Automatic detection and segmentation of membrane-stained subcellular components

Richard Giuly, Steve Lamont, Maryann E. Martone, Mark H. Ellisman
Center for Research in Biological Systems, University of California, San Diego, 92093.

Segmentation and Annotation Framework

This work addresses automatic segmentation and annotation of subcellular components, specifically mitochondria and vesicles in membrane-stained neuropil. Automatic segmentation of these components is challenging because the images are densely packed with various components, and algorithms tuned to segment one particular type of component often give incorrect results for images with other types of components present.

The segmentation process uses three stages, (1) image enhancement, (2) object detection, and (3) object segmentation. In stage 1, image enhancement is performed on the original volume to detect pixels that belong to membranes. For mitochondria membrane enhancement, a voxel classifier based on supervised learning is used. For vesicle membrane enhancement a standard sharpening filter is used. In stage 2, contours in X-Y planes of the volume are detected, and a naive Bayes classifier is used to classify the contours as mitochondrion, vesicle, or other, based on geometric properties. In stage 3, scaled contours of mitochondria are used to initialize a fast march operation which produces 3D blobs. These blobs tend to fill the mitochondria. The final output is a set of 3D blobs that represent types of cell components, along with 2D contours that were used to identify the components. Future work will address increasing the accuracy of the fast march filling operation. Objects are automatically labeled as belonging to the class Mitochondrion or Vesicle according to the subcellular anatomy ontology (SAO), and output can be viewed, corrected, or edited manually in Jinx, an ontologically aware manual segmentation tool. This project's code is open source and available at http://cytoseg.googlecode.com.
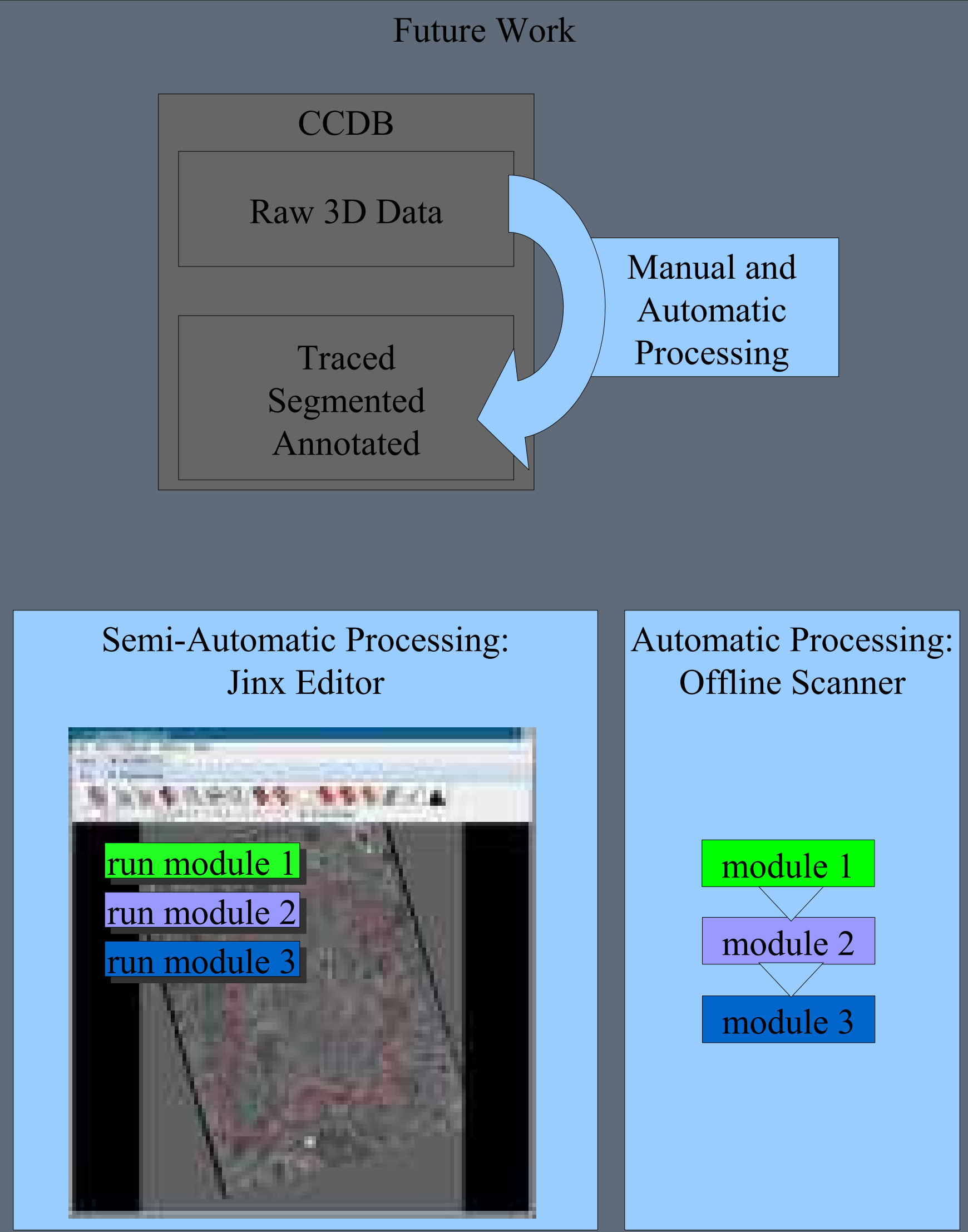
Segmentation and Annotation Framework

Semi-Automatic and Automatic Segmentation

Levels of Segmentation Automation:
● **Manual:** extremely labor intensive but necessary for generating training data and when no automatic techniques exist
● **Semi-automatic segmentation:** data can be corrected at each stage of the pipeline; high accuracy with reduced user interaction
● **Totally automatic processing:** typically more error but useful in cases where manual processing is infeasible (very large data sets) or total accuracy is not required (such as when searching a data set for a structure or constelation of structures)
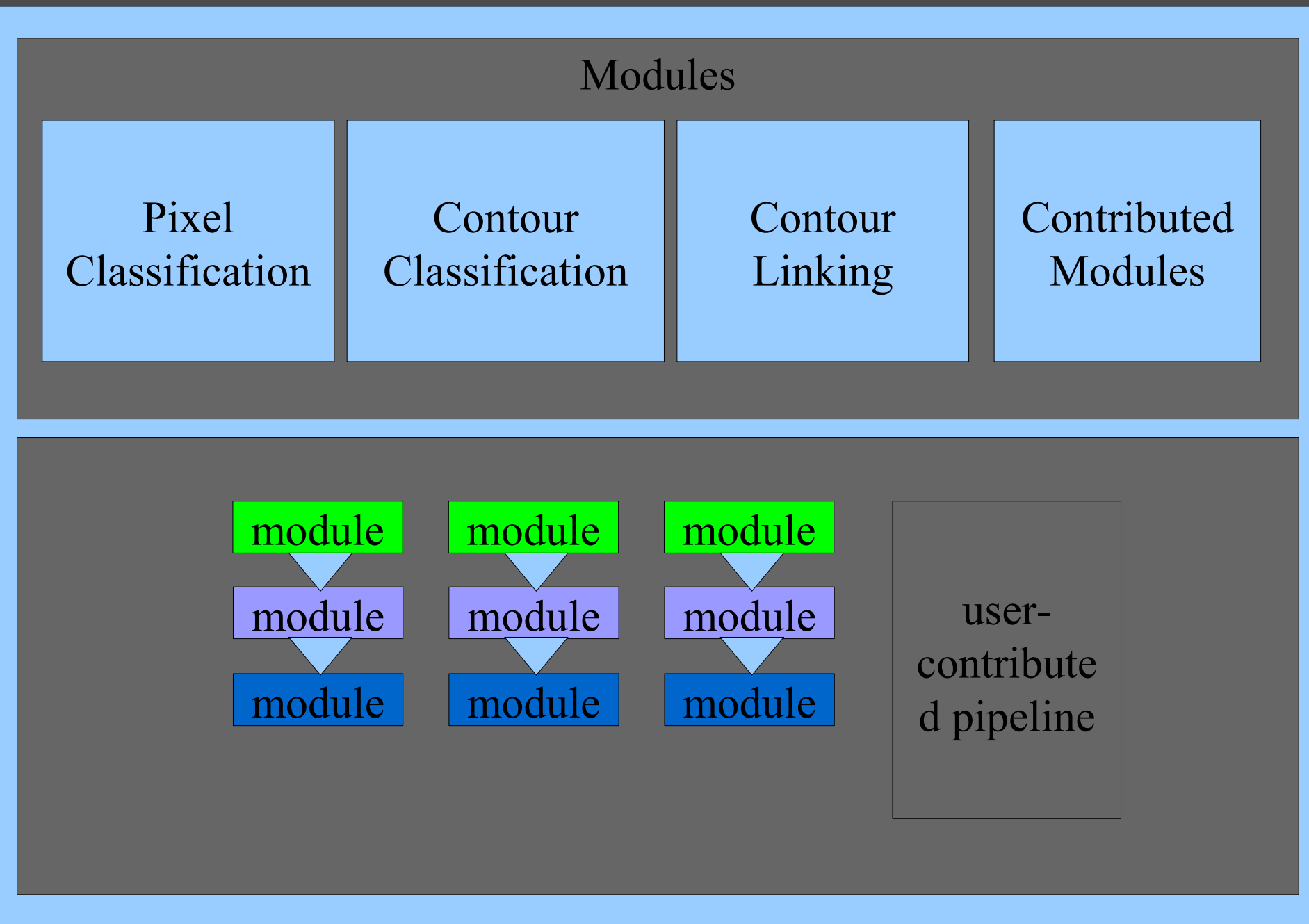
Future Uses of Segmented and or Annotated Data:
● **Visualization** of annotated data in Jinx (2D) and the Whole Brain Catalog (3D)
● **Convenient analysis of objects** that have been labeled and stored in the Cell Centered Database (CCDB)
● **Searching** of very large 3D datasets for particular objects or constellations
● **Automatically generated surface geometry** of structures to be used for modeling and simulation
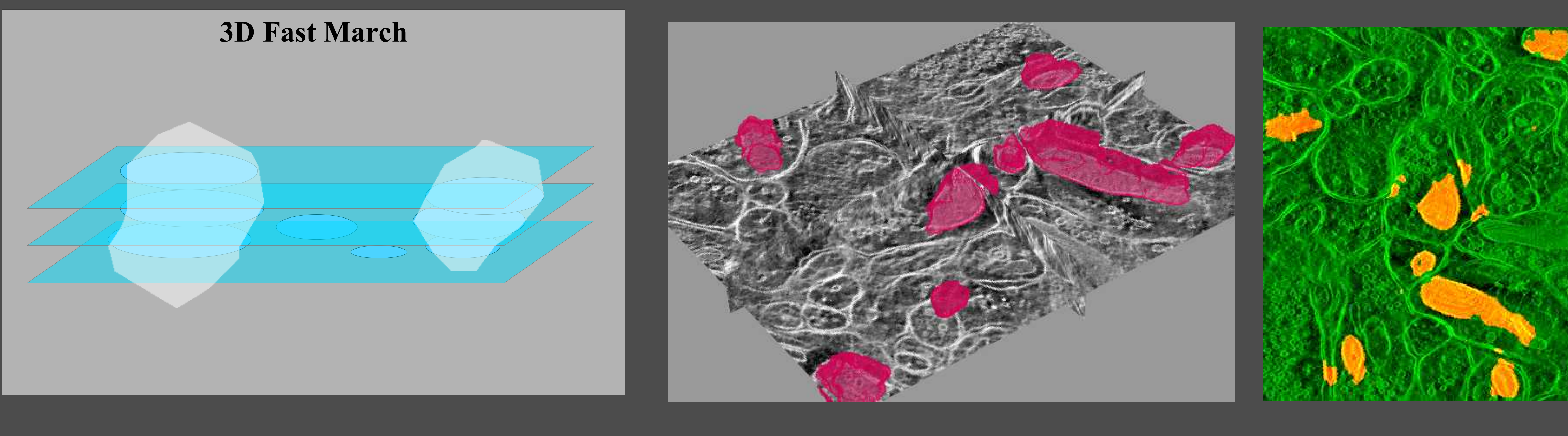
Future Work

CCDB
Raw 3D Data → Manual and Automatic Processing
Traced Segmented Annotated

Semi-Automatic Processing: Jinx Editor

run module 1
run module 2
run module 3

Automatic Processing: Offline Scanner
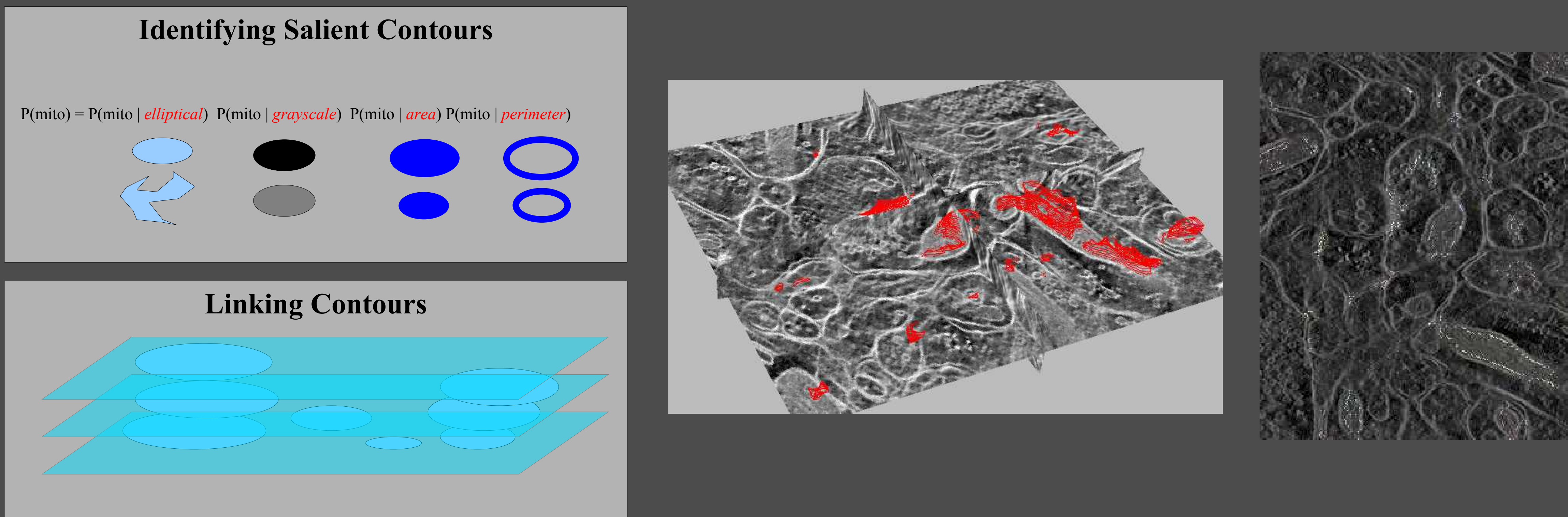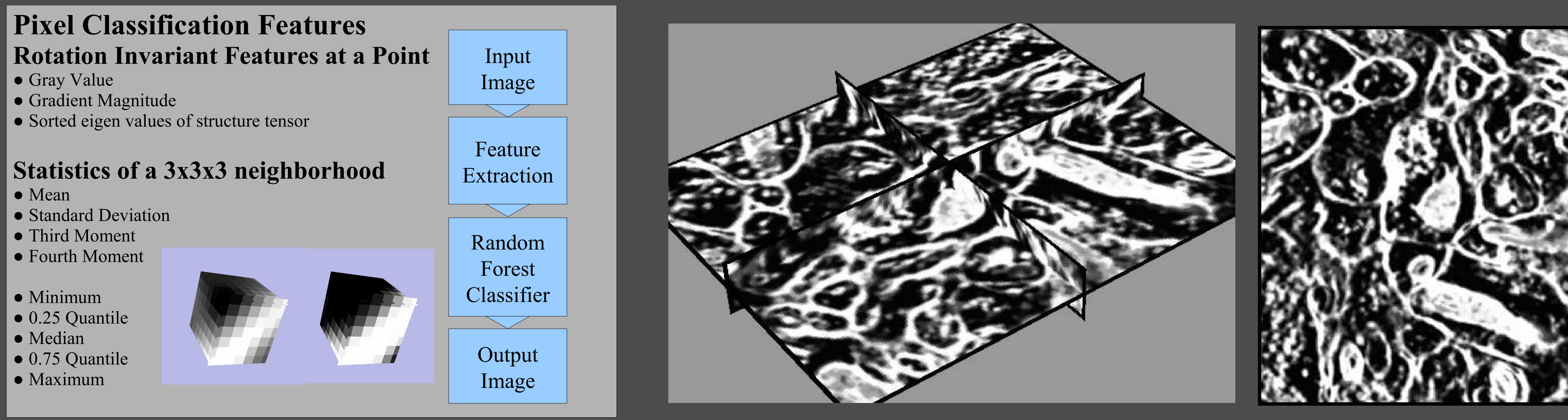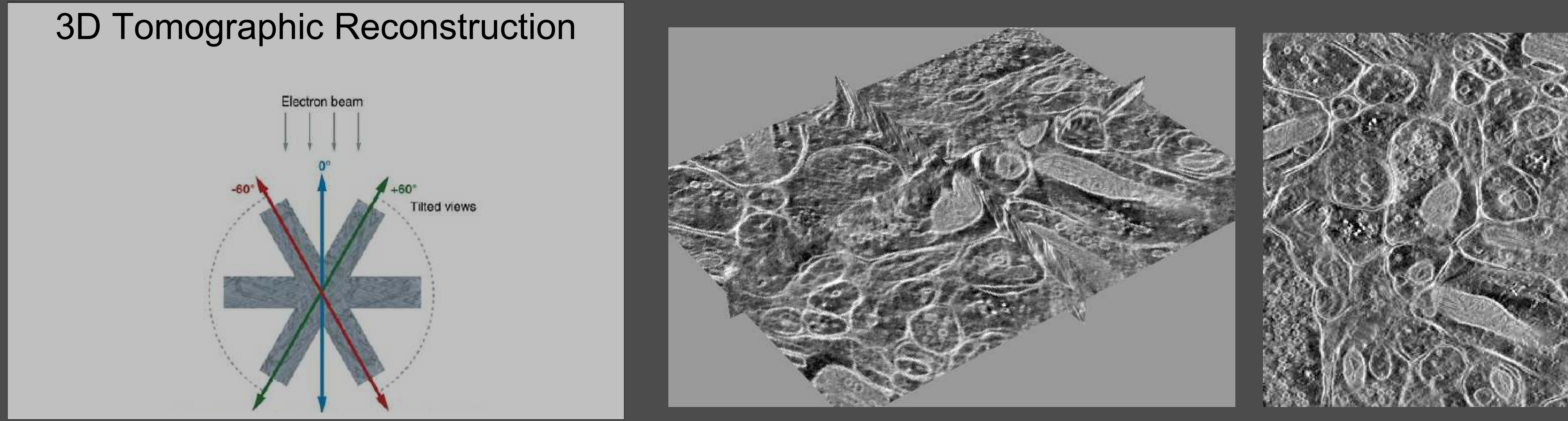
module 1
module 2
module 3

Cytoseg Prototyping Environment
Modular Open Source Architecture

The goal of the Cytoseg project is to produce a tool for automatic segmentation of 3D biological datasets, with emphasis on 3D electron microscopy. The project is written in Python and uses the pythonxy platform (which includes scipy and ITK image processing tools). Cytoseg is currently in a pre-alpha stage.

Modules

| Pixel Classification | Contour Classification | Contour Linking | Contributed Modules |

module module module
module module module   user-contributed pipeline
module module module

Pipeline Example: Mitochondria Detection and Segmentation

3D Tomographic Reconstruction

Electron beam
Tilted views

Pixel Classification Features
Rotation Invariant Features at a Point
● Gray Value
● Gradient Magnitude
● Sorted eigen values of structure tensor

Statistics of a 3x3x3 neighborhood
● Mean
● Standard Deviation
● Third Moment
● Fourth Moment

● Minimum
● 0.25 Quantile
● Median
● 0.75 Quantile
● Maximum

Input Image → Feature Extraction → Random Forest Classifier → Output Image

Identifying Salient Contours

$P(mito) = P(mito \mid elliptical) \; P(mito \mid grayscale) \; P(mito \mid area) \; P(mito \mid perimeter)$

Linking Contours

3D Fast March

Background

Cytoseg	Jain & Denk	Mishchenko	Jurrus Axon Tracking	Andres
Voxel feature extraction	3D voxel classification (neural network)	Voxel feature extraction	Denoising	Voxel feature extraction
3D voxel classification		2D voxel classification		3D voxel classification (random forest)
Post Classification Filter				Post Classification Filter
Contour		Contour	User Initialized Contours	Super Voxel
Contour feature extraction		Contour feature extraction	Kalman Filter	Super Voxel feature extraction
Fast March				

Pipelines Under Development

Vesicle Contours
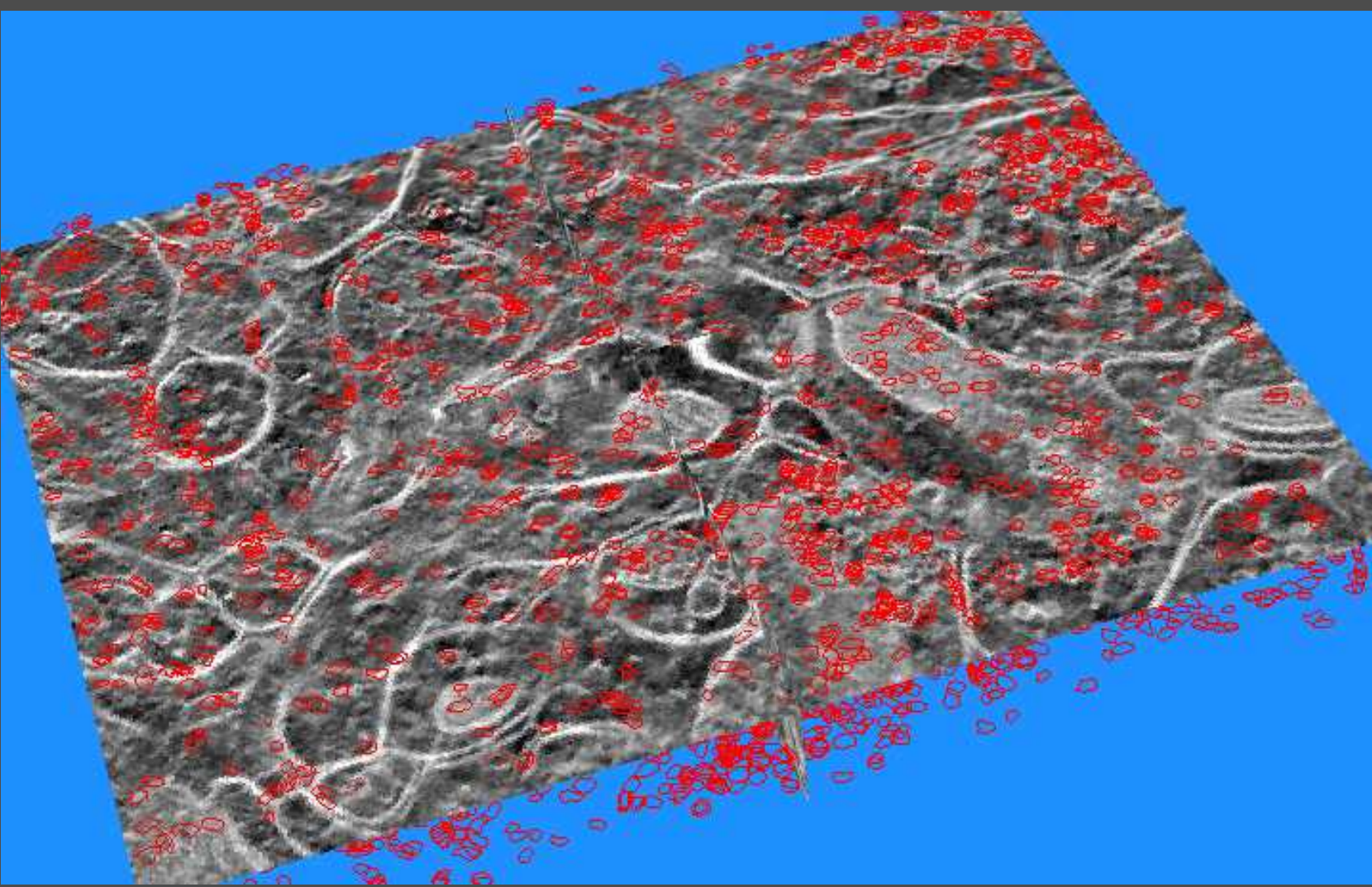
Cytosol Contours
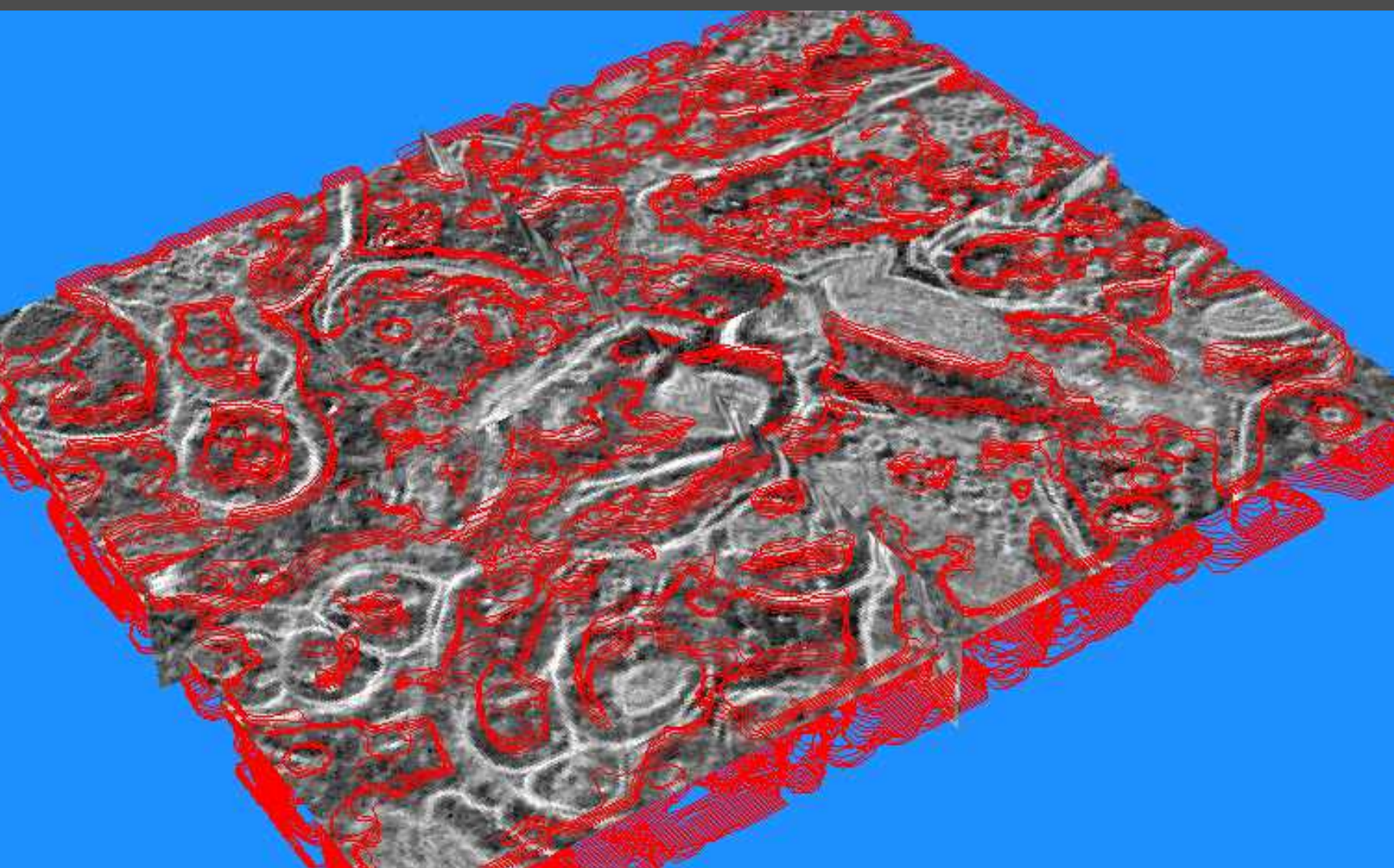
Data Integration

Future work: The scale, subject matter, type of imaging, and type of staining determine the appropriate pipeline and set of parameters for automatic segmentation. Future work involves integrating data set annotation and traced examples so the pipeline and parameters (and training data for machine learning) can be selected automatically or semiautomatically for a given data set.

High Throughput Microscopy
Voxel feature extraction — Algorithm Selection Based on Image Type
3D voxel classification — Training Data
Post Classification Filter — Algorithm Selection Based on Image Type
Contour Detection — Size Exclusion
Contour feature extraction
Contour Linking and Classification — Training Data
- Mesh Cleaning
- Display
- Simulation

Cell Centered Data Base
● Examples of traced objects

● Subject
● Stain type
● Scale

Cellular Knowledge Base
● manually generated descriptions of objects
● suggested algorithms
● preselected filters to recognize objects
● suggested pipelines