Cryo-EM Structure Determination through Eigenvectors of Sparse Matrices

Yoel Shkolnisky

Department of Mathematics
Program in Applied Mathematics
Yale University

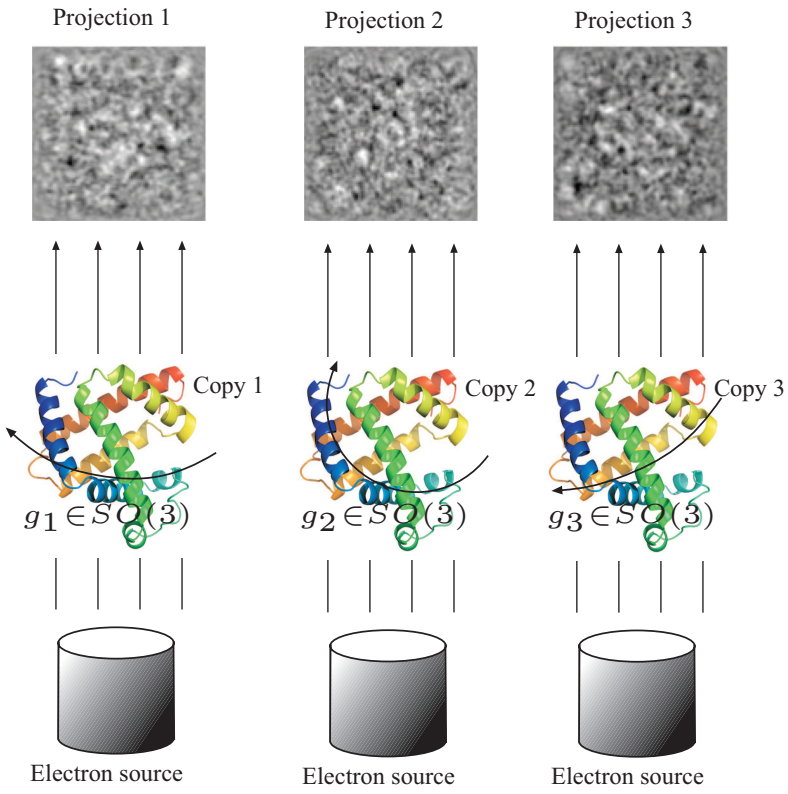
Image Analysis Challenges in Molecular Microscopy – IPAM

January 28 – February 1, 2008

Joint work with...

- *Amit Singer* (Yale University, Applied Mathematics)
- *Ronald R. Coifman* (Yale University, Applied Mathematics)
- *Fred Sigworth* (Yale School of Medicine, Cellular & Molecular Physiology)

Physical setting – CryoEM imaging



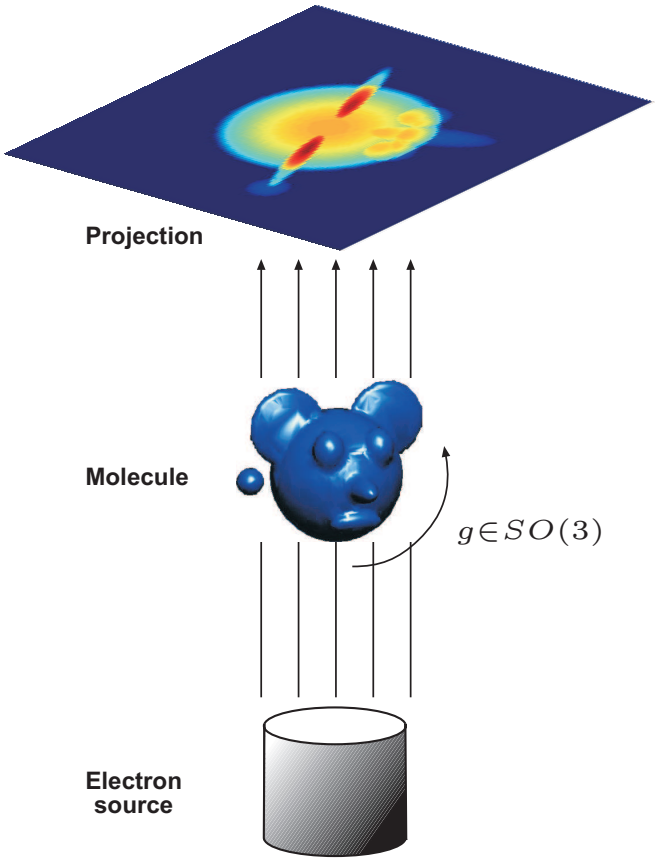
- CryoEM: Put sample in liquid medium, freeze, take images using an Electron Microscope.
- Many images: each image corresponds to a different copy of the molecule in a different spatial orientation.
- Orientations are random and unknown: there might a preferred orientation!

Goal: Reconstruct the 3D molecule from its projections taken at random unknown directions

Physical setting – CryoEM imaging (cont.)

- Imaging process destroys the sample: a single molecule can be imaged only once.
- Images are small (100×100) and noisy (signal to noise ratio < 1).
- If orientations were known then reconstruction could be done by the classical tomography algorithms.
- BUT, orientations are unknown!
- The structure of the molecule is also unknown.
- Can the orientations be revealed from the images themselves, without knowing what is being imaged?
- Assumption: the molecule has no symmetries.

What is being measured by the electron microscope? – The Radon transform

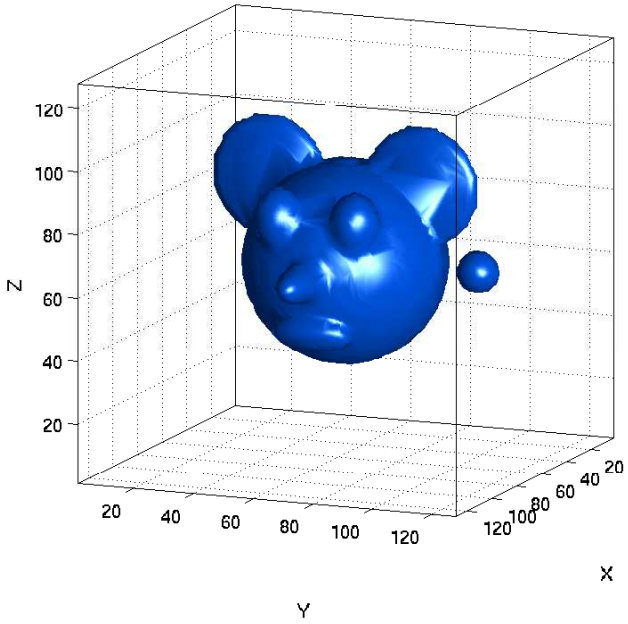


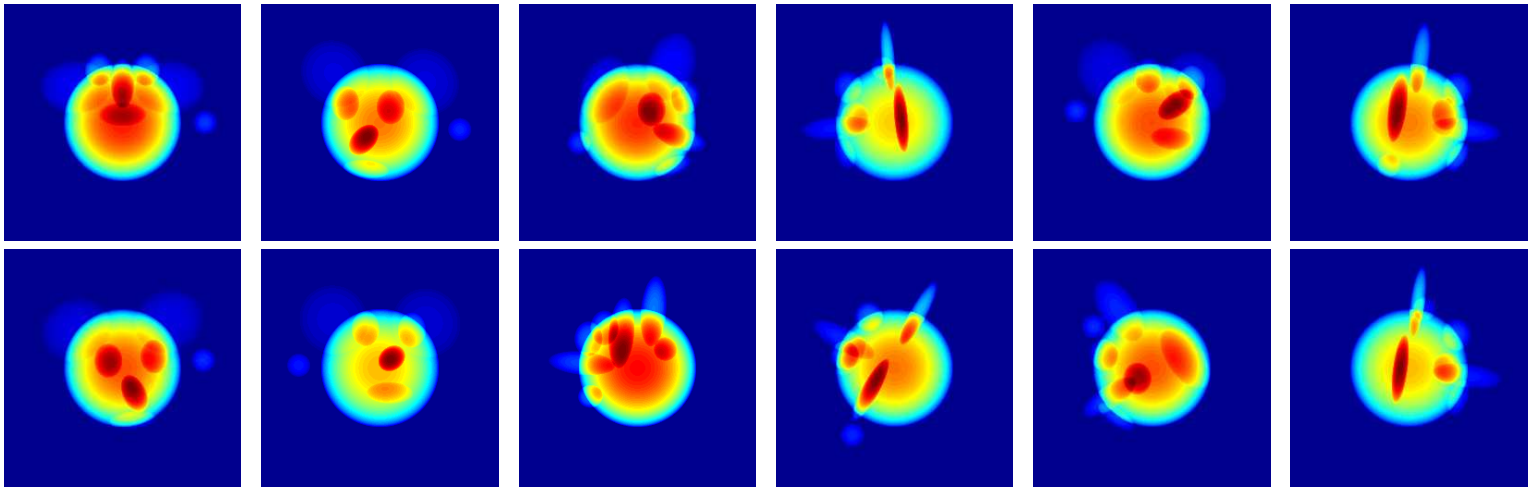
- $\phi(r)$ is the electric potential of the molecule ($r \in \mathbb{R}^3$).
- The molecule is rotated by $g \in SO(3)$: $\phi_g(r) = \phi(g^{-1}r)$.
- A projection image is

$$P_g(x, y) = \int_{-\infty}^{\infty} \phi_g(x, y, z)\, dz.$$

- Each projection is a collections of line integrals through the molecule in a direction determined by $g \in SO(3)$.

Projection Images: Toy Example





The Fourier projection-slice theorem

- The Radon transform and the Fourier transform are related by the projection-slice theorem.
- $\theta \in S^2$ beaming direction, image is formed on the orthogonal plane $\theta^\perp$.
- The 2D FT of the projection image is the double integral

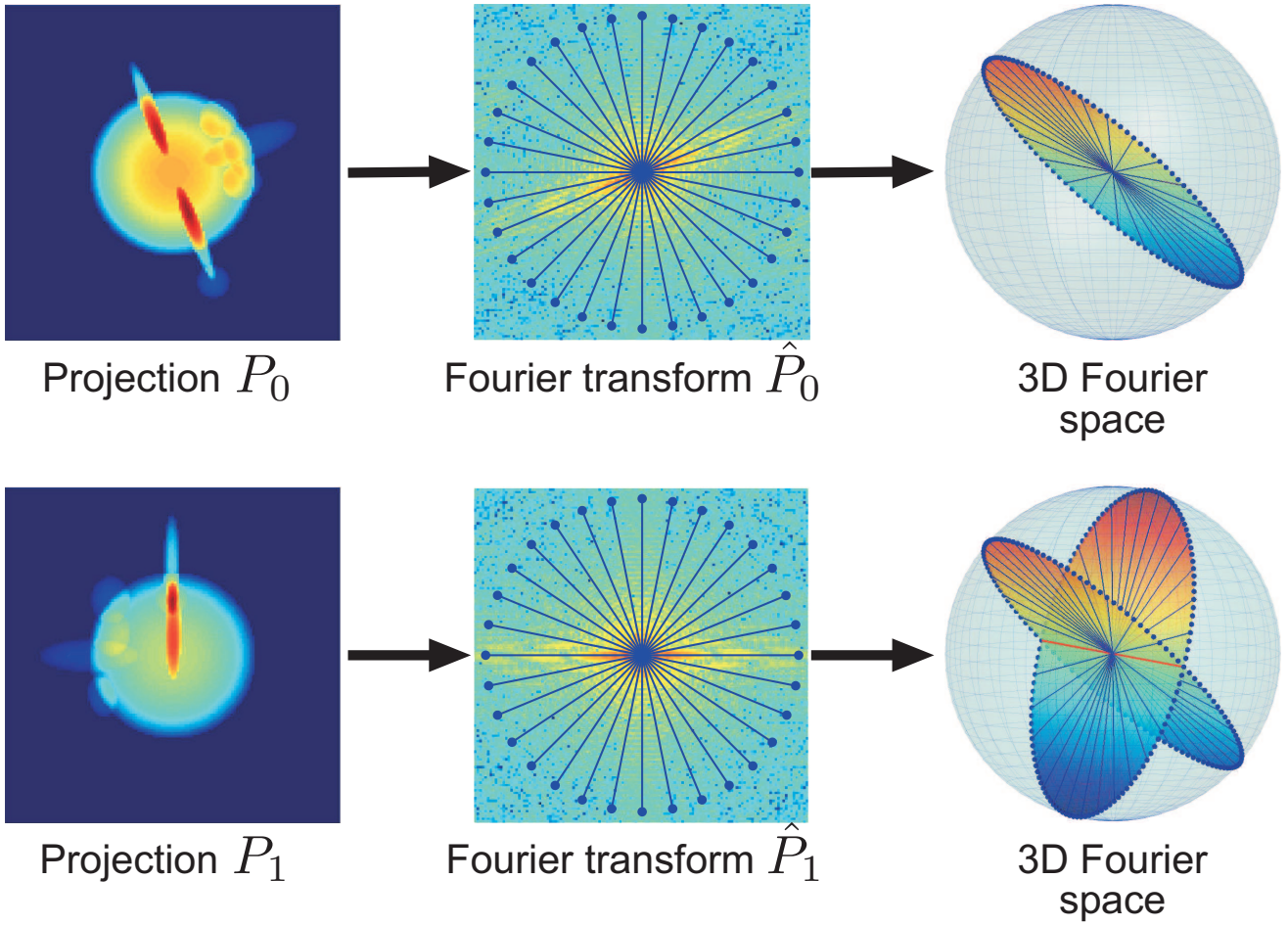
$$\hat{P}_\theta(\xi) = \int_{\theta^\perp} e^{-ir\cdot\xi} P_\theta(r)\, dr.$$

- The 3D FT of the molecule is the triple integral

$$\hat{\phi}(\xi) = \int_{\mathbb{R}^3} e^{-ir\cdot\xi} \phi(r)\, dr.$$

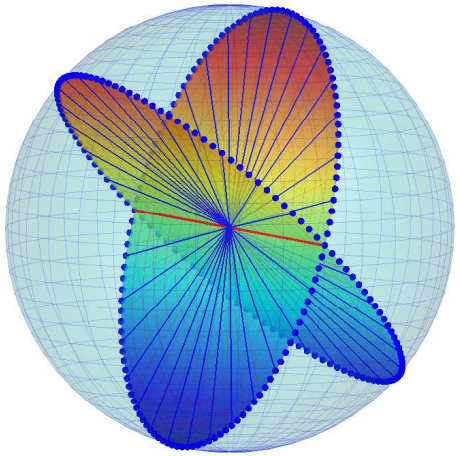
- Projection-slice Theorem: $\hat{P}_\theta(\eta) = \hat{\phi}(\eta), \quad \eta \in \theta^\perp.$

Geometry of the projection-slice theorem



The 2D Fourier transform of a projection image is a slice of the 3D Fourier transform of the molecule.

Geometry of the projection-slice theorem (cont.)



- The Fourier transform of any two projections has a common line – a ray in 3D Fourier space of the molecule.
- Each ray of the 3D Fourier transform of the molecule corresponds to a point on S^2 (the direction of the ray).
- Every image corresponds to a great circle over S^2.

Molecular reconstruction puzzle

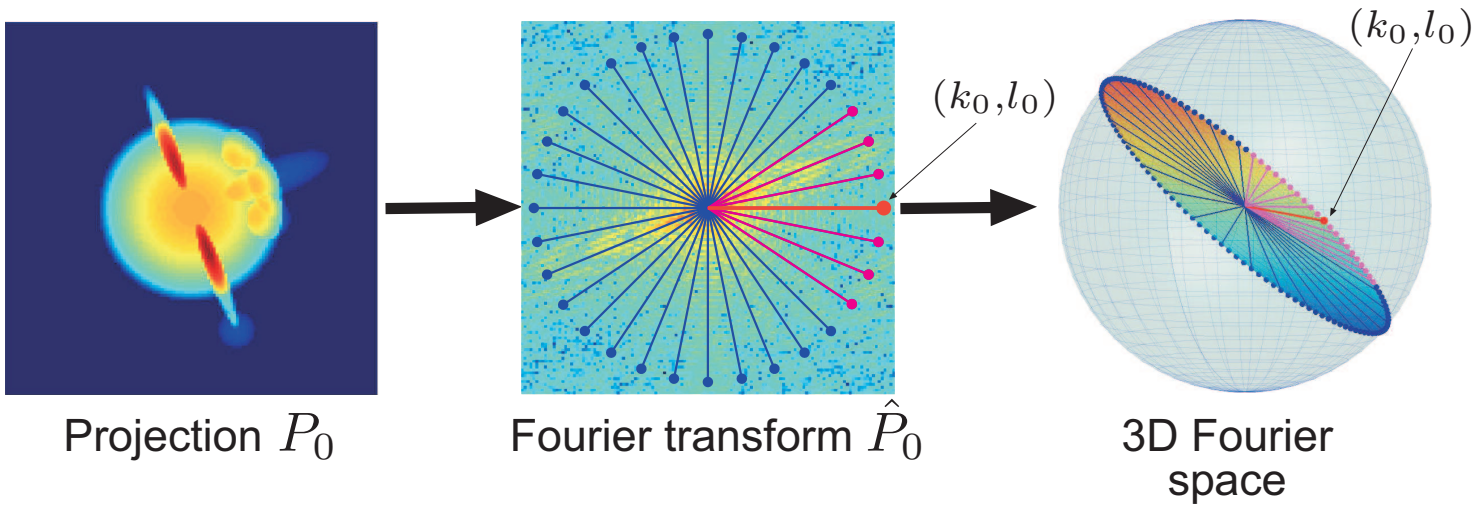
- The radial Fourier rays are the puzzle pieces – not the projection images.
- Every image is a circular chain of pieces – Fourier projection-slice theorem.
- Common line: meeting point
- Goal: determine the location of each piece – orientation of each radial line.

Main idea

Construct an operator on the dataset (Fourier rays) whose eigenvectors reveal the orientation of each ray

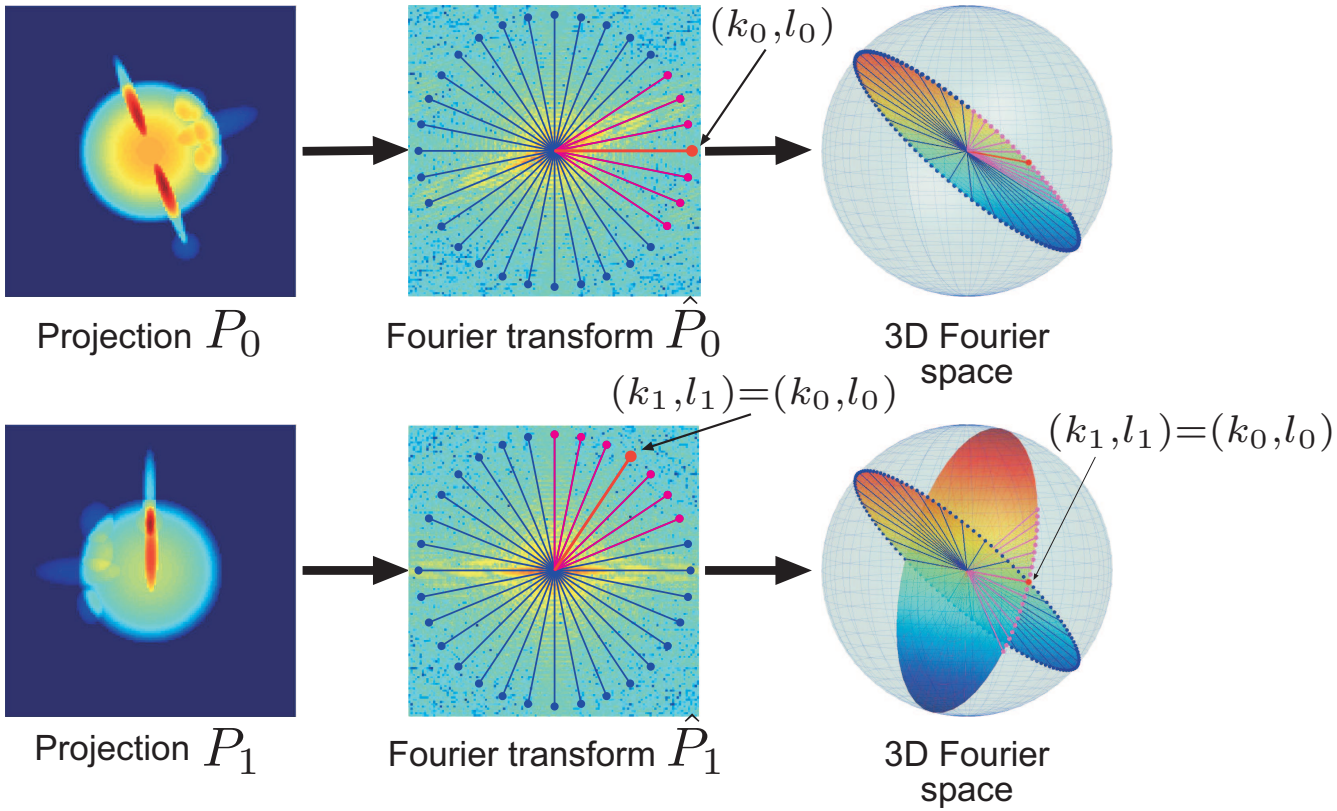
- Construct a graph where each Fourier ray is a node.
- Connect edges using the Fourier projection-slice theorem.
- Explore the spectrum and eigenvectors of the graph – eigenvectors will reveal orientations.

Identify local geometry



- The Fourier transform of a projection P_0 corresponds to a plane in the 3D Fourier space, that is , a great circle on S^2.
- The neighbors of (k_0, l_0) on $\hat{P}_0$ lie on a geodesic arc on S^2 centered at (k_0, l_0).
- Identify d such neighbors (say 10) on each side of (k_0, l_0) – write down their indices $(k_0, l_{-d}), \ldots, (k_0, l_d)$.

Identify local geometry – two projections with a common line

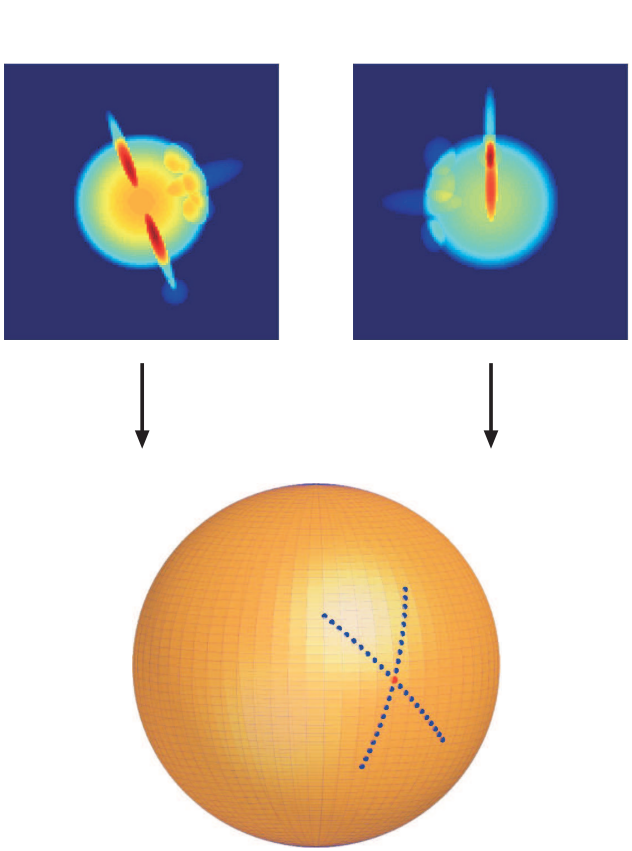


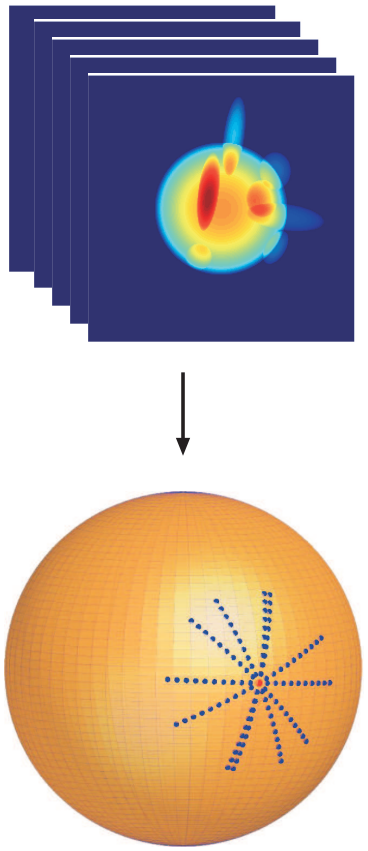
- $\hat{P}_1$ gives another geodesic circle through (k_0, l_0).
- Identify d neighbors on each side of (k_0, l_0) on this circle – $(k_1, l_{-d}), \ldots, (k_1, l_d)$.

Identify local geometry – "spider" neighborhood

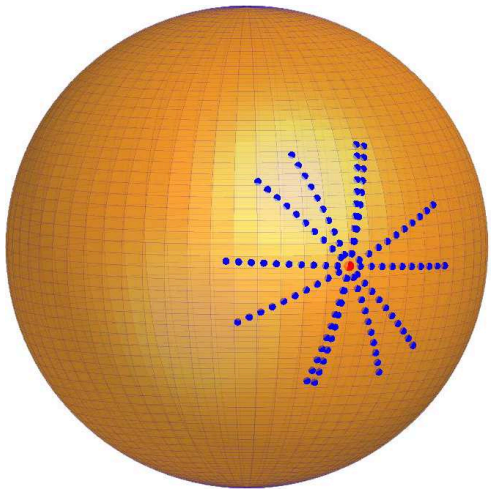
Two projections with a common line give intersecting arcs on S^2

Repeat for several projections having (k_0, l_0) as a common line – "spider" neighborhood





Properties of the "spider" neighborhood



- The projection-slice theorem allows to identify for each ray its "spider" neighborhood on S^2.
- The "spider" neighborhood is symmetric around (k_0, l_0).
- The average of their coordinates on S^2 is a multiple of the coordinate of (k_0, l_0) – each pair of opposite legs balance each other.

Orientation revealing operator

- Given K projection images, compute L equally-spaced polar Fourier rays.

- The dataset of the problem is the set of KL Fourier rays (and no longer the projection images).

- The "orientation revealing operator" is a matrix W of size $KL \times KL$ – each row corresponds to a Fourier ray (a point on S^2).

- For each row (Fourier ray) put 1's at columns that correspond to its "spider" neighbors – W is sparse.

Orientation revealing operator

- Normalize W to be row-stochastic – divide each row by its sum

$$D_{ii} = \sum_{j=1}^{KL} W_{ij},$$

$$A = D^{-1}W.$$

- A is not symmetric – spectrum may be complex.
- $\lambda_0 = 1$. Other eigenvalues satisfy $|\lambda| < 1$.
- $(1, \ldots, 1)^T$ is an eigenvector.
- Each "spider" may have a different number of legs.
- Any symmetric weights would work.

A as an averaging operator

- Take a vector $f = (f_1, \ldots, f_{KL})^T$.
- View f as a function on the dataset of the rays: f_j is the value of the function for ray j.
- Af is a vector whose j entry is the average of f over the "spider" neighborhood of ray j.

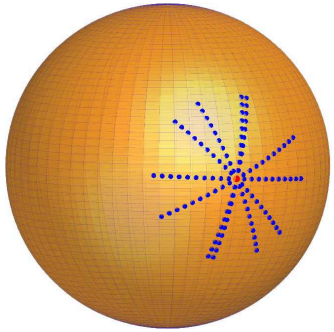
$$(Af)_j = \begin{pmatrix} & & \vdots & & \\ 0 & \frac{1}{|S_j|} & 0 & \frac{1}{|S_j|} & \ldots & 0 \\ & & \vdots & & \end{pmatrix} \begin{pmatrix} f_1 \\ \vdots \\ f_{KL} \end{pmatrix} = \frac{1}{|S_j|} \sum_{l \in S_j} f_l.$$

- A is an averaging operator on the dataset of Fourier rays - the geometry of the averaging is determined by "spider" neighborhoods.

Eigenvectors of A reveal orientations

- Let $p_j = (x_j, y_j, z_j) \in S^2$ be the direction vector of ray j.
- Since the "spider" neighborhood is symmetric around p_j

$$\frac{1}{|S_j|} \sum_{l \in S_j} p_l = \lambda p_j,$$



or, in coordinates:

$$\frac{1}{|S_j|} \sum_{l \in S_j} x_l = \lambda x_j, \quad \frac{1}{|S_j|} \sum_{l \in S_j} y_l = \lambda y_j, \quad \frac{1}{|S_j|} \sum_{l \in S_j} z_l = \lambda z_j.$$

- Reminder:

$$(Ax)_j = \frac{1}{|S_j|} \sum_{l \in S_j} x_l, \quad (Ay)_j = \frac{1}{|S_j|} \sum_{l \in S_j} y_l, \quad (Az)_j = \frac{1}{|S_j|} \sum_{l \in S_j} z_l.$$

Eigenvectors of A reveal orientations (cont.)

- So,
$$(Ax)_j = \lambda x_j, \quad (Ay)_j = \lambda y_j, \quad (Az)_j = \lambda z_j.$$

- The coordinate functions of the dataset $x = (x_1, \ldots, x_{KL})$, $y = (y_1, \ldots, y_{KL})$, $z = (z_1, \ldots, z_{KL})$ are eigenvectors of A.

- The direction vector of each Fourier ray $p_j = (x_j, y_j, z_j) \in S^2$ is given by the eigenvectors of A.

- If
$$A\psi_1 = \lambda\psi_1, \quad A\psi_2 = \lambda\psi_2, \quad A\psi_3 = \lambda\psi_3,$$
then
$$p_j = (x_j, y_j, z_j) = (\psi_1(j), \psi_2(j), \psi_3(j)) .$$

Algorithm

- Compute the polar Fourier transform of all projections.
- Find common lines for all pairs of Fourier images.
- Use common lines to construct the "spider" matrix A.
- Compute eigenvectors $A\psi_i = \lambda_i \psi_i$.
- Embed each Fourier ray r_j into the three linear eigenvectors (ψ_1, ψ_2, ψ_3)

$$r_j \mapsto (\psi_1(j), \psi_2(j), \psi_3(j)).$$

- The embedding reveals the projection orientations up to rotation and reflection.

Eigenvectors unmixing

- A is not symmetric – eigenvectors not orthogonal.
- x, y, z are eigenvectors (with the same eigenvalue), but so any linear combination of x, y, z.
- Computed eigenvectors of A give some arbitrary linear combination of x, y, z – unmix x, y, z from the given eigenvectors.
- Let ψ_1, ψ_2, ψ_3 be the computed eigenvectors, and let M be the 3×3 unmixing matrix such that

$$M \begin{pmatrix} — & \psi_1^T & — \\ — & \psi_2^T & — \\ — & \psi_3^T & — \end{pmatrix} = \begin{pmatrix} — & x^T & — \\ — & y^T & — \\ — & z^T & — \end{pmatrix}$$

Eigenvectors unmixing (cont.)

- Set

$$\Psi = \begin{pmatrix} — & \psi_1^T & — \\ — & \psi_2^T & — \\ — & \psi_3^T & — \end{pmatrix}, \qquad X = \begin{pmatrix} — & x^T & — \\ — & y^T & — \\ — & z^T & — \end{pmatrix}.$$

  We are looking for M such that $M\Psi = X$.

- x, y, z are coordinates on S^2 and so

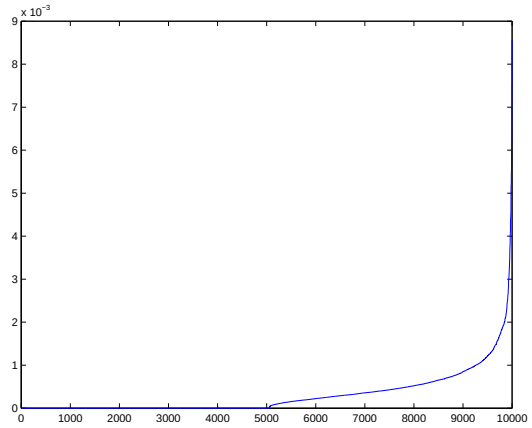
$$(X^T X)_{ii} = x_i^2 + y_i^2 + z_i^2 = 1, \quad \text{or,} \quad \left(\Psi^T M^T M \Psi\right)_{ii} = 1.$$

- Since Ψ is known, this gives KL linear equations for the 9 entries of $M^T M$ – solve using least-squares.

- Once $M^T M$ is given, factor M using Cholesky decomposition or SVD.

Improvements to the algorithm

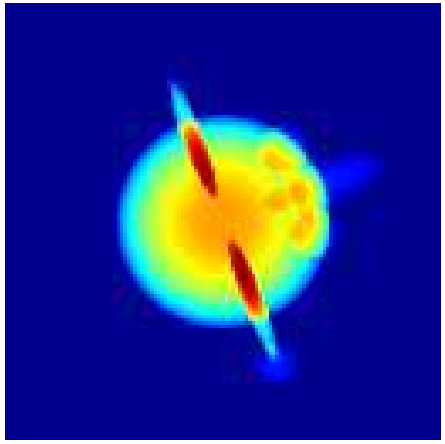
- Use Fourier space geometry again ("cosmetics"):
  PCA same image Fourier lines and equally space them.
- Do not use all intersection of all projection images. Use only certain common lines. This gives fewer arcs in each "spider" (sparser matrix) while increasing the accuracy of the embedding (due to fewer errors in A).

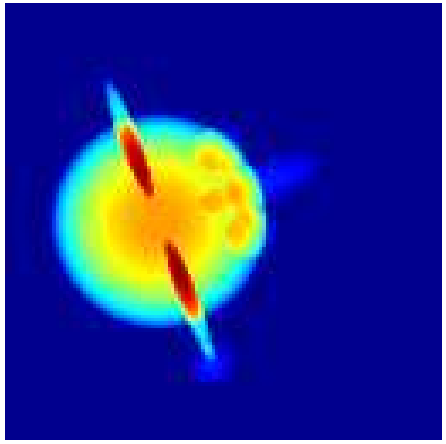


How "common" is the common line between images – sorted from best to worst

Improvements to the algorithm (cont.)

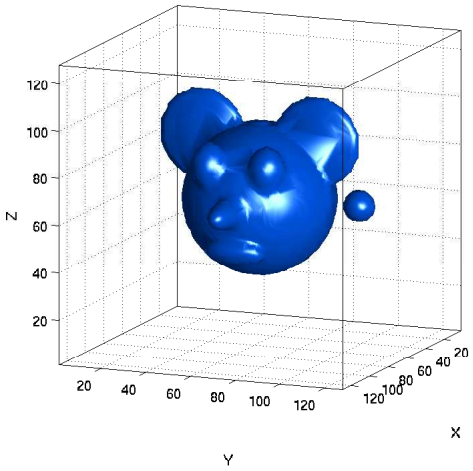
- Real-life projection images are never centered. The algorithm can be modified to estimate image translations as well as orientations simultaneously – essential for reconstruction.

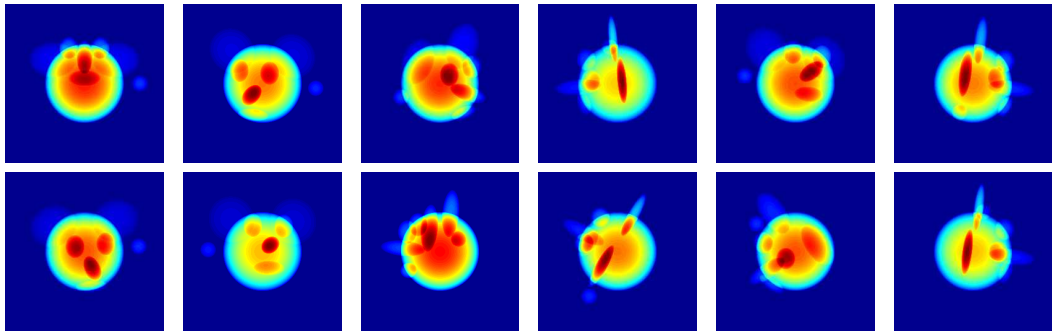




Results

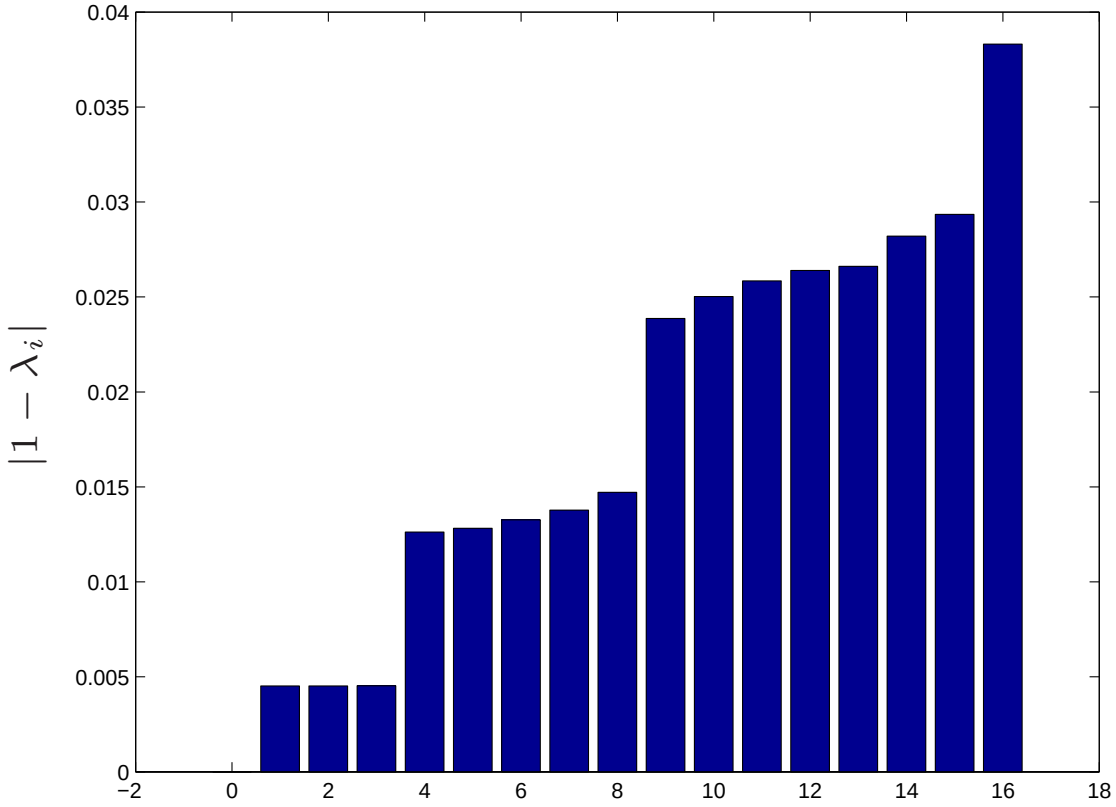
- Toy molecule: $K = 200$ projections; $L = 400$ Fourier rays in each projection.





Results (cont.)

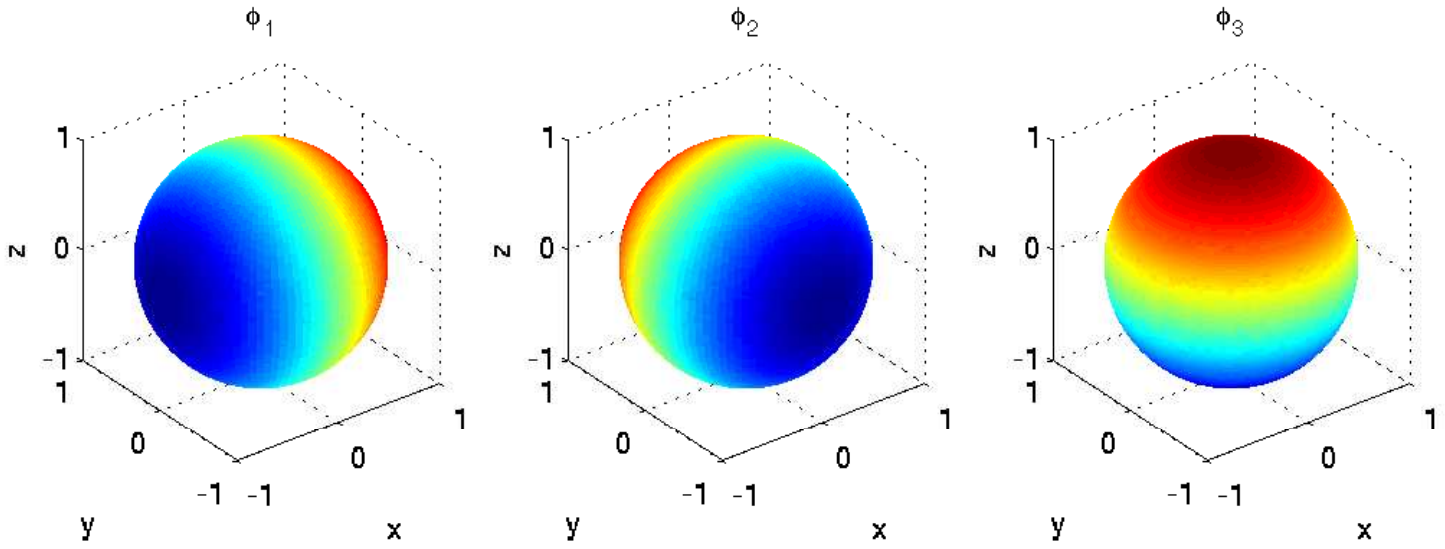
- Spectrum of the A



The eigenspace of dimension 3 is apparent.

Results (cont.)

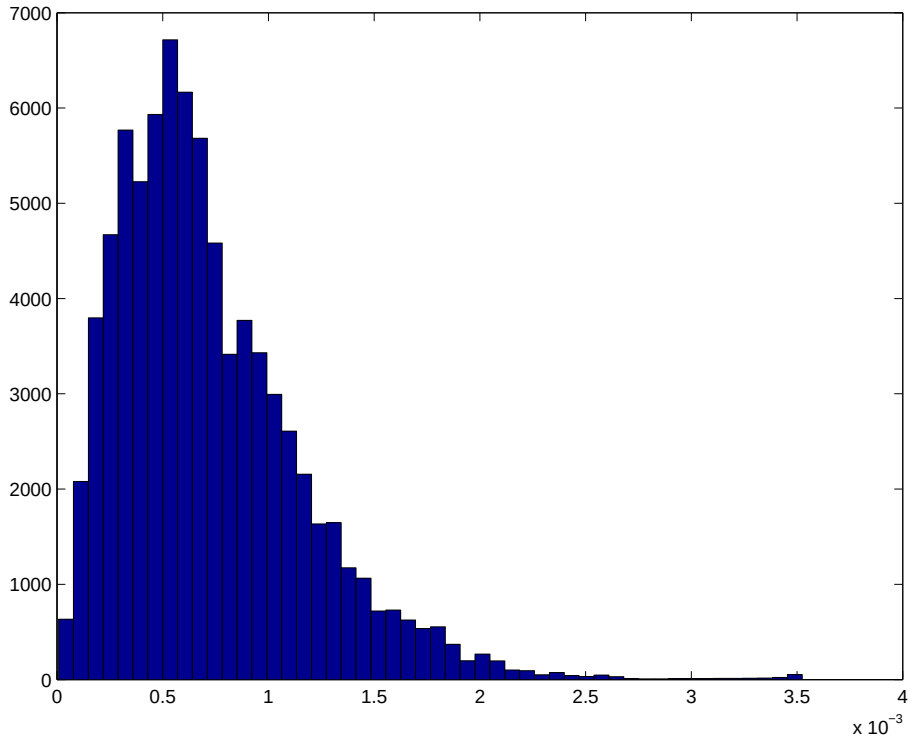
- Three linear eigenfunctions plotted on S^2 (after unmixing)



Clearly, these eigenfunctions correspond to the x, y, and z coordinates on S^2.

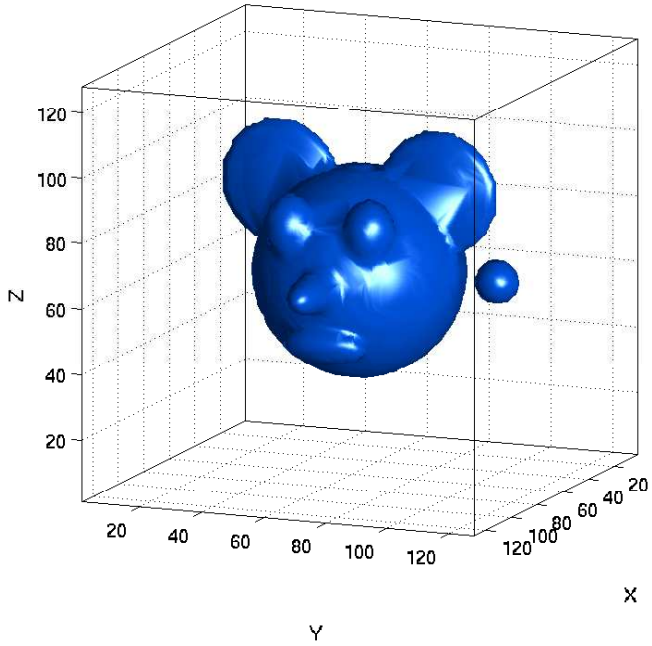
Results (cont.)

- Angle estimation error (angle between the true direction of each Fourier ray and its estimated direction)

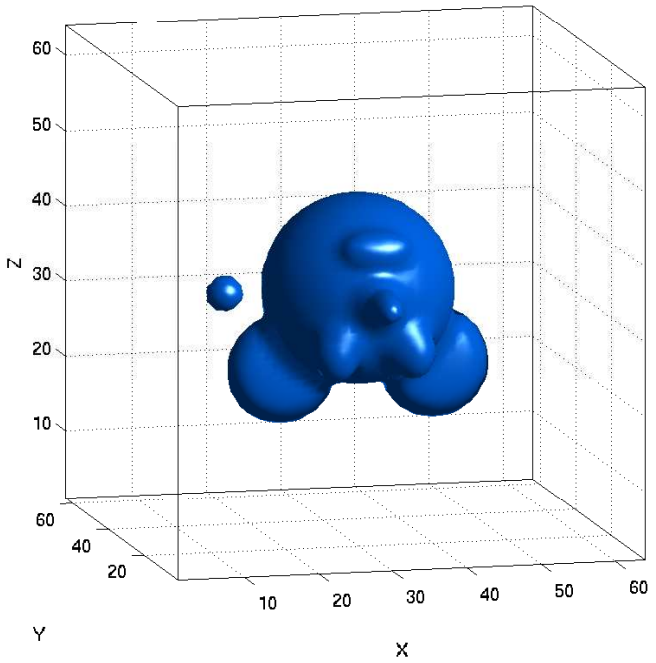


Maximal estimation error is less than 0.2 degrees.

Results (cont.)



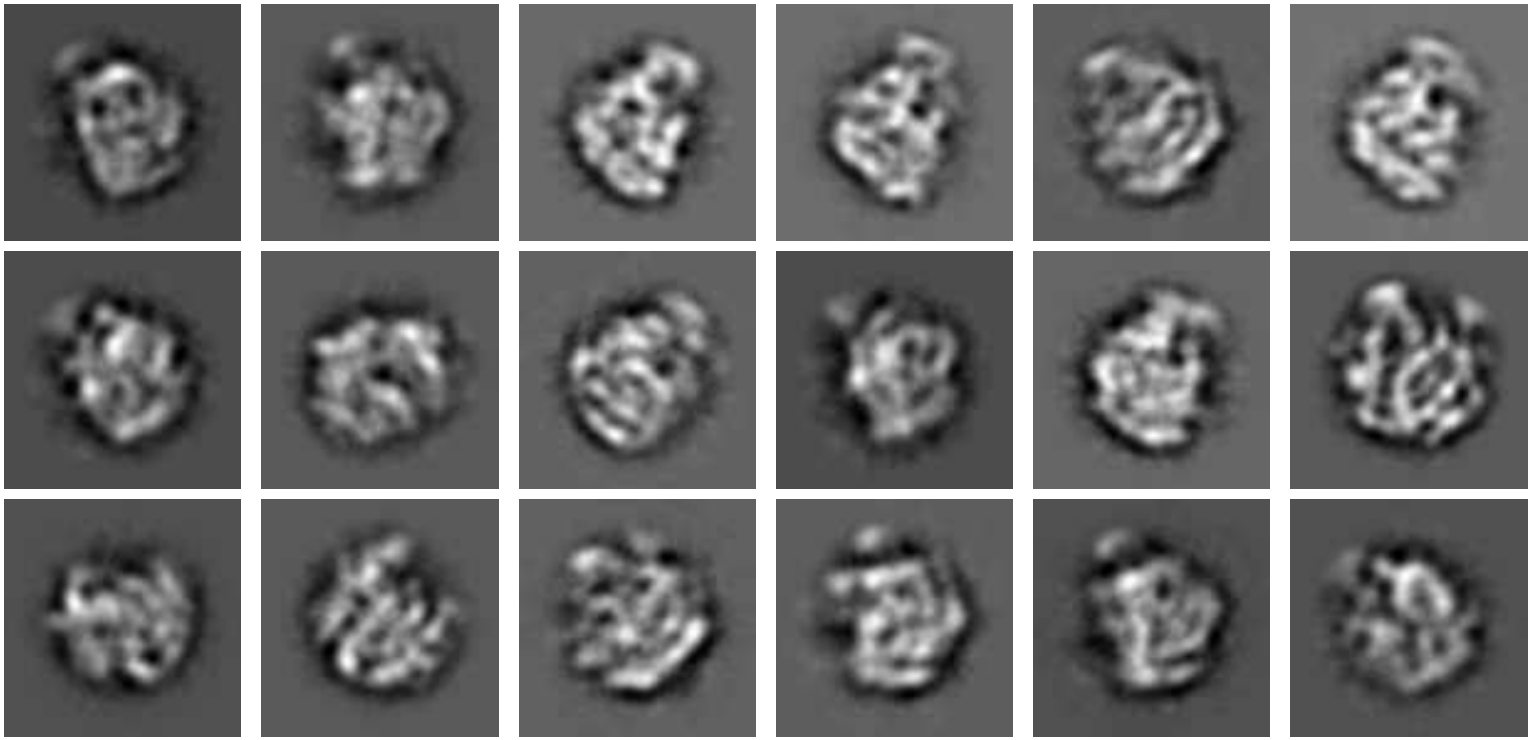
Original



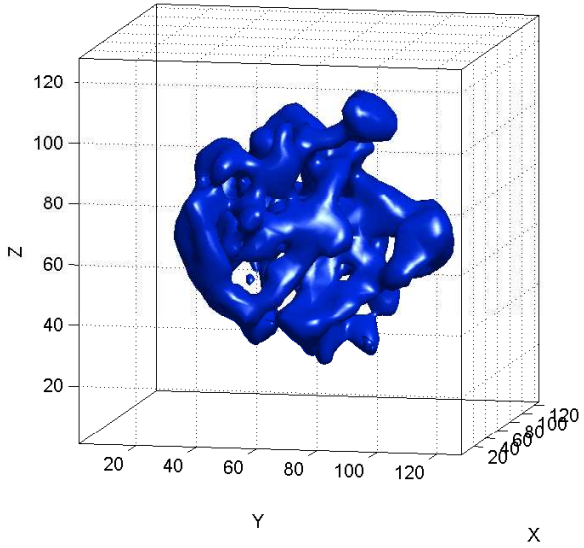
Reconstructed

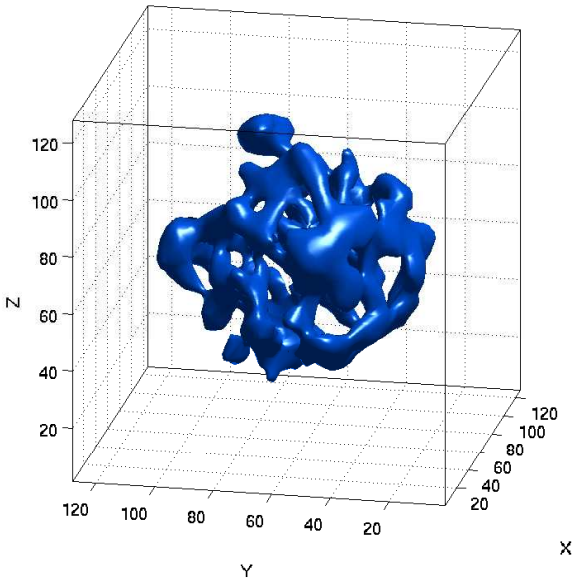
Another example

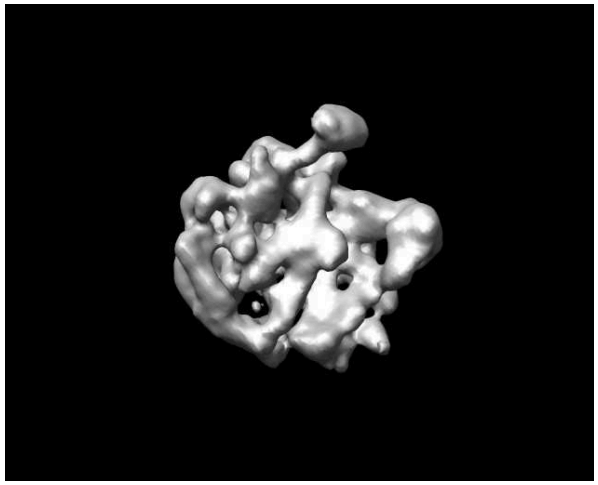
- 100 images
- Each image 96×96 pixels
- Images are not centered

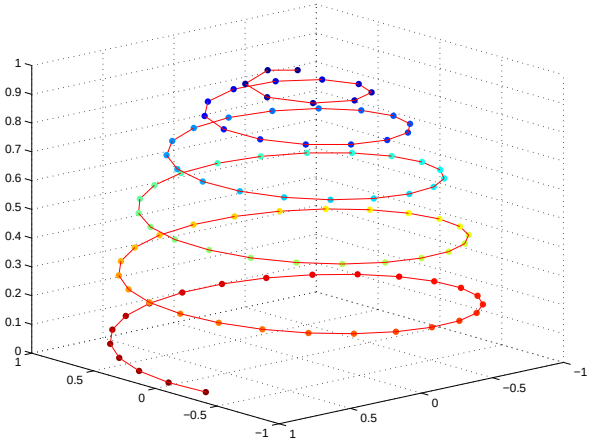


Another example (cont.)

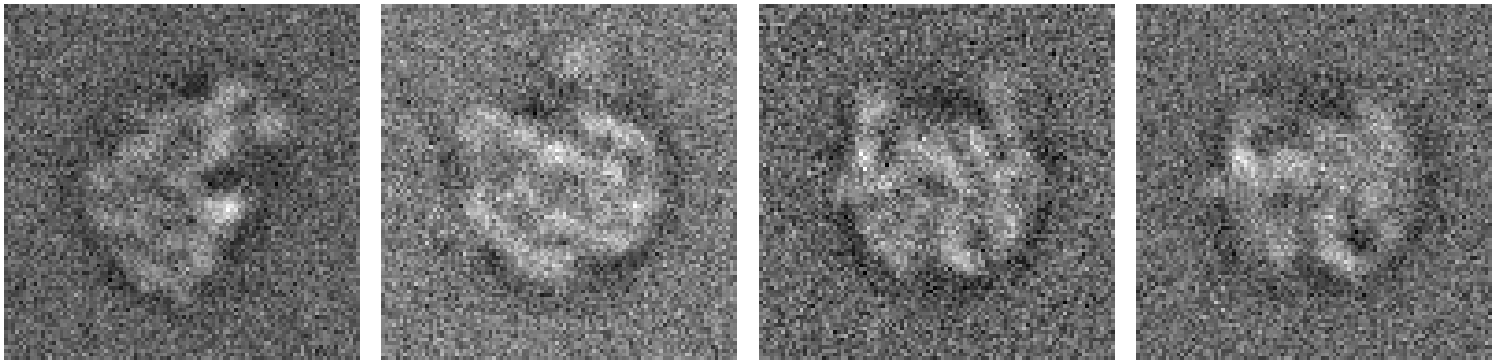


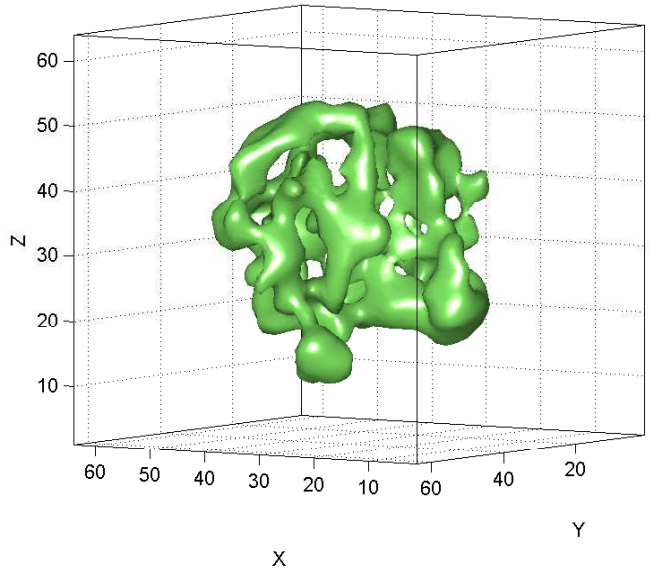






Noisy projections

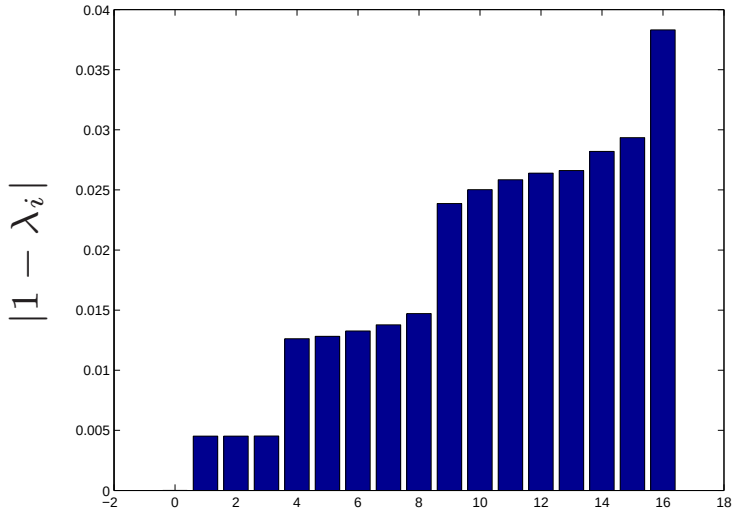


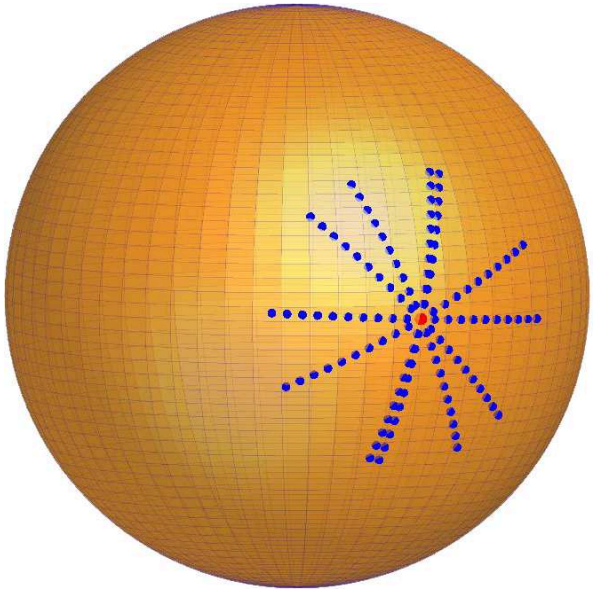


Advantages of the algorithm

- Global: all Fourier rays are linked together.
- Fast: linear in data size KL and intersection points $\binom{K}{2}$.
- Averaging: all geometric information is averaged.
- Robust: errors due to false detections of common lines are smoothed out (can be viewed as matrix perturbation).
- Independent of the distribution of the orientations.
- Insensitive to in-plane rotations of the projection images.

Further properties of the eigenvectors – spherical harmonics





- The spherical harmonics Y_l^m are the eigenfunctions of the Laplacian on the sphere

$$\Delta_{S^2} Y_l^m = -l(l+1)Y_l^m, \quad l = 0, 1, 2, \ldots, \quad m = -l, \ldots, l.$$

Further properties of the eigenvectors – spherical harmonics (cont.)

- Funk-Hecke: The spherical harmonics are the eigenfunctions of any integral operator that commutes with rotations

$$\begin{aligned}(\mathcal{K}f)(\beta) &= \int_{S^2} k(\langle\beta,\beta'\rangle) f(\beta')\, dS_{\beta'},\\ \mathcal{K}Y_l^m &= \lambda_l Y_l^m.\end{aligned}$$

- The operator A ("spider kernel") commutes with rotations only in the limit, so spherical harmonics are not guaranteed.

- The three linear spherical harmonics are exact eigenfunctions of A.

Center determination

- For two shifted projections

$$Q_1(x^1, y^1) = P_1(x^1 + \Delta x^1, y^1 + \Delta y^1),$$
$$Q_2(x^2, y^2) = P_2(x^2 + \Delta x^2, y^2 + \Delta y^2)$$

  we have

$$\hat{Q}_j(\omega_x^j, \omega_y^j) = \hat{P}_j(\omega_x^j, \omega_y^j) e^{\imath(\Delta x^j \omega_x^j + \Delta y^j \omega_y^j)}.$$

- Along the common line

$$\hat{P}_1(r\cos\theta^1, r\sin\theta^1) = \hat{P}_2(r\cos\theta^2, r\sin\theta^2),$$

  or,

$$\hat{Q}_1(r\cos\theta^1, r\sin\theta^1) e^{-\imath r(\Delta x^1 \cos\theta^1 + \Delta y^1 \sin\theta^1)}$$
$$= \hat{Q}_2(r\cos\theta^2, r\sin\theta^2) e^{-\imath r(\Delta x^2 \cos\theta^2 + \Delta y^2 \sin\theta^2)}.$$

Center determination (cont.)

- Equation for the unknowns $(\Delta x^1, \Delta y^1)$ and $(\Delta x^2, \Delta y^2)$

$$\mu_{1,2} = \Delta x^1 \cos\theta^1 + \Delta y^1 \sin\theta^1 - \Delta x^2 \cos\theta^2 - \Delta y^2 \sin\theta^2.$$

  with

$$\frac{1}{r} \arg \frac{\hat{Q}_1(r\cos\theta^1, r\sin\theta^1)}{\hat{Q}_2(r\cos\theta^2, r\sin\theta^2)} = \mu_{1,2}$$

- One equation for each common line

- System is very sparse (4 unknowns per equation)

- Solve using least-squares

Thank You!