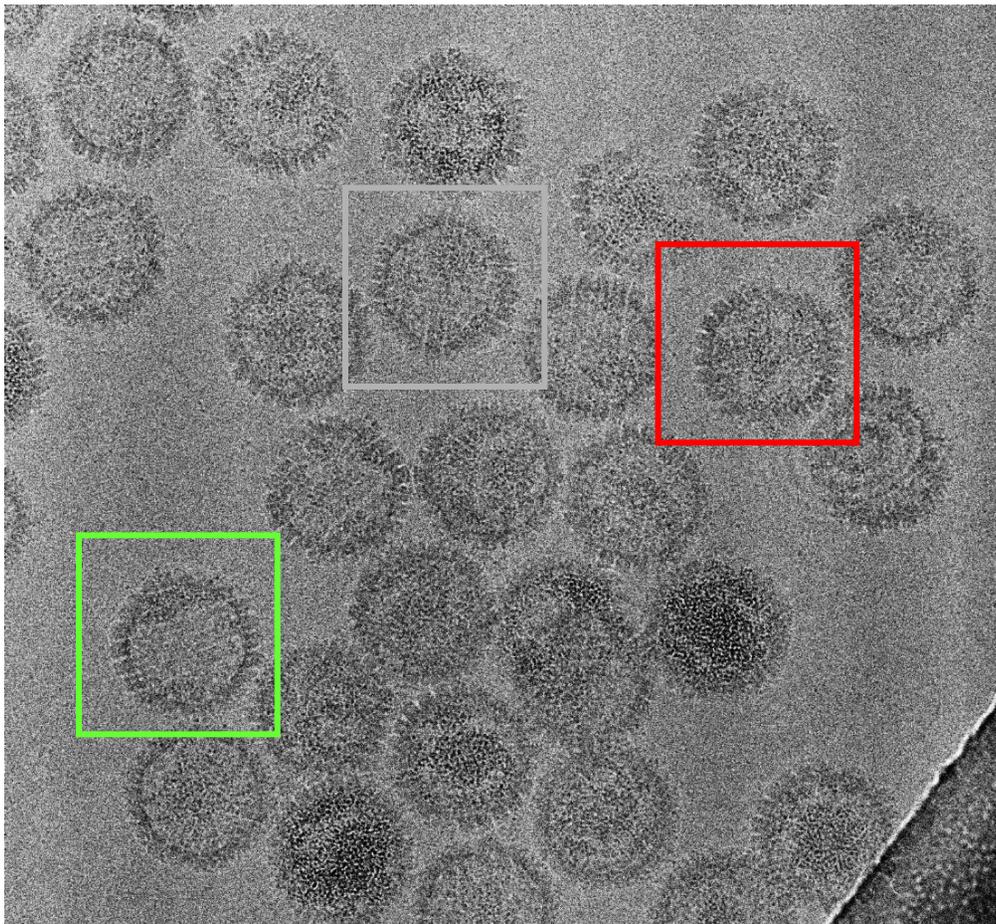
*“An observed section of the transform intersects an identical symmetry-related section in a **line**, along which the transform must have the **same value in both sections**”*

“The common line lies in the original section.”

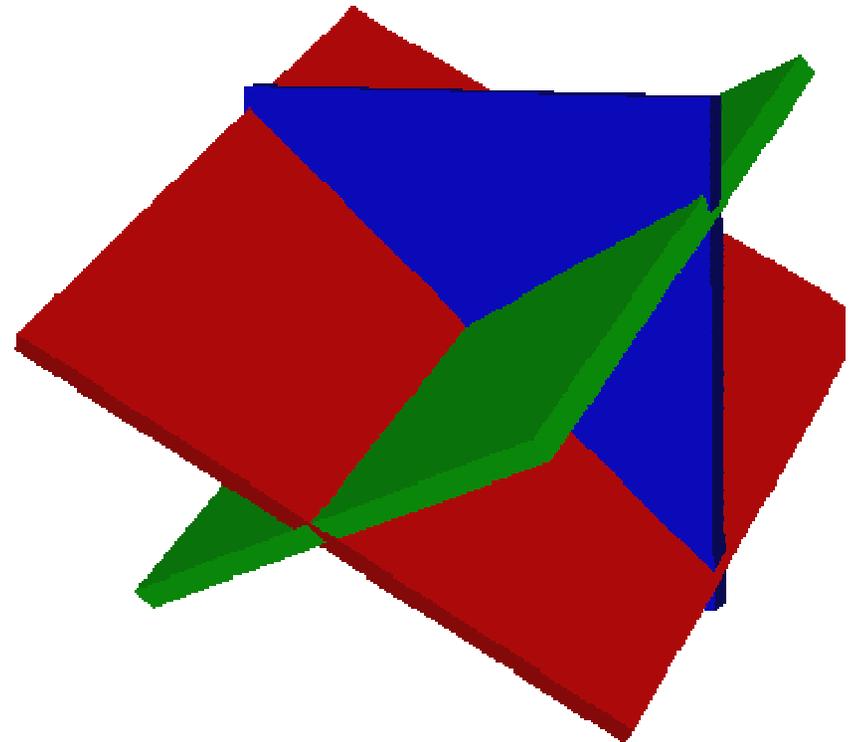
*“However, regarded as lying in the symmetry-related section it must have been generated by the symmetry operation from **some other line** in the original section.”*

Common Lines

Electron Images of
Virus Particles



Equivalent data in
Fourier space

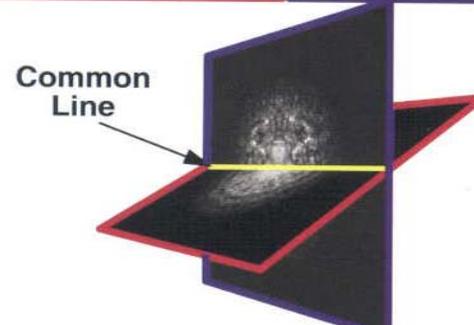
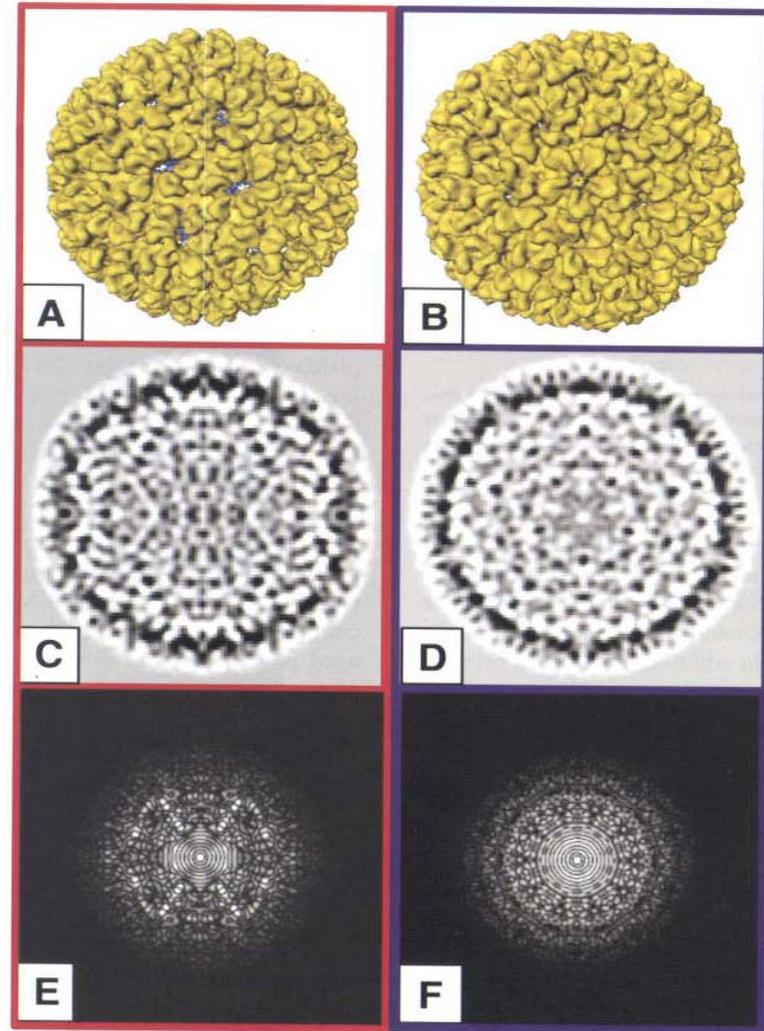


Common Lines

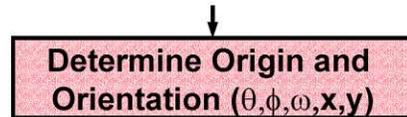
3D Object

Projection Image

Fourier Transform



Icosahedral Particle Reconstruction Scheme



Common Lines

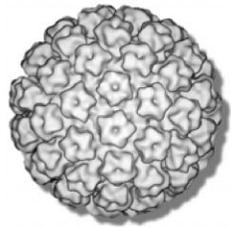
The 'gospel' continued:

“We therefore have a pair of lines in the original transform plane along which the transform must have identical values”

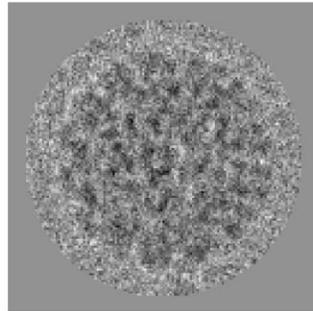
“A similar pair of lines will be generated by each possible choice of pairs of symmetry operations”

“The angular positions of these lines are dependent on the orientation of the particle.”

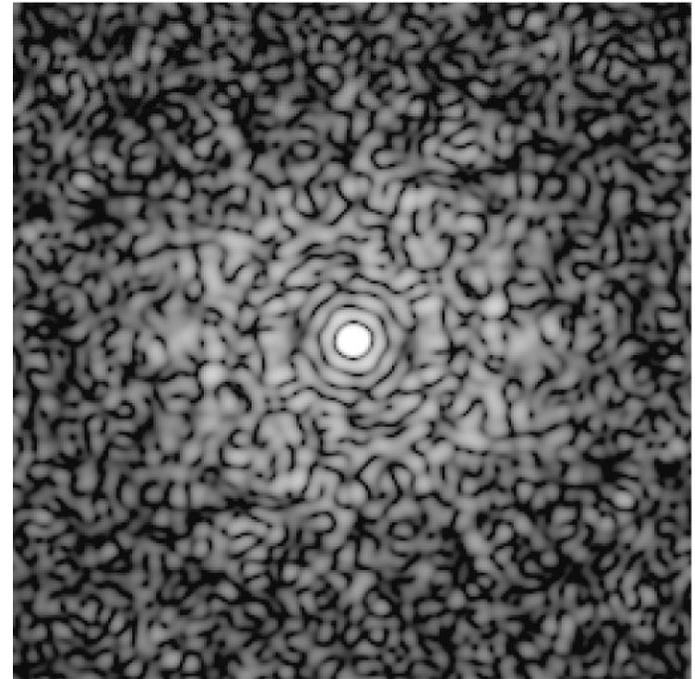
Orientation Determination by Common Lines



3D Object



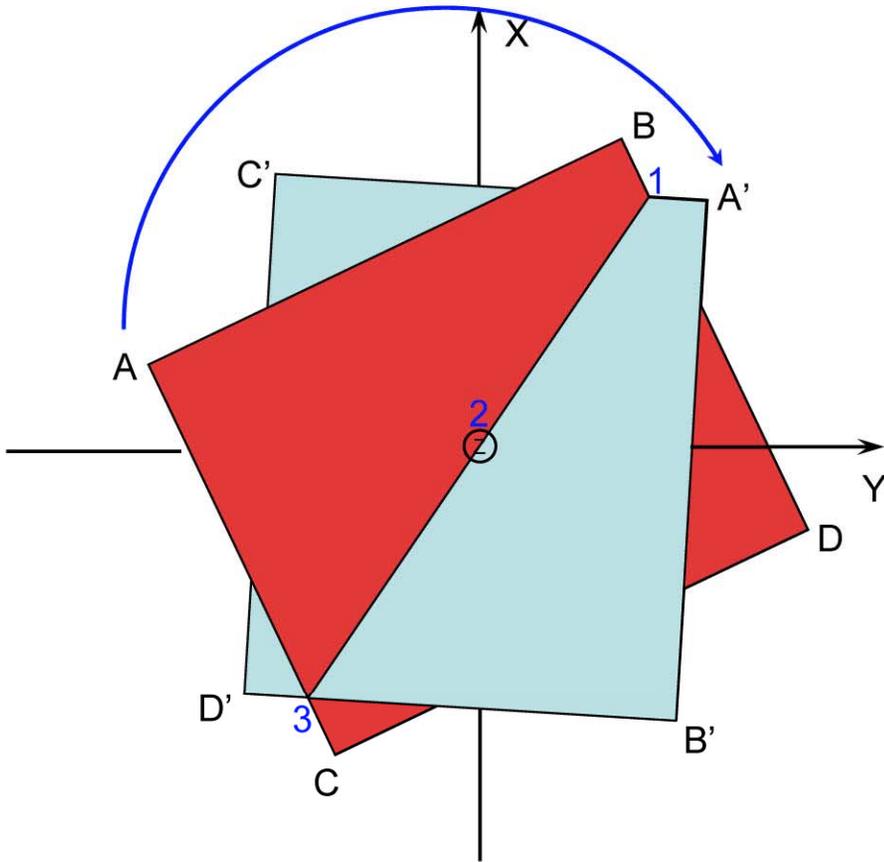
2D Projection
(θ, ϕ, ω)



2D Fourier Transform

Orientation Determination by Common Lines

Simple example: object with single three-fold axis of symmetry



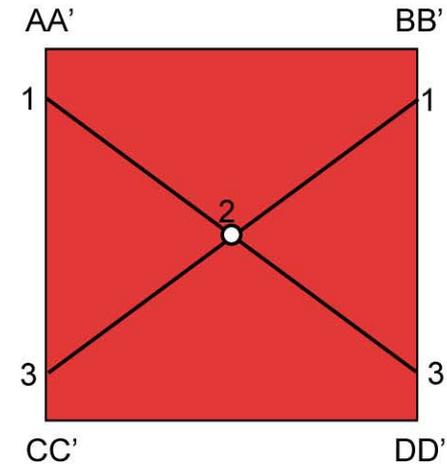
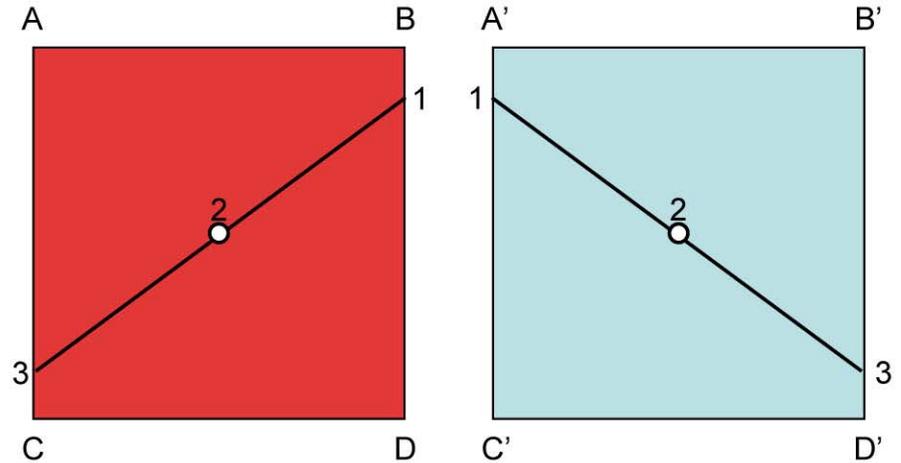
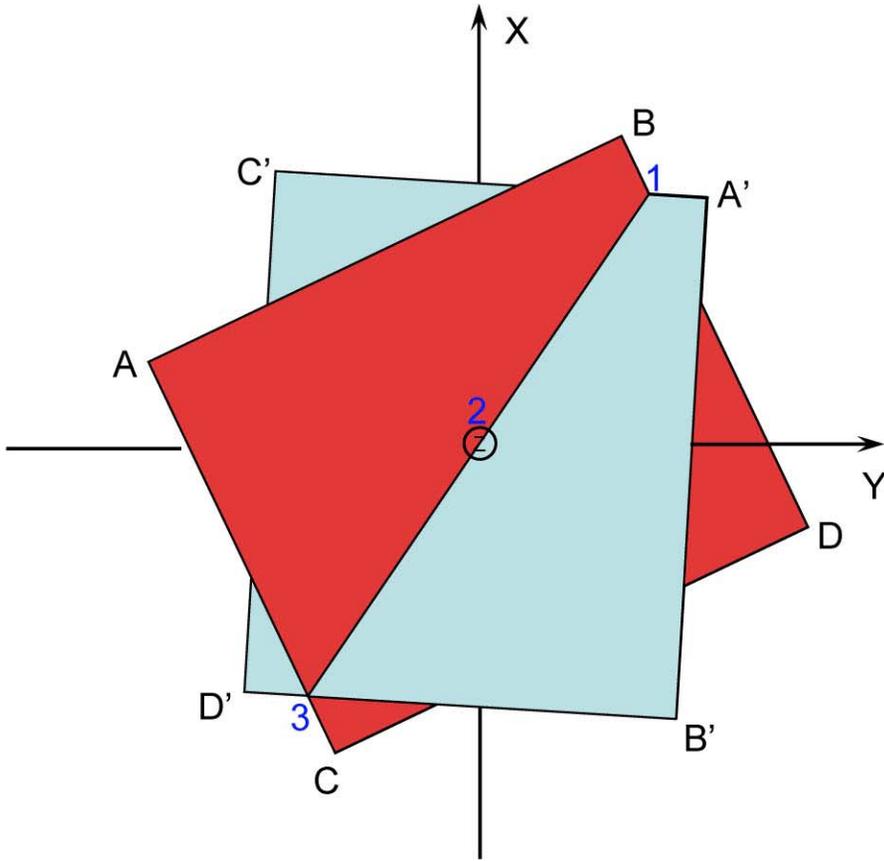
ABCD = 2D transform of image from particle **not** viewed along an axis of symmetry

Let z-direction coincide with **3-fold** axis of symmetry

3-fold operation generates **two** additional FT sections (only A'B'C'D' shown)

Both planes have **common values** along the **line** (1,2,3) of their intersection

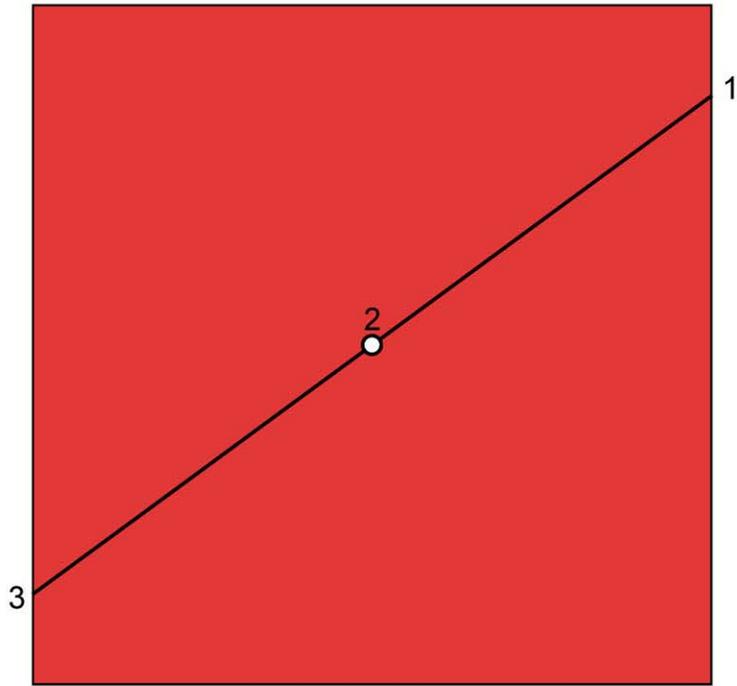
Orientation Determination by Common Lines



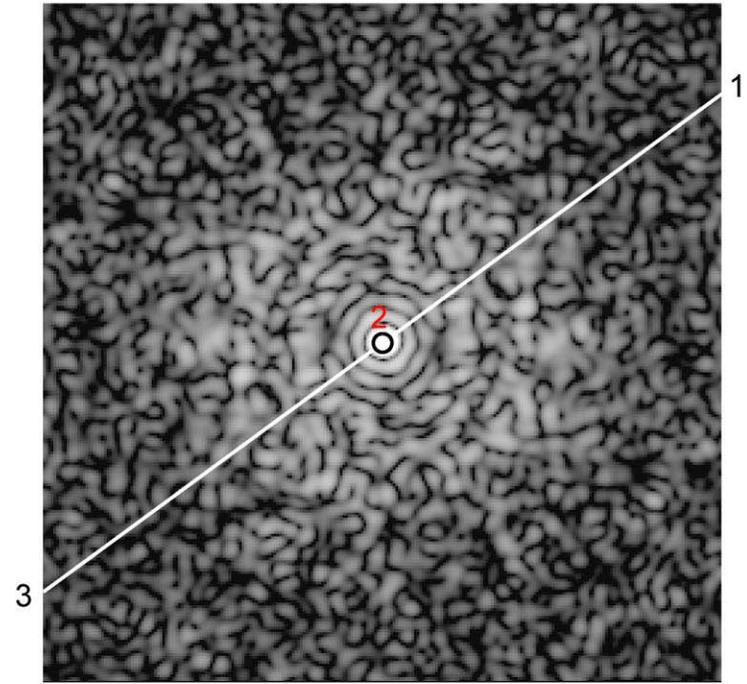
Adapted from Moody (1990) Fig. 7.68, p.245

Adapted from Moody (1990) Fig. 7.69, p.246

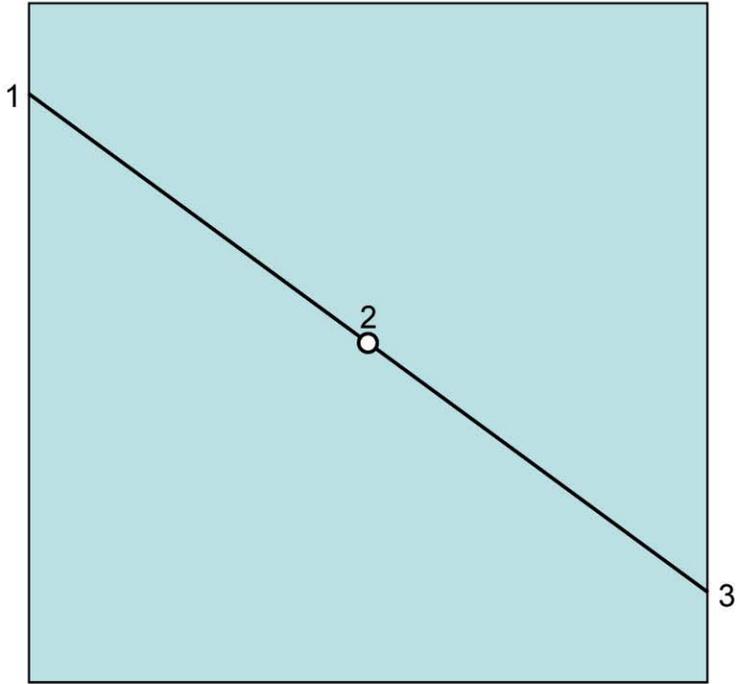
Orientation Determination by Common Lines



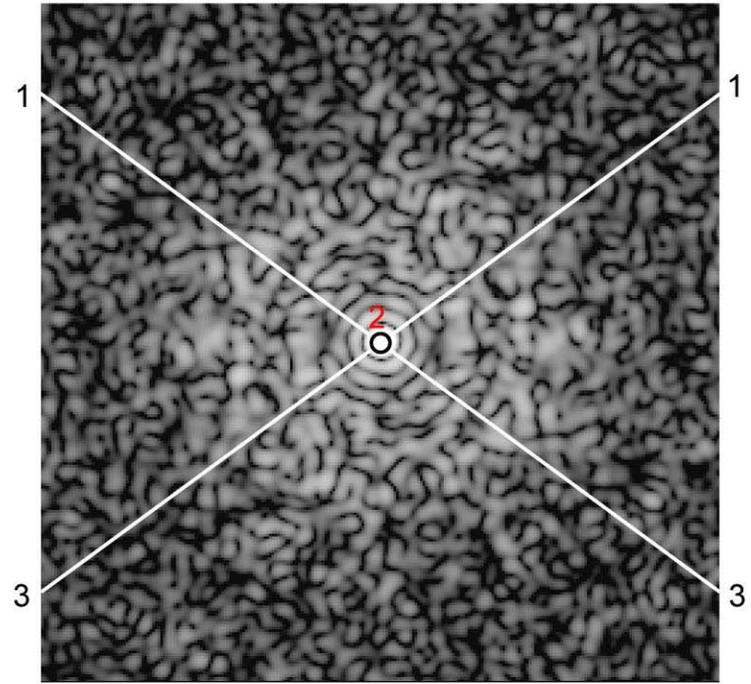
Original Transform Plane



Orientation Determination by Common Lines



Symmetry-Related
Transform Plane



Orientation Determination by Common Lines

Ok, that's easy (simple object with single 3-fold axis)

What about an object with 532 symmetry?

For a **general view**, icosahedral symmetry generates:

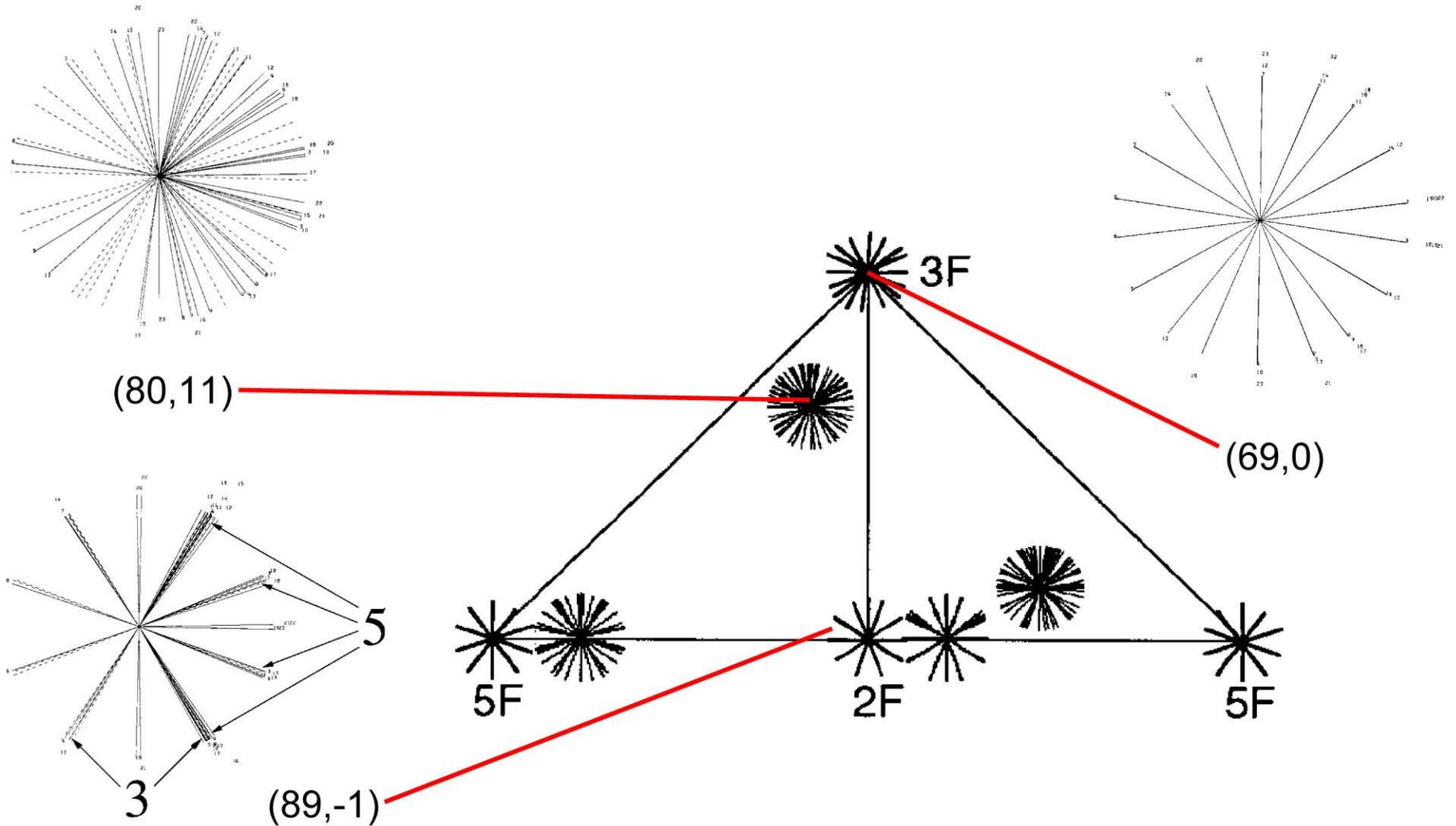
$$\text{5-folds: } \frac{12}{2} \times 2 = 12 \text{ pairs}$$

$$\text{3-folds: } \frac{20}{2} \times 1 = 10 \text{ pairs}$$

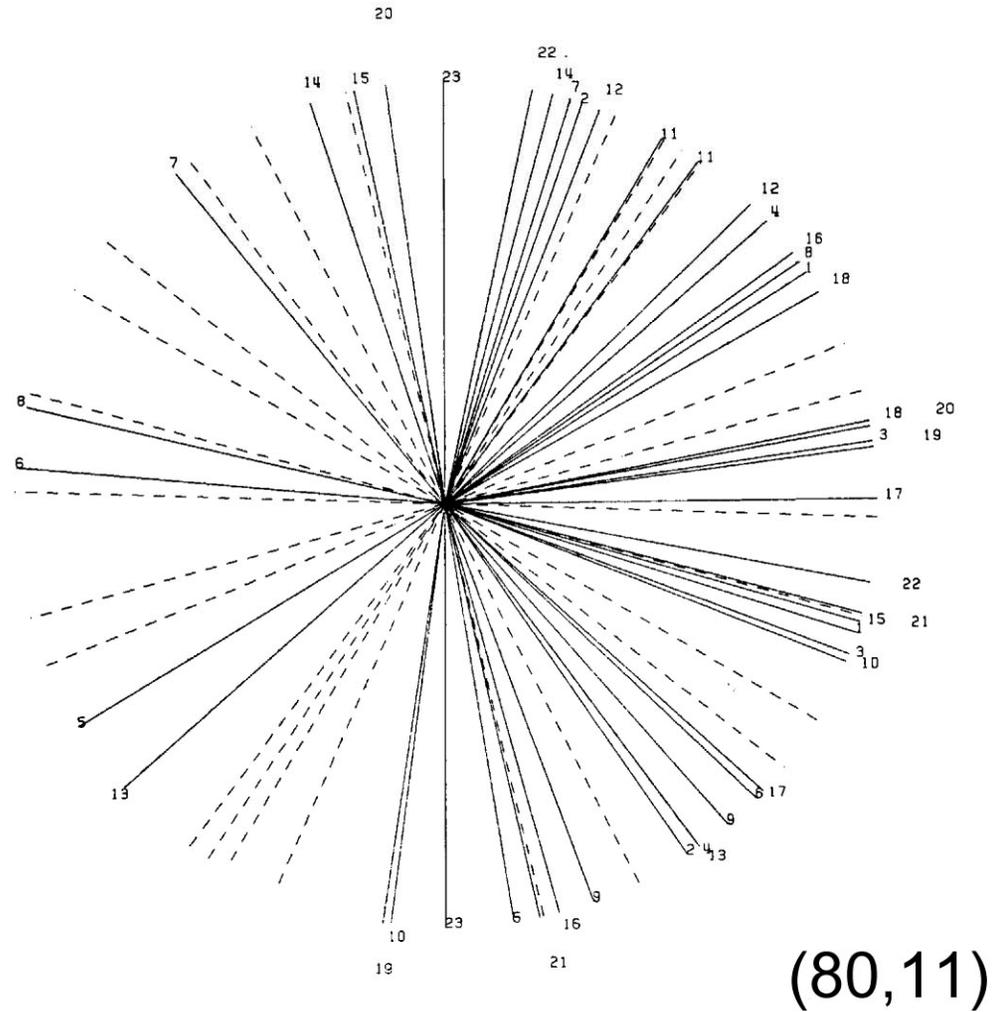
$$\text{2-folds: } \frac{30}{2} \times 1 = \underline{15} \text{ real lines}$$

37 common lines

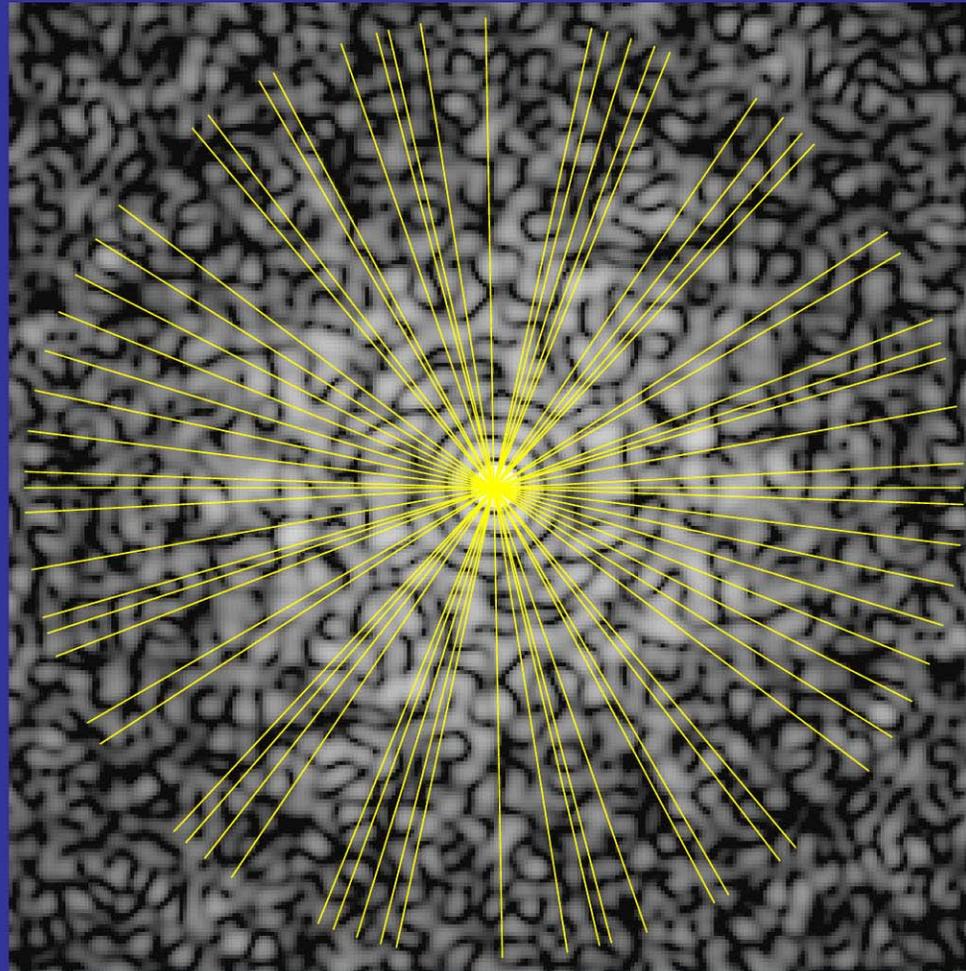
Orientation Determination by Common Lines



Orientation Determination by Common Lines



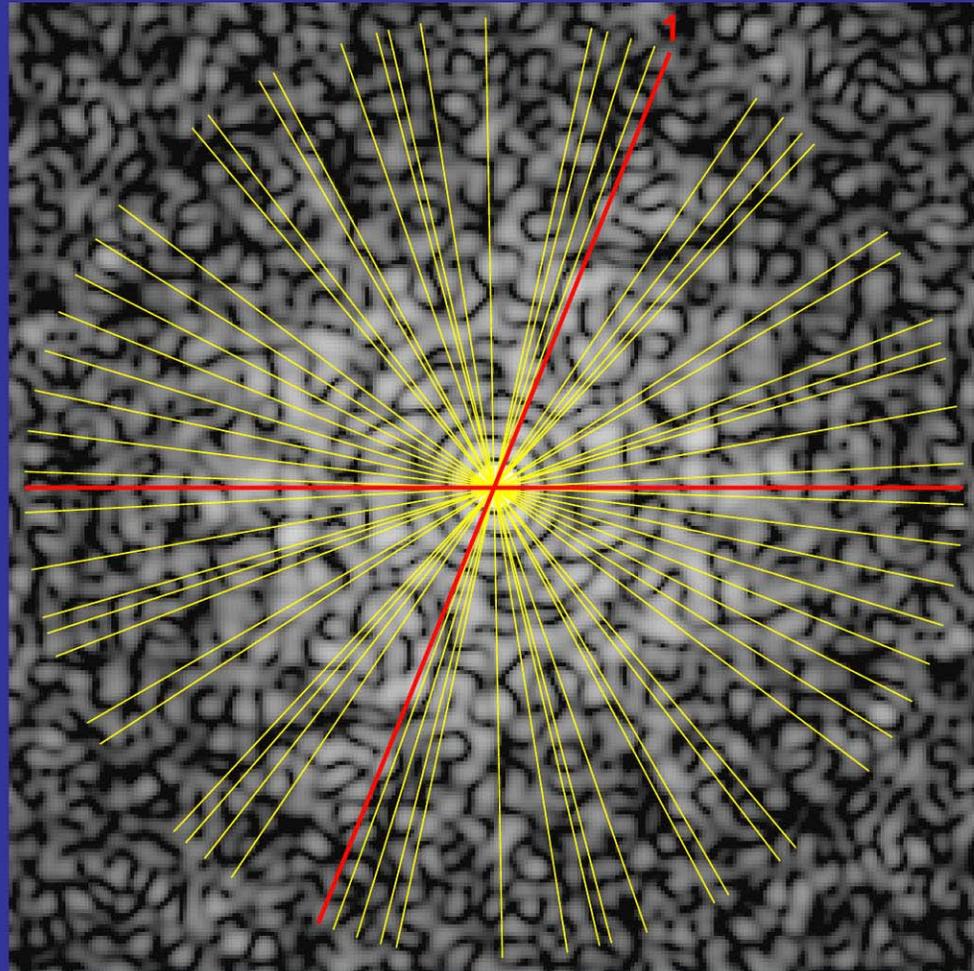
Orientation Determination by Common Lines



(80,11)

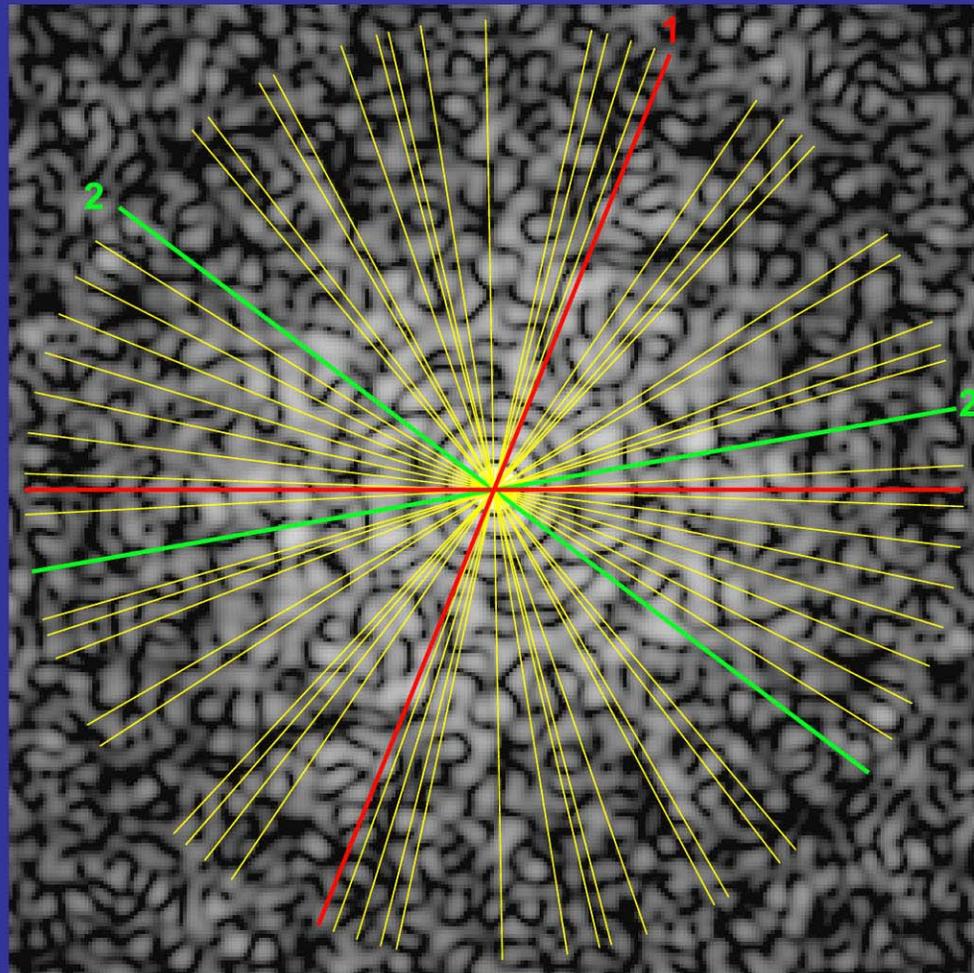
What is (θ, ϕ, ω) for this particle?

Orientation Determination by Common Lines



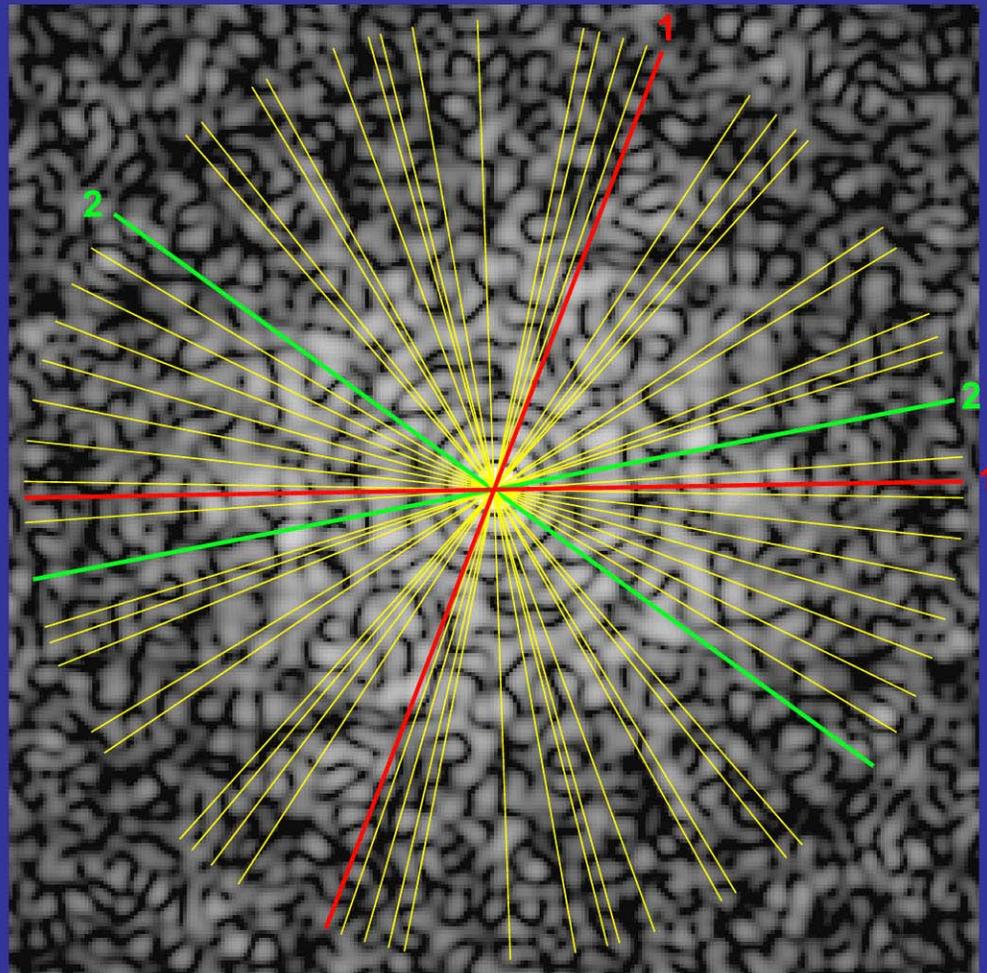
ω
↓
(80, 11, 0)

Orientation Determination by Common Lines



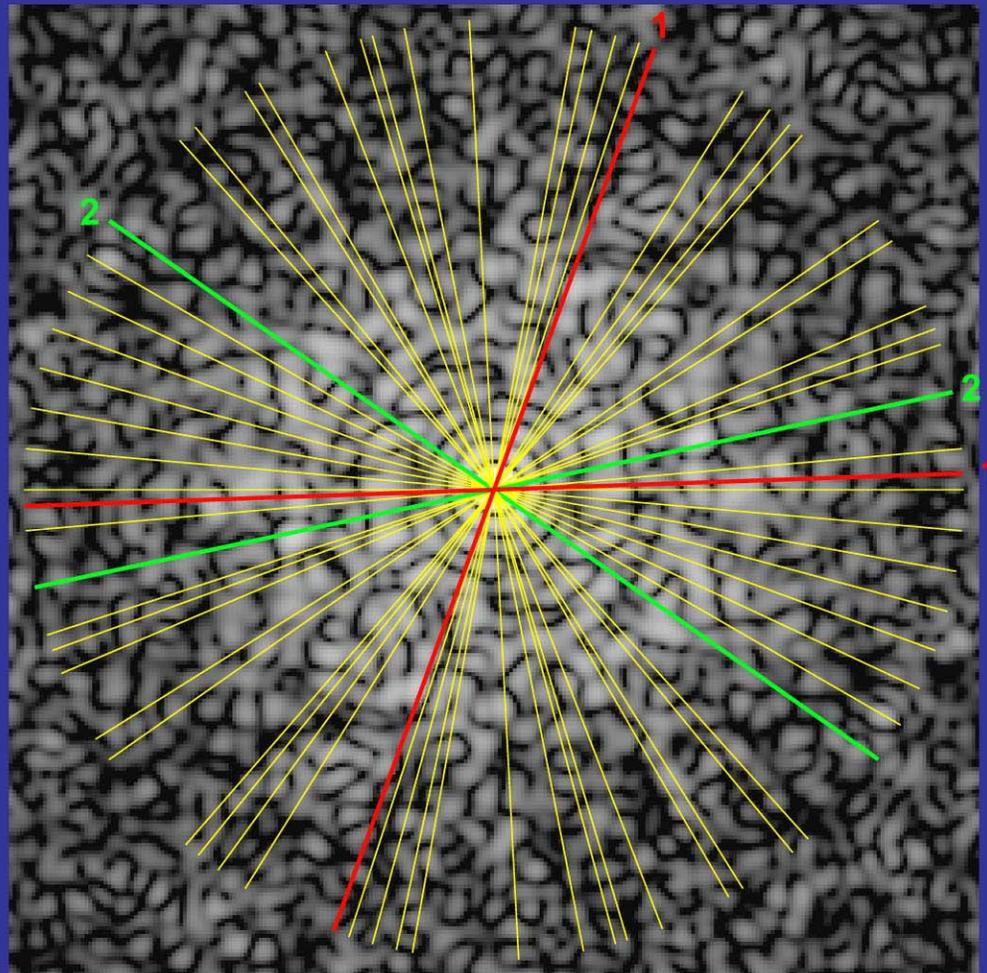
ω
↓
(80, 11, 0)

Orientation Determination by Common Lines



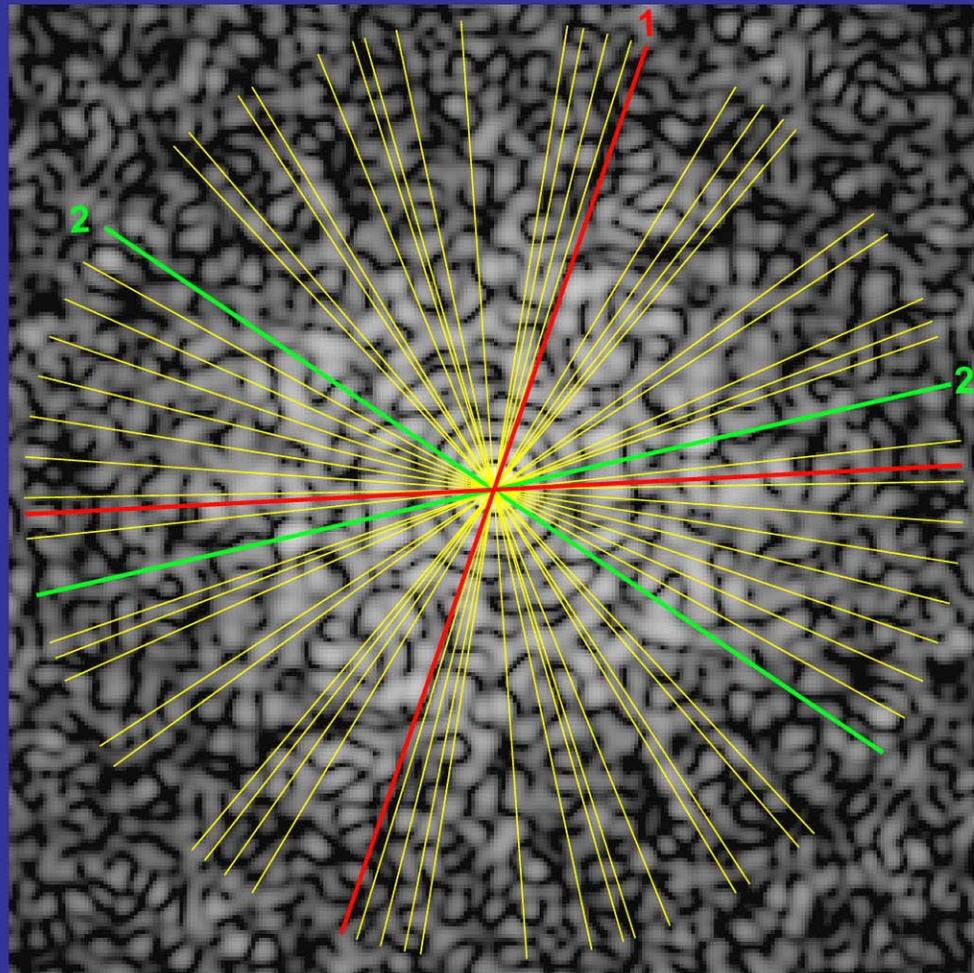
ω
↓
(80, 11, 1)

Orientation Determination by Common Lines



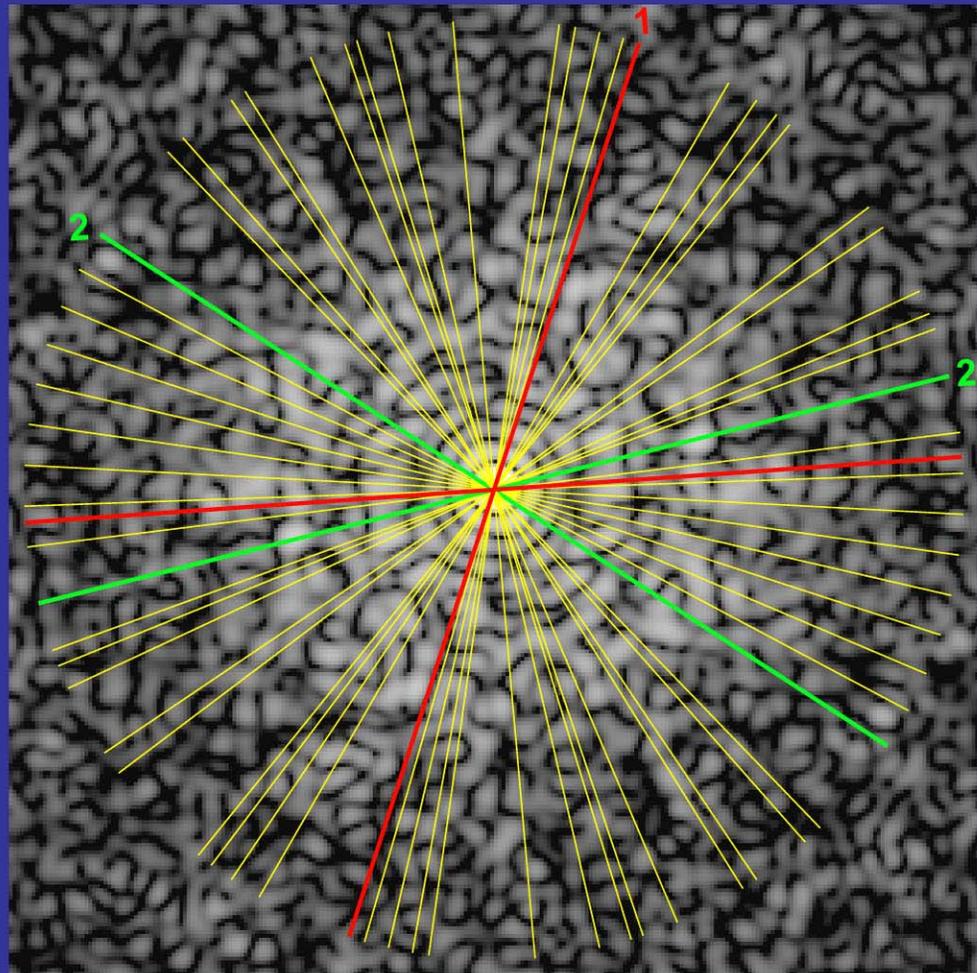
(80, 11, 2)

Orientation Determination by Common Lines



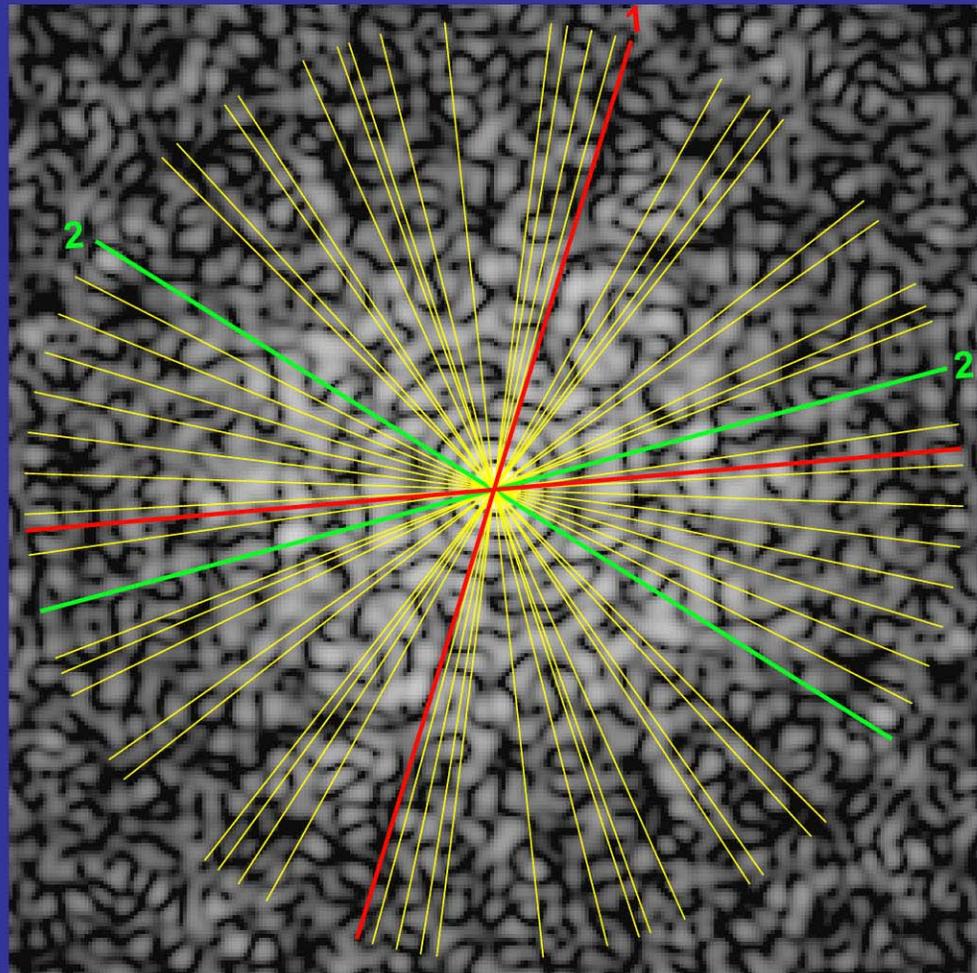
ω
↓
(80, 11, 3)

Orientation Determination by Common Lines



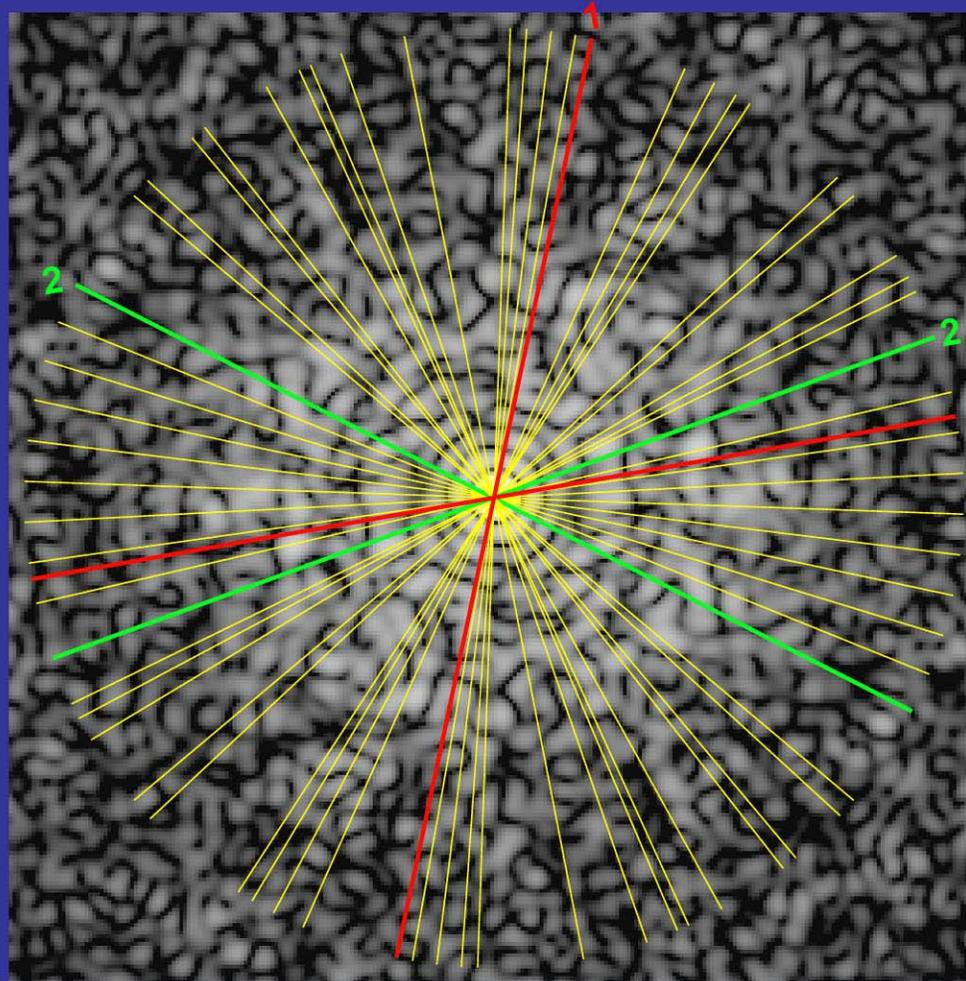
ω
↓
(80, 11, 4)

Orientation Determination by Common Lines



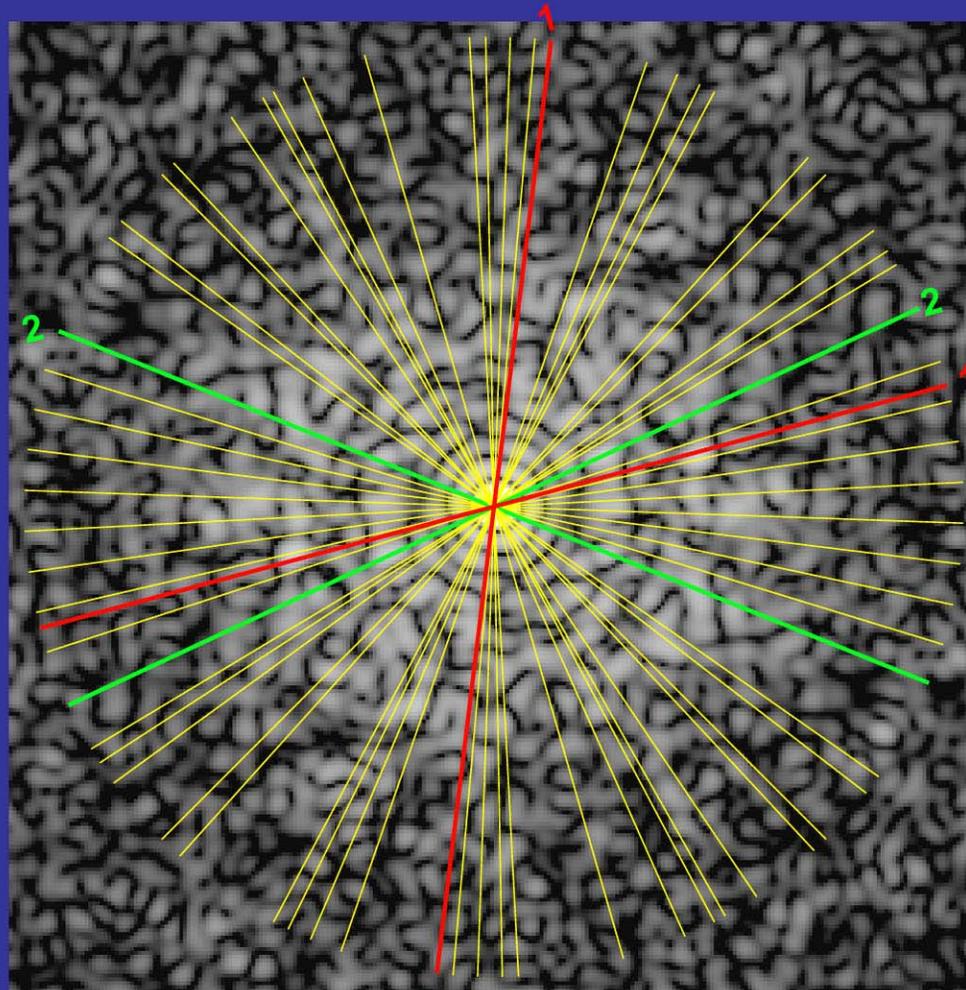
ω
↓
(80, 11, 5)

Orientation Determination by Common Lines



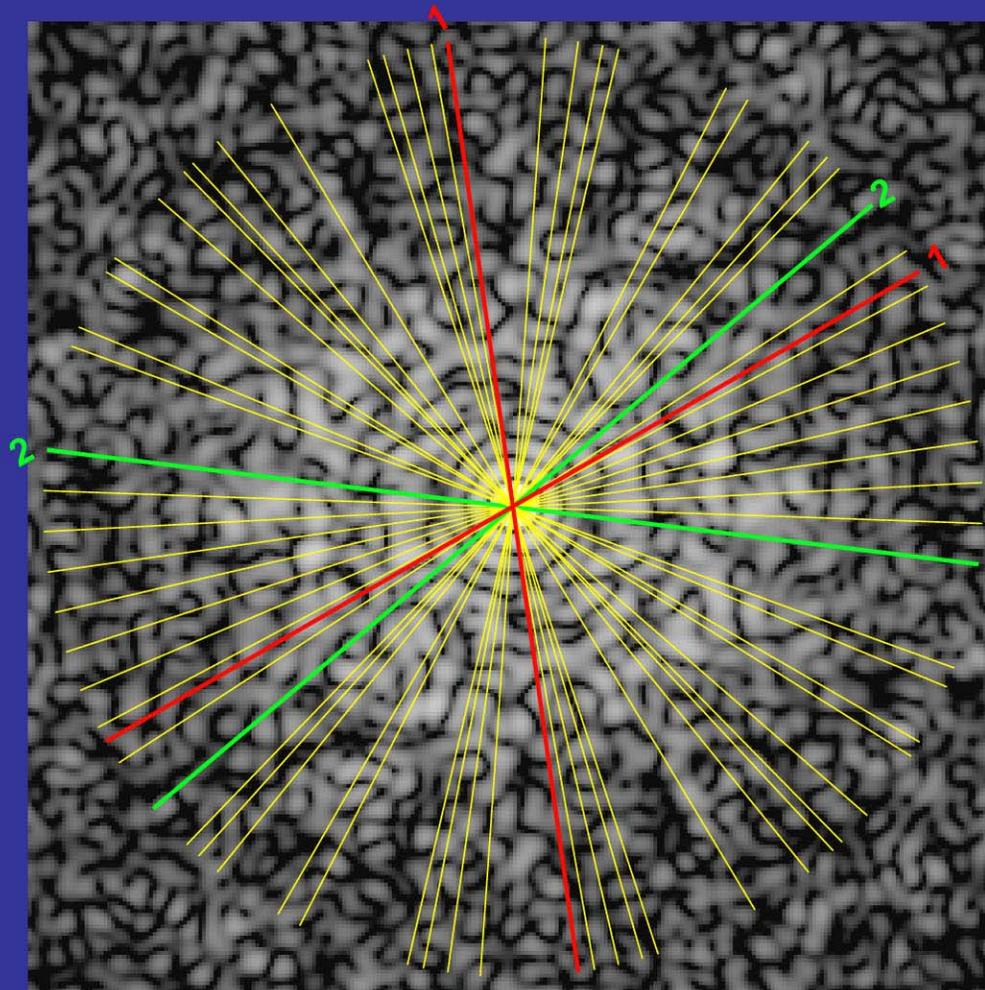
ω
↓
(80, 11, 10)

Orientation Determination by Common Lines



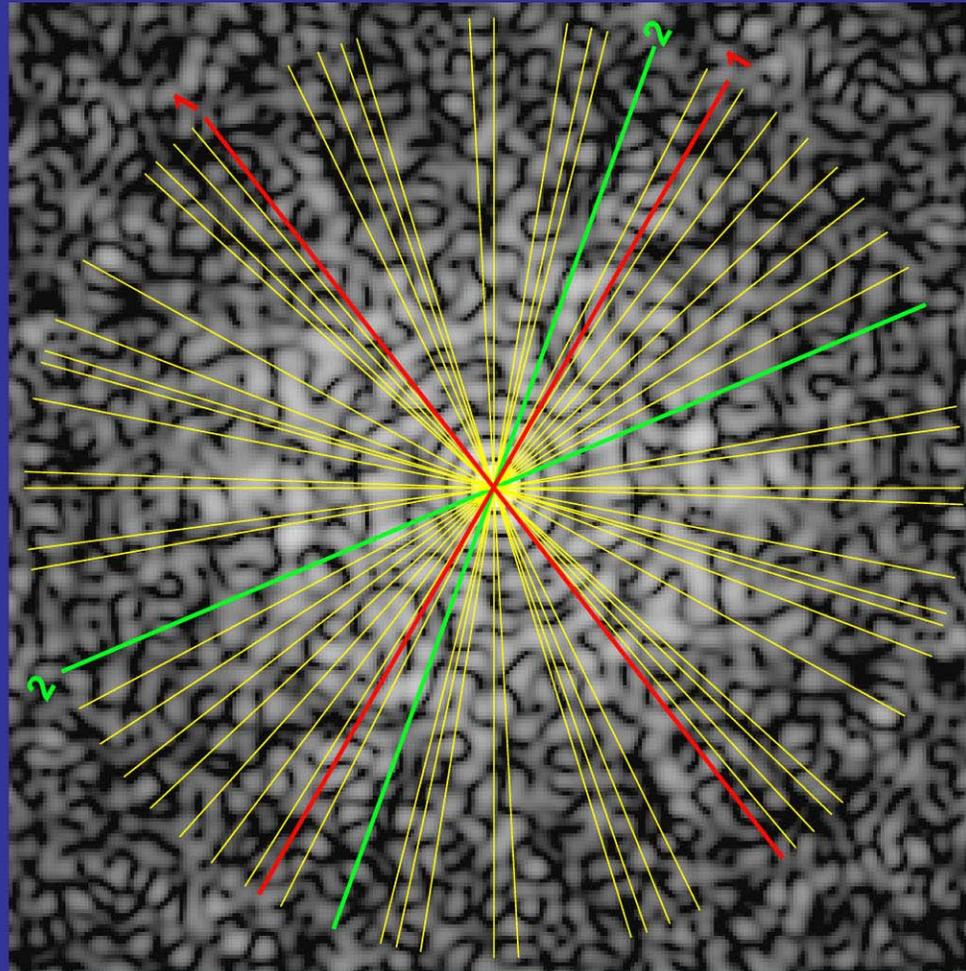
ω
↓
(80, 11, 15)

Orientation Determination by Common Lines



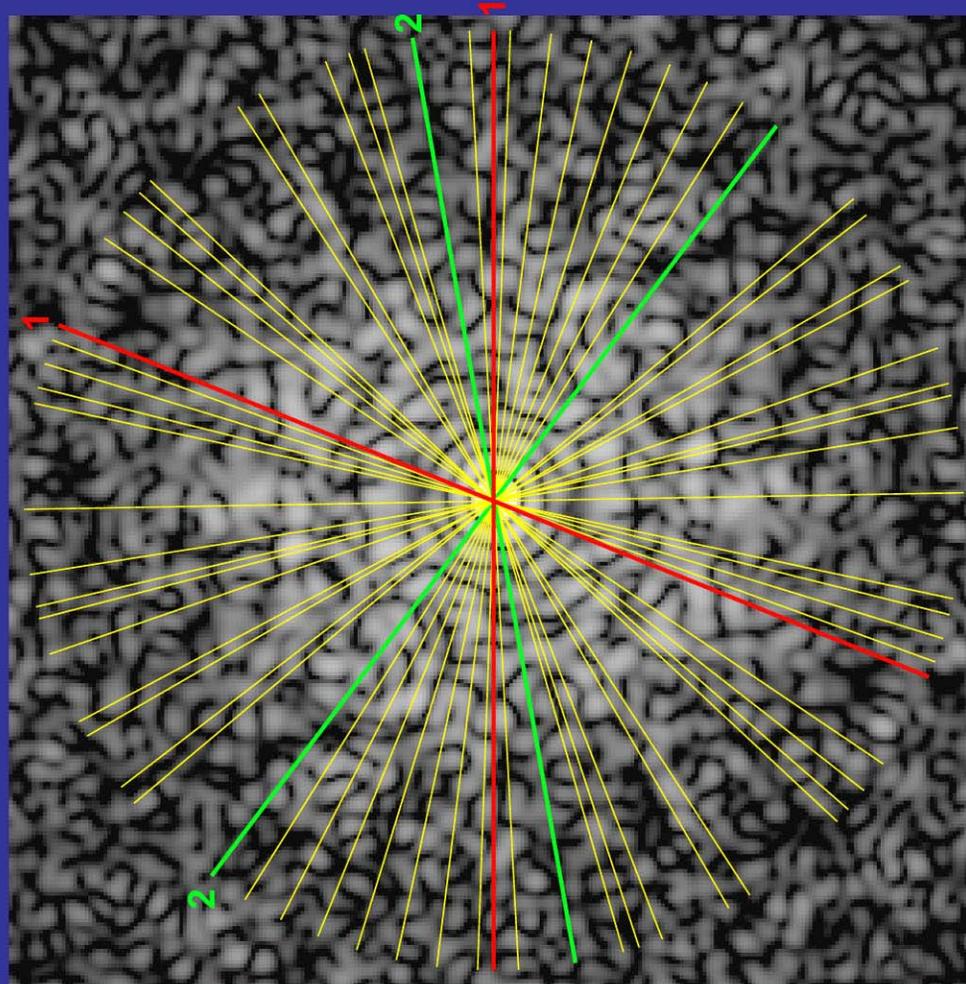
ω
↓
(80, 11, 30)

Orientation Determination by Common Lines



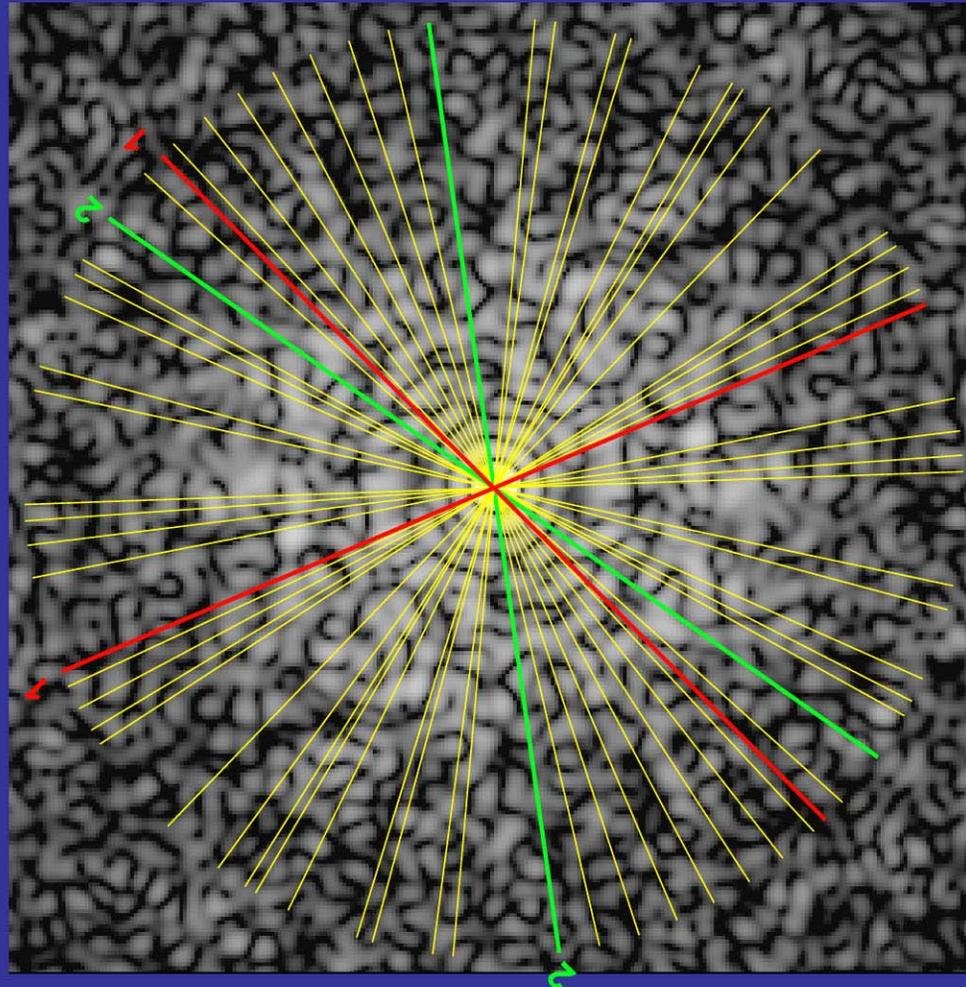
ω
↓
(80, 11, 60)

Orientation Determination by Common Lines



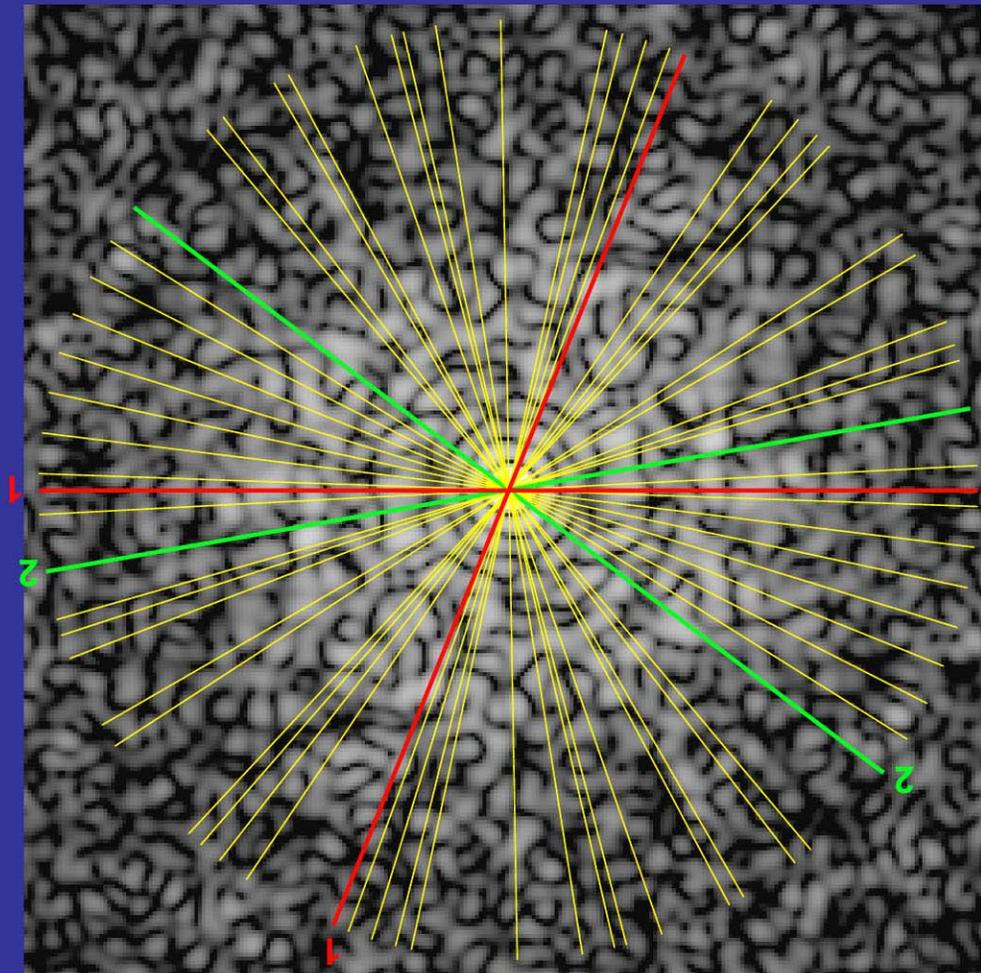
ω
↓
(80, 11, 90)

Orientation Determination by Common Lines



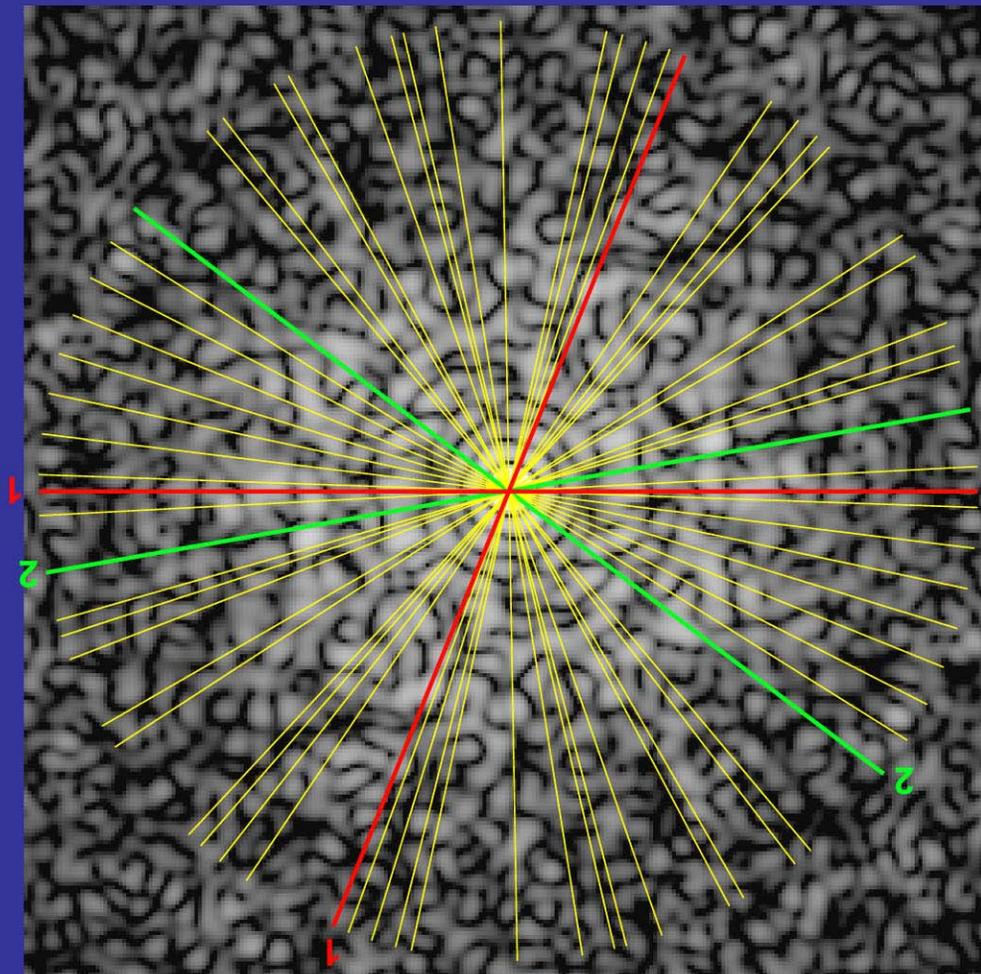
ω
↓
(80, 11, 135)

Orientation Determination by Common Lines



ω
↓
(80, 11, 180)

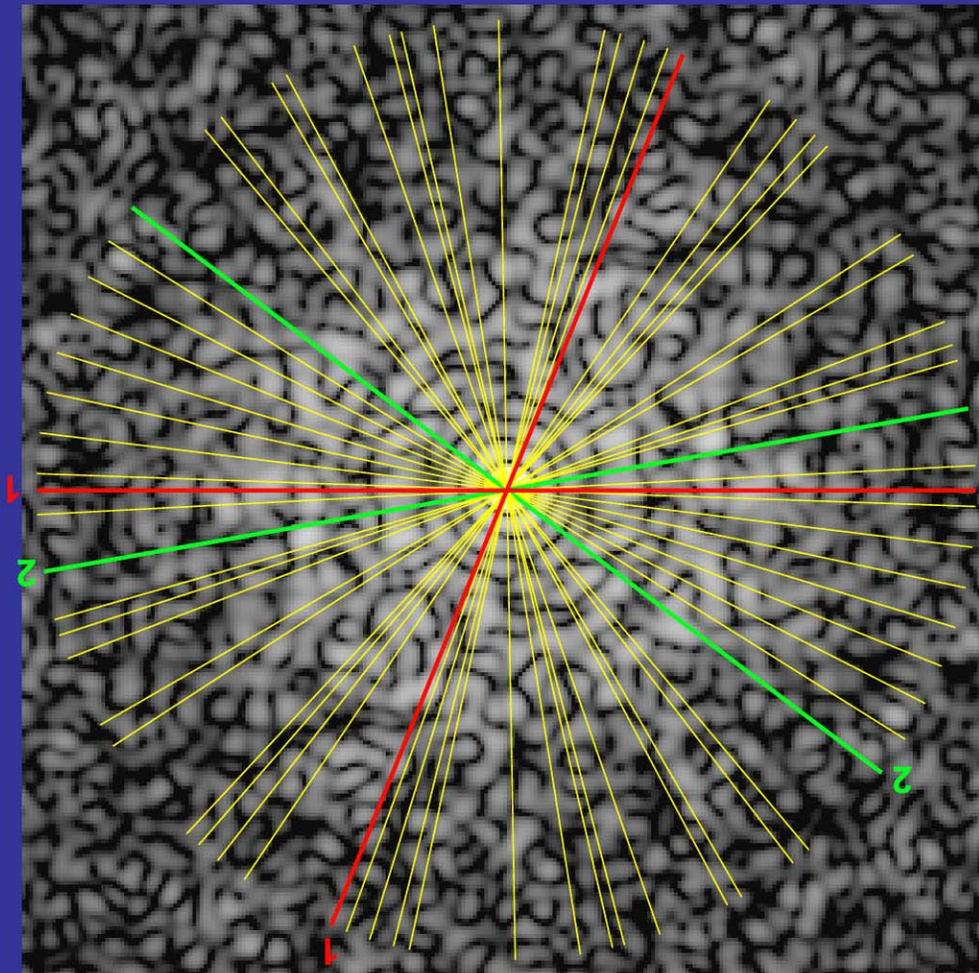
Orientation Determination by Common Lines



(80, 11, ω)

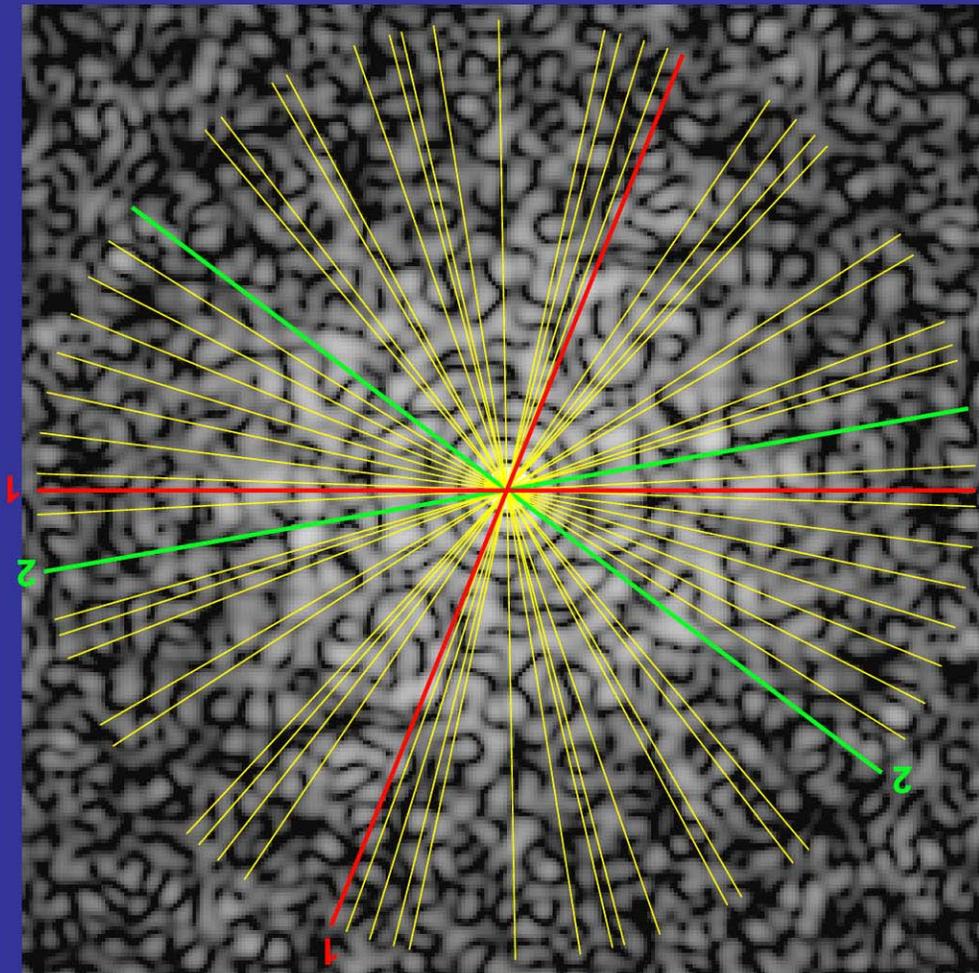
Metric: Identify ω that gives lowest phase residual

Orientation Determination by Common Lines



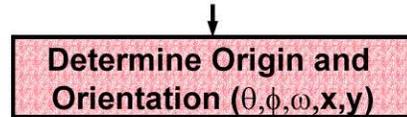
Repeat process for all possible (θ, ϕ, ω) combinations

Orientation Determination by Common Lines



> 250,000 combinations for 1° angular search intervals

Orientation Determination by Common Lines



Common Lines

The (θ, ϕ, ω) that results in the lowest phase residual is selected as the best estimate for the particle view orientation

The 'common lines' procedure is similarly used to determine the particle phase origin (x, y)

Minimizing Phase Differences Among Common Lines

$$P_i(\phi, \theta, \omega, x, y) = \frac{\sum_{j=1}^N \sum_{k=1}^{k_{\max}(j)} \sum_{R=R_{\min}}^{R_{\max}} \left| \psi_i(R, x_i, y_i, \alpha_{i,j,k}) - \psi_j(R, x_j, y_j, \alpha_{j,i,k}) \right| \times w(R, \alpha_{i,j,k}, \alpha_{j,i,k})}{(R_{\max} - R_{\min}) \sum_{j=1}^N k_{\max}(j)}$$

P is the phase residual

Ψ is the phase value in Fourier space

i and j refer to particles or symmetry-related sections

x and y define the phase origin

k refers to the common line

α_{ijk} are the angles of the k-th common-lines

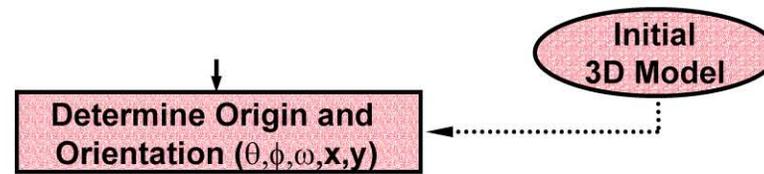
R_{\min} and R_{\max} define the frequency range within which the phase residuals are evaluated

k_{\max} is total number of common lines (here: 37)

N is the number of particles

w defines a weighting function for the Fourier elements at different frequency and orientations

Icosahedral Particle Reconstruction Scheme



Recall: two methods to determine $(\theta, \phi, \omega, x, y)$:

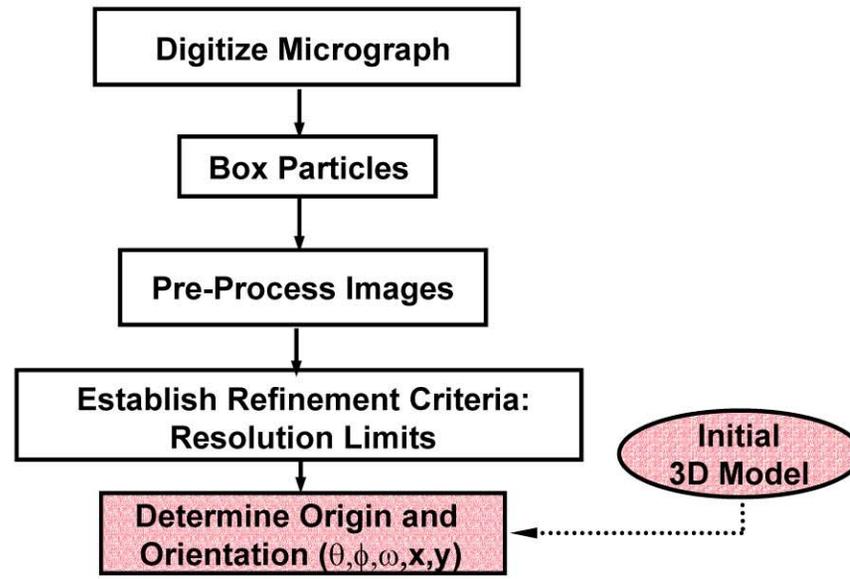
1. Common lines

2. Model-based (template) matching

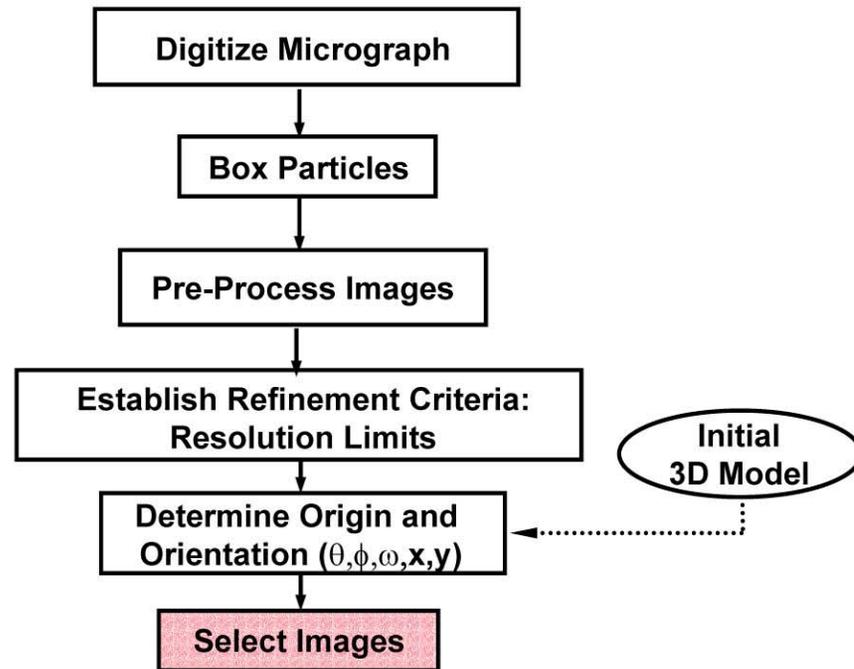
- Many structures now solved this way
- Same as reference based alignment in single-particle processing in earlier session...

3D Reconstruction

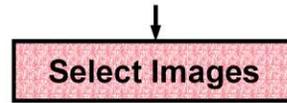
Icosahedral Particle Reconstruction Scheme



Icosahedral Particle Reconstruction Scheme

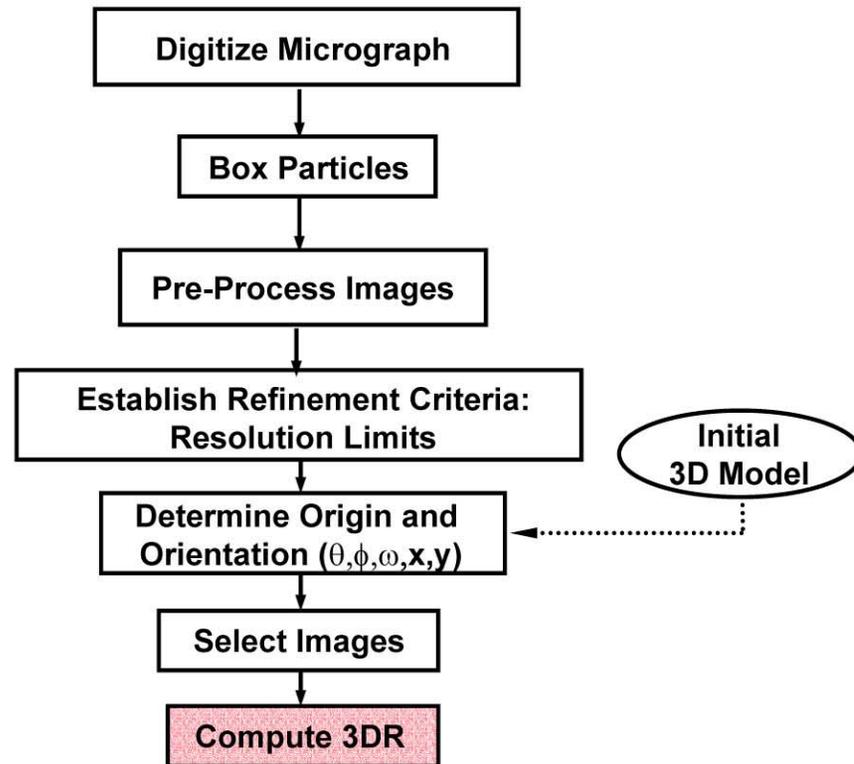


Icosahedral Particle Reconstruction Scheme

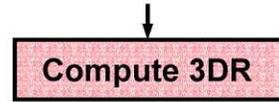


Goal: weed out 'bad' particle images before computing 3D reconstruction

Icosahedral Particle Reconstruction Scheme



Icosahedral Particle Reconstruction Scheme

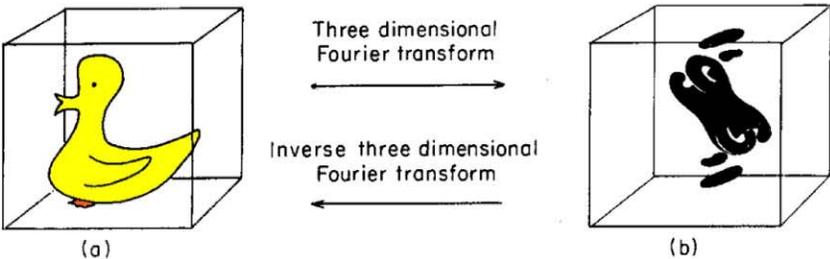


Goal: combine “good” particle images to compute a 3D density map

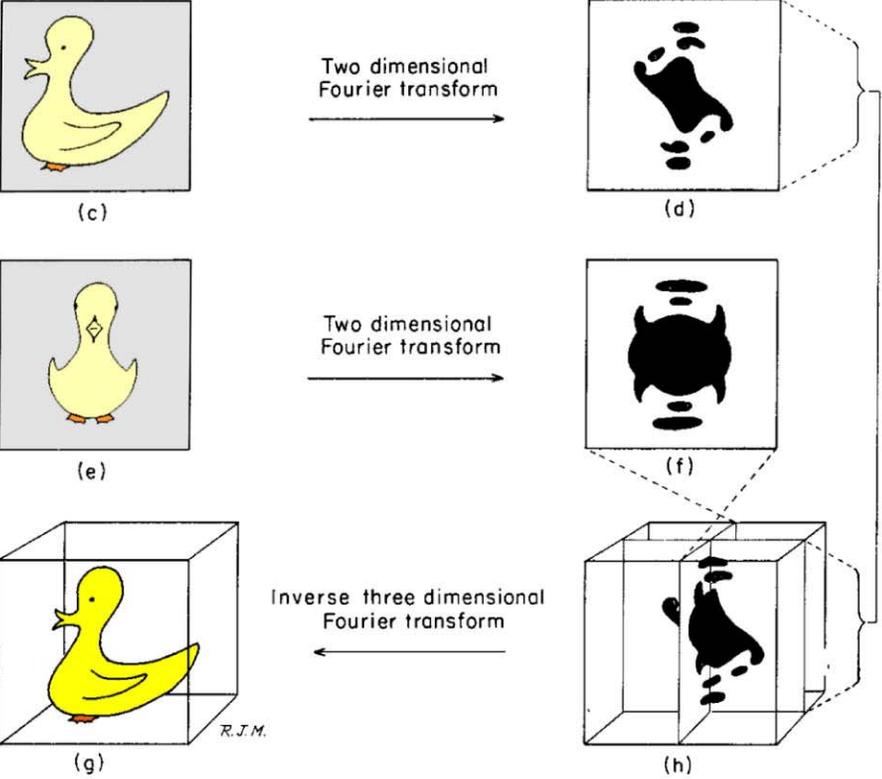
Icosahedral Particle Reconstruction Scheme

↓
Compute 3DR

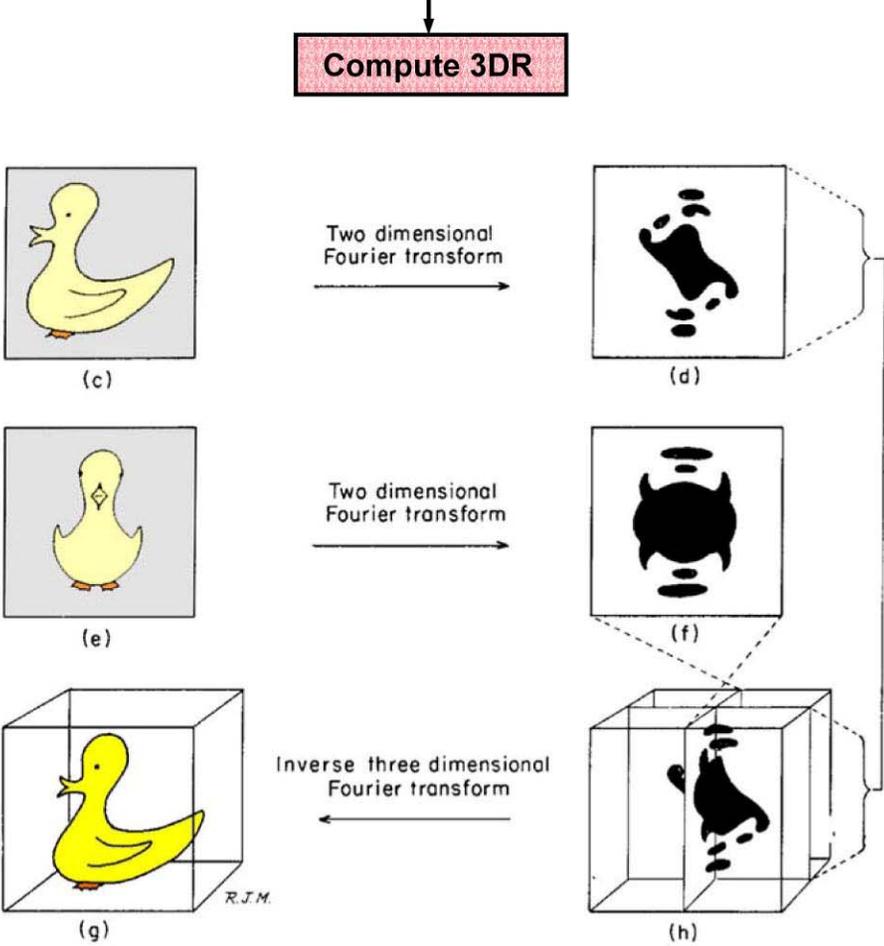
In theory



In practice



Icosahedral Particle Reconstruction Scheme



Overall scheme: $\rho \leftarrow g \leftarrow G \leftarrow F$

Remember? Fourier-Bessel Formalism

$$\rho(r, \varphi, z) = \sum_{n=-\infty}^{\infty} \int_{-\infty}^{\infty} g_n(r, Z) e^{in\varphi} e^{2\pi izZ} dZ$$

$$g_n(r, Z) = \int_0^{\infty} G_n(R, Z) J_n(2\pi Rr) 2\pi R dR$$

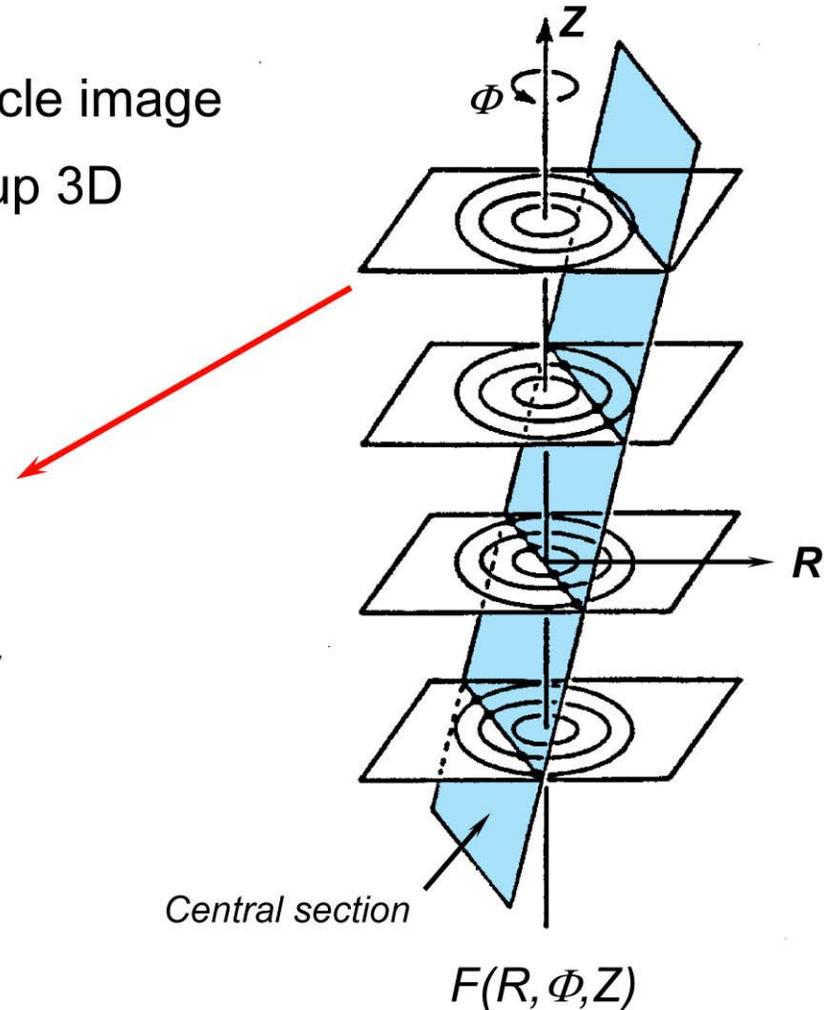
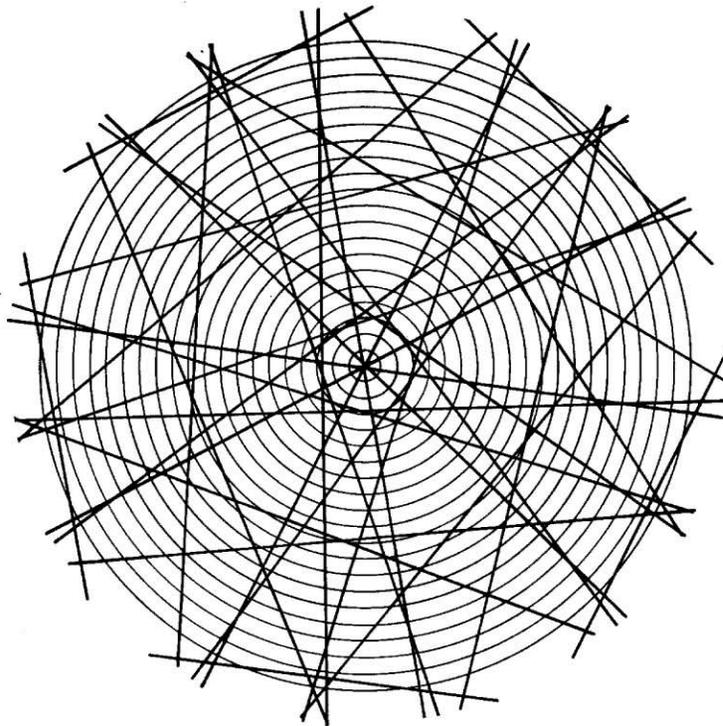
$$F(R, \Phi, Z) = \sum_{n=-\infty}^{\infty} G_n(R, Z) i^n e^{in\Phi}$$

Icosahedral Particle Reconstruction Scheme

↓
Compute 3DR

Steps:

1. Compute 2D FFT of each particle image
2. Combine all 2D FFTs to build up 3D Fourier-Bessel transform



Icosahedral Particle Reconstruction Scheme

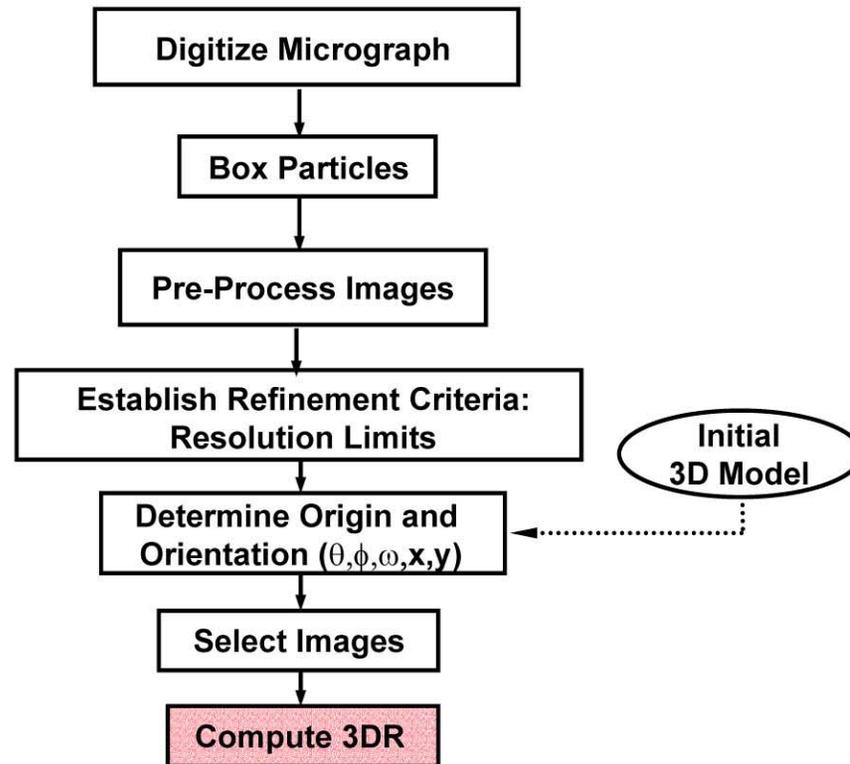
↓
Compute 3DR

$$\rho \leftarrow \mathbf{g} \leftarrow \mathbf{G} \leftarrow \mathbf{F}$$

Steps:

1. Compute 2D FFT of each particle image
2. Combine all 2D FFTs to build up 3D Fourier-Bessel transform
3. Compute G_n 's on each annulus $G = (B^\dagger B)^{-1} B^\dagger F$ solve linear system of equations
4. Compute g_n 's from G_n 's (Fourier-Bessel transform)
5. Compute polar density map ($\rho(r, \phi, z)$) from g_n 's
6. Convert from polar to Cartesian map ($\rho(r, \phi, z) \rightarrow \rho(x, y, z)$)

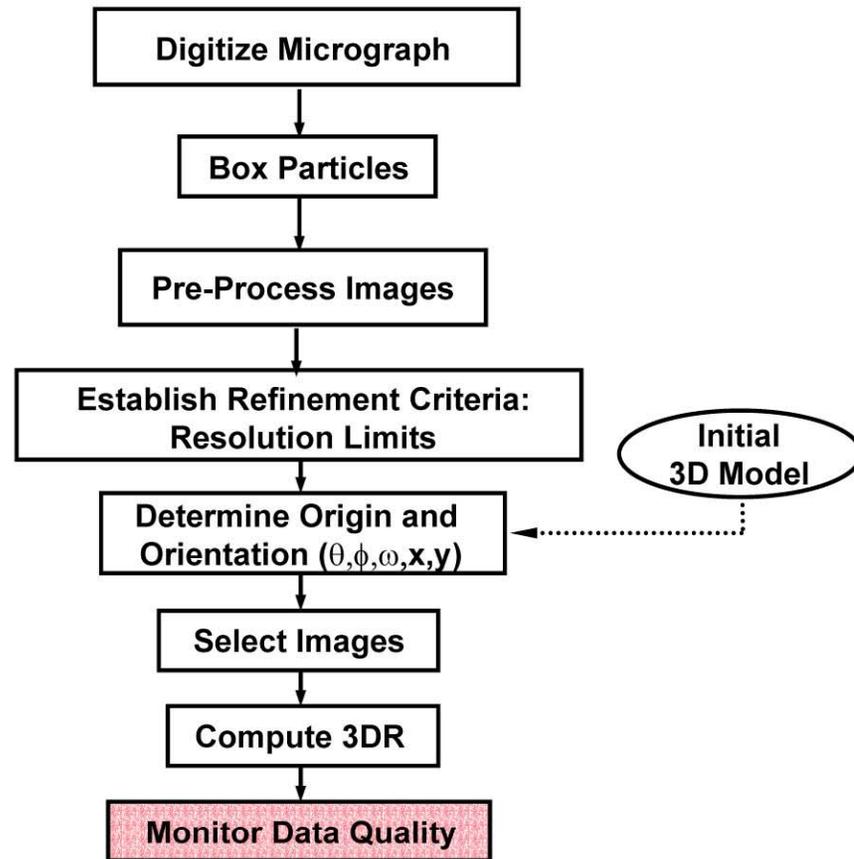
Icosahedral Particle Reconstruction Scheme



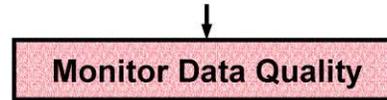
Option: correct for CTF effects in particle FFTs before FFTs are merged to form the 3D FFT

Resolution Estimation and Quality Control

Icosahedral Particle Reconstruction Scheme

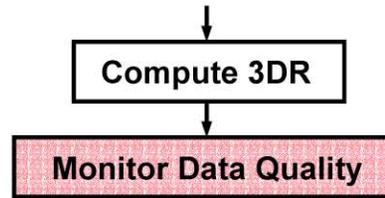


Icosahedral Particle Reconstruction Scheme

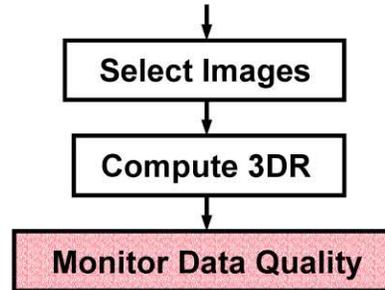


Goal: assess resolution of 3D density map
to determine what to do next

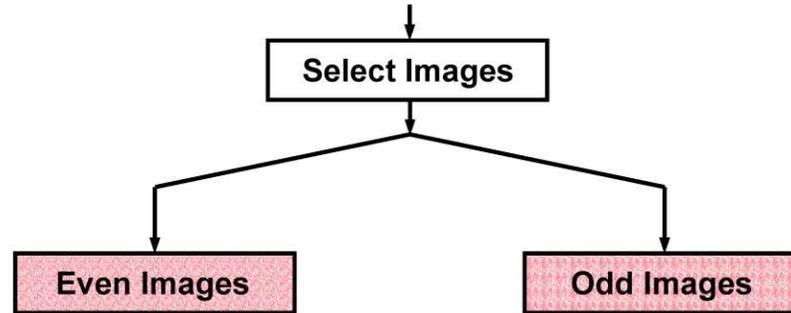
Icosahedral Particle Reconstruction Scheme



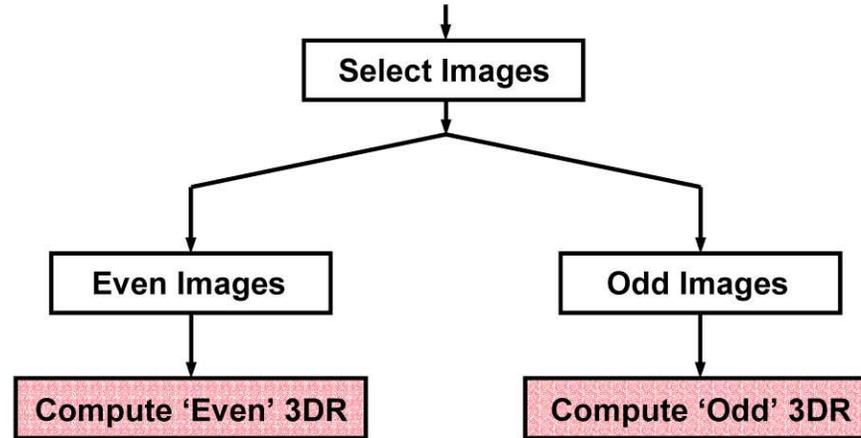
Icosahedral Particle Reconstruction Scheme



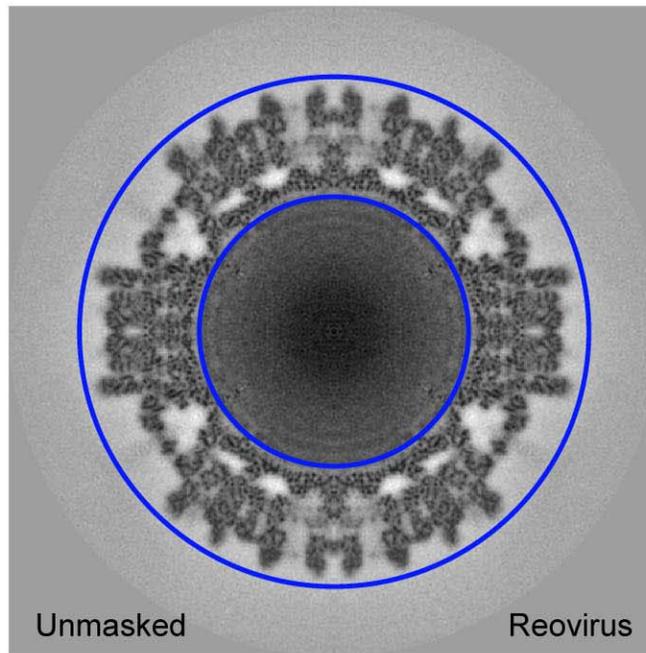
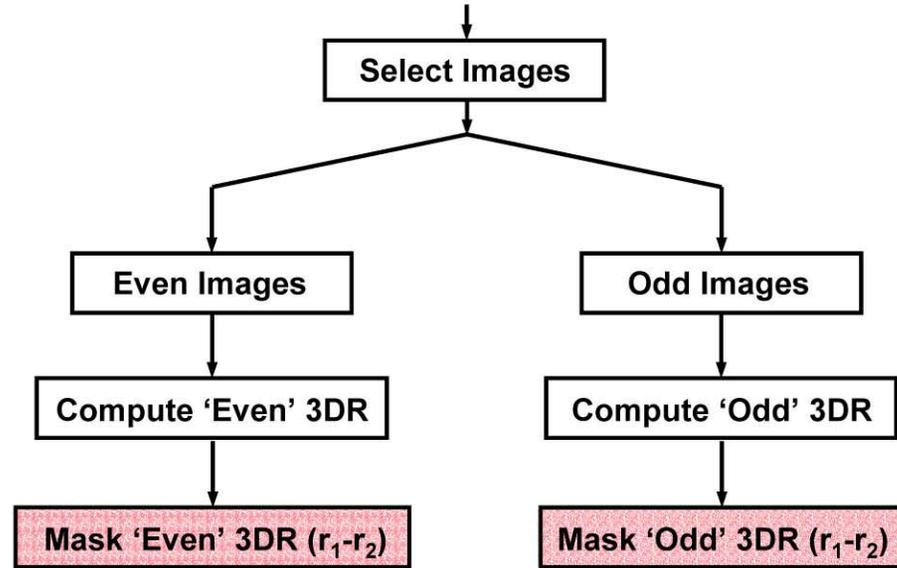
Icosahedral Particle Reconstruction Scheme



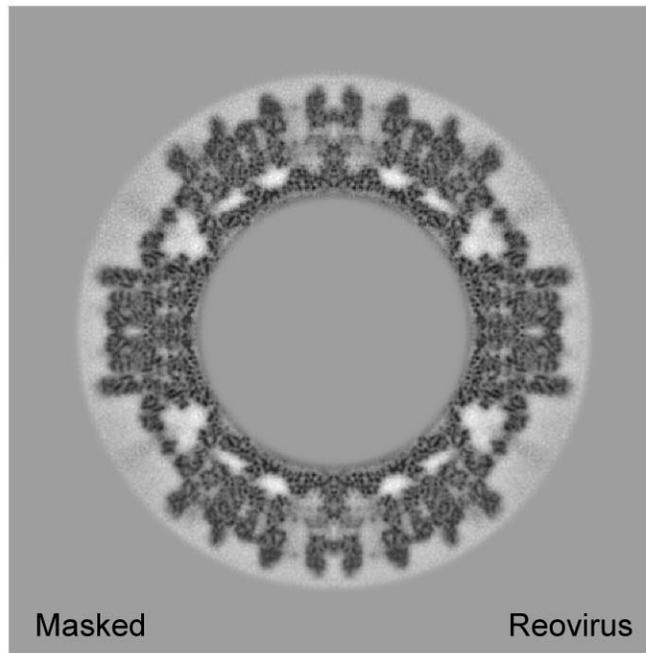
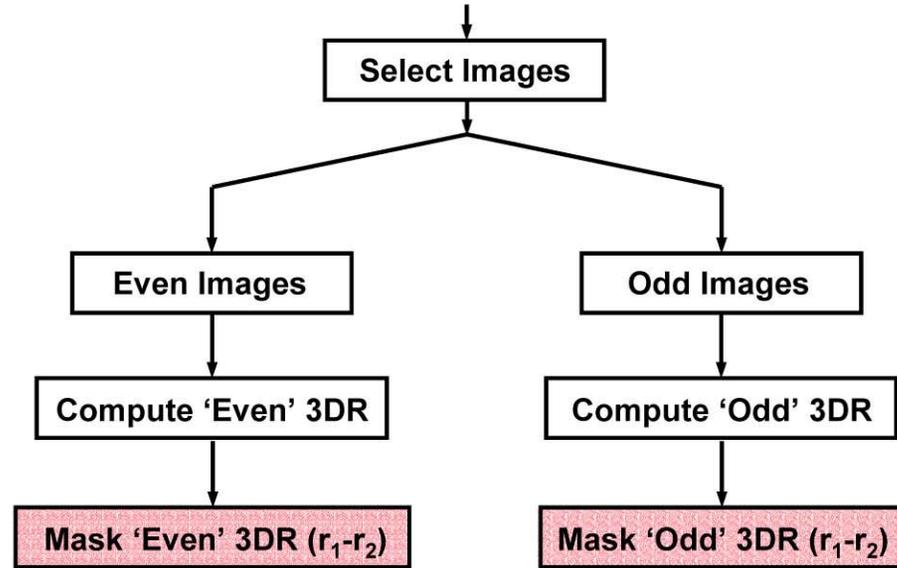
Icosahedral Particle Reconstruction Scheme



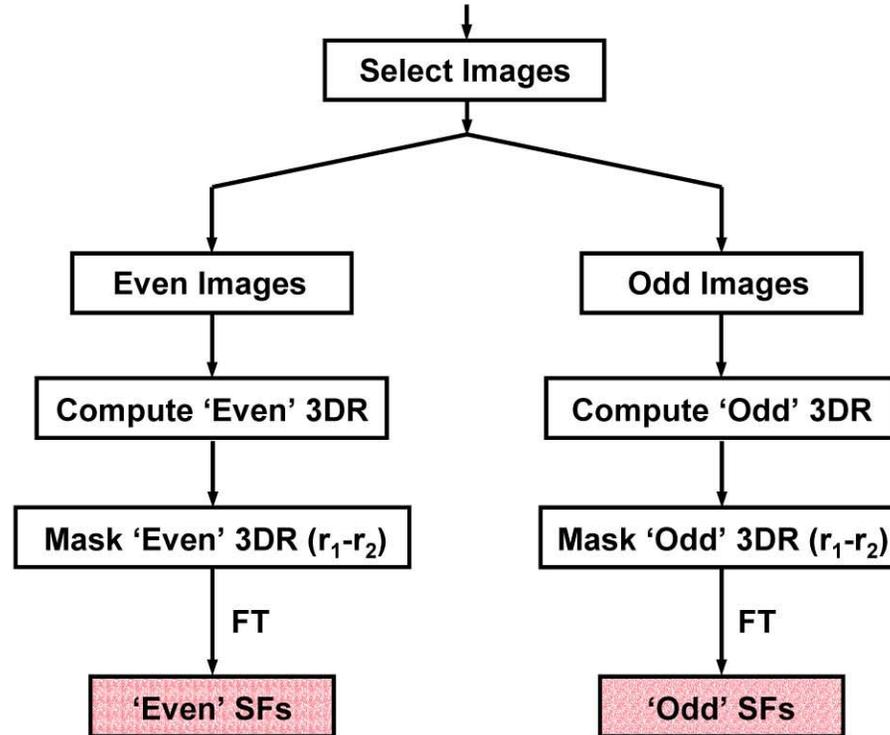
Icosahedral Particle Reconstruction Scheme



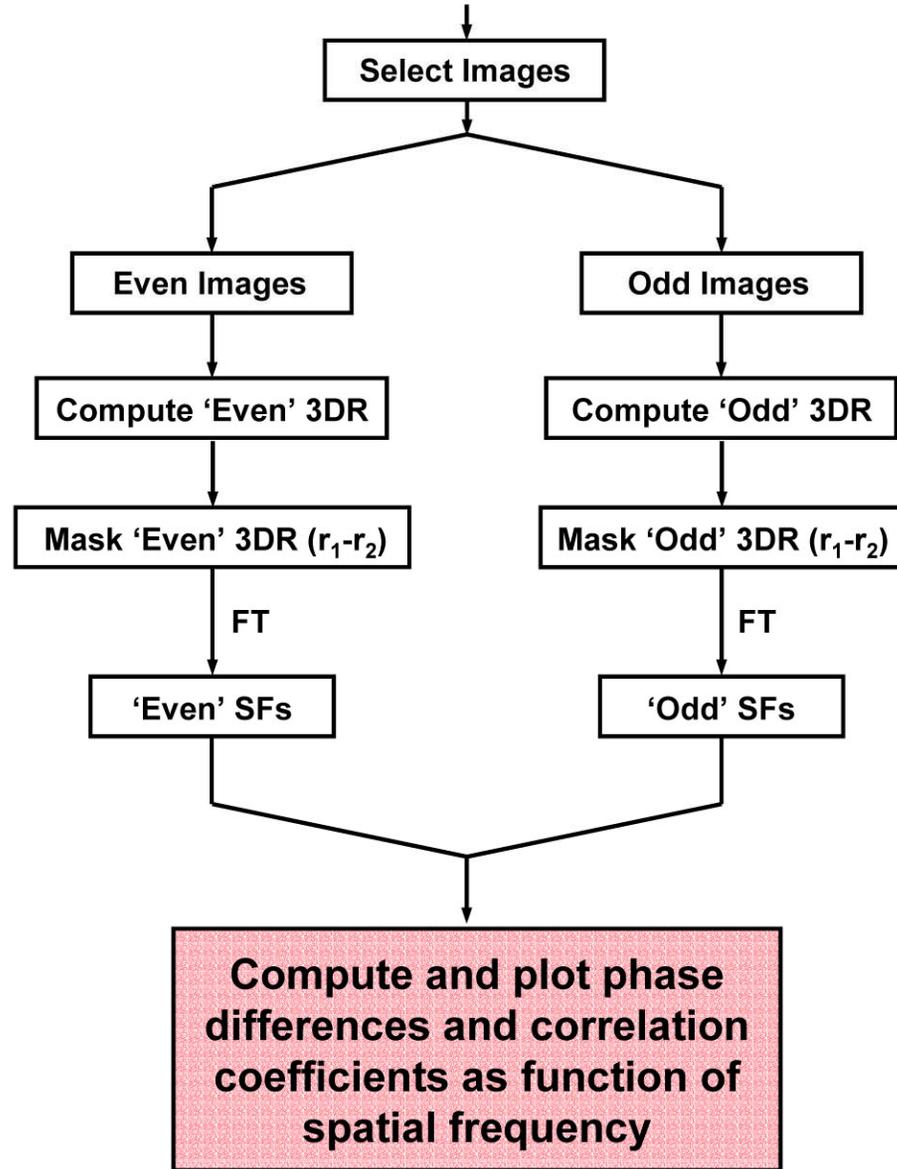
Icosahedral Particle Reconstruction Scheme



Icosahedral Particle Reconstruction Scheme

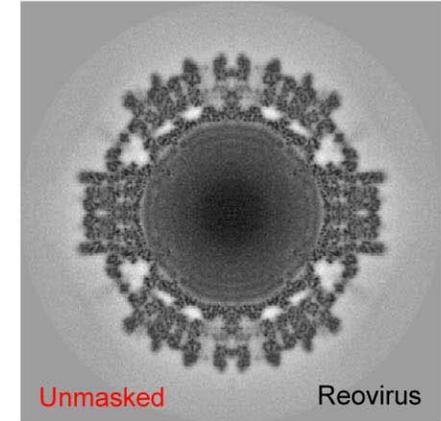
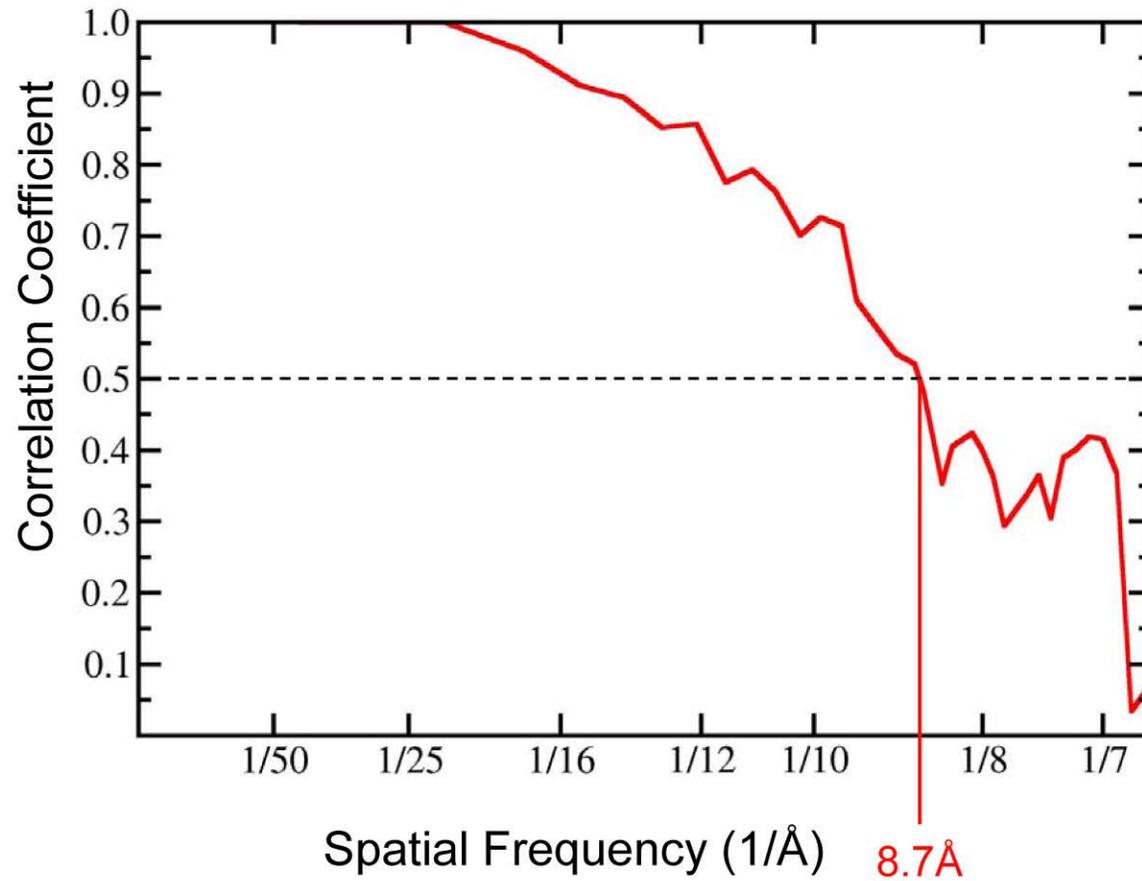


Icosahedral Particle Reconstruction Scheme



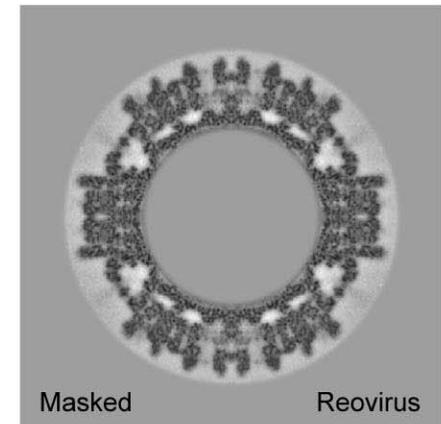
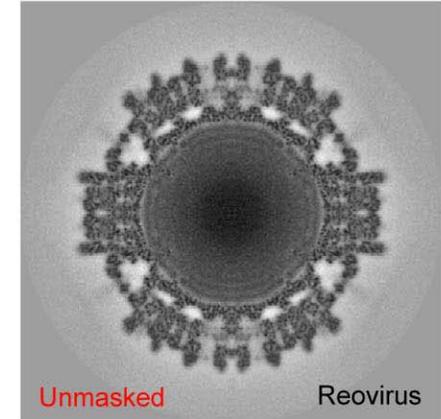
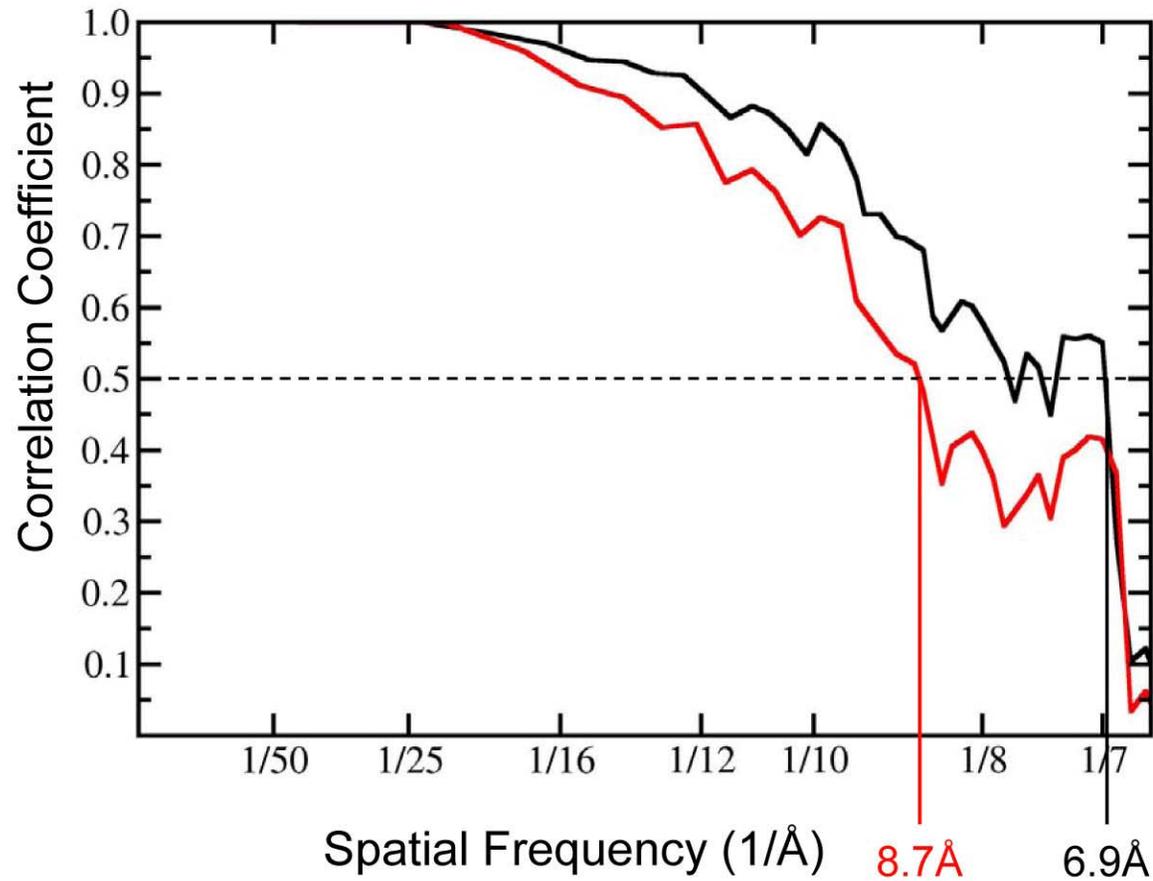
Icosahedral Particle Reconstruction Scheme

Monitor Data Quality



Icosahedral Particle Reconstruction Scheme

Monitor Data Quality



Icosahedral Particle Reconstruction Scheme

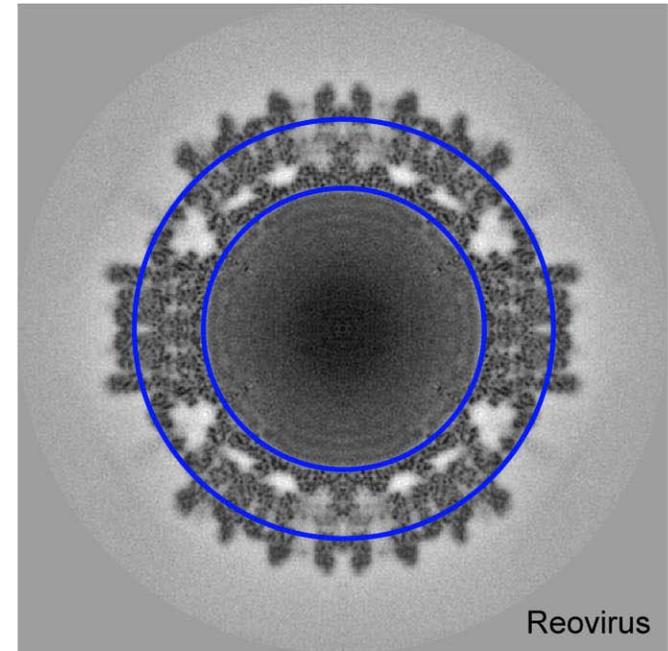
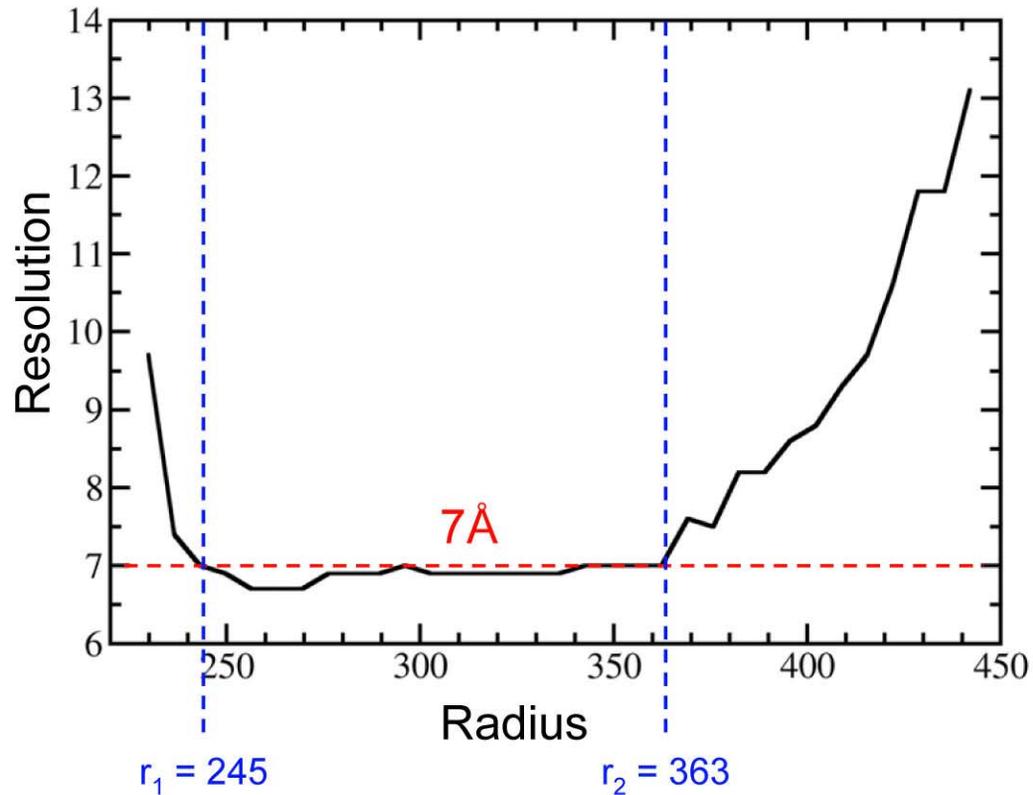


Monitor Data Quality

Note: quality of 3D density map is not the identical throughout the map

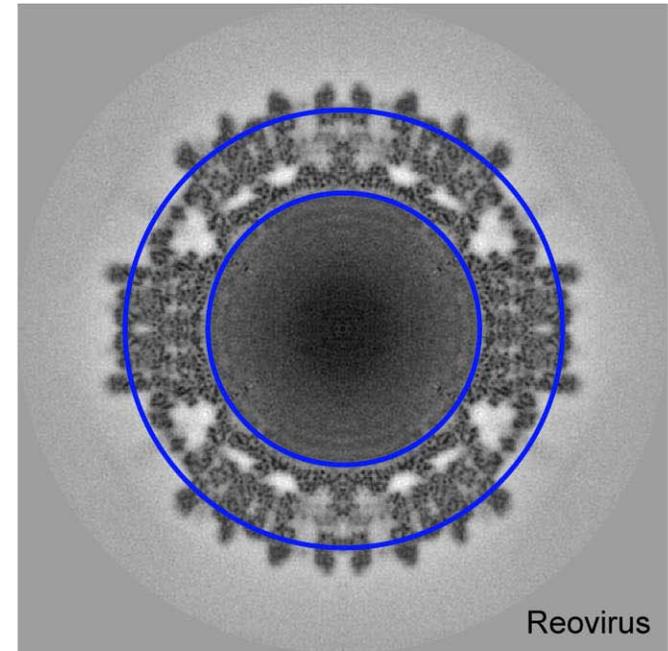
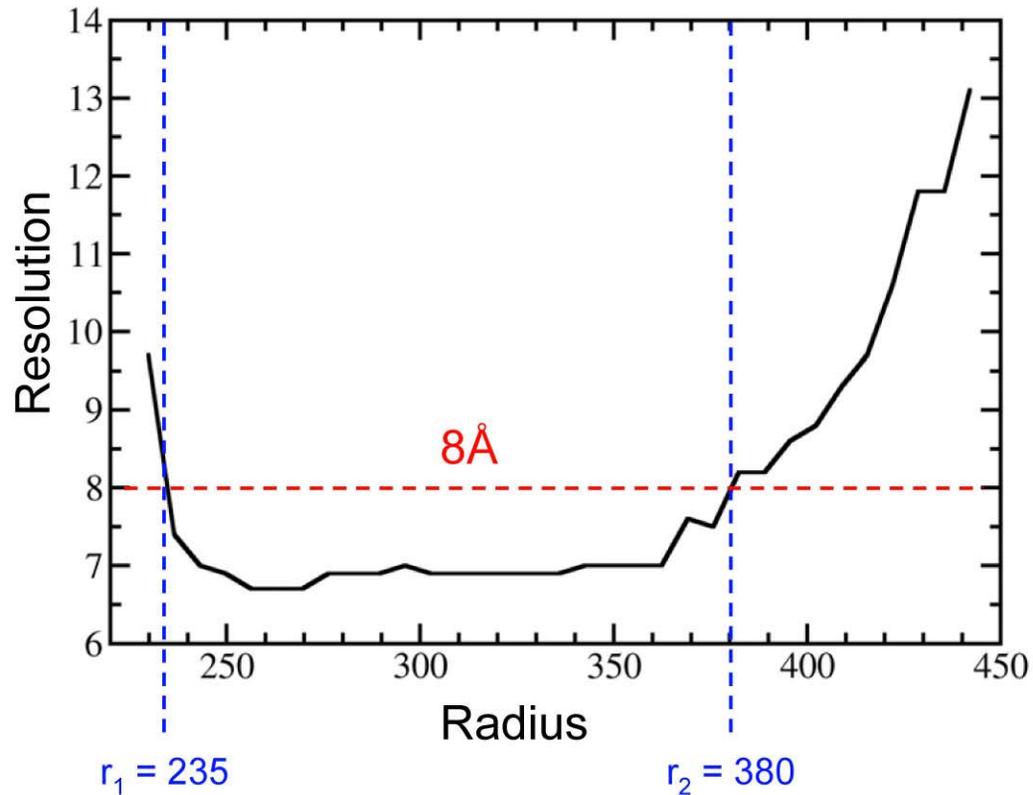
Icosahedral Particle Reconstruction Scheme

Monitor Data Quality



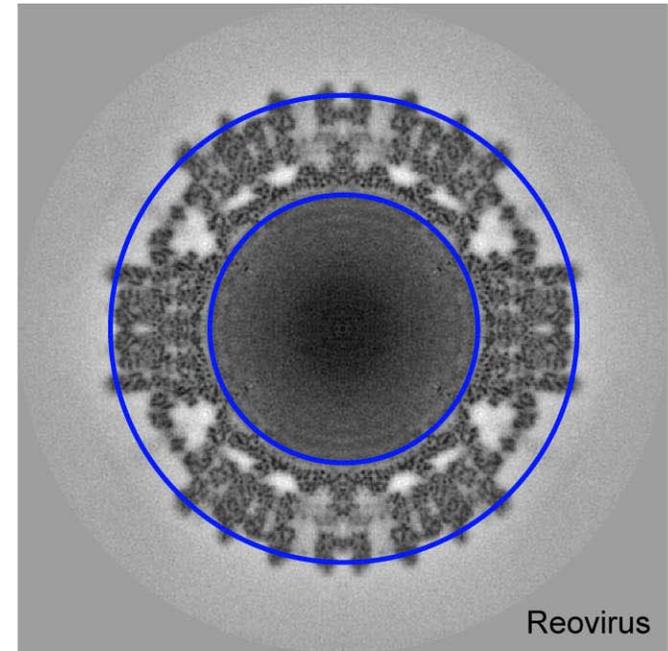
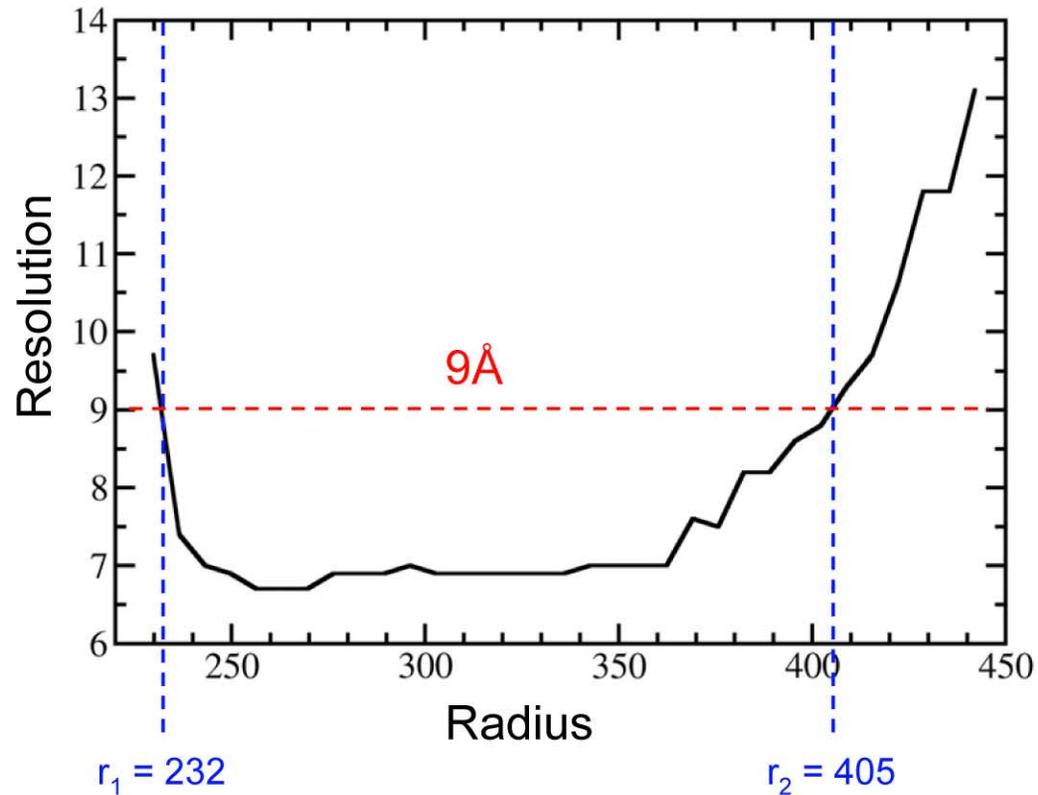
Icosahedral Particle Reconstruction Scheme

Monitor Data Quality

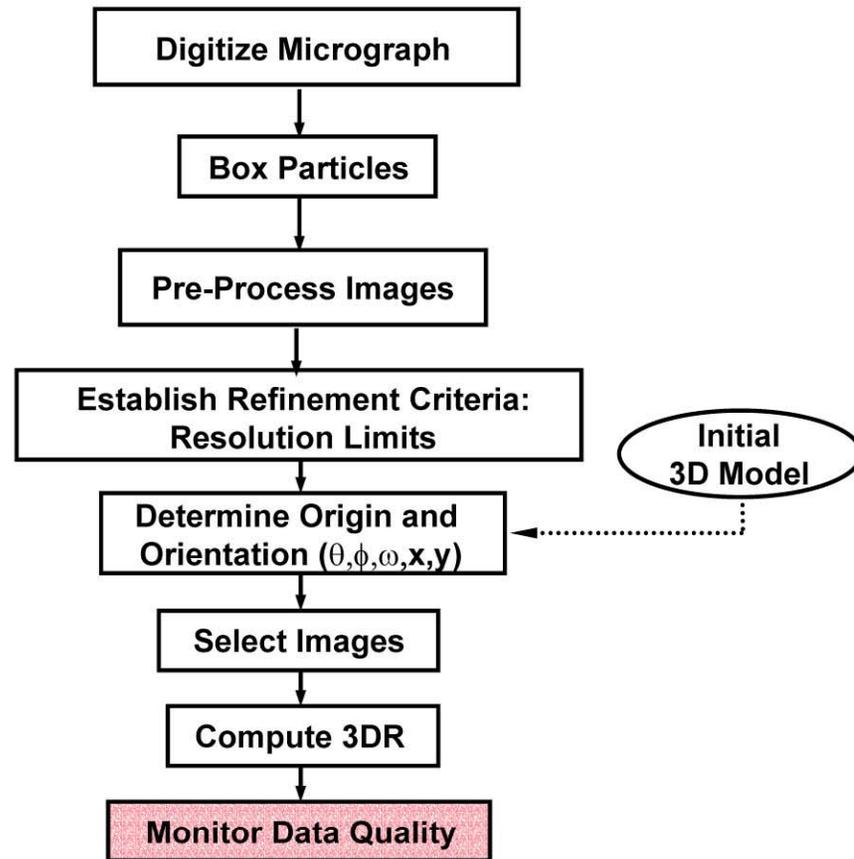


Icosahedral Particle Reconstruction Scheme

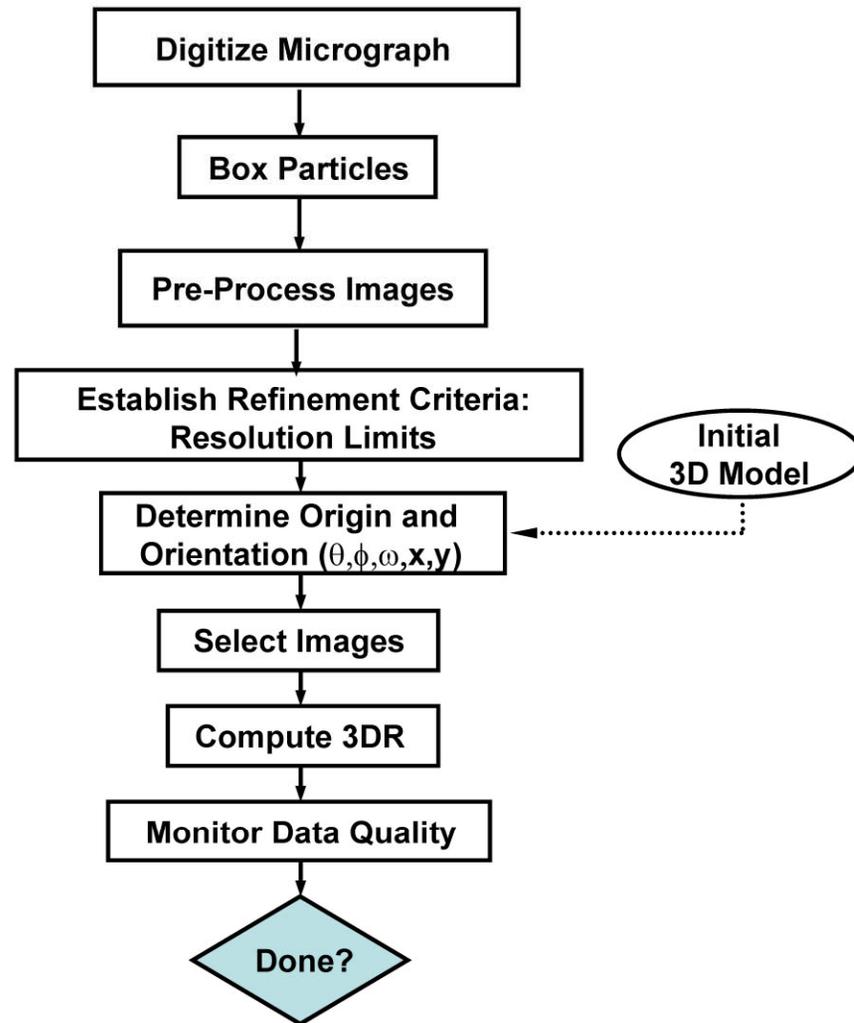
Monitor Data Quality



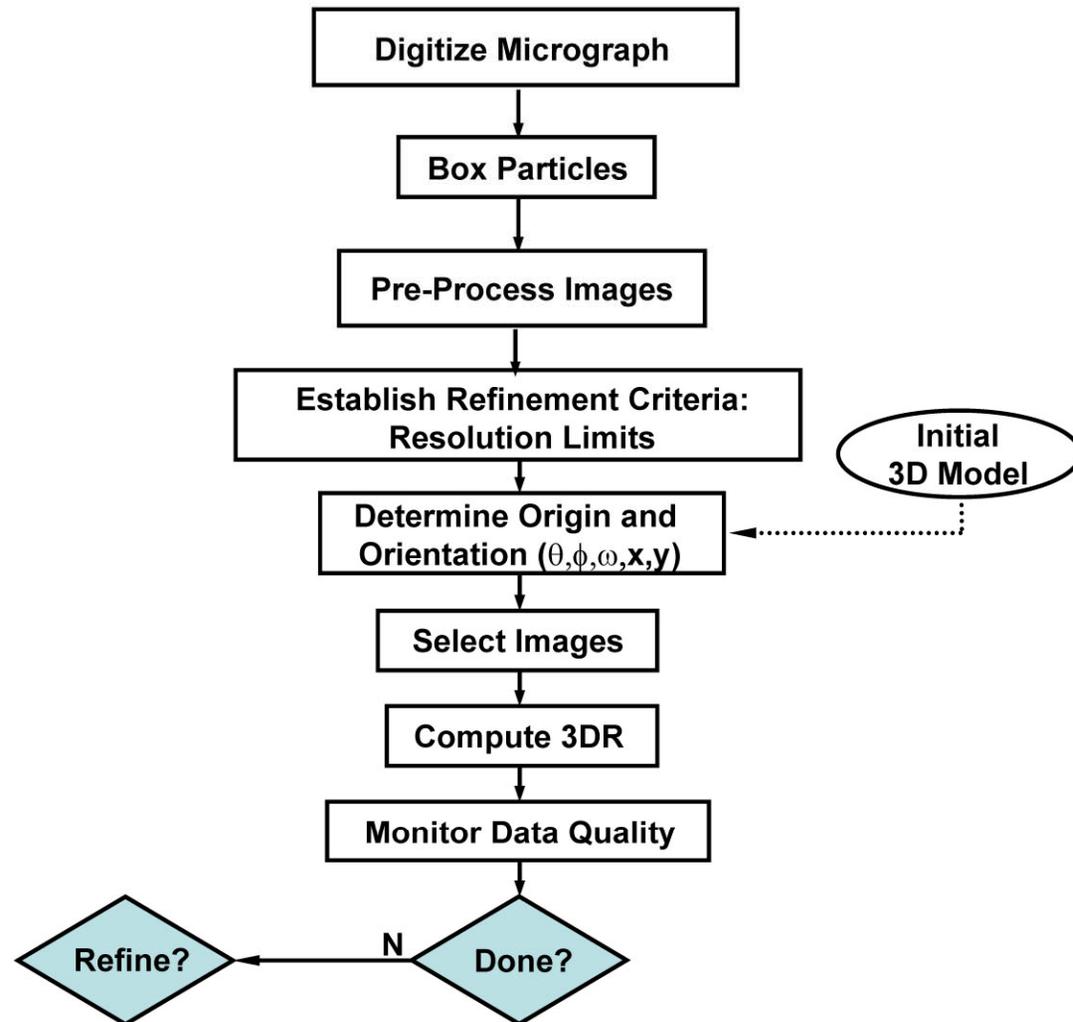
Icosahedral Particle Reconstruction Scheme



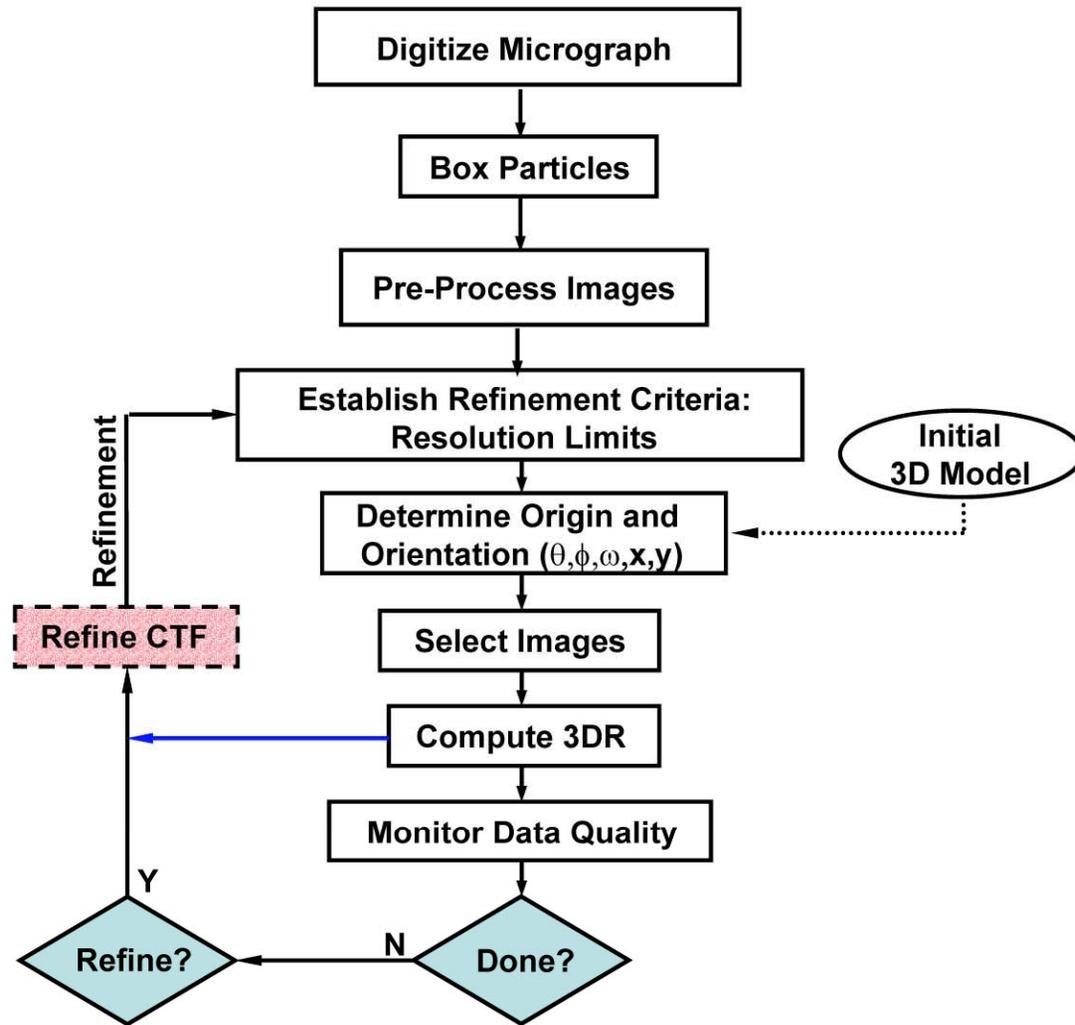
Icosahedral Particle Reconstruction Scheme



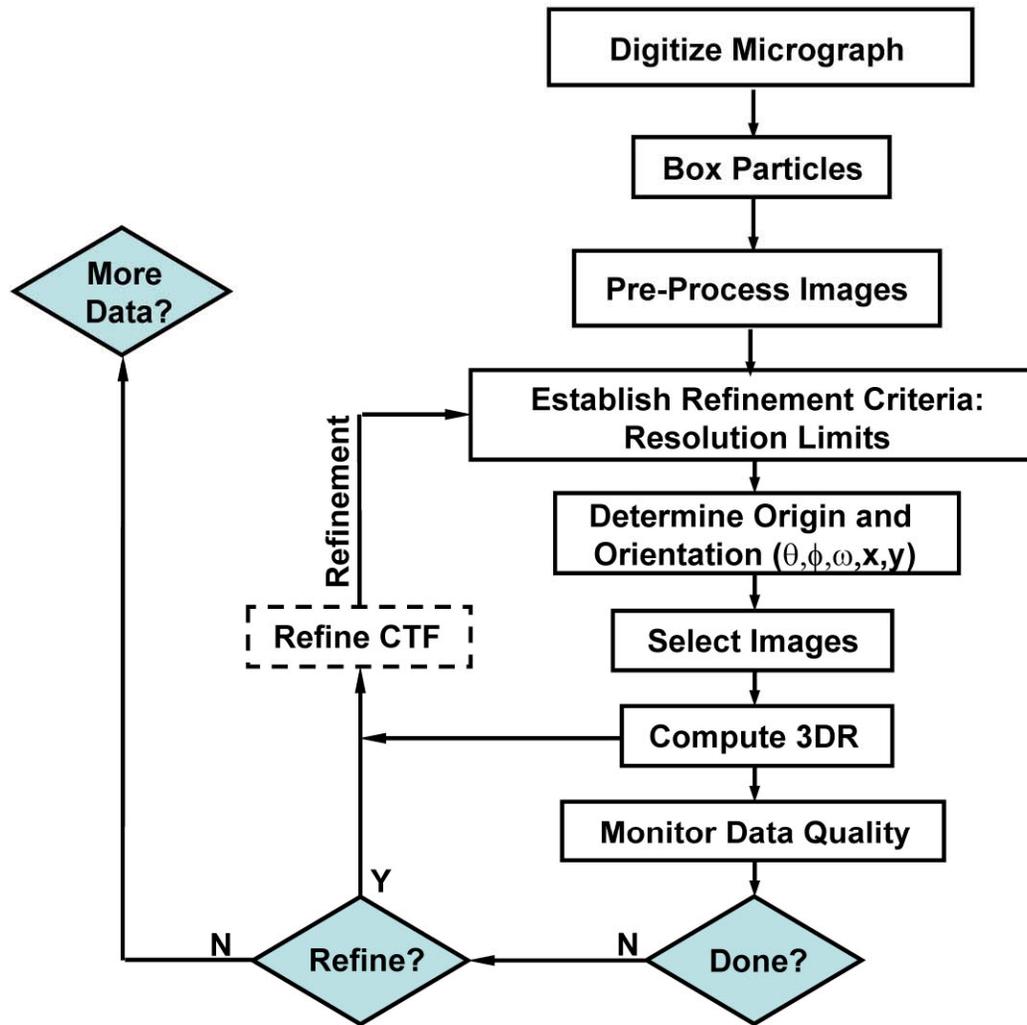
Icosahedral Particle Reconstruction Scheme



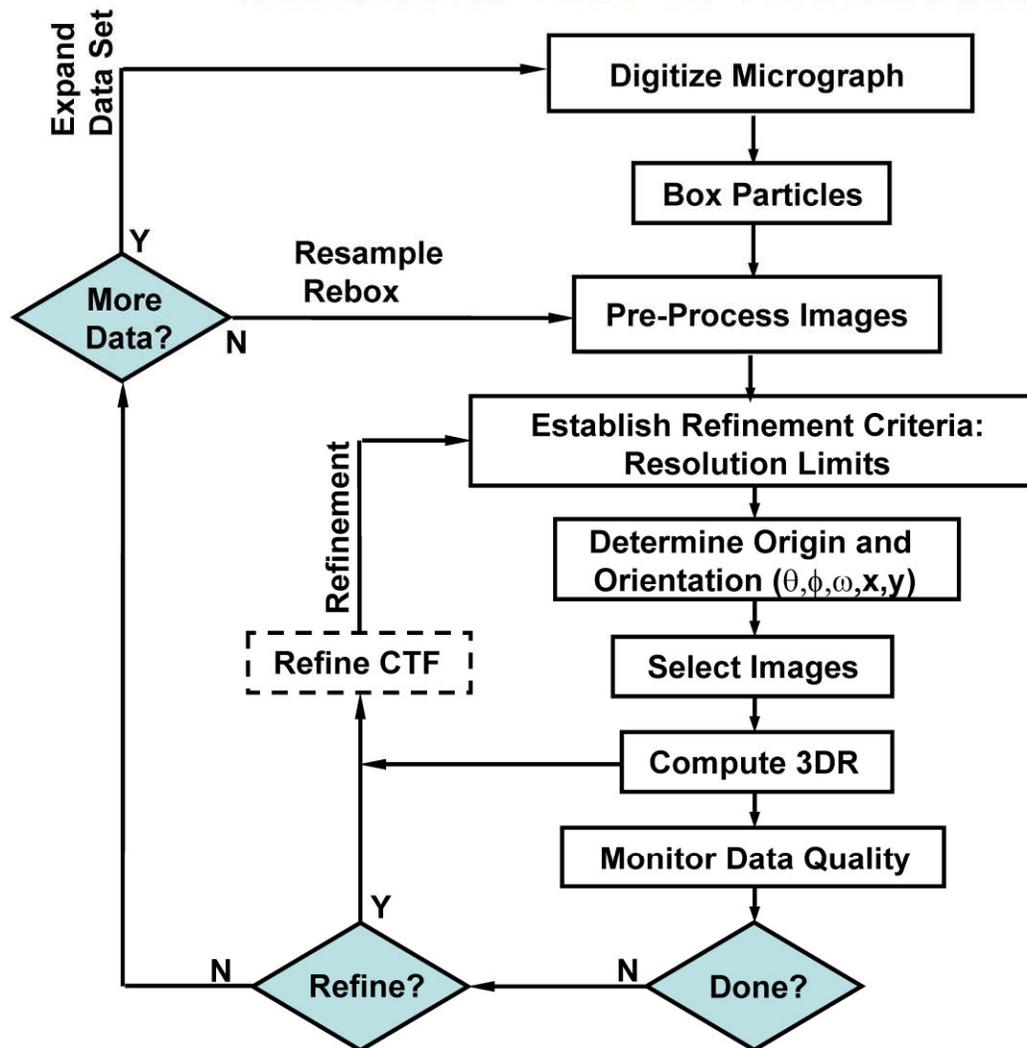
Icosahedral Particle Reconstruction Scheme



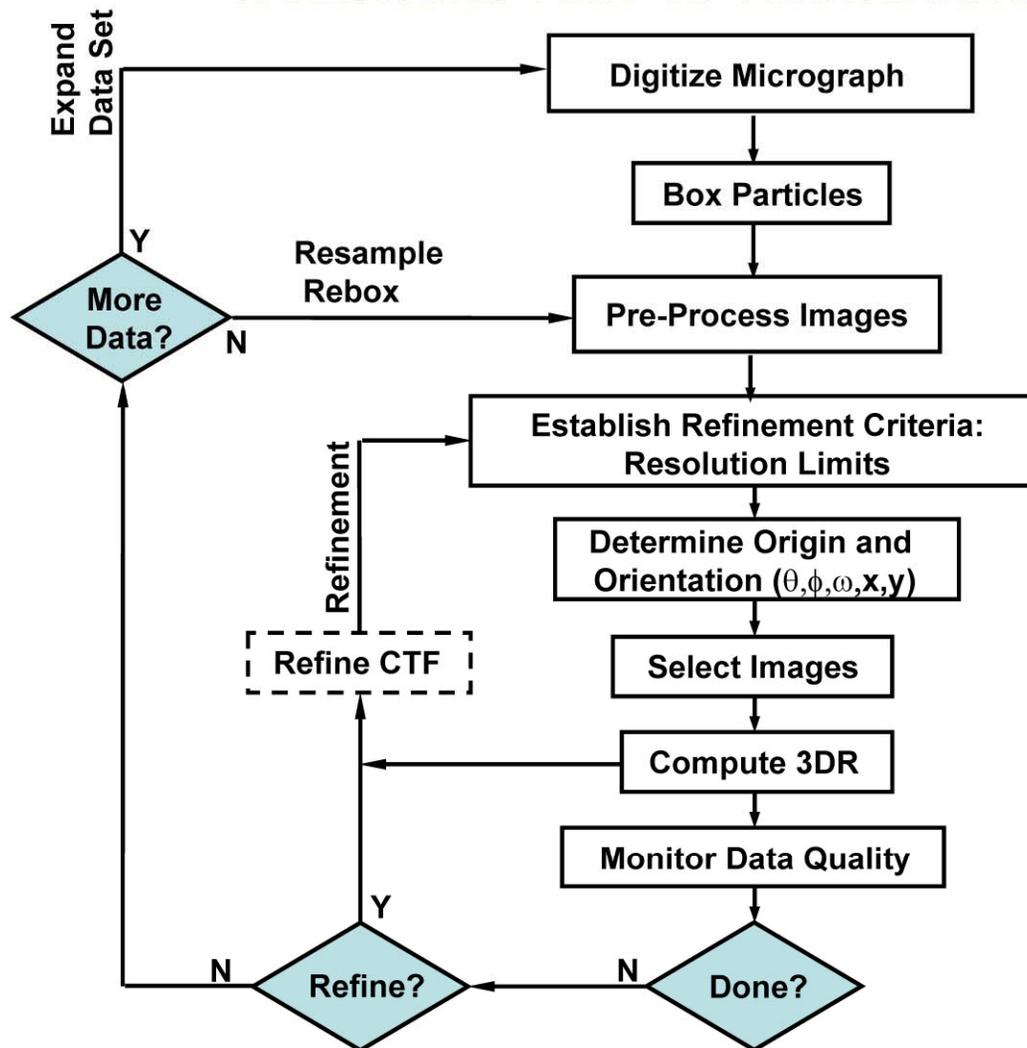
Icosahedral Particle Reconstruction Scheme



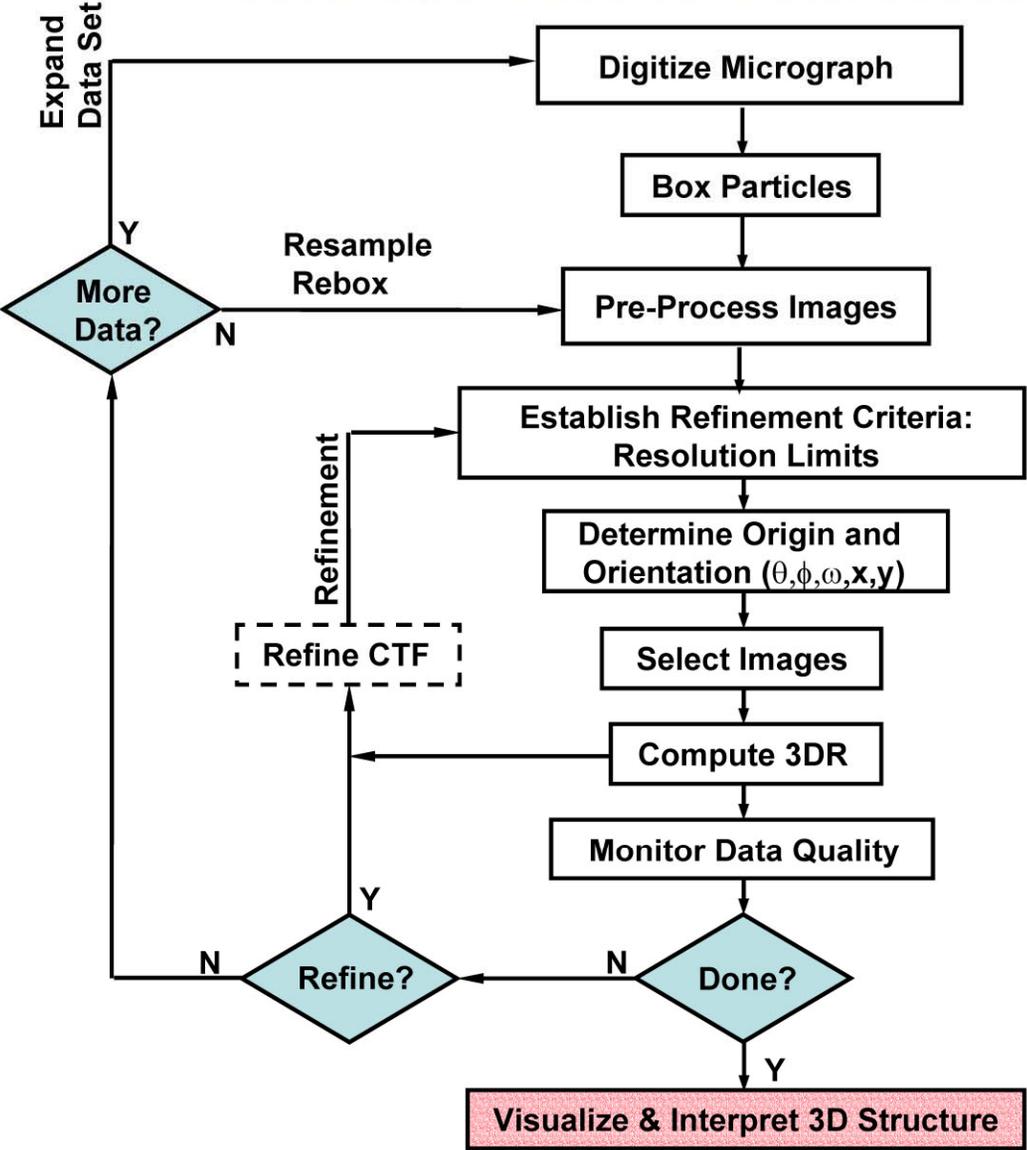
Icosahedral Particle Reconstruction Scheme



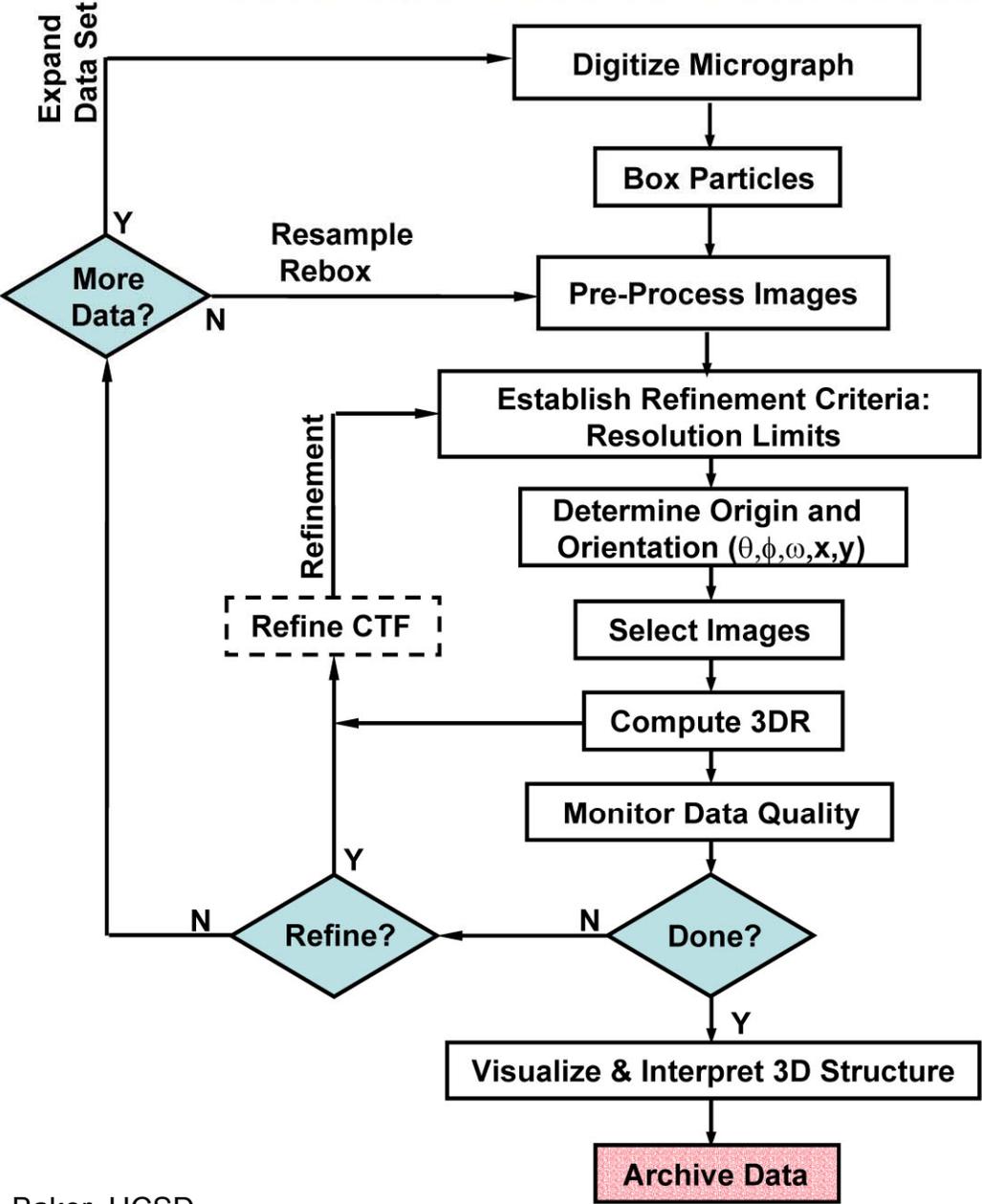
Icosahedral Particle Reconstruction Scheme



Icosahedral Particle Reconstruction Scheme



Icosahedral Particle Reconstruction Scheme



References

- Crowther, R.A., Amos, L.A., Finch, J.T., DeRosier, D.J. & Klug, A. Three dimensional reconstructions of spherical viruses by Fourier synthesis from electron micrographs. *Nature* **226**, 421-5 (1970).
- Crowther, R.A. Procedures for three-dimensional reconstruction of spherical viruses by Fourier synthesis from electron micrographs. *Philos Trans R Soc Lond B Biol Sci* **261**, 221-30 (1971).
- Böttcher, B., Wynne, S.A. & Crowther, R.A. Determination of the fold of the core protein of hepatitis B virus by electron cryomicroscopy. *Nature* **386**, 88-91 (1997).
- Conway, J.F., Cheng, N., Zlotnick, A., Wingfield, P.T., Stahl, S.J. & Steven, A.C. Visualization of a 4-helix bundle in the hepatitis B virus capsid by cryo-electron microscopy. *Nature* **386**, 91-4 (1997).
- Zhou, Z.H., Dougherty, M., Jakana, J., He, J., Rixon, F.J. & Chiu, W. Seeing the herpesvirus capsid at 8.5 Å. *Science* **288**, 877-80 (2000).
- Mancini, E.J., Clarke, M., Gowen, B.E., Rutten, T. & Fuller, S.D. Cryo-electron microscopy reveals the functional organization of an enveloped virus, Semliki Forest virus. *Mol Cell* **5**, 255-66 (2000).
- Baker, T.S. & Cheng, R.H. A model-based approach for determining orientations of biological macromolecules imaged by cryoelectron microscopy. *J Struct Biol* **116**, 120-30 (1996).
- Baker, T.S., Olson, N.H. & Fuller, S.D. Adding the third dimension to virus life cycles: three-dimensional reconstruction of icosahedral viruses from cryo-electron micrographs. *Microbiol & Mol Biol Rev* **63**, 862-922 (1999).
- Thuman-Commike, P.A. & Chiu, W. Reconstruction principles of icosahedral virus structure determination using electron cryomicroscopy. *Micron* **31**, 687-711 (2000).
- Zhou, Z.H. and Chiu, W. (2003). Structural determination of icosahedral viruses by electron cryomicroscopy at sub-nanometer resolution. *Adv Protein Chem* **64**: 93-130.