Motion Adapted Reconstruction

Stuart Rowland

Sherif Gadoue

Keith Schubert

Reinhard Schulte

Yair Censor

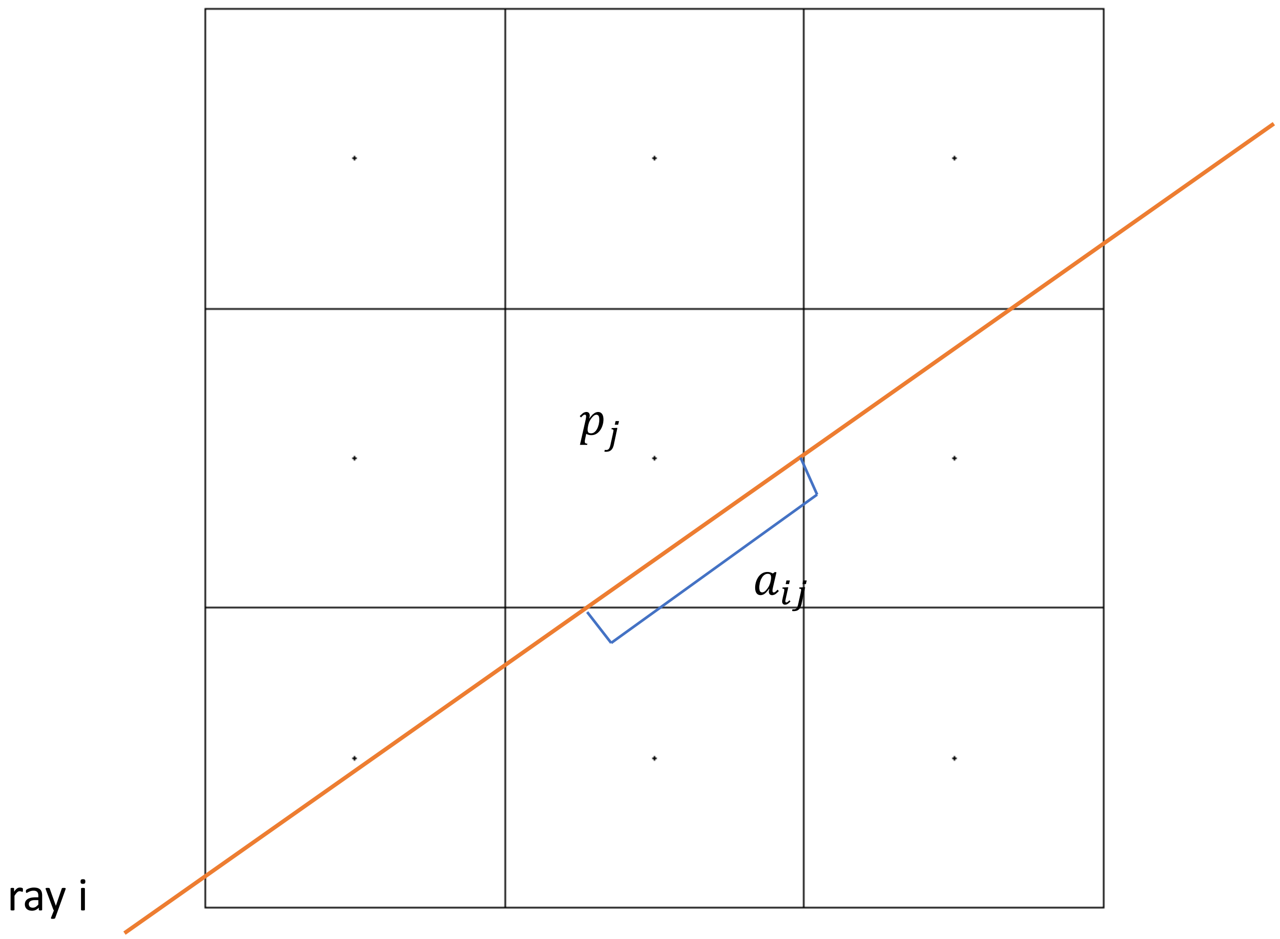
Problem statement

Can we reconstruct a reference image from projections of distorted versions of that image if we know how the voxels of the reference are related to those of the distorted images and know which rays come from which distorted image.

The reconstruction problem

- We want to solve (approximately) the system of equations given by $Ax = b$, where
- A is a sparse $I \times J$ matrix with elements a_{ij},
- x is a J dimensional vector representing the voxelated output image,
- b is a I dimensional vector representing the projection measurements along known rays,
- a_{ij} is contribution of voxel x_j to projection measurement b_i, and
- p_j is the position in space of the center of voxel j.

Calculation of the a_{ij}



Motion model

We assume that there is a set of physiological parameters that is sufficient to create a motion model.

We assume there is a voxelated reference image and a motion model which when given an instance of the physiological parameters will give as an output the location of every voxel center under those conditions.

Let p_j be the coordinates of the center of voxel j in the reference image.

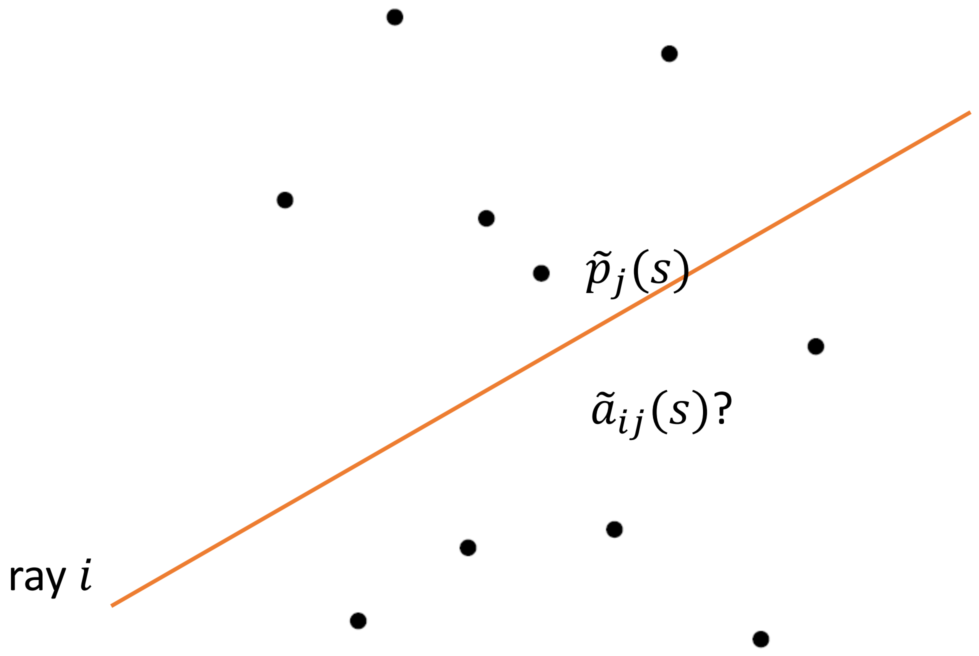
Let s be values of the physiological parameters at some instant.

Let M be the motion model.

Let $\tilde{p}_j(s)$ be the position of p_j in state s.

Then $\tilde{p}_j(s) = M(p_j, s)$.

Effect of motion model



We assume that the motion model distorts the shape of the voxels such that they fill space and there are no gaps and no overlaps.

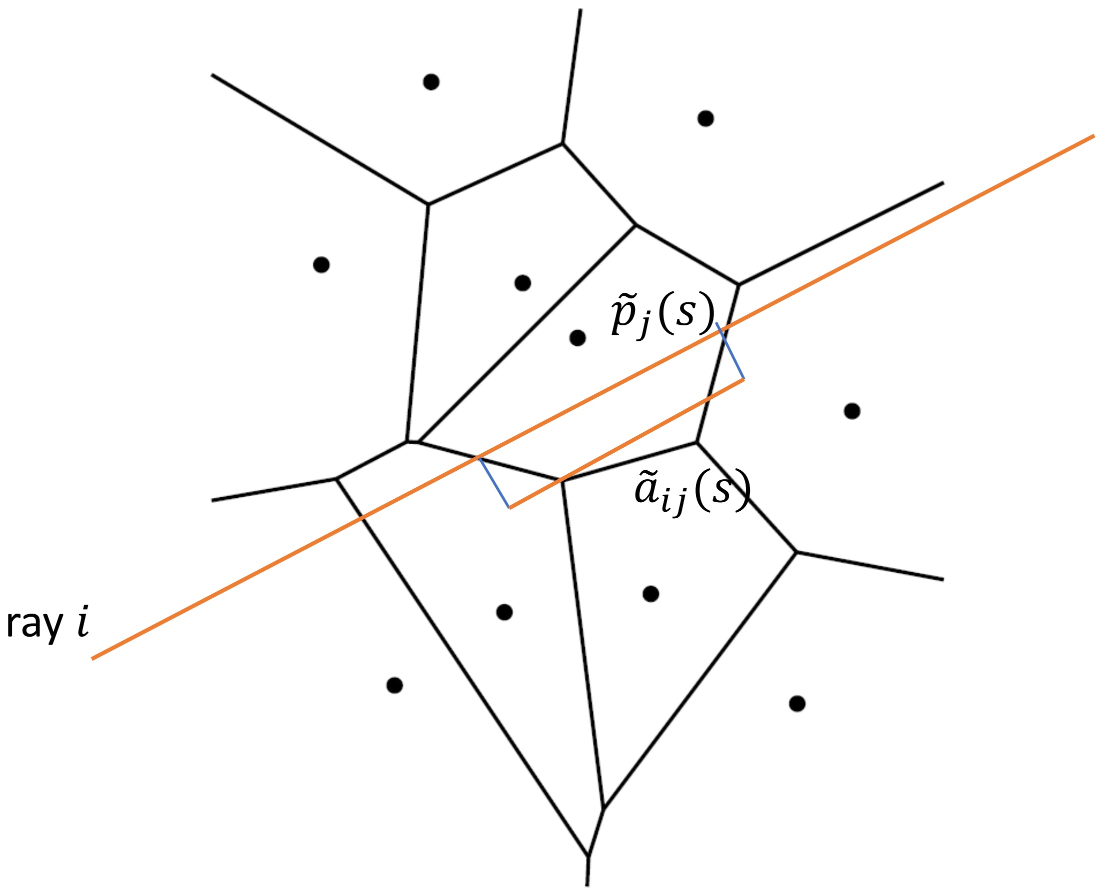
Voronoi diagram

From Wikipedia:

We are given a finite set of points $\{p_1, ..., p_n\}$ *in the Euclidean plane. In this case each site* p_k *is simply a point, and its corresponding Voronoi cell* R_k *consists of every point in the Euclidean plane whose distance to* p_k *is less than or equal to its distance to any other* p_k*. Each such cell is obtained from the intersection of half-spaces, and hence it is a (convex) polyhedron. The line segments of the Voronoi diagram are all the points in the plane that are equidistant to the two nearest sites. The Voronoi vertices (nodes) are the points equidistant to three (or more) sites.*

- This definition easily extends to an n-dimensional Euclidean space.
- In 3D, each Voronoi cell is a convex polyhedron.
- Cubic voxels are the Voronoi cells formed by the simple cubic lattice.

Calculation of $\tilde{a}_{ij}(s)$



Delaunay triangulation

From Wikipedia:

*In mathematics and computational geometry, a **Delaunay triangulation** (also known as a **Delone triangulation**) for a given set **P** of discrete points in a general position is a triangulation DT(**P**) such that no point in **P** is inside the circumcircle of any triangle in DT(**P**).*

- The triangulation is unique unless 4 points lie on a circumcircle.
- The triangulation extends to 3D with triangles replaced by tetrahedrons and circles with spheres.
- In 3D the Delaunay tetrahedralization is unique unless 5 points lie on a circumsphere.
- For our purposes, uniqueness is unimportant. When multiple solutions exist, any one of them is acceptable.

Voronoi diagram and Delaunay triangulation are duals

- Two problems are duals when the solution of one provides the solution to the other.
- Given the Voronoi diagram of a set of points, the Delaunay triangulation is created by connecting each center point to the centers of the cells that border the given cell.
- Given the Delaunay triangulation of a set of points, the Voronoi diagram is created by the perpendicular bisectors of the line segments in the Delaunay triangulation.

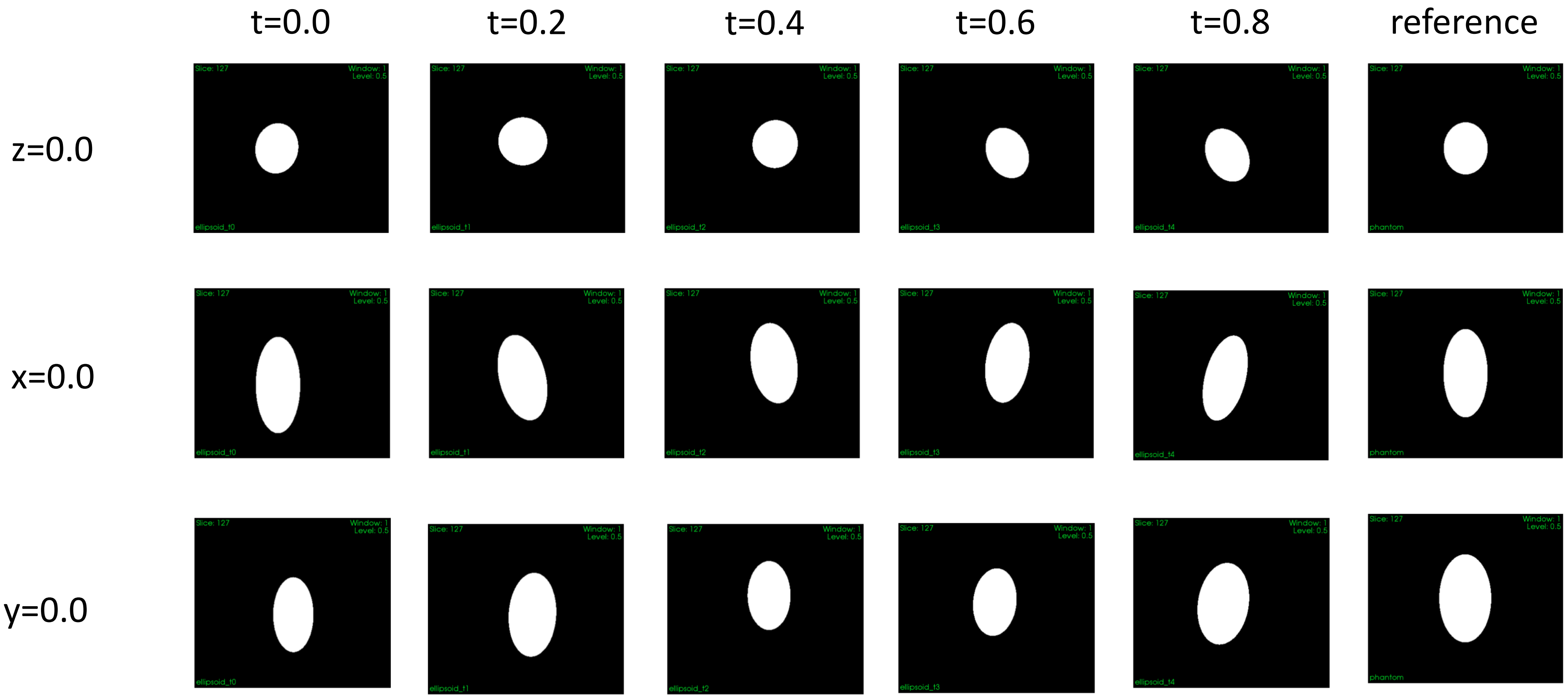
First test

- We have made a number of assumptions and did not know if the process will work.
- Pick a simple phantom (an ellipsoid).
- Use a small number (5) of affine transformations of the phantom for the motion states.
- Generate noiseless projection data for the reconstruction.
- If the process does not work under these conditions, it can not possibly work with real data.

Phantom specification - a single time varying ellipsoid

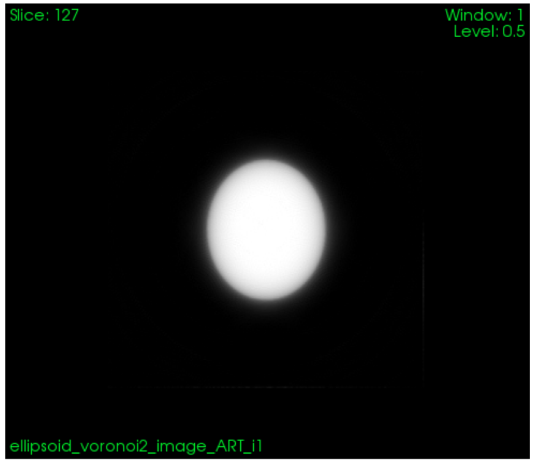
cx	$-3.6\cos(2\pi t)$
cy	$-1.8\sin(2\pi t)$
cz	$-3\cos(2\pi t)$
u	$0.5\sin(2\pi t) + 5.5$
v	$0.5\sin(2\pi t + \pi) + 6.5$
w	$\sin(2\pi t + \pi/2) + 11$
z_1	$15\sin(2\pi t + \pi/4)$
y	$15\sin(2\pi t + \pi)$
z_2	$20\sin(2\pi t)$

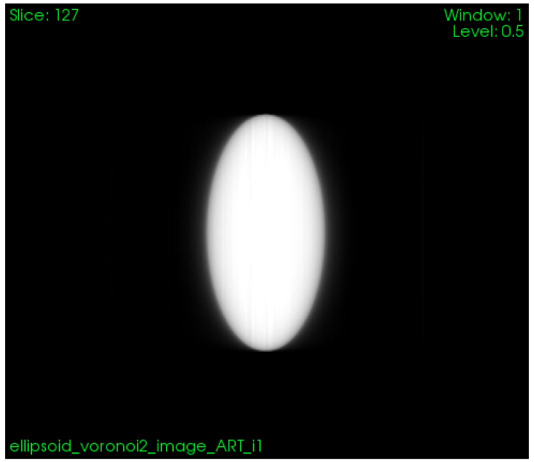
Orthogonal slices of the moving ellipsoid

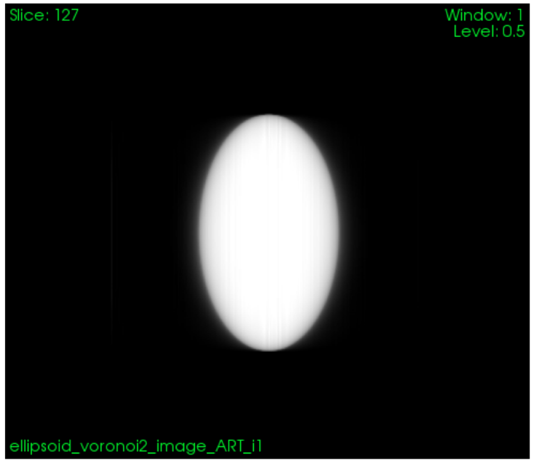


Reconstruction of moving ellipsoid

window=1.0
level=0.5





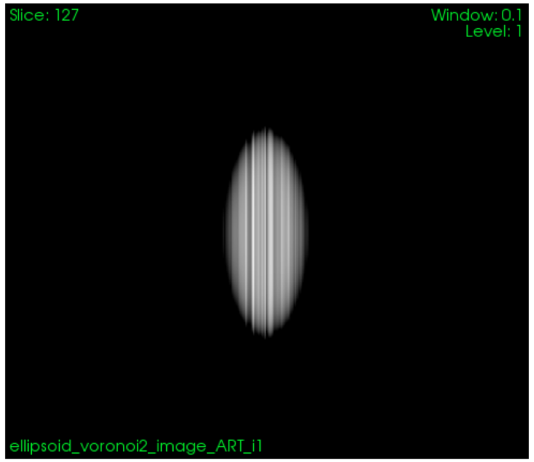


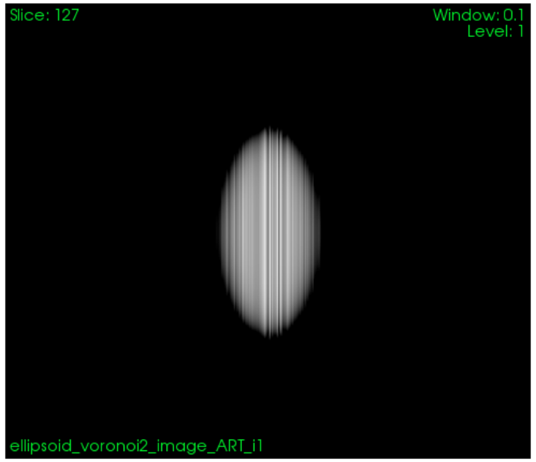
z=0.0

x=0.0

y=0.0

window=0.1
level=1.0



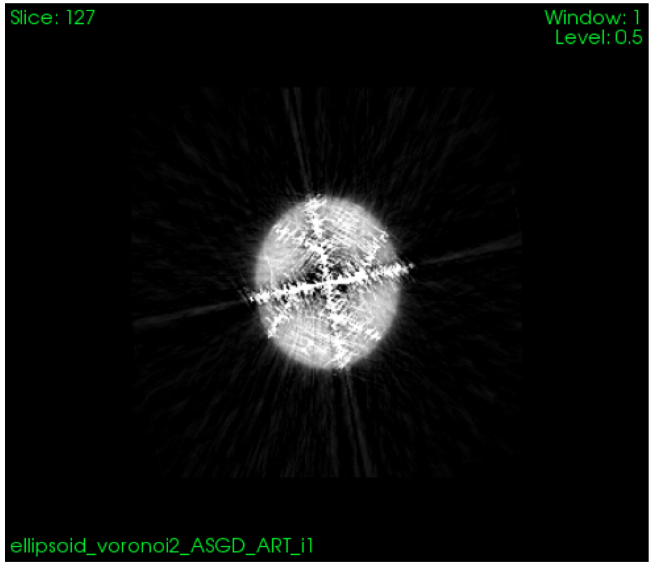


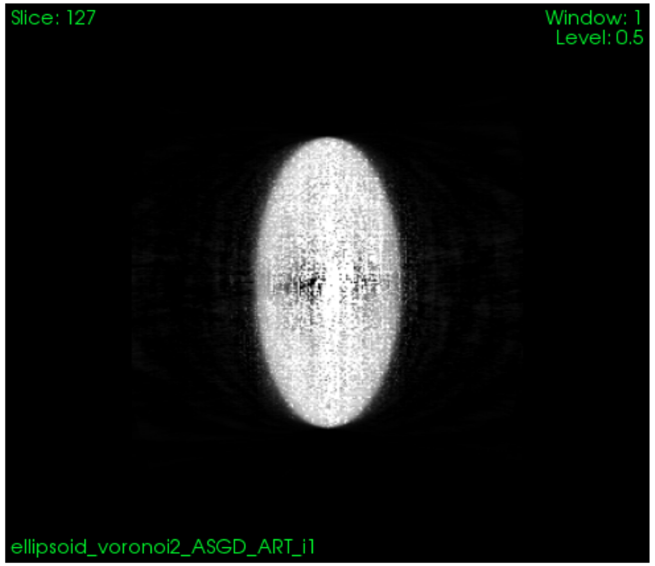
Where did artifacts come from?

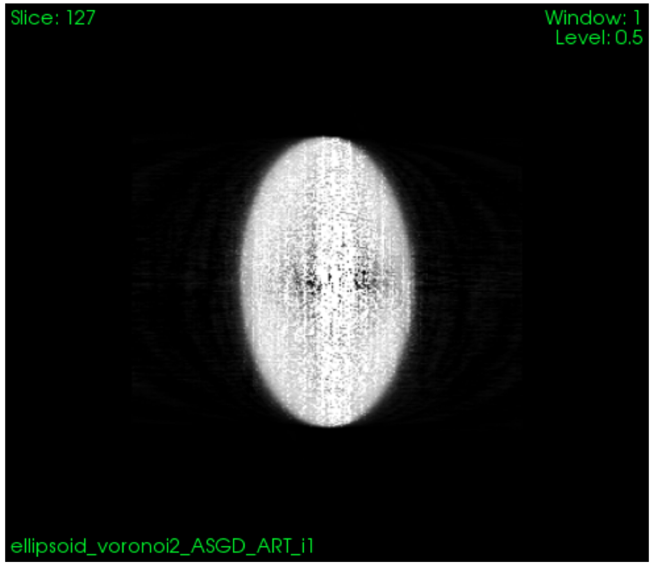
- Simplify problem.
- Deformation is an affine transformation.
- Separate transformation into the three types of affine transformation:
  - translation
  - scaling
  - rotation
- Use only 1 motion state. That is all the projections come from a single instance of the transformed ellipsoid.

Rotation

Reconstruction of rotated ellipsoid.







Rotation caused the largest artifacts.

Aspect Ratio

- Keith suggested examining the aspect ratio of the Voronoi cells.
- $aspect\ ratio = \frac{distance\ to\ furthest\ neighbor}{distance\ to\ closest\ neighbor}$

Cubic grid	$\sqrt{3}$
Translated grid	$\sqrt{3}$
Scaled grid	$\frac{max}{min}\sqrt{3}$
Rotated grid	8.9
Restored grid	8.9

Maximum difference between cubic grid points and restored grid points is about 10^{-15}.

Delaunay triangulation algorithm in CGAL is numerically unstable

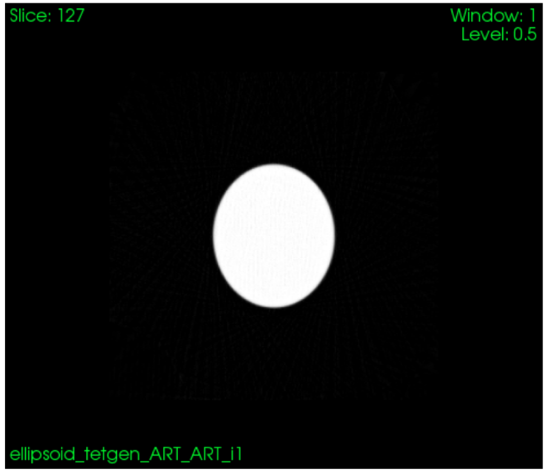
- Had been using the Delaunay triangulation algorithm in the Computational Geometry Algorithms Library – http://www.cgal.org.
- Tested MATLAB implementation of Delaunay triangulation. It is also numerically unstable.
- Searched internet. Found Maur and Kolingerova, COPING WITH NUMERICAL PROBLEMS IN IMPLEMENTATION OF 3D DELAUNAY TRIANGULATION, Proceedings of ALGORITMY 2002, Conference on Scientific Computing, pp. 153-161.
- Dr. Kolingerova pointed us to Dr. Shewchuk of UC Berkeley whose faculty web page points to http://www.tetgen.org, a numerically stable implementation of Delaunay triangulation.

Projection data generation of moving ellipsoid

- tangent cone beam
- source to origin=57, source to detector=104
- number of rays=513
- detector spacing=0.117187
- number of detector rows=32
- detector element height=0.117187
- number of pulses=10800
- helix axis=Z
- helix start=-11 helix end=11
- number of turns=30
- start time=0.0
- pulse time=0.2

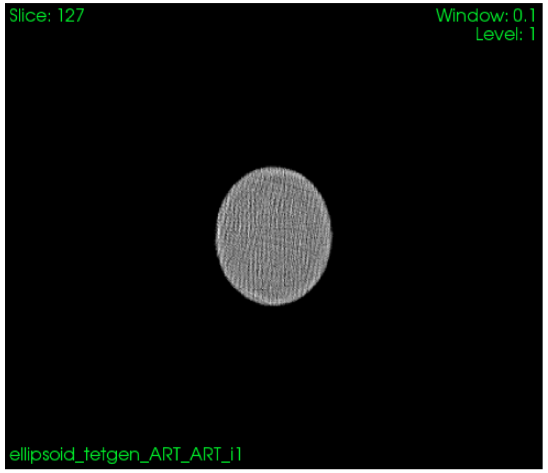
Reconstruction of moving ellipsoid using tetgen

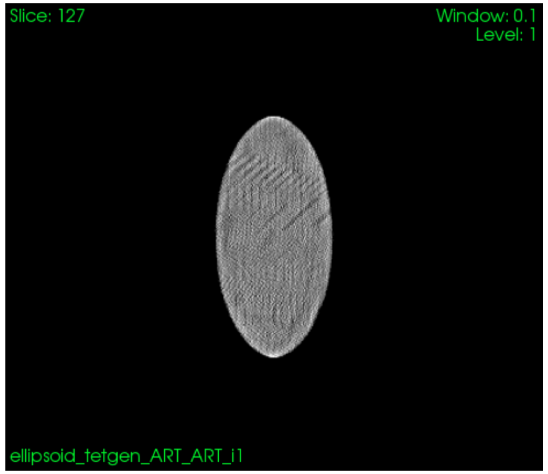
window=1.0
level=0.5



z=0.0 x=0.0 y=0.0

window=0.1
level=1.0





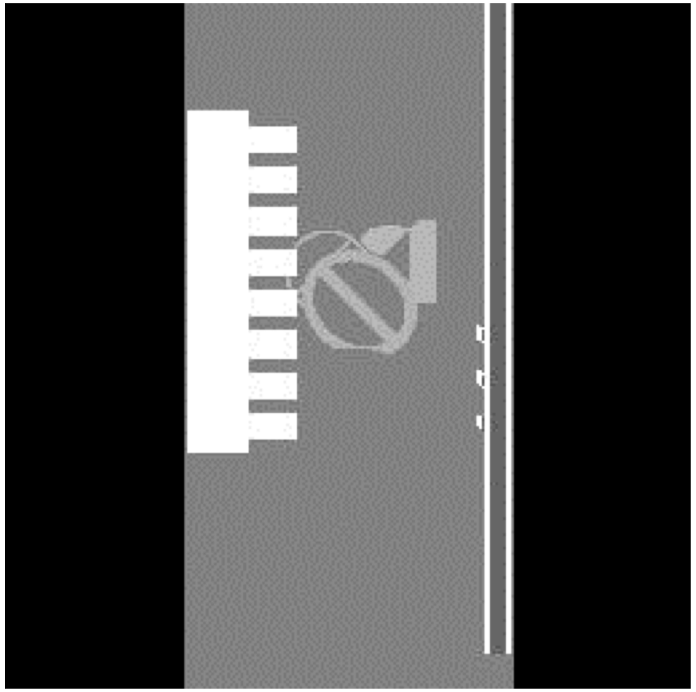
Preliminary Conclusions

- Using the tetgen program reduces the magnitude of the noise in the narrow window images.
- In addition, the noise is no longer structured, but salt and pepper.
- This method works for any row action reconstruction algorithm. The only change was how a_{ij} is calculated.
- Any motion model that meets our conditions can be used.

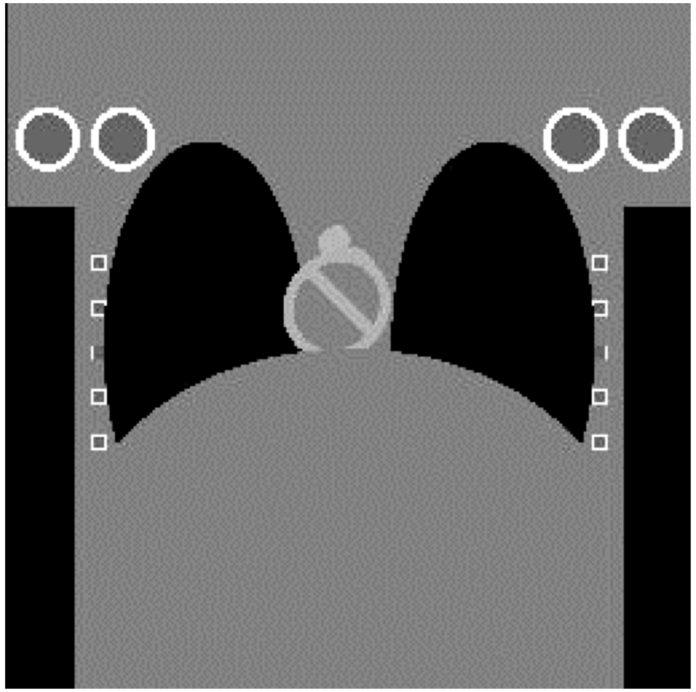
Moving FORBILD Thorax

- The original FORBILD thorax phantom was of a single “patient”.
- Graduate student Lawrence Orijuela wrote a program to modify the phantom to create many variations of the phantom, i.e., multiple patients.
- He also added a 4-chamber heart and pulmonary veins and arteries.
- I extended his modifications to add motion to the diaphragm and heart.
- The following are videos of orthogonal slices of the moving thorax.
- All videos are 1 minute long with 10 images per second.
- There are 13 breath cycles and 79 heart cycles.

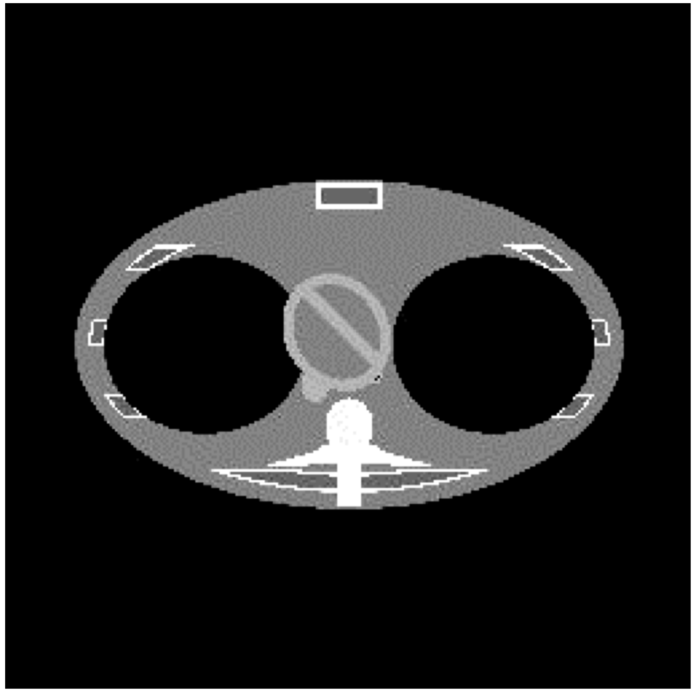
Thorax



X=0.0



Y=0.9



Z=2.7

Future work

- Pick a small number (5 to 10) of instances of the moving thorax phantom.
- Create a reference image that is different from all of the above instances.
- Create the deformation fields that transform the instances to the reference image.
- Create tetrahedrons for the deformed voxels.
- Create projection data from the various instances.
- Using the projection data and the tetrahedrons, try to reconstruct the reference image.