EM Reconstruction Roadmap

**Milestone 1: Implement low-level voxel classification based on deep convolutional neural networks**

1.1. Implement 2D-slice data provider for single GPU convnet

1.2. Implement 2 class version of single GPU convnet (‘in’ vs. ‘out’ classification) using k-fold cross validation

1.3. Implement formatting of network output

1.4. Measure training data through-put rate for single GPU convnet (training samples per unit time/k-folds)

1.5. Implement segmentation-based error metrics (rand error, warping error, splits vs. mergers, etc..) to measure performance

1.6. Understand effective field of view (context defined by kernel size, etc..) for given network architecture

1.7. Understand benefit of various optimizations used by deep convnet code (dropout, pooling, logistic function, etc…)

1.8. Measure error rates on single vs. dual GPU card network

1.9. Explore 3D convolution in the future

**Milestone 2: Implement data storage standard**

2.1 Define raw data format (HDF5?)

2.2 Define cube-based hierarchy (with adjacency overlap?)

2.3. Define storage for final segmentation results

2.4 Define storage/queue for client-returned annotations

**Milestone 3: Implement segmentation of large-scale (terabyte-sized) datasets**

3.1. Implement voxel-level classification using multi-level thresholding of NN model, measure data through-put rate

3.2. Implement efficient supervoxel segmentation algorithm of voxel-level classification (connected components, graph cut, etc…)

**Milestone 4: Implement meshing of segmented objects**

4.1. Determine mesh format required by front-end client (e.g., WebGL)

4.2. Implement mesh rendering of segmentation results

**Milestone 5: Implement storage of client-generated annotations**

5.1. Implement storage of contour-based annotations

5.2. Implement storage of supervoxel-based agglomeration annotations

**Milestone 6: Implement multi-user consolidation of client-generated annotations**

6.1. Implement consolidation of contour-based annotations

6.2. Implement consolidation of supervoxel-based agglomeration annotations

**Milestone 7: Implement super-voxel agglomeration learning algorithm**

7.1 Literature review of current methods

**Milestone 8: Implement queuing system to supply data/segmentations to client**

8.1. Flexible queuing that can be prioritized for a specific region of a dataset or for a particular neuron

**Milestone 9: Implement scoring system for client**

9.1. Explore scoring possibilities (consensus based, confirmation of correct segmentations, etc..)