

# Serial Sectioning, Image Segmentation, and Analysis



Dave Rowenhorst

Special thanks: Alexis Lewis, Leroy Levenberry (NRL)

Marc DeGraef, Greg Rohrer (CMU), Begum Gulsoy (NU), Andy Deal (GE Global Research), The Mikes (Uchic, Greober, (AFRL), Jackson (BlueQuartz))



# Outline

Concentrate on one case study, serial sectioning of a single phase Ti-alloy that combines optical images with EBSD images.

- Slice alignment
- Image segmentation
- Data analysis and representation
- Integrating EBSD data (including coordinate transforms)

How to improve what we did.

- Better imaging to reduce image segmentation methods
  - Image distortions
  - Automated data fusion methods.



# Evolution of Grain Structures

At elevated temperatures grain boundaries evolve so as to reduce the total interfacial area of the system.

The velocity of the boundaries is proportional to the curvature of the boundary.

Here we focus on the  $\beta$ -grain (BCC) boundary structure in Ti-21S titanium alloy.

Going to need 1000's of grains, with a resolution on the order of 1 micron.

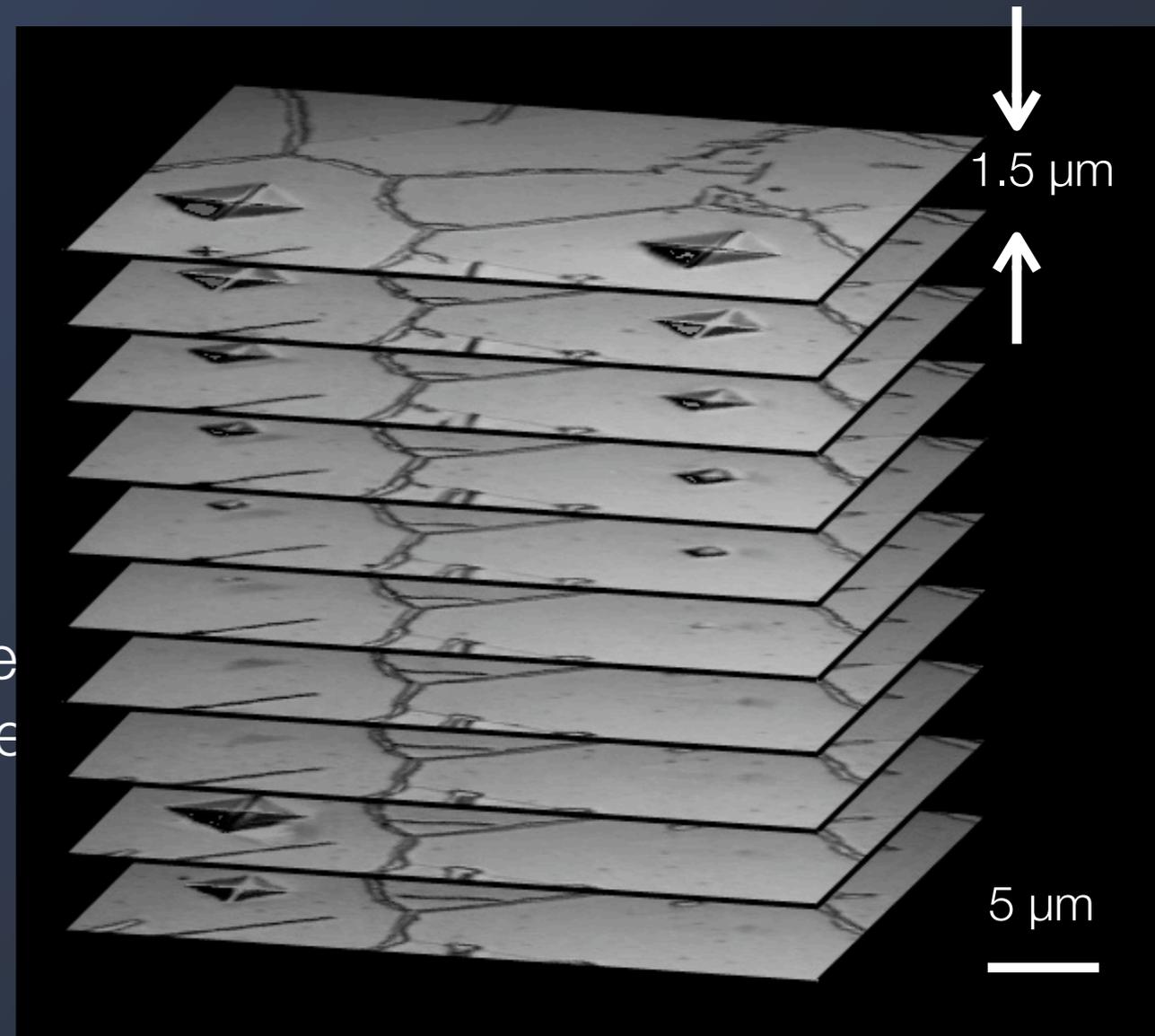


700 $\mu$ m



# 3D Reconstruction Technique

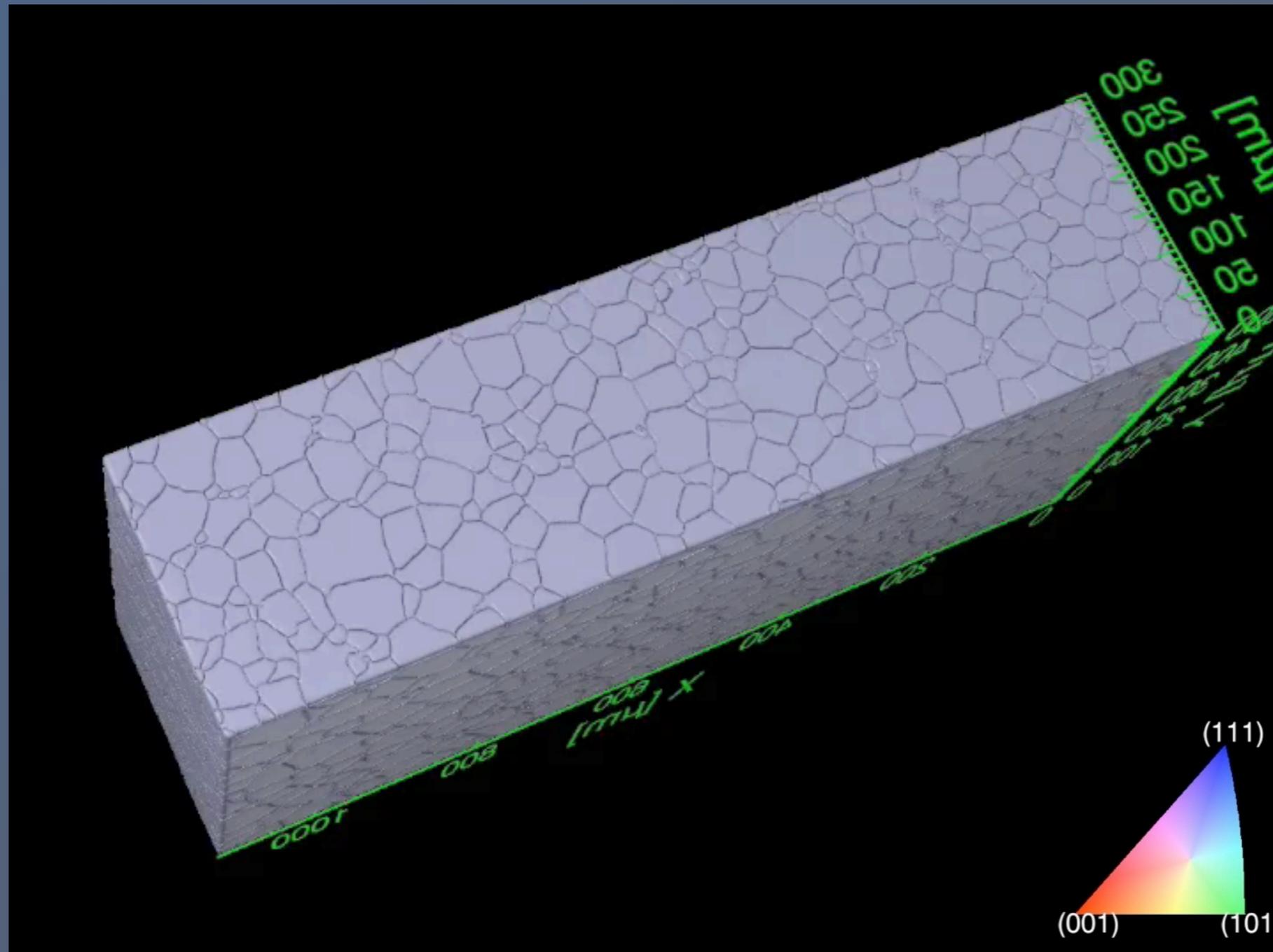
- 1) Polish away 1.5  $\mu\text{m}$
- 2) Etch for contrast
- 3) Apply fiducial marks
- 4) Acquire image digitally
  - 4a.) Collect EBSD data every 10th section
- 5) Repeat (e.g. >100 times).....
- 6) Transfer images to stack
- 7) Align all sections (using fiducial marks)
- 8) Image processing to segment the structure
- 9) Combine EBSD data with the optical image stack.
- 10) Reconstruct 3D volume and analyze on computer (IDL, vtk)



## Suggestions/advice:

- 1.) 10 Section per feature of interest.
- 2.) Image with whatever gives you contrast on the features you are interested in.
- 3.) Do as much alignment during collection, rather than after.
- 4.) Get a decent workstation computer (lots of CPU, RAM)

# Reconstruction of $\beta$ Grains in Ti-21S



4300 Grains in the box

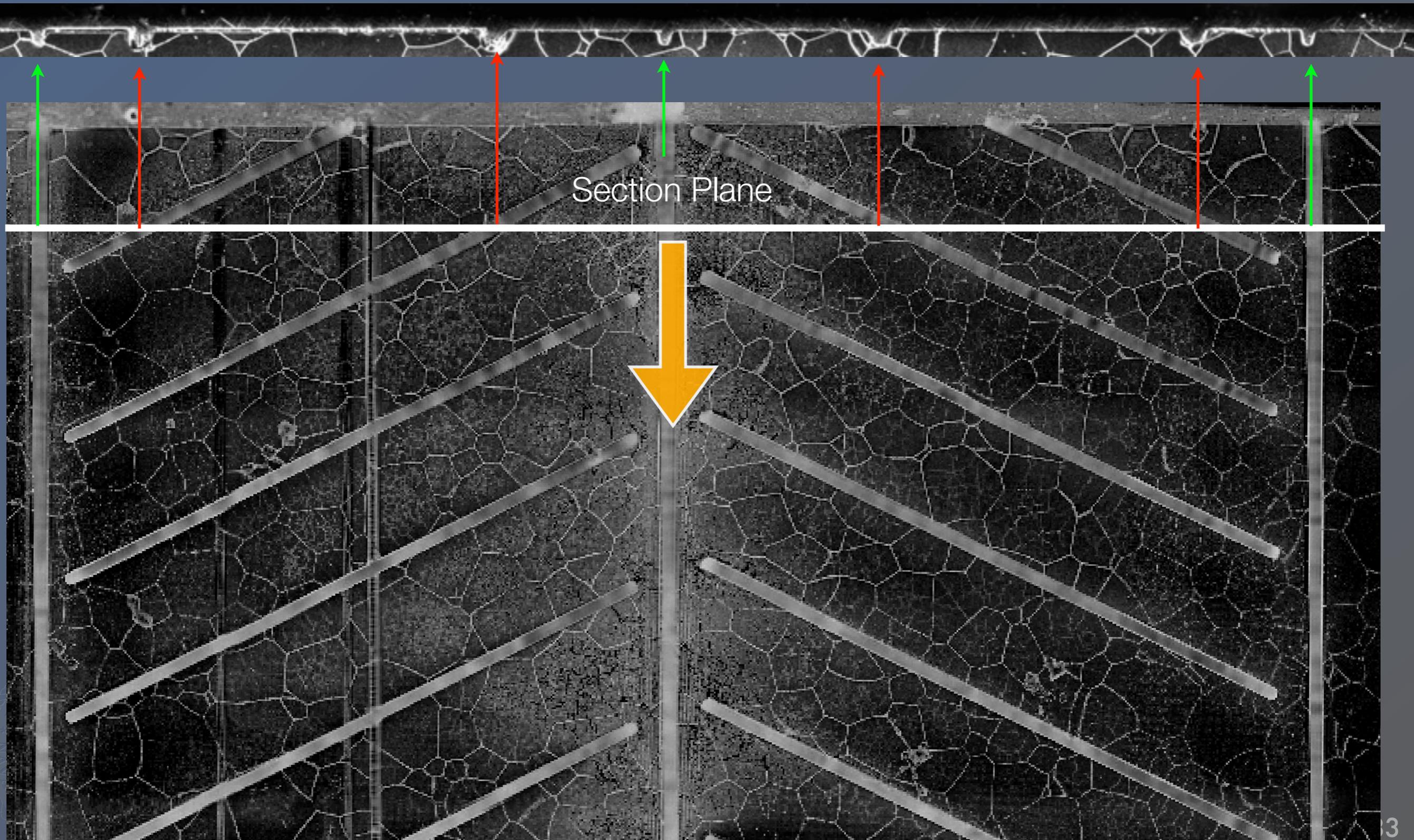
200 Sections

1.5 $\mu$ m section spacing

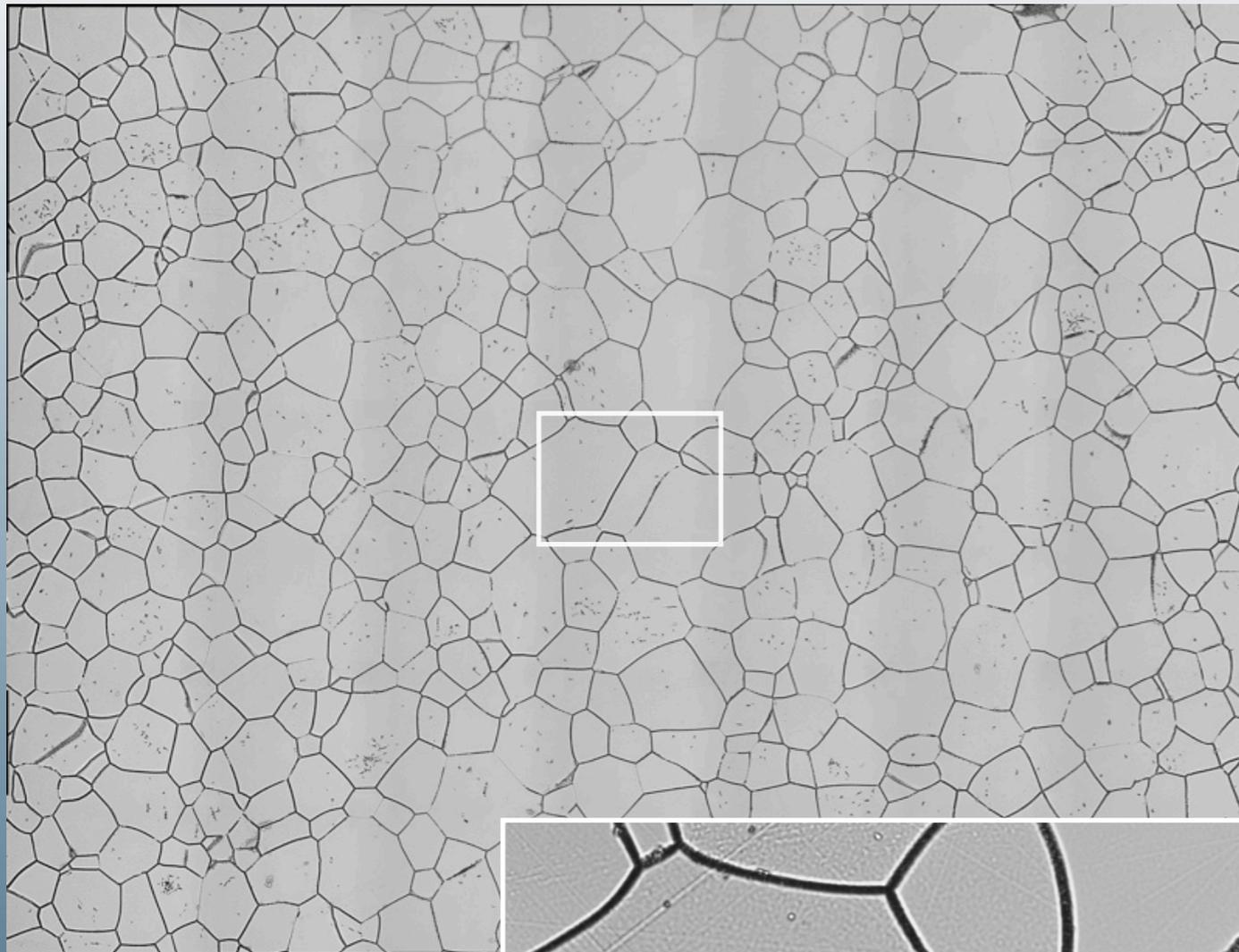
20 EBSD sections



# Alignment: Fiducial Marks Coarse Scale



# Original Image

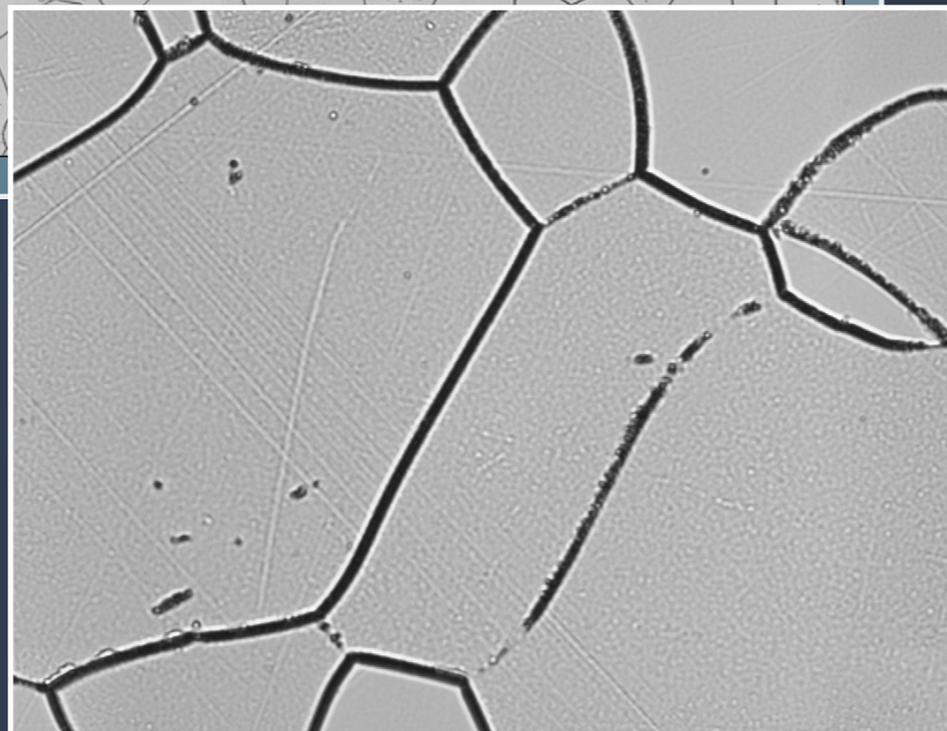


Ti  $\beta$ 21-S after 725C for 30 min -  $\alpha$  coats the grain boundaries.

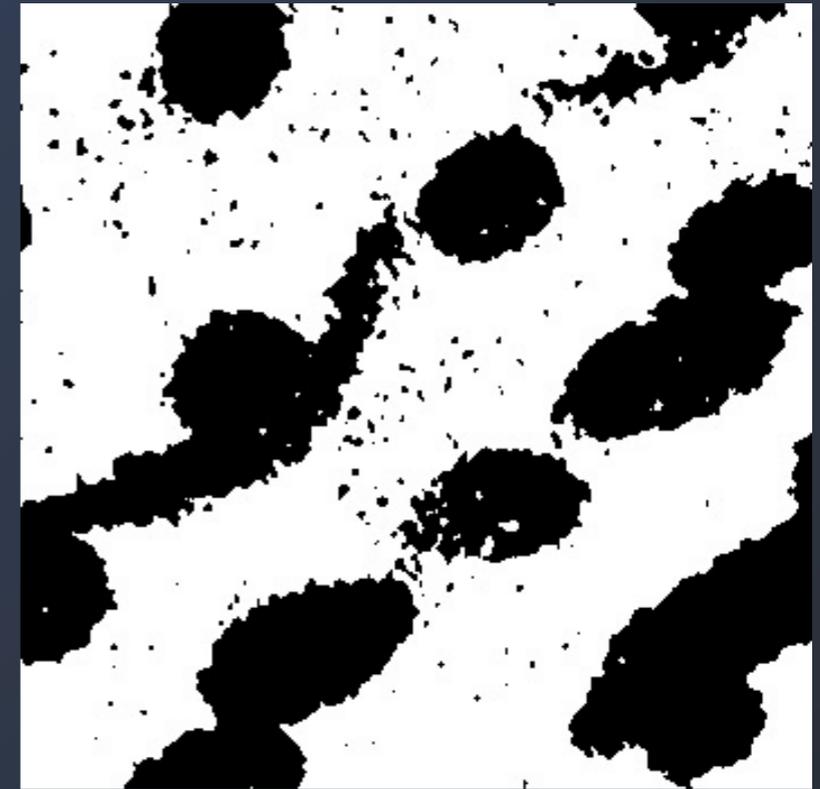
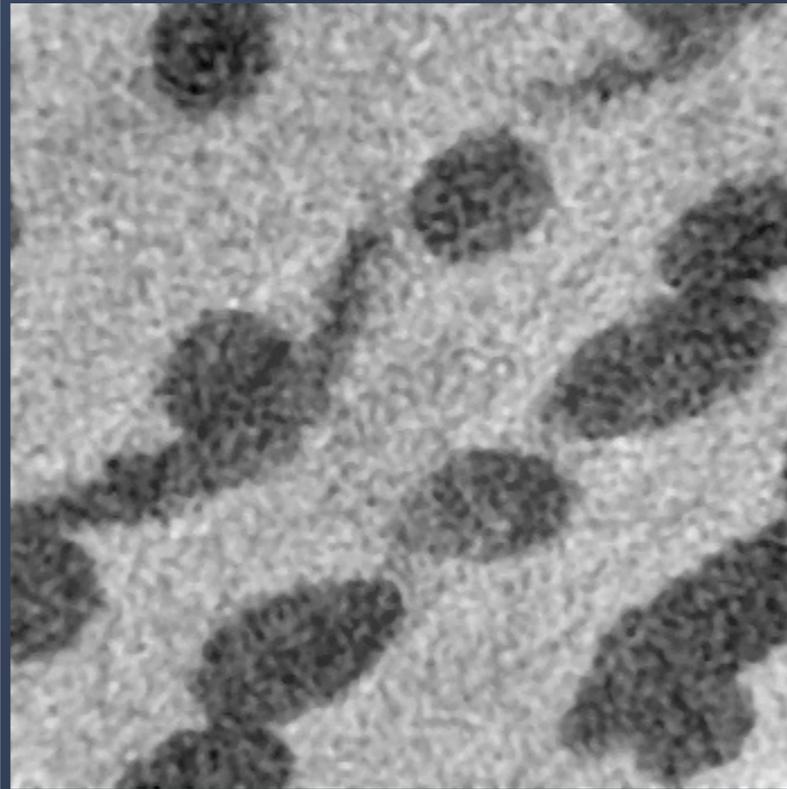
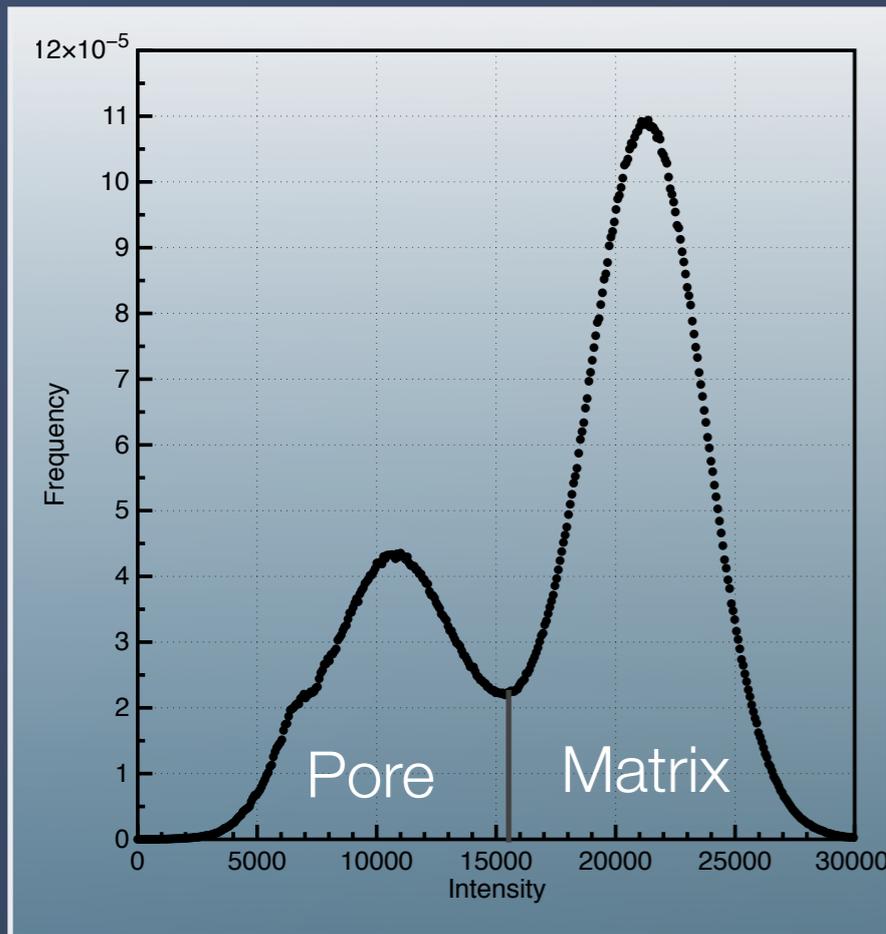
Montage of 8x8 images taken at 500x

A single threshold can be applied to the entire image.

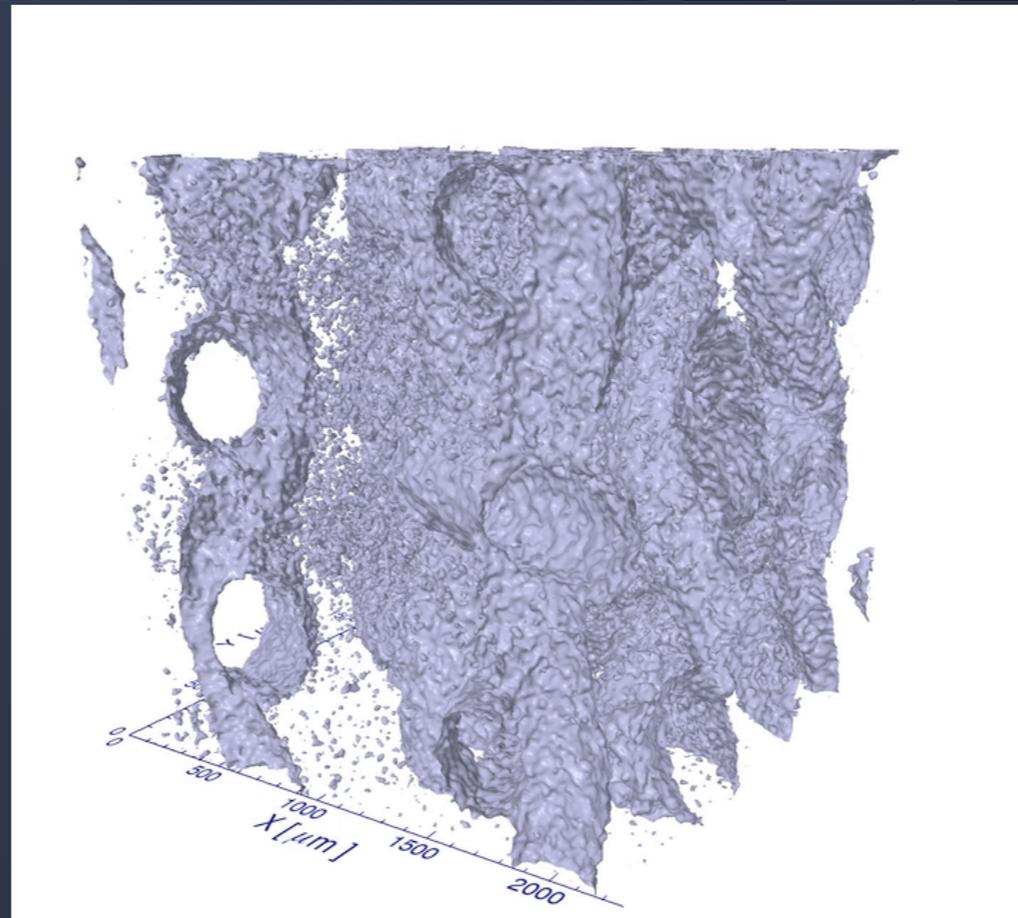
Further image processing will lead to a segmented image.



# Image segmentation example from tomography



**NiTi metallic foam:** A direct segmentation causes a large number of pixels within the image to be misclassified, ignores any of the information from the pixel neighborhood.



# EM/MPM

## EM/MPM segmentation algorithm

J P Simmons, P Chuang, M Comer, J E Spowart, M D Uchic and M De Graef Modeling and Simulation in Mat. Sci. & Eng. (2009)

### Expectation Maximization/Maximization of Posterior Marginals

EM/MPM seeks to minimize the number of pixels that are misclassified pixels assuming:

The pixels of the same region typically will neighbor each other.

The distribution in the intensities for each class are normally distributed.

The number of regions are know (set by the user).

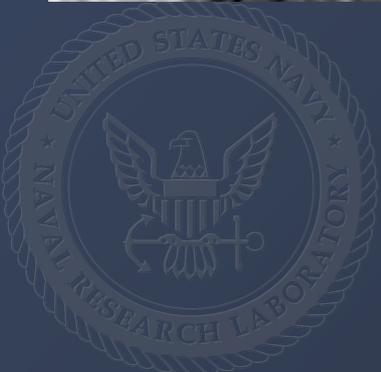
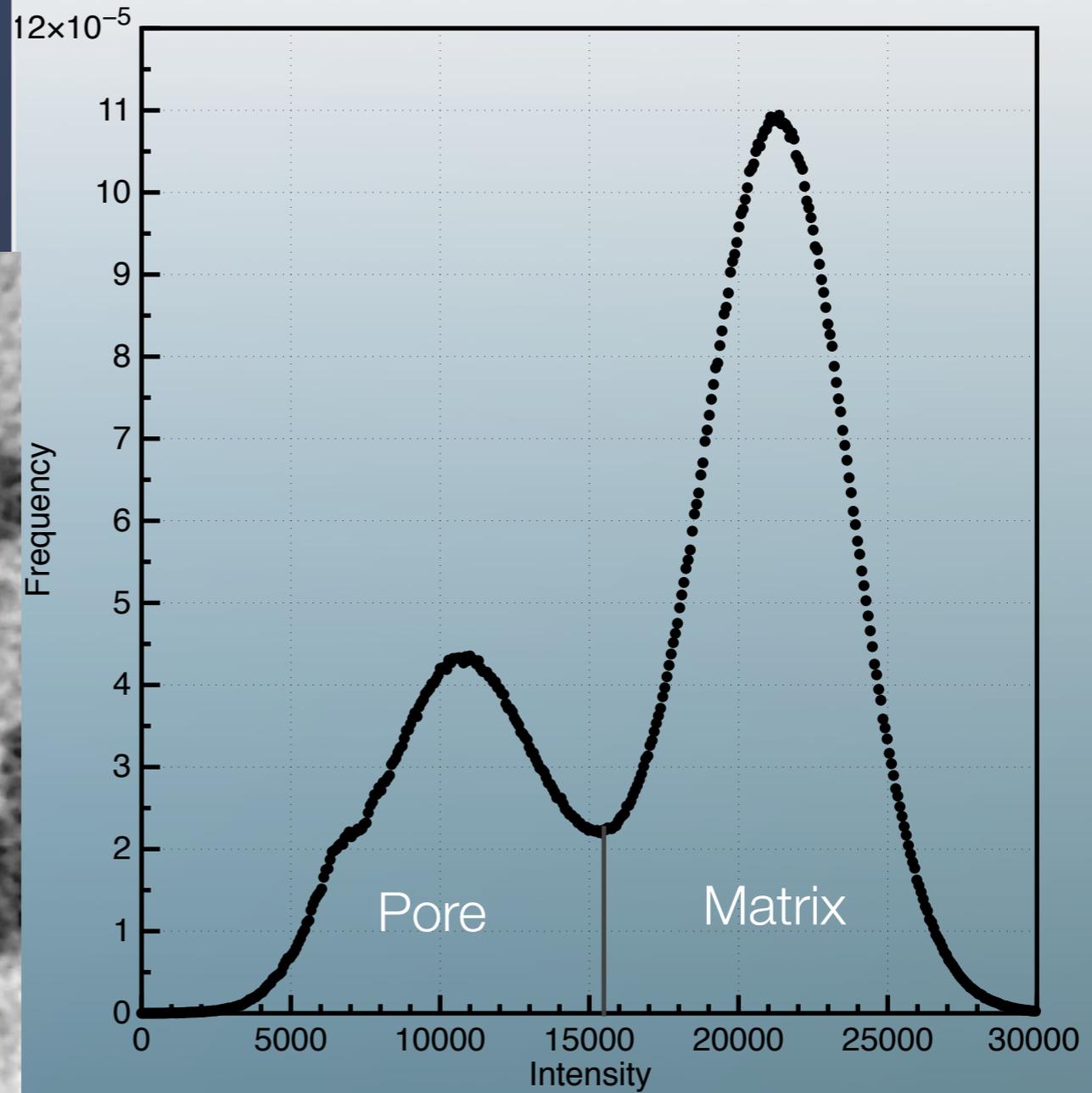
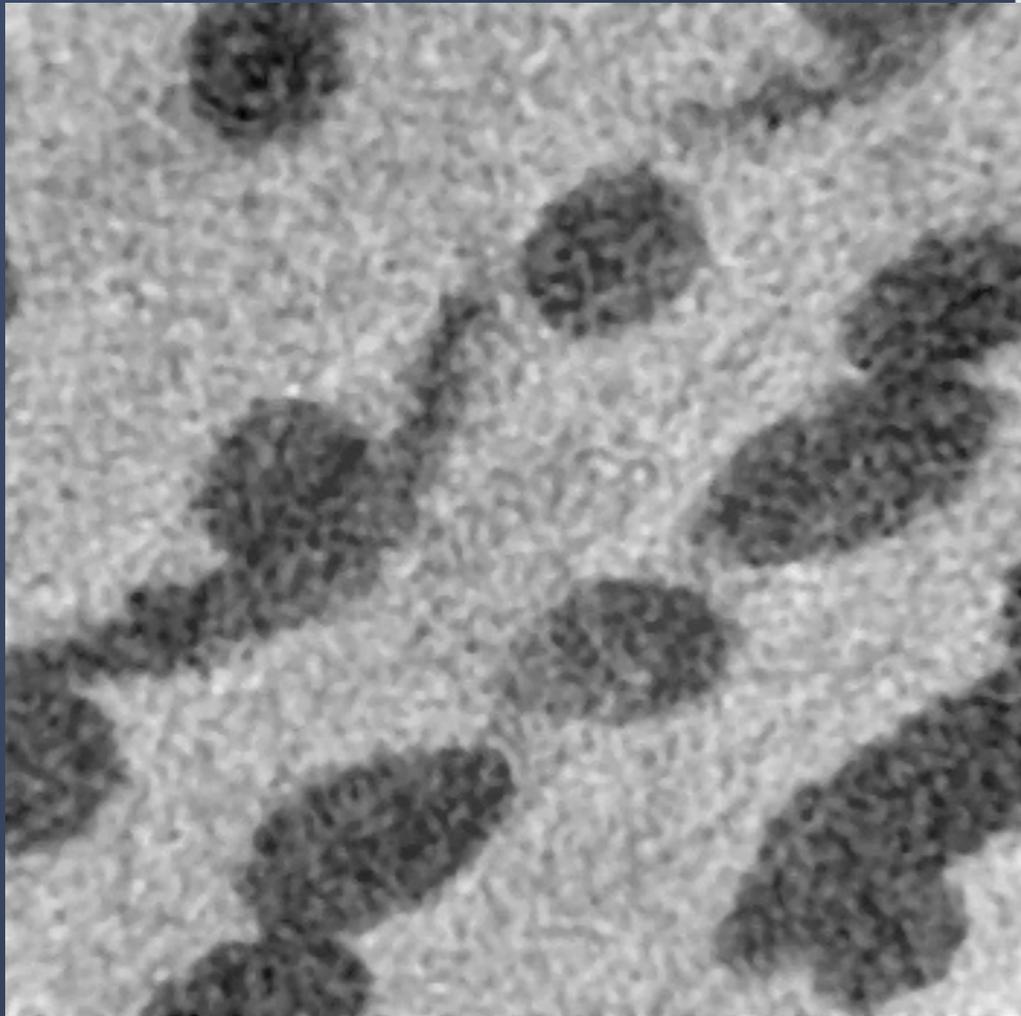
MPM: Evaluate the probability for each pixel in the image that it belongs to a particular class.  
(how closely does the mean and variance match the global values of that class)

- Evaluate the change in probability if the class number is changed, then keep the most probable (multi-spin Ising model Monte-Carlo)

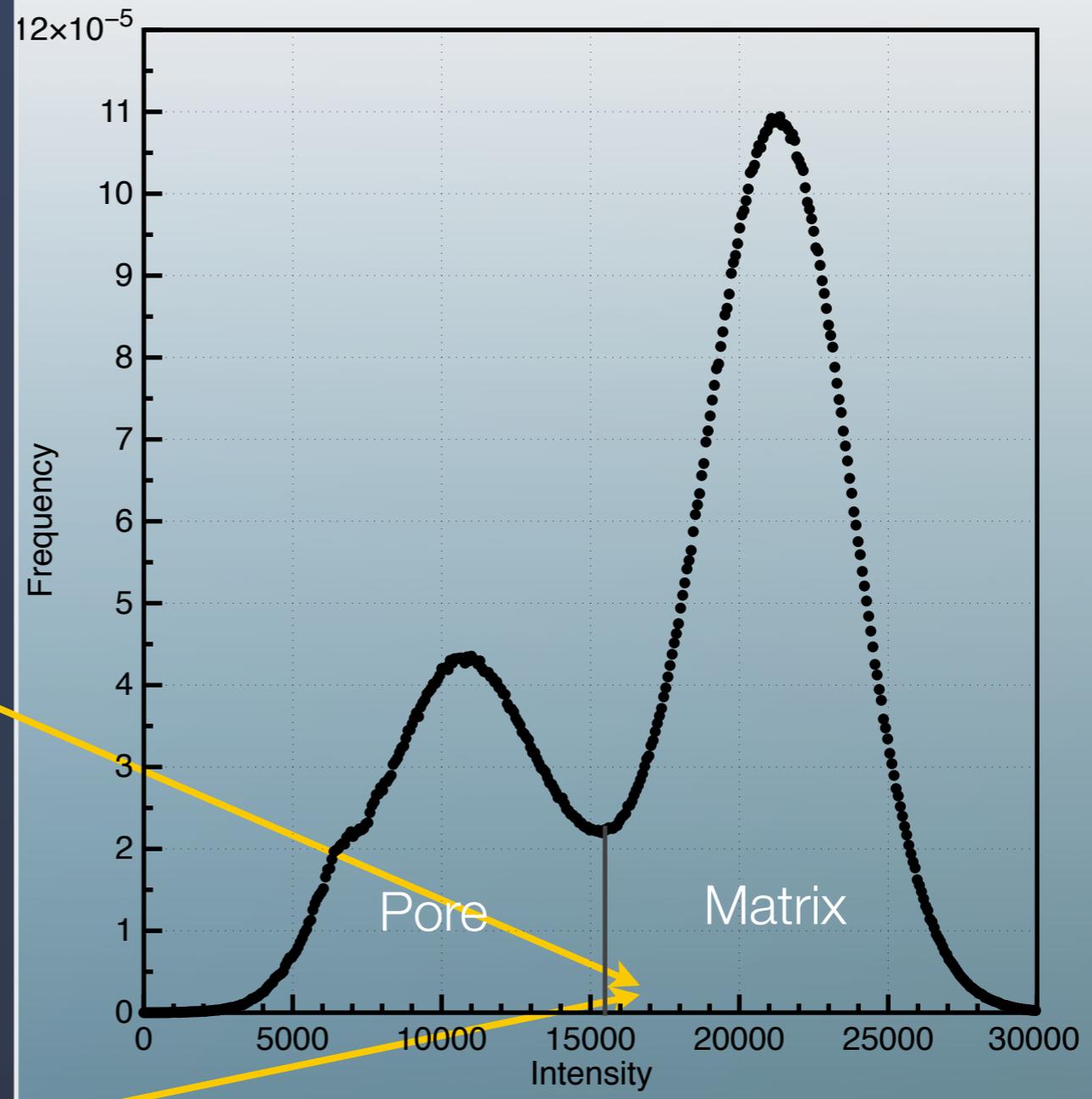
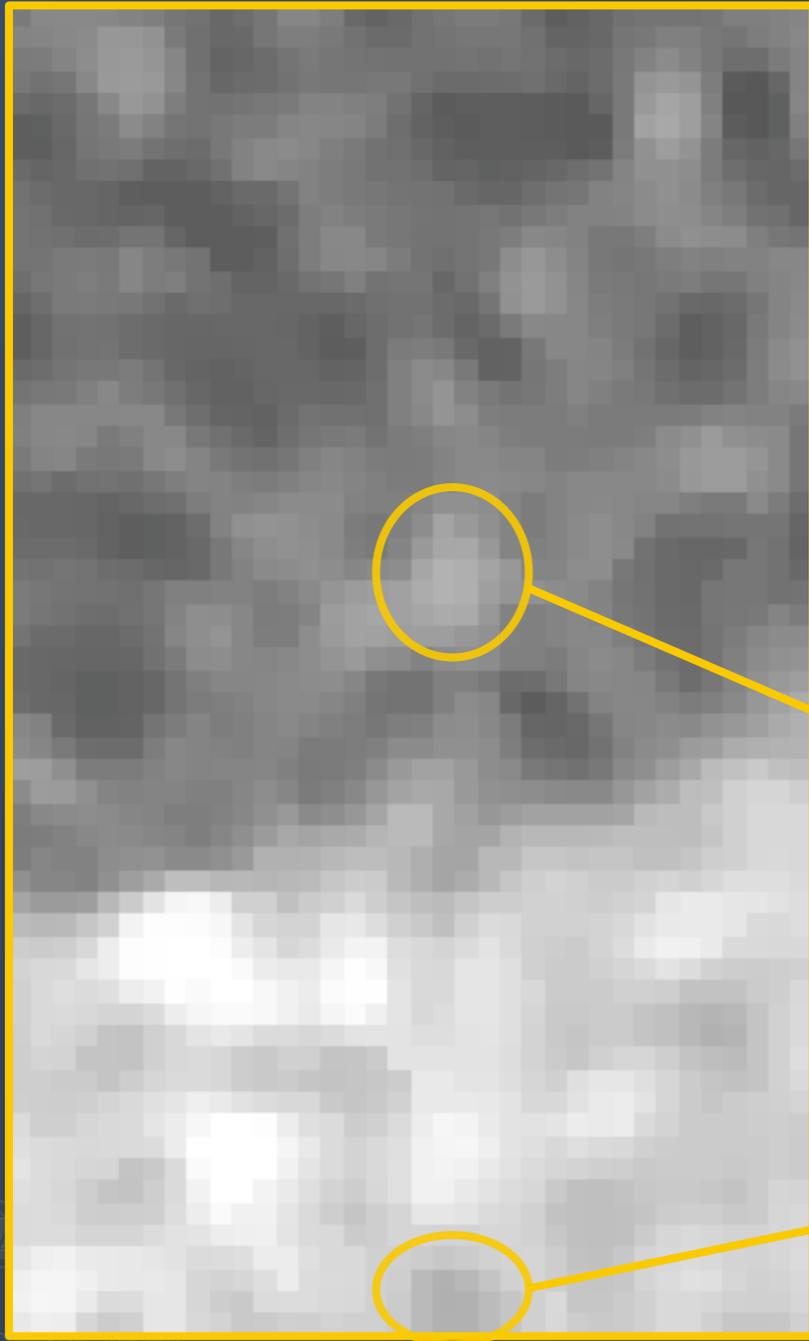
EM: Update the new values of the global distribution for each class and pass that back to MPM.



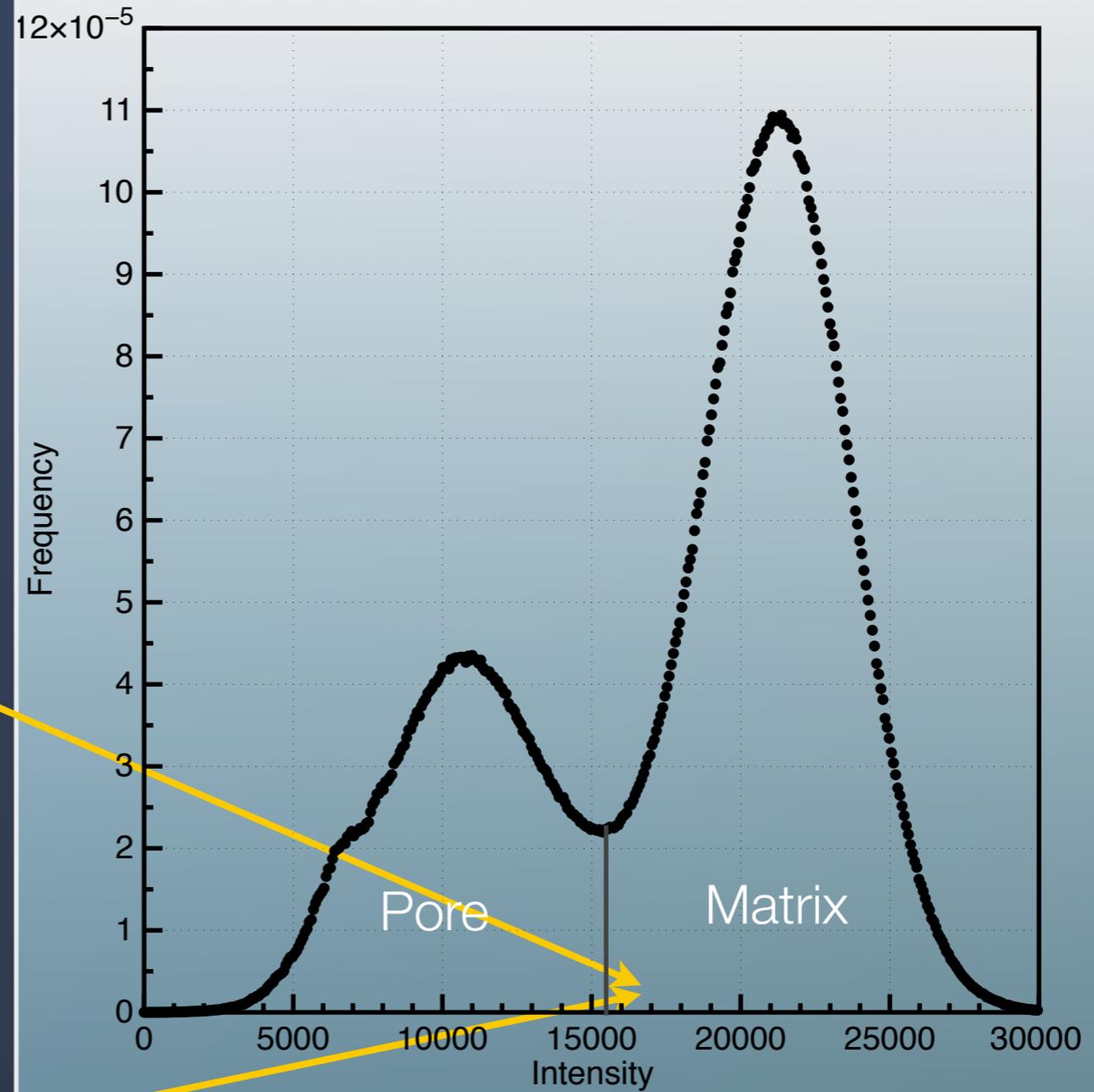
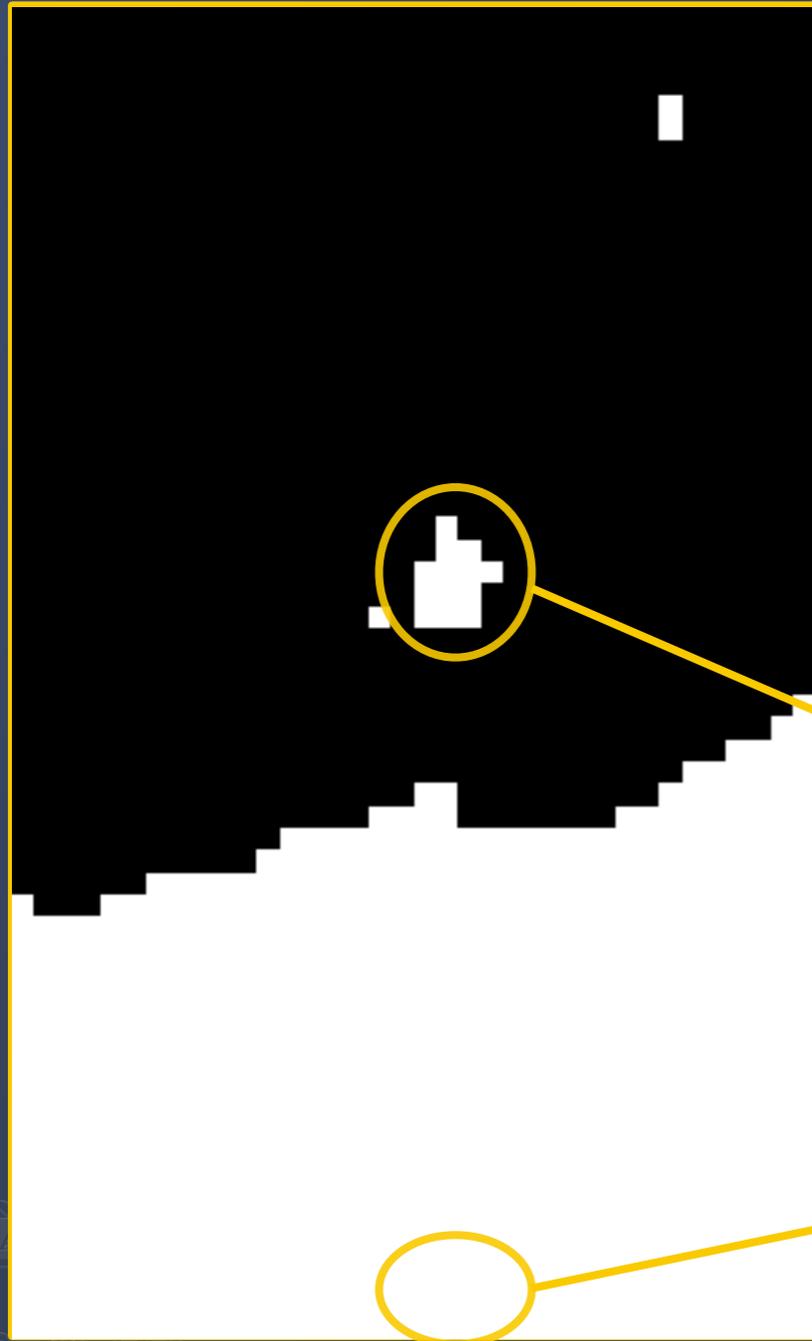
# EM/MPM



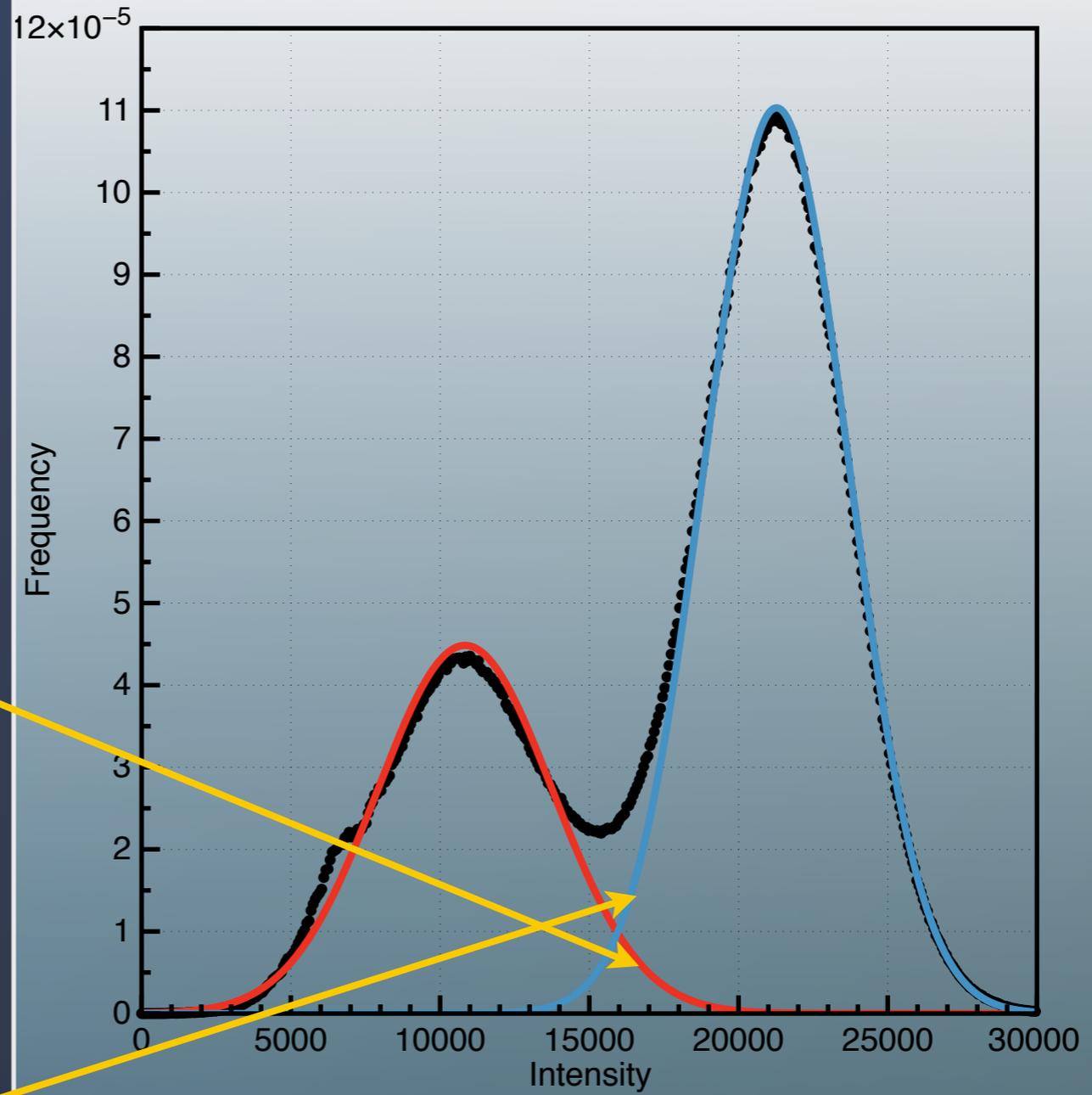
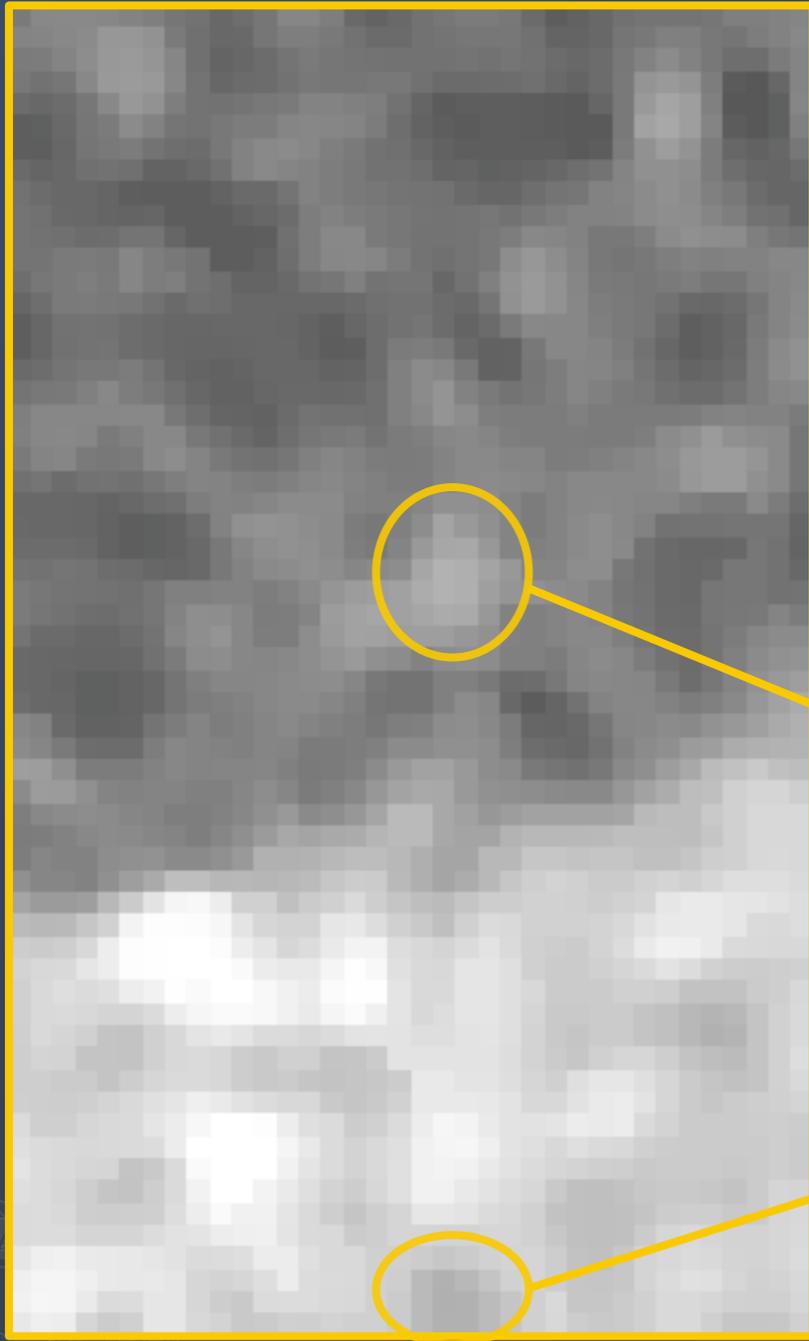
# EM/MPM



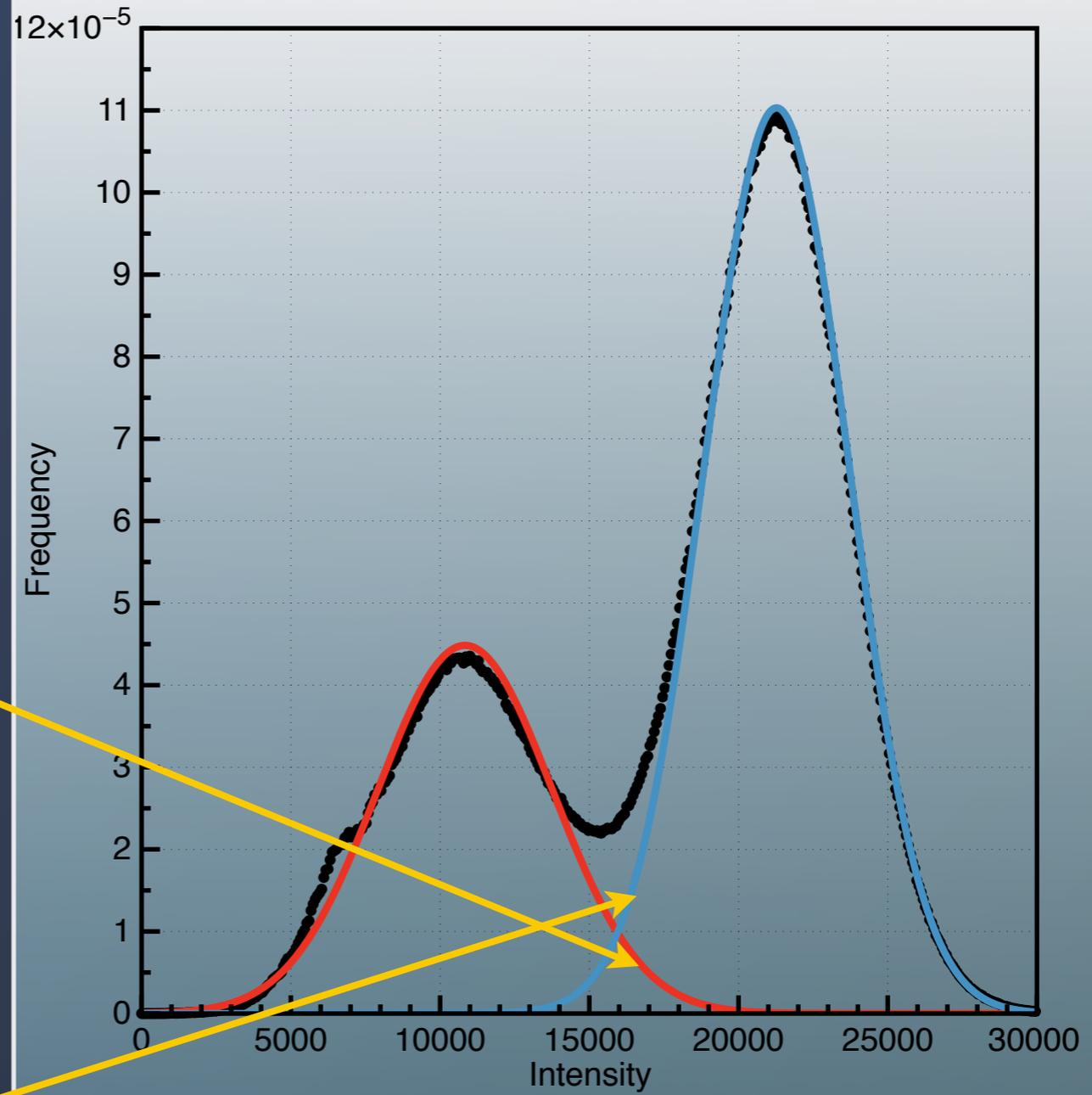
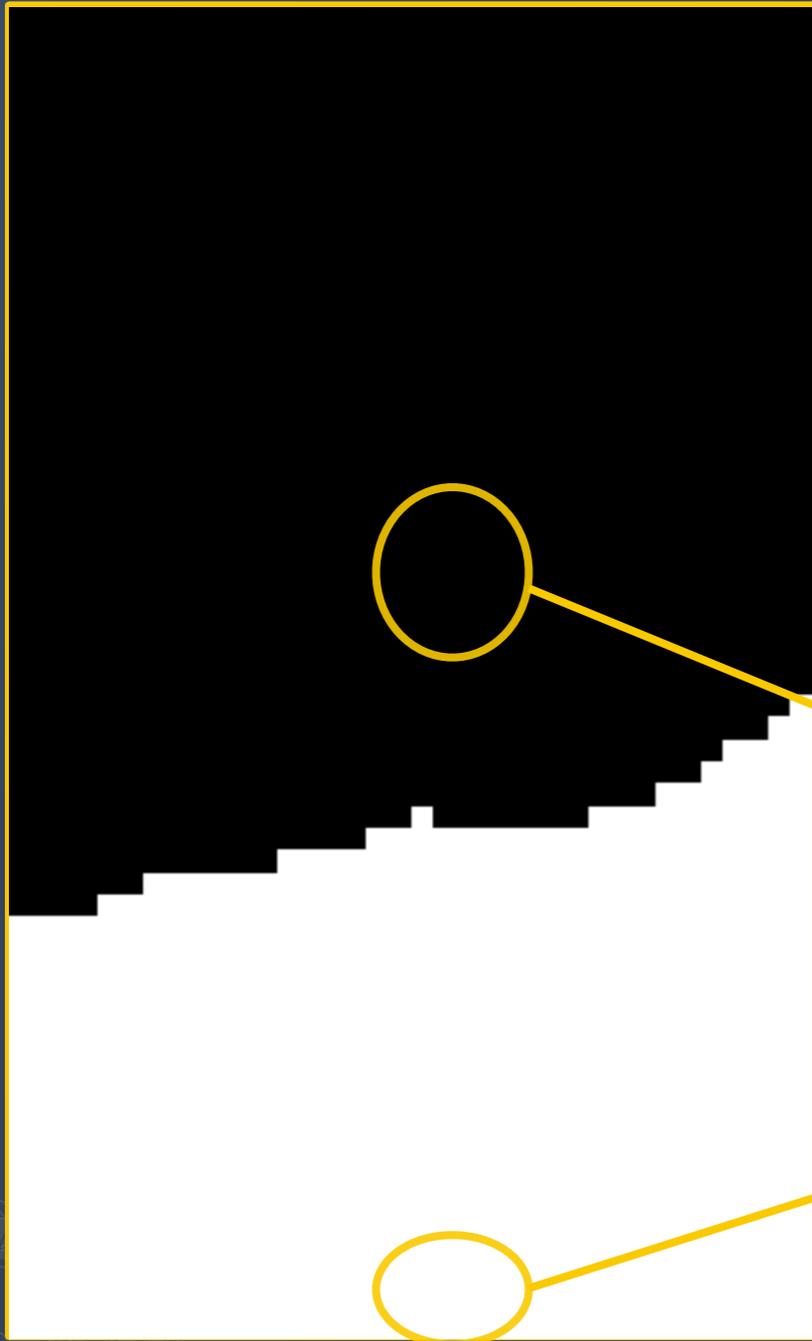
# EM/MPM



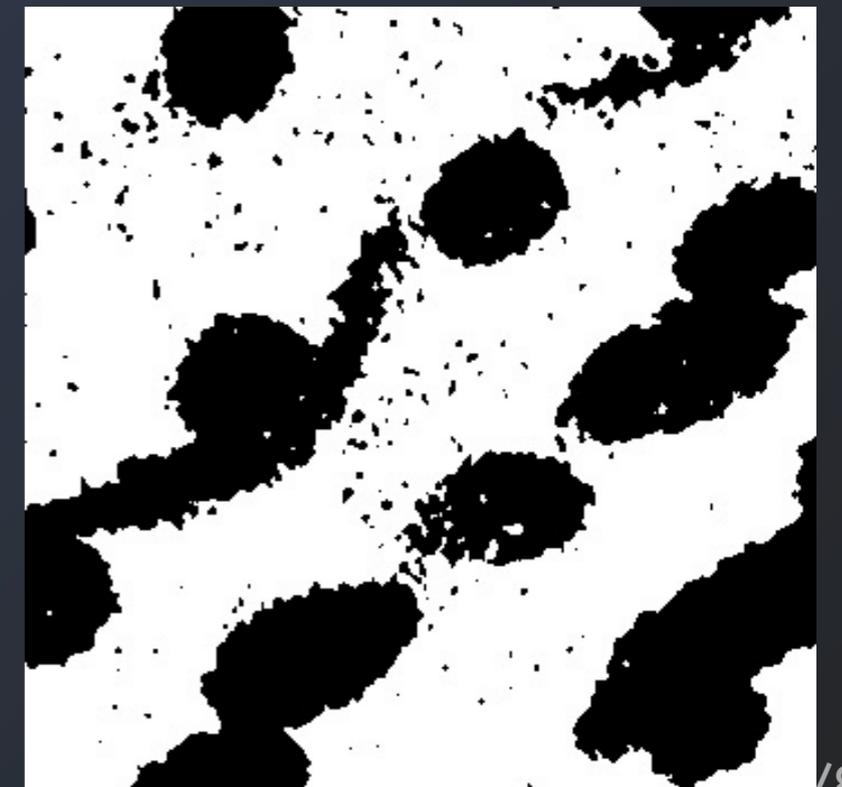
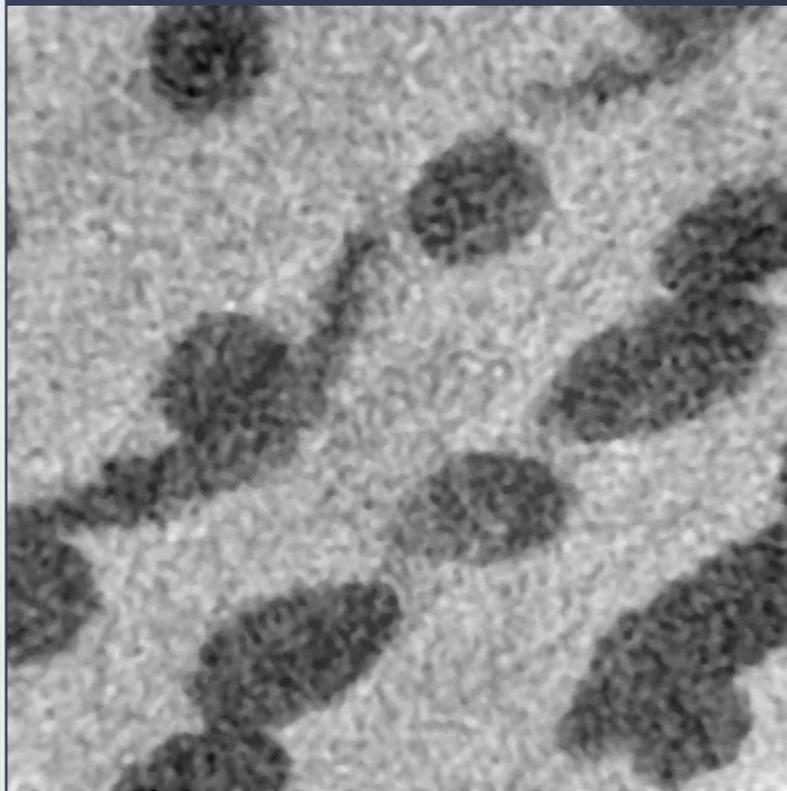
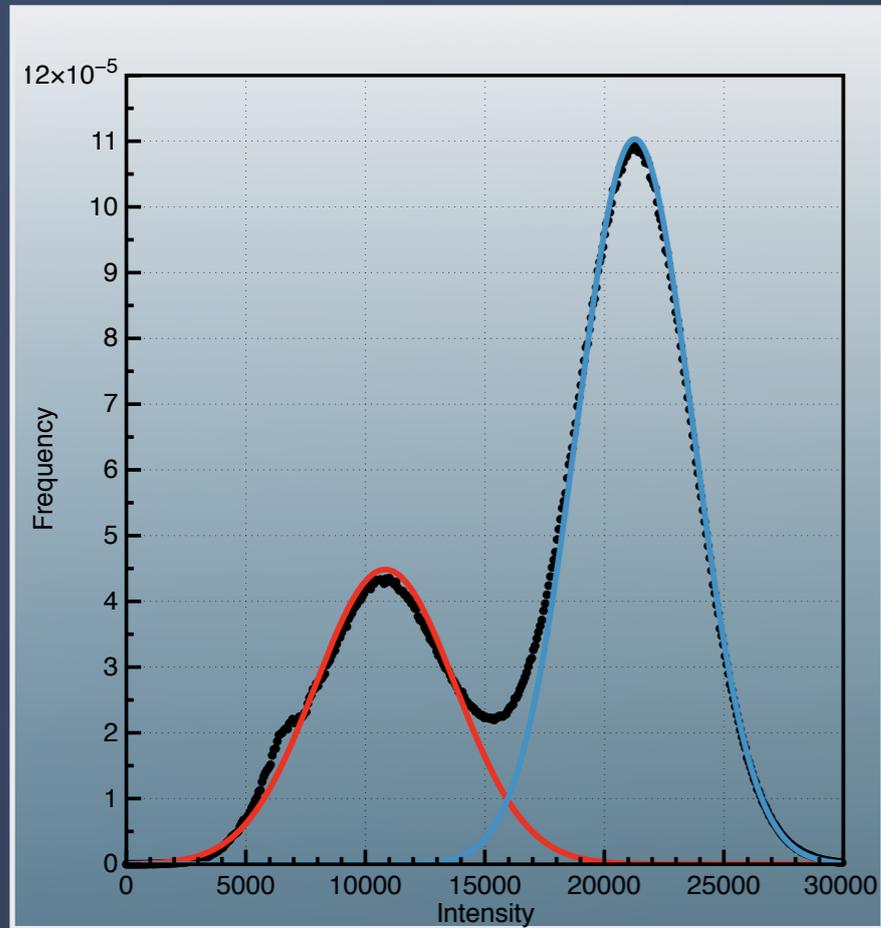
# EM/MPM



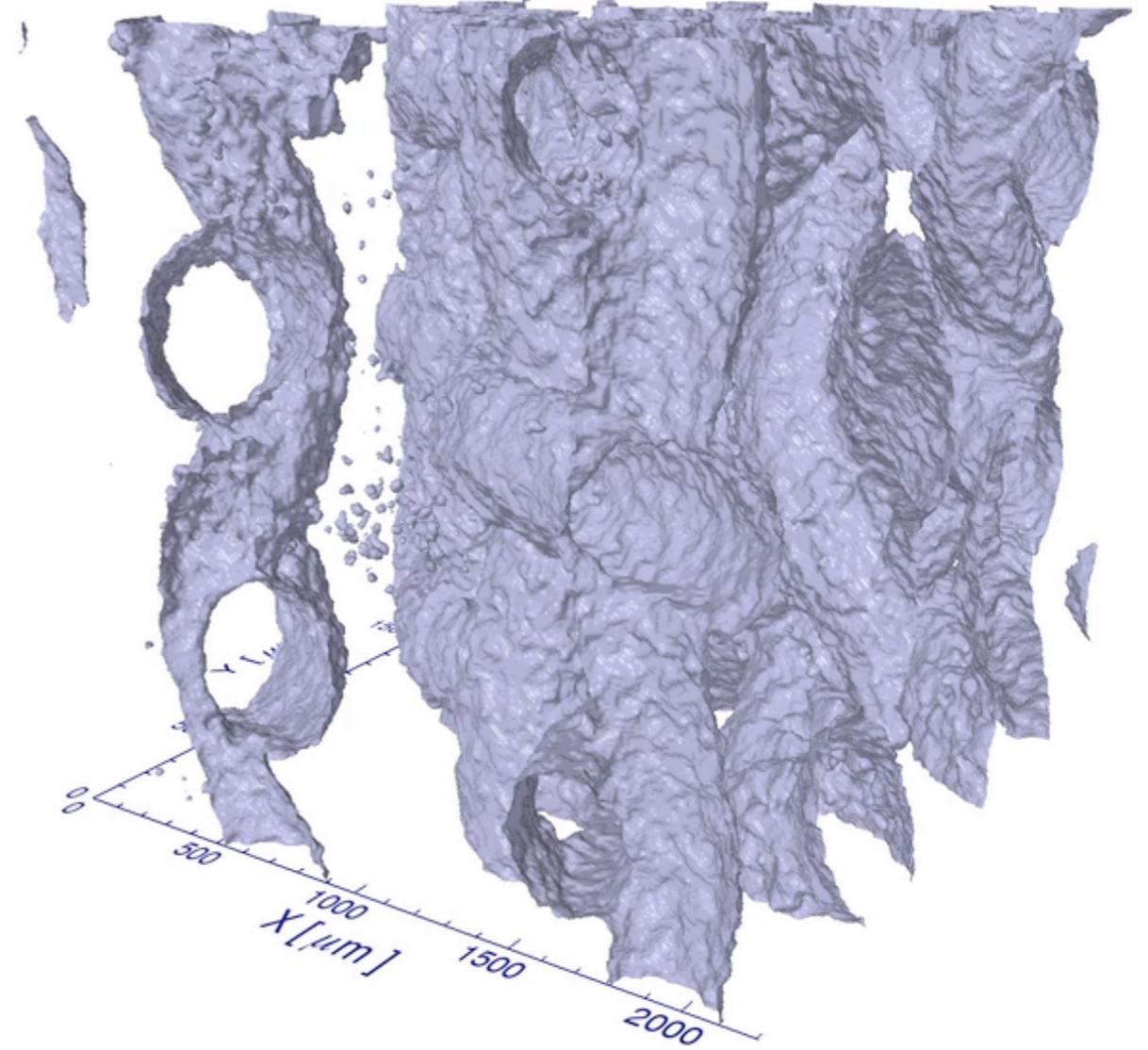
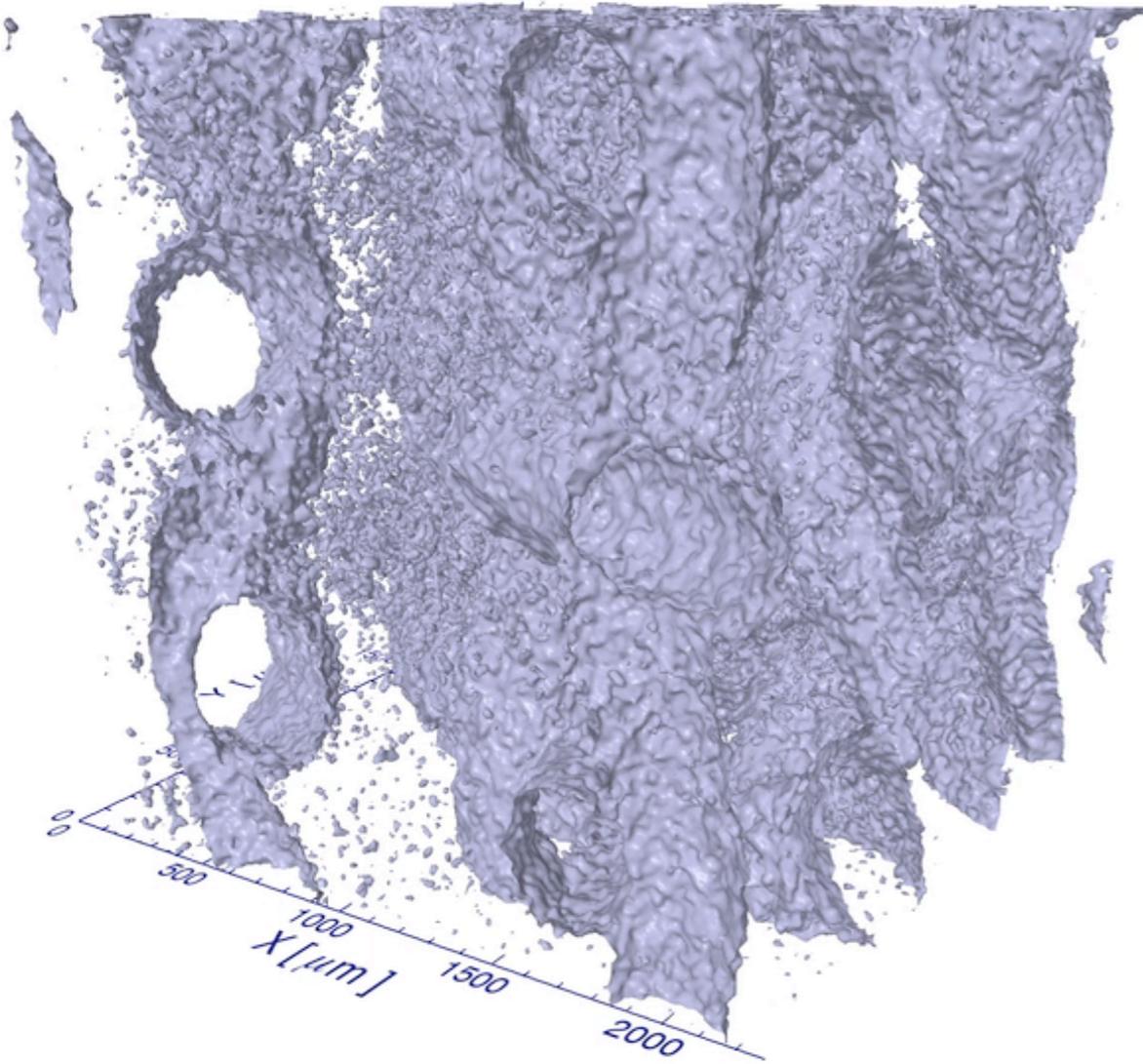
# EM/MPM



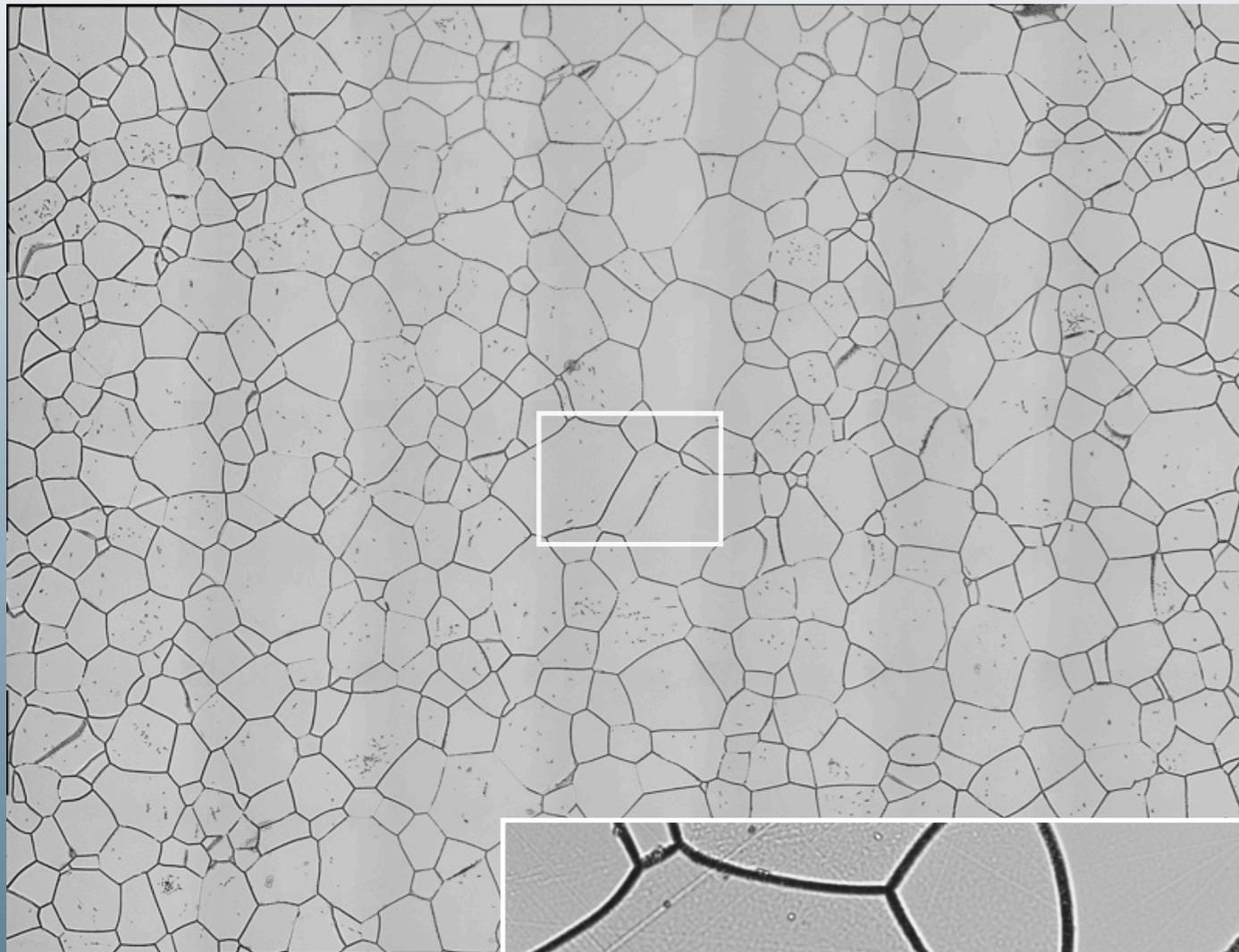
# EM/MPM Image Segmentation



# EM/MPM Segmentation



# Original Image

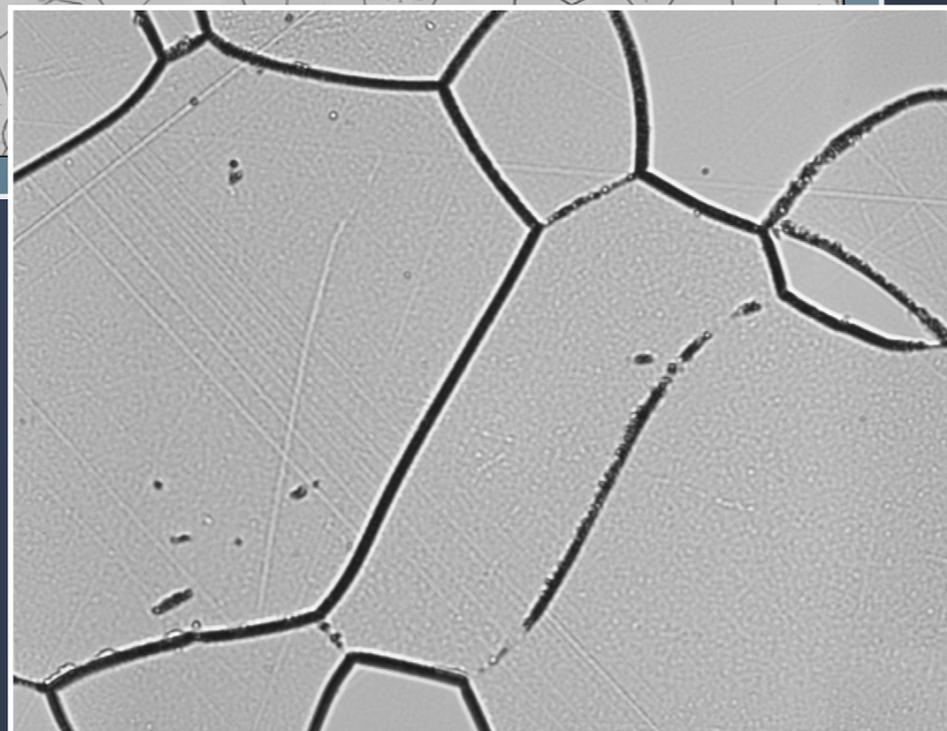


Ti  $\beta$ 21-S after 725C for 30 min -  $\alpha$  coats the grain boundaries.

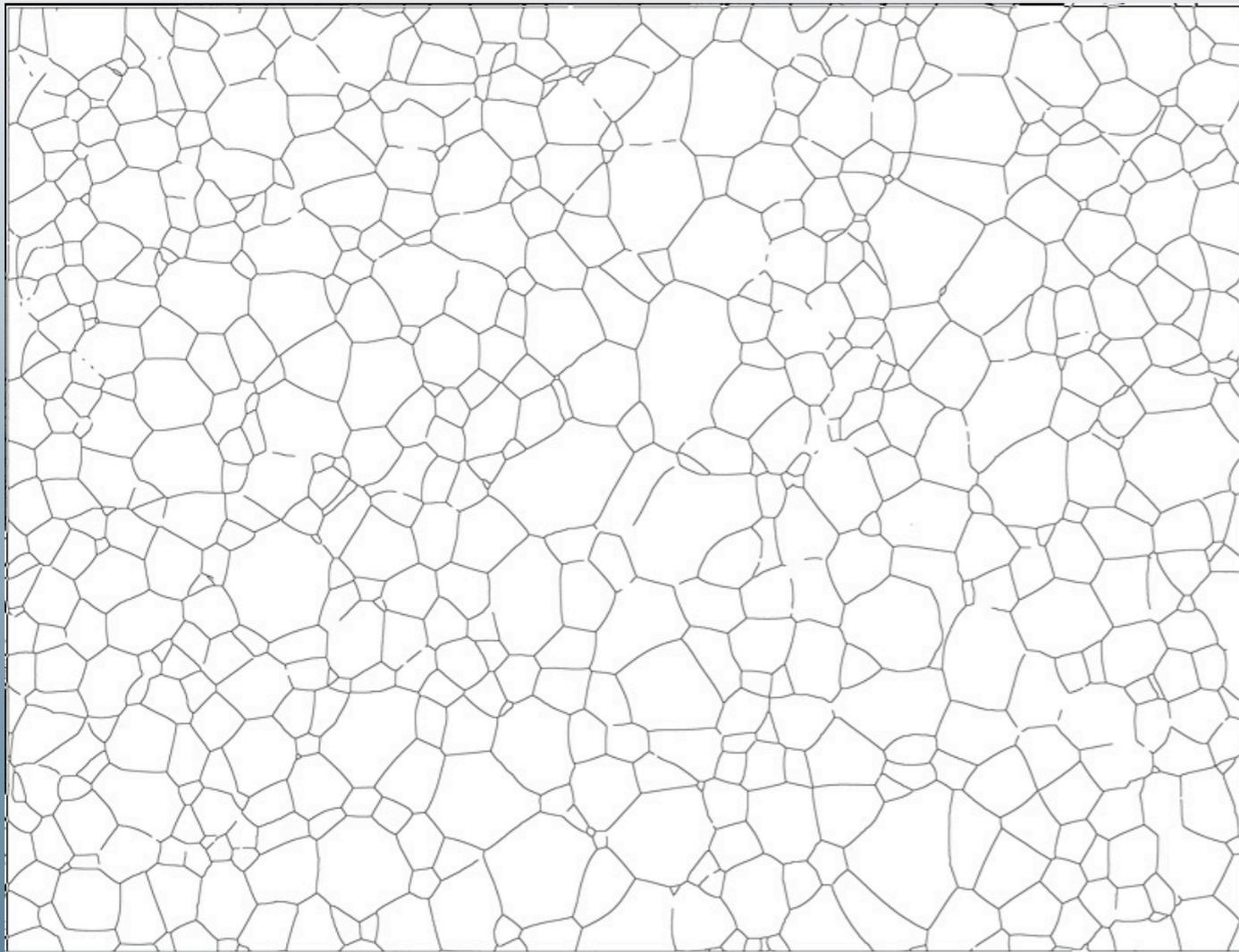
Montage of 8x8 images taken at 500x

A single threshold can be applied to the entire image.

Further image processing will lead to a segmented image.



# Image Filtering

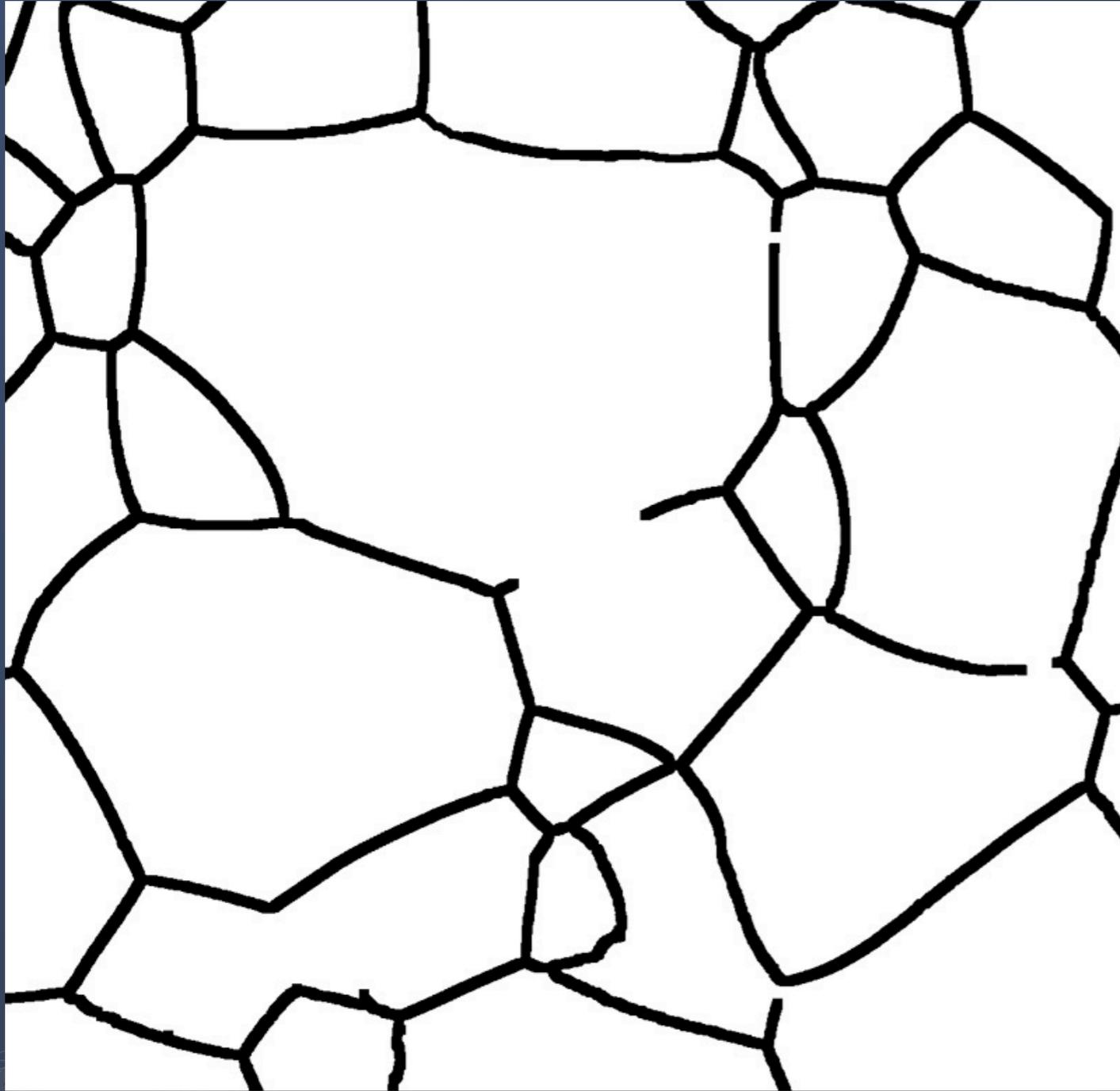


The internal alpha regions are removed from the image.

To fill in the gaps in the boundary we use a distance function in combination with the watershed algorithm



# Grain Segmentation and Labeling

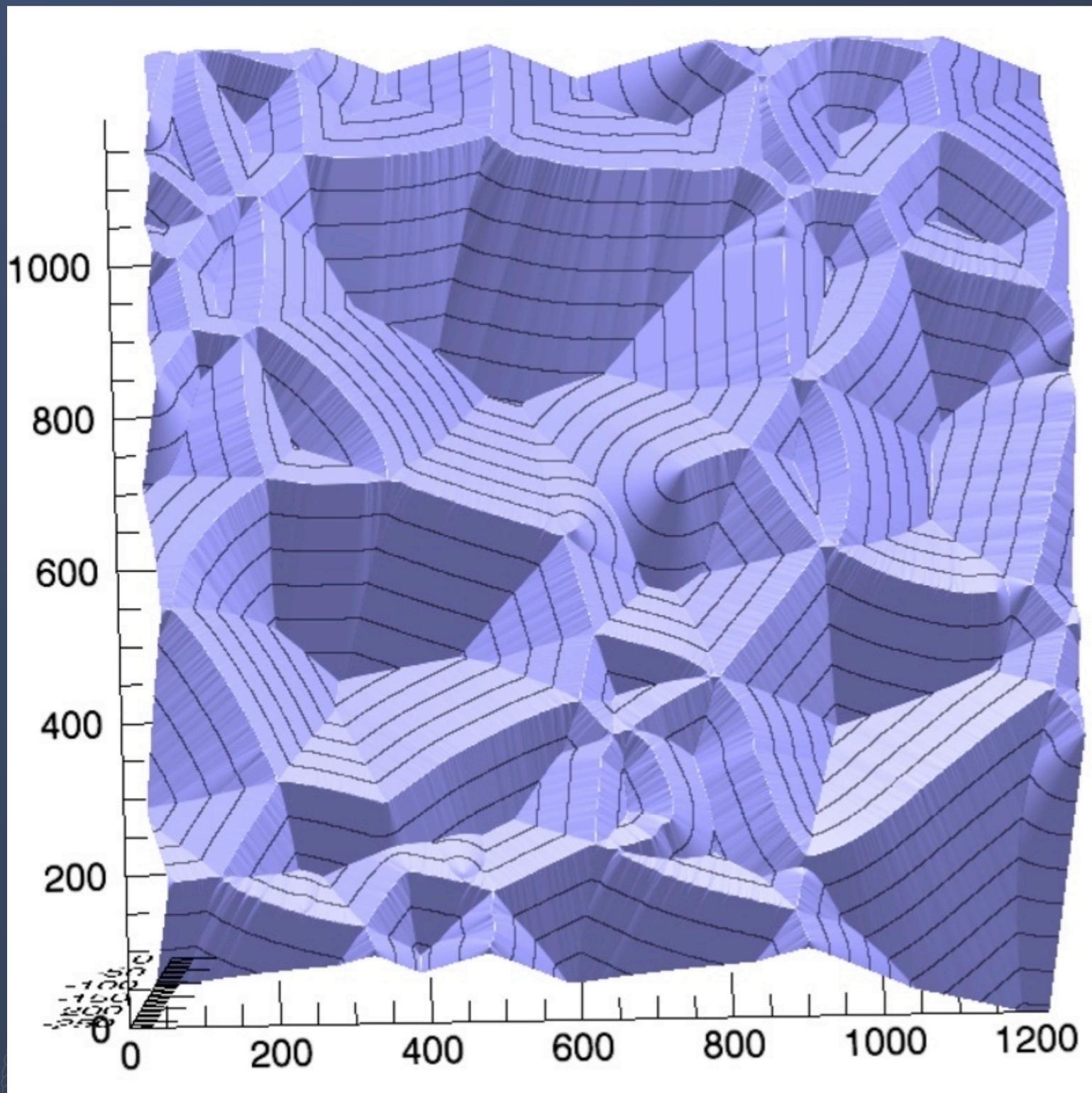


Thresholding and Grain Decoration is not perfect.  
Breaks in the boundary need to be filled using a watershed algorithm.

The WS starts with a Distance Function (DF): The value of each pixel is labeled as the minimum distance from an interface.  
Each grain becomes a “basin”

A distance function is created of the boundary network

# Grain Segmentation and Labeling



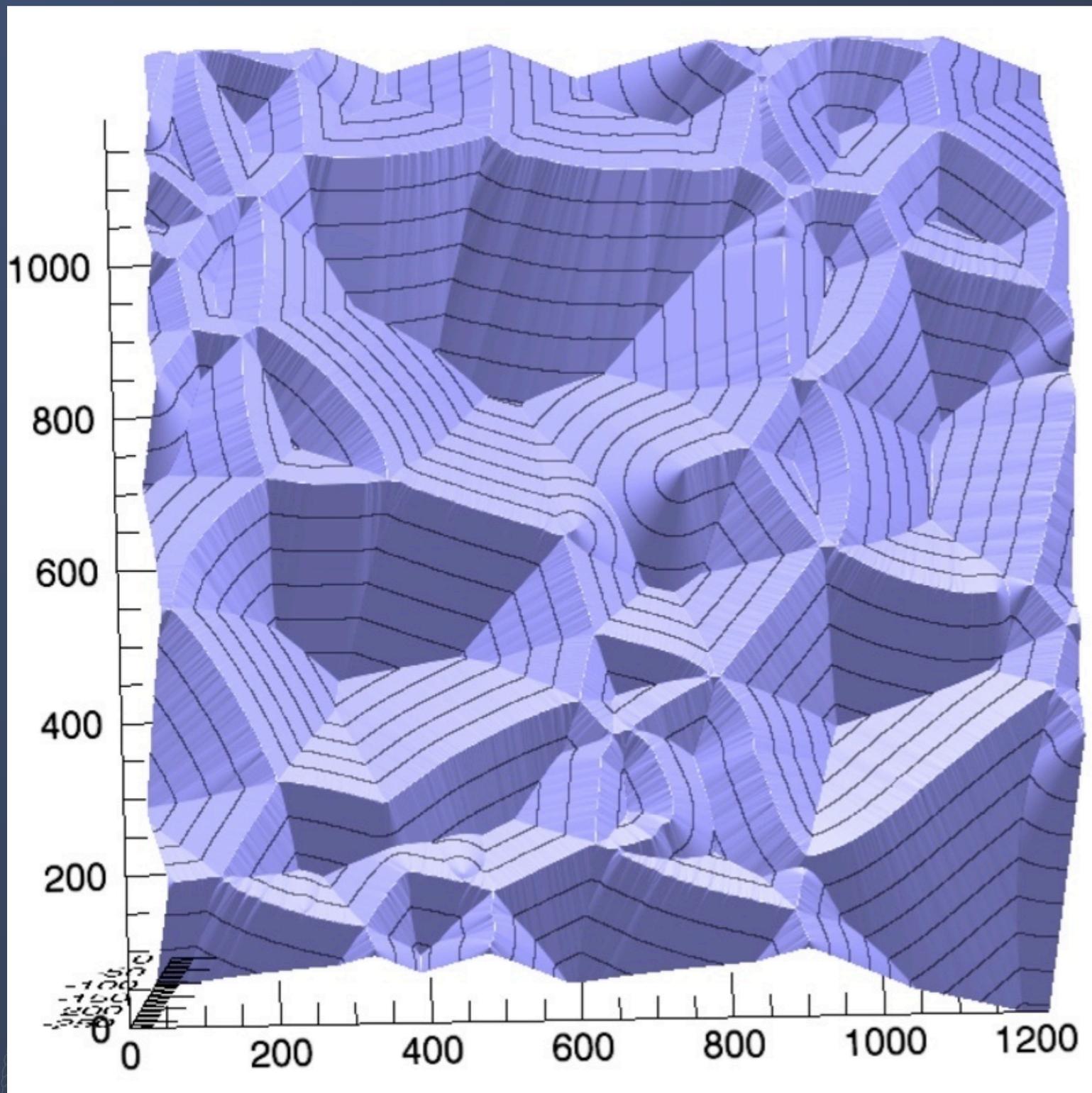
Thresholding and Grain Decoration is not perfect.  
Breaks in the boundary need to be filled using a watershed algorithm.

The WS starts with a Distance Function (DF): The value of each pixel is labeled as the minimum distance from an interface.  
Each grain becomes a “basin”

A distance function is created of the boundary network

(Huttenlocher, D. P., & Felzenszwalb, P. F. (2004). Distance Transforms of Sampled Functions ( No. TR20041963) (pp. 1–15).

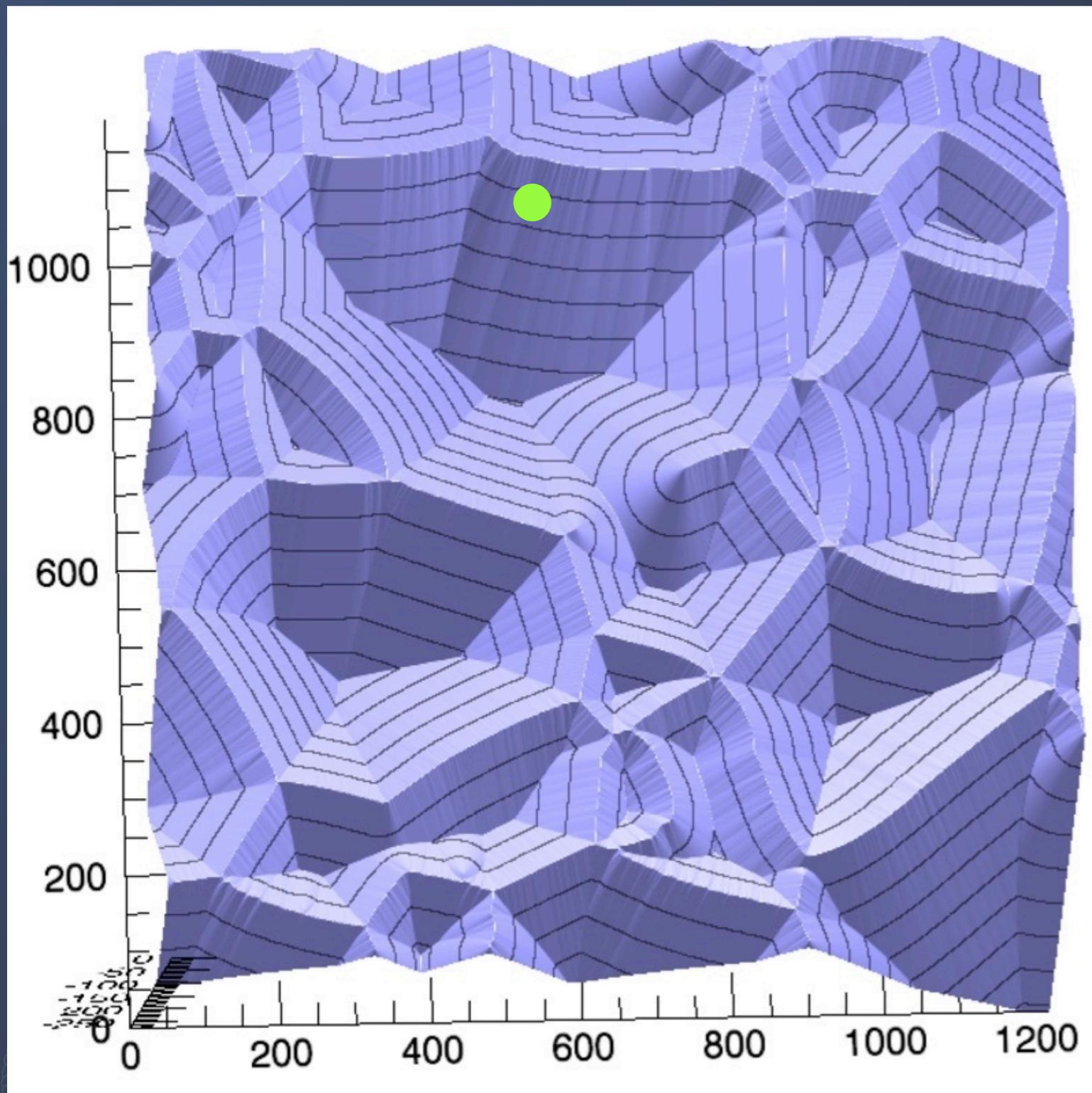
# Watershed Segmentation



Imagine placing a water-drop on each of the pixels of the DF. Droplets that collect in the same basin are labeled with a unique identifier.

We developed a 3D “rainfalling” watershed algorithm that gives superior results to traditional algorithms.

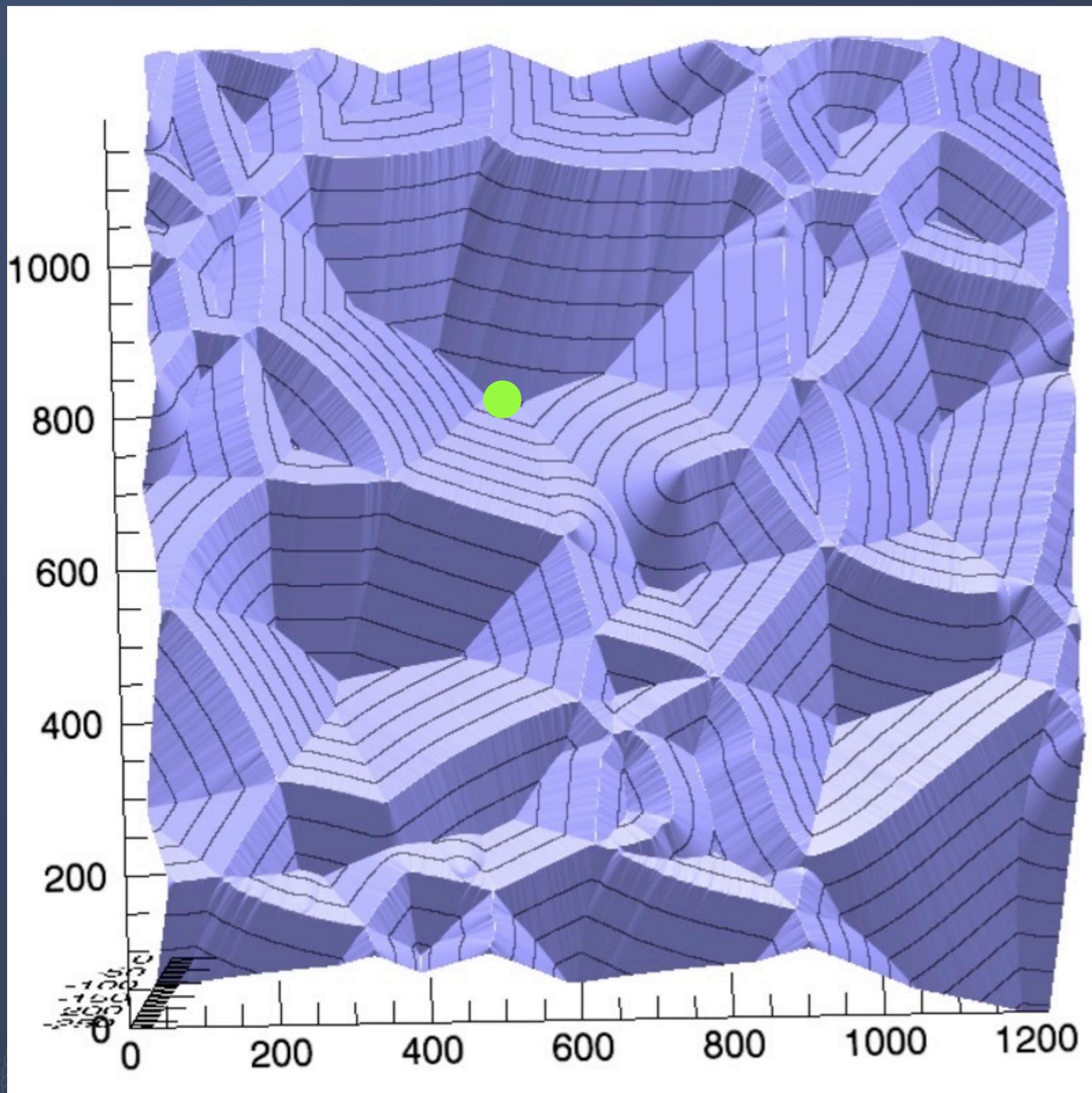
# Watershed Segmentation



Imagine placing a water-drop on each of the pixels of the DF. Droplets that collect in the same basin are labeled with a unique identifier.

We developed a 3D “rainfalling” watershed algorithm that gives superior results to traditional algorithms.

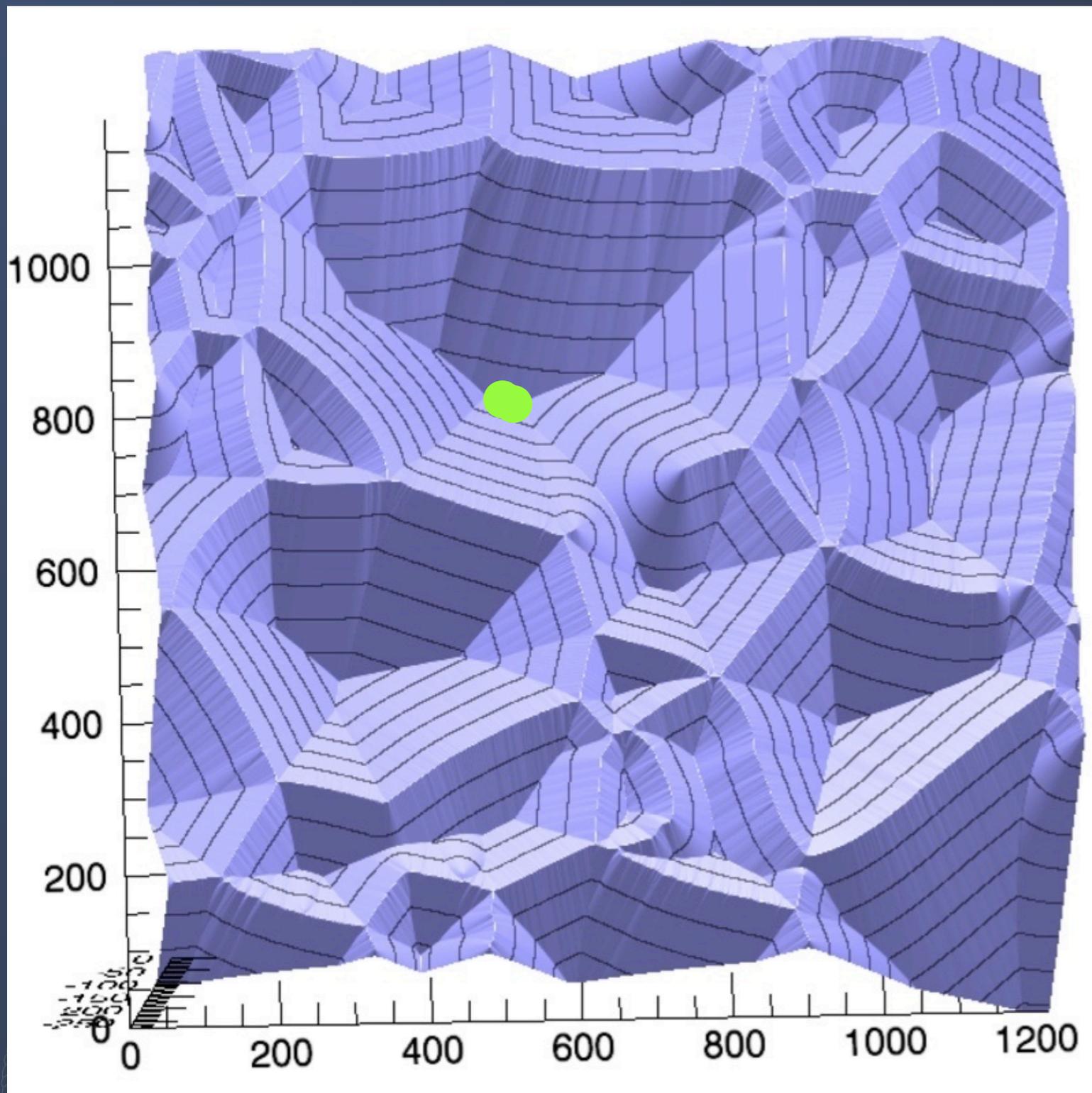
# Watershed Segmentation



Imagine placing a water-drop on each of the pixels of the DF. Droplets that collect in the same basin are labeled with a unique identifier.

We developed a 3D “rainfalling” watershed algorithm that gives superior results to traditional algorithms.

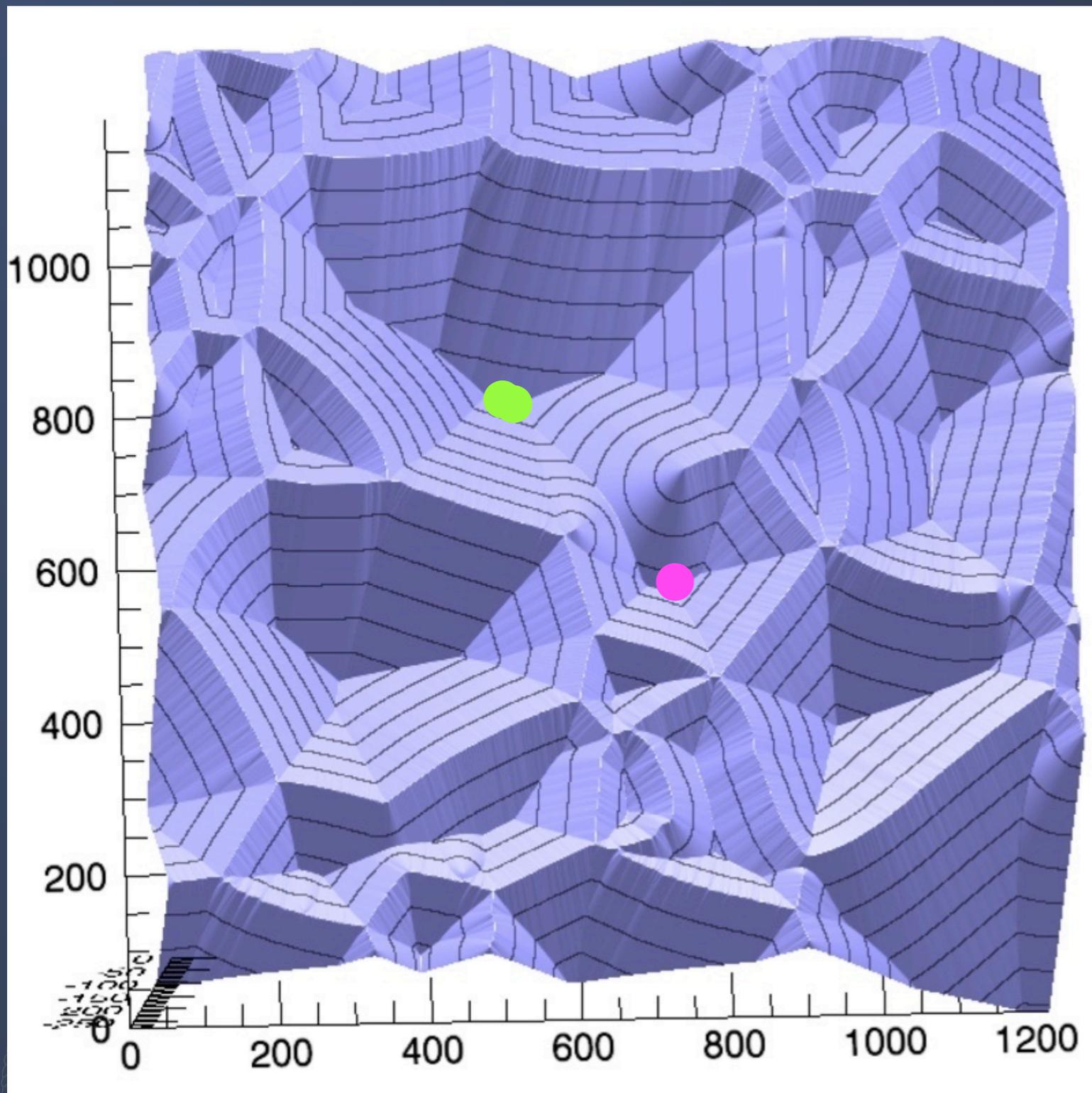
# Watershed Segmentation



Imagine placing a water-drop on each of the pixels of the DF. Droplets that collect in the same basin are labeled with a unique identifier.

We developed a 3D “rainfalling” watershed algorithm that gives superior results to traditional algorithms.

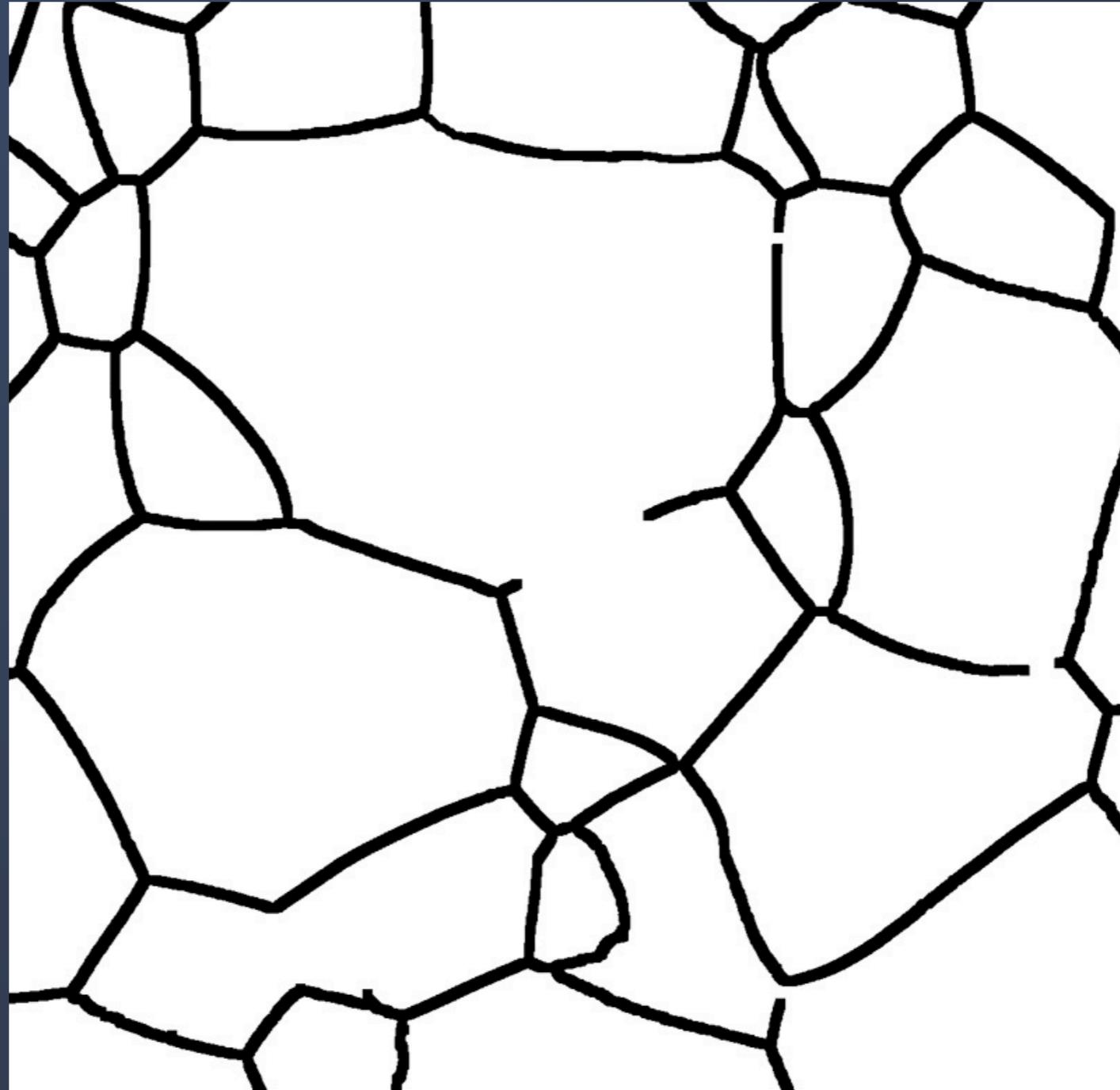
# Watershed Segmentation



Imagine placing a water-drop on each of the pixels of the DF. Droplets that collect in the same basin are labeled with a unique identifier.

We developed a 3D “rainfalling” watershed algorithm that gives superior results to traditional algorithms.

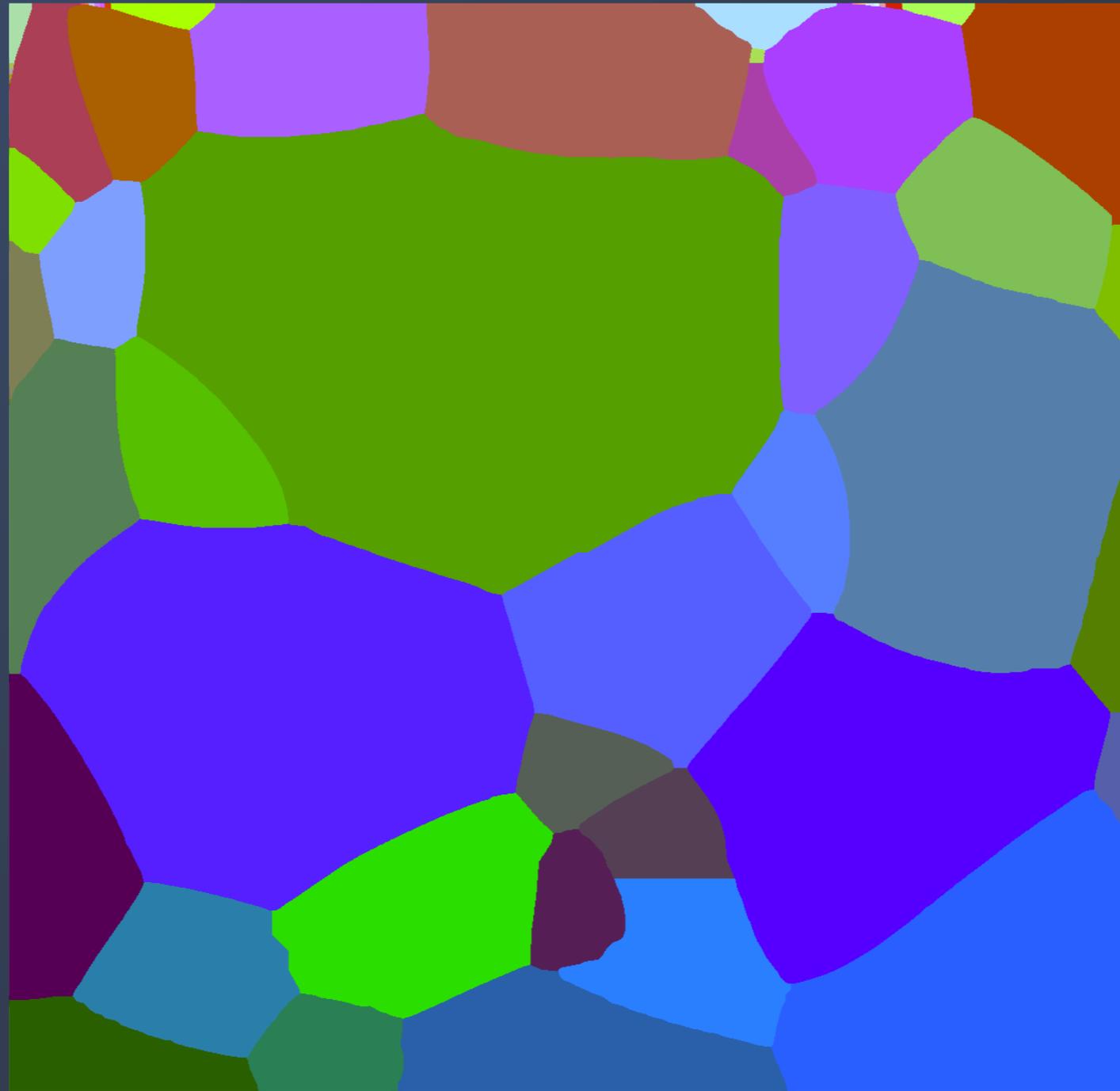
# Watershed - Final Result



The segmented voxel structure makes it easy to count grain volumes, aspect ratios (and other higher order moments), nearest neighbor correlations, ...



# Watershed - Final Result

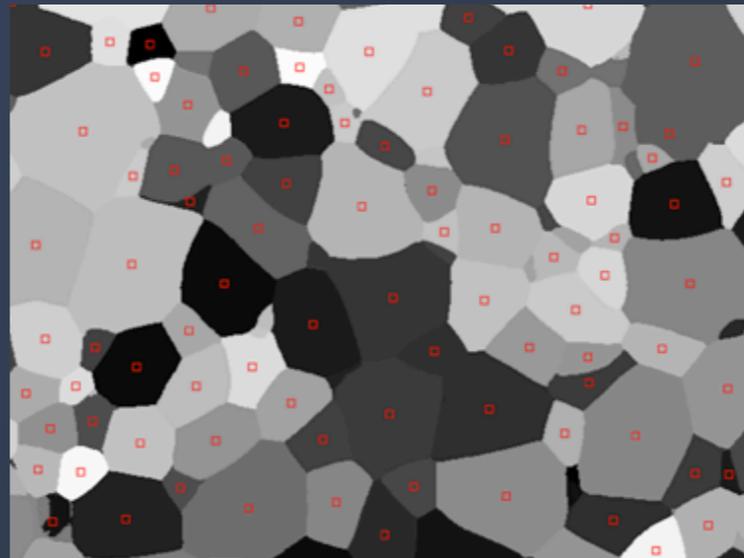
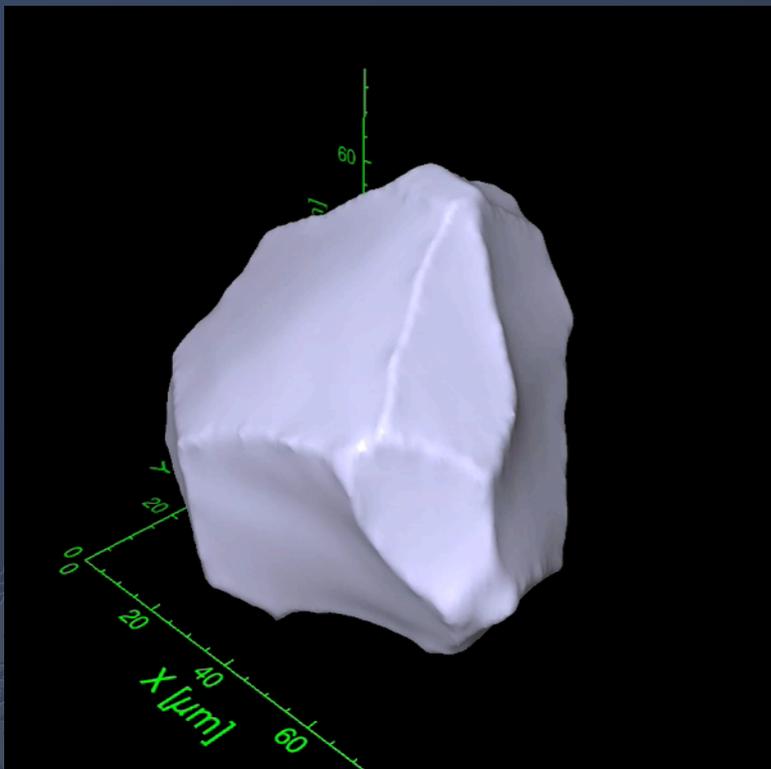
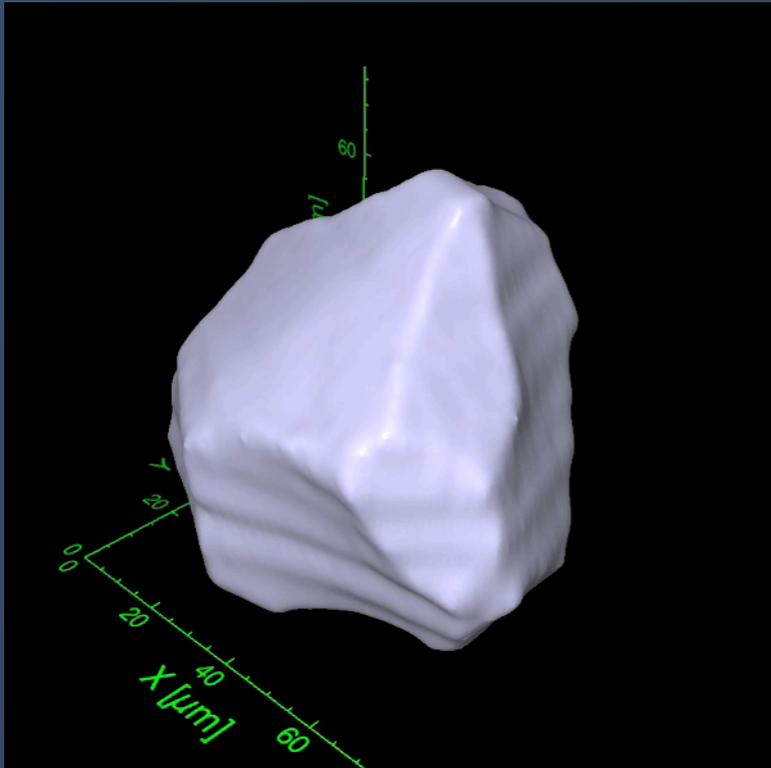


The segmented voxel structure makes it easy to count grain volumes, aspect ratios (and other higher order moments), nearest neighbor correlations, ...

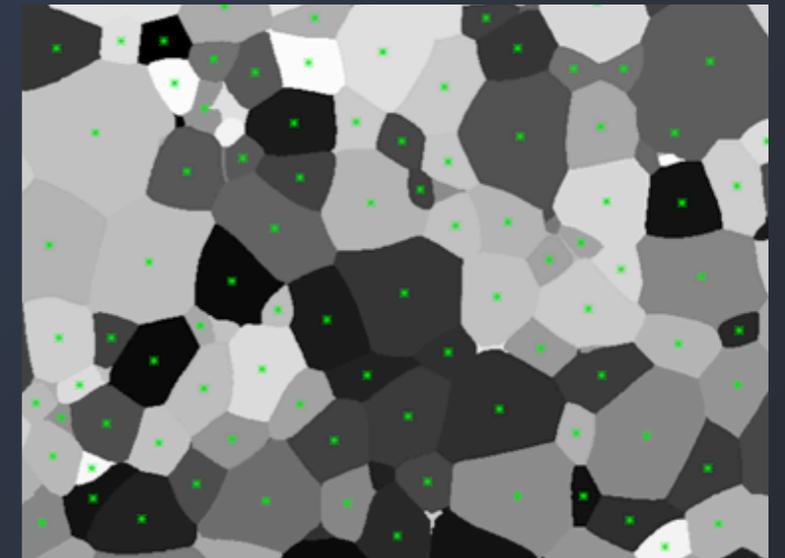


# Fine Scale Alignment

Find the center of area for each large grain in a section.  
Find the average deviation for each grain and adjust each section accordingly.



Section 0



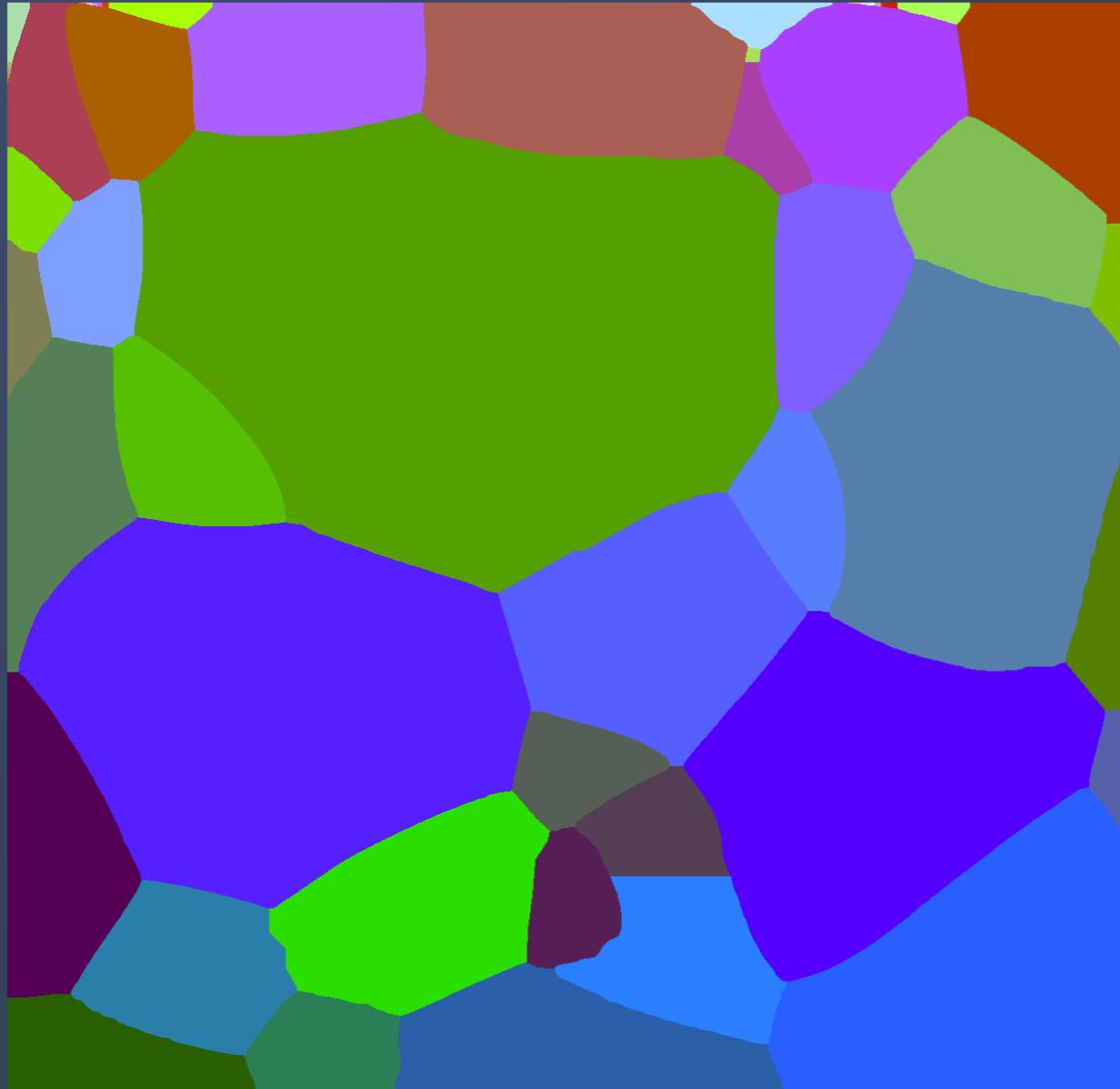
Section 1

$$|\Delta X_t| = 0.33\text{px}$$

$$|\Delta Y_t| = 0.43\text{px}$$

$$|\Delta\theta| = 0.03^\circ$$

# Calculating Interface Properties



Each region is described by discrete voxels.

The edges of the voxels do not represent the interfaces well:

- Interface normal/curvature is only comprised of multiples of  $90^\circ$
- Surface areas are over estimated



# Calculating Interface Properties

2D: replace interface pixels with discrete line segments.

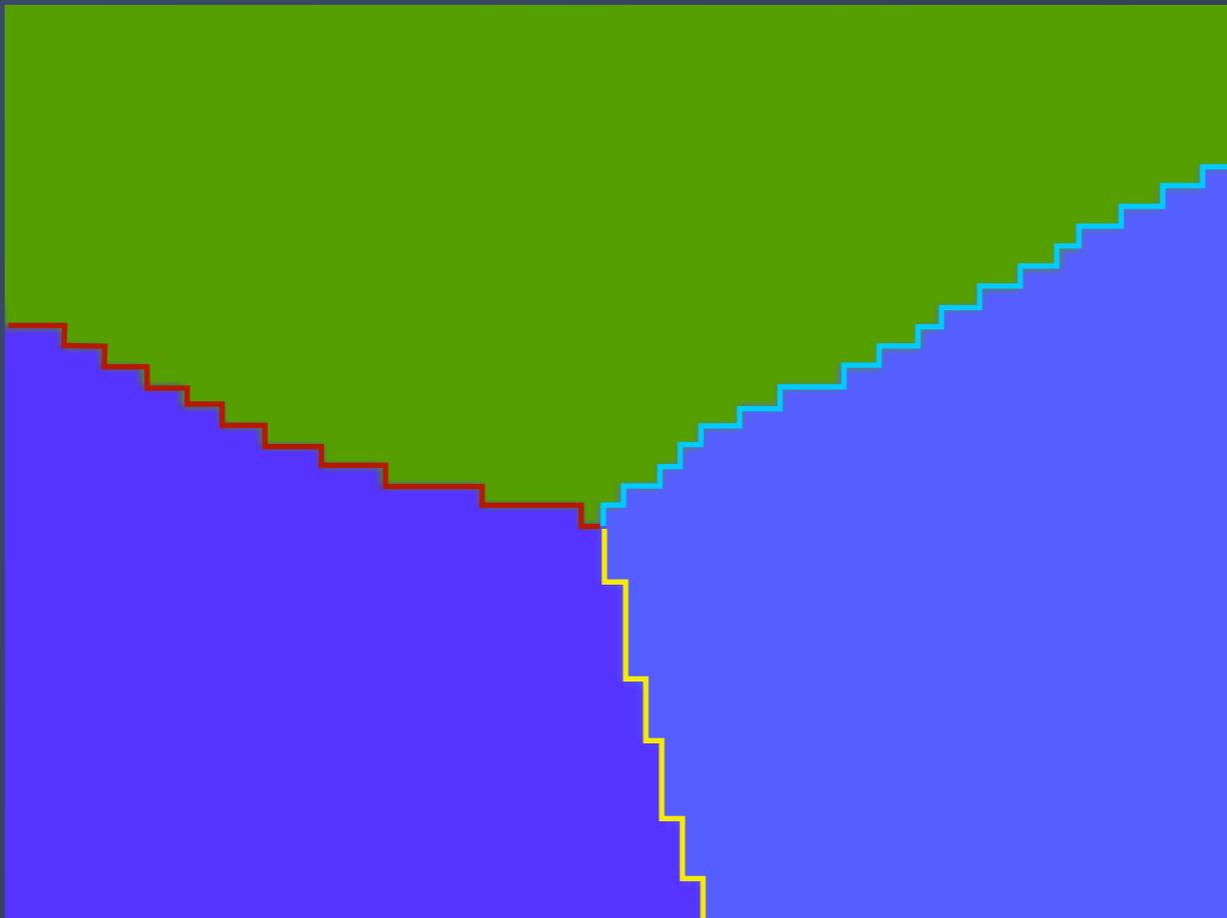
Each line segment is described by two vertex points.

list of vertex points:

$$v = [[x_1, y_1], [x_2, y_2], \dots, [x_N, y_N]]$$

connectivity list that list that lists the vertex points that make each line segment:

$$t = [[0, 3], [3, 1], [2, 101] \dots]$$



# Calculating Interface Properties



Remove pixel edges by a curvature based smoothing:

$$v_i^{\text{smooth}} = v_i + \frac{\lambda}{M} \sum_{j=0}^M (v_j - v_i)$$

Where  $j$  loops over the  $M$  connected vertices of vertex  $i$

Iterate this until the desired result is obtained. Alternatively  $\lambda$  can be a function of the vertex point (set  $\lambda = 0$  for triple junctions).

$$v_i^{\text{smooth}} = v_i + \frac{1}{M} \sum_{j=0}^M \lambda_i (v_j - v_i)$$



# Calculating Interface Properties

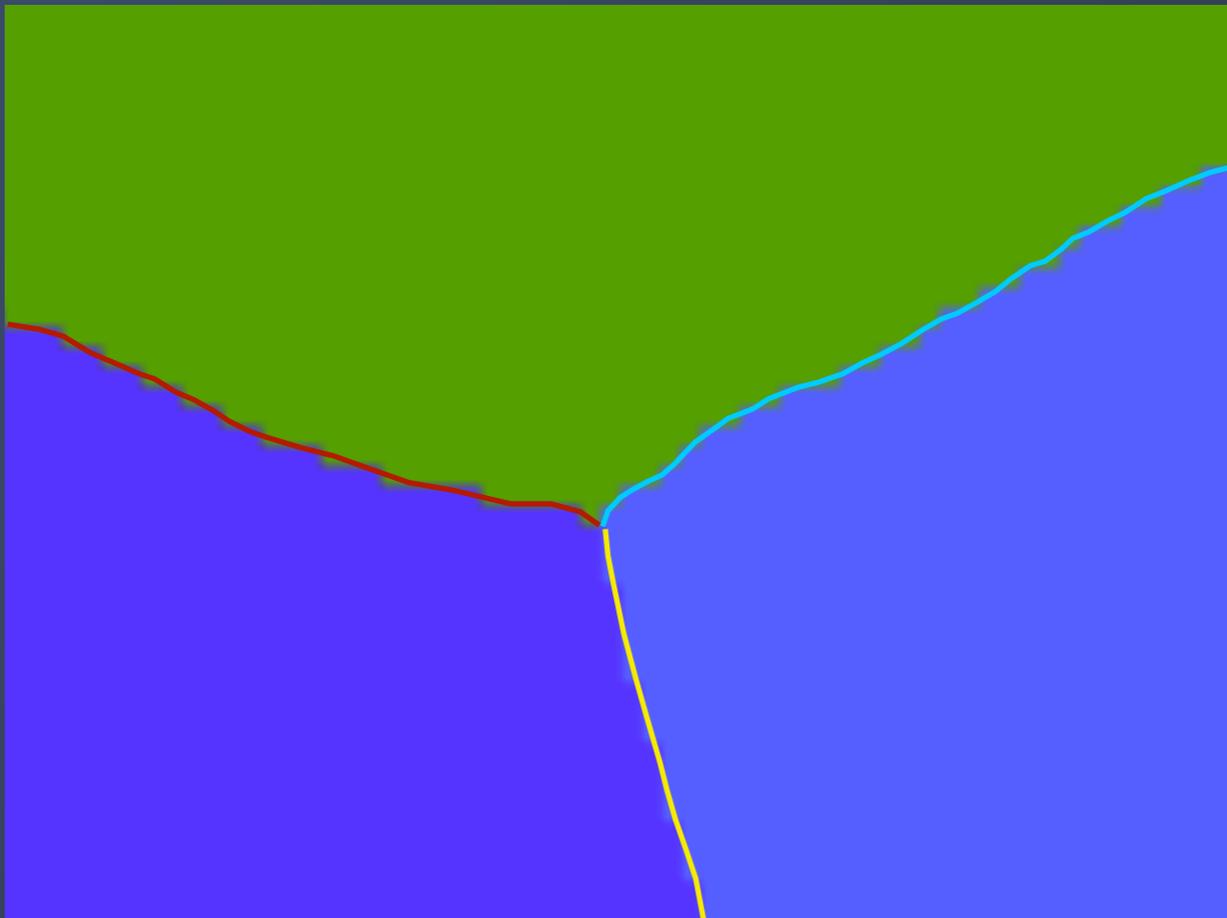
Remove pixel edges by a curvature based smoothing:

$$v_i^{\text{smooth}} = v_i + \frac{\lambda}{M} \sum_{j=0}^M (v_j - v_i)$$

Where  $j$  loops over the  $M$  connected vertices of vertex  $i$

Iterate this until the desired result is obtained. Alternatively  $\lambda$  can be a function of the vertex point (set  $\lambda = 0$  for triple junctions).

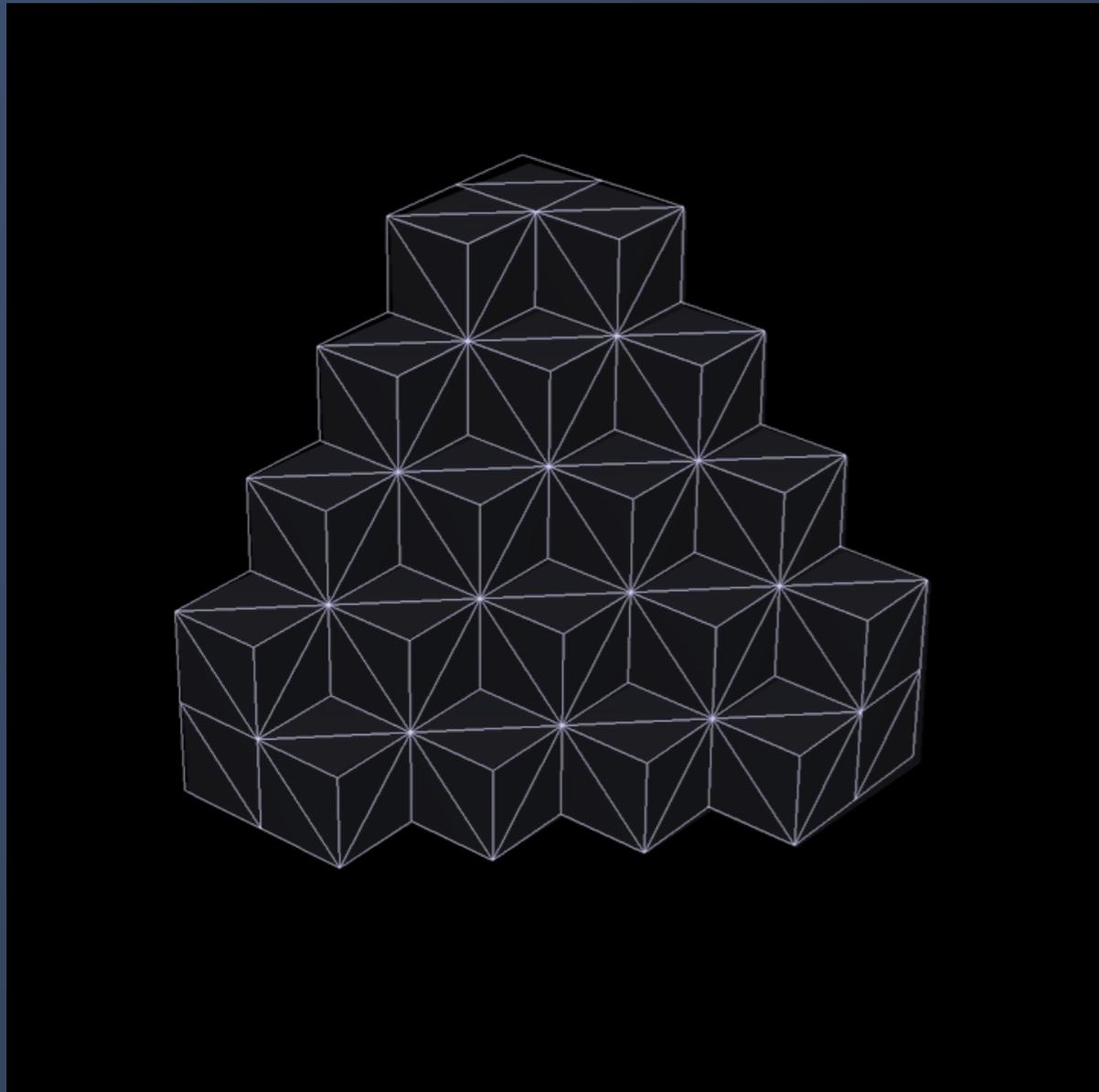
$$v_i^{\text{smooth}} = v_i + \frac{1}{M} \sum_{j=0}^M \lambda_i (v_j - v_i)$$



Provides a much more accurate estimate of the interface shape.



# From Pixels to Voxels: Lines to Triangles



In 3D:

Pixels  $\rightarrow$  Voxels

Line segments  $\rightarrow$  polygons (triangles)

List of vertex points:

$$v = [[x_1, y_1, z_1], [x_2, y_2, z_2], \dots [x_N, y_N, z_N]]$$

Connectivity list that lists the vertex points that make each triangle:

$$t = [[0, 3, 2], [2, 3, 1], [5, 2, 101] \dots]$$

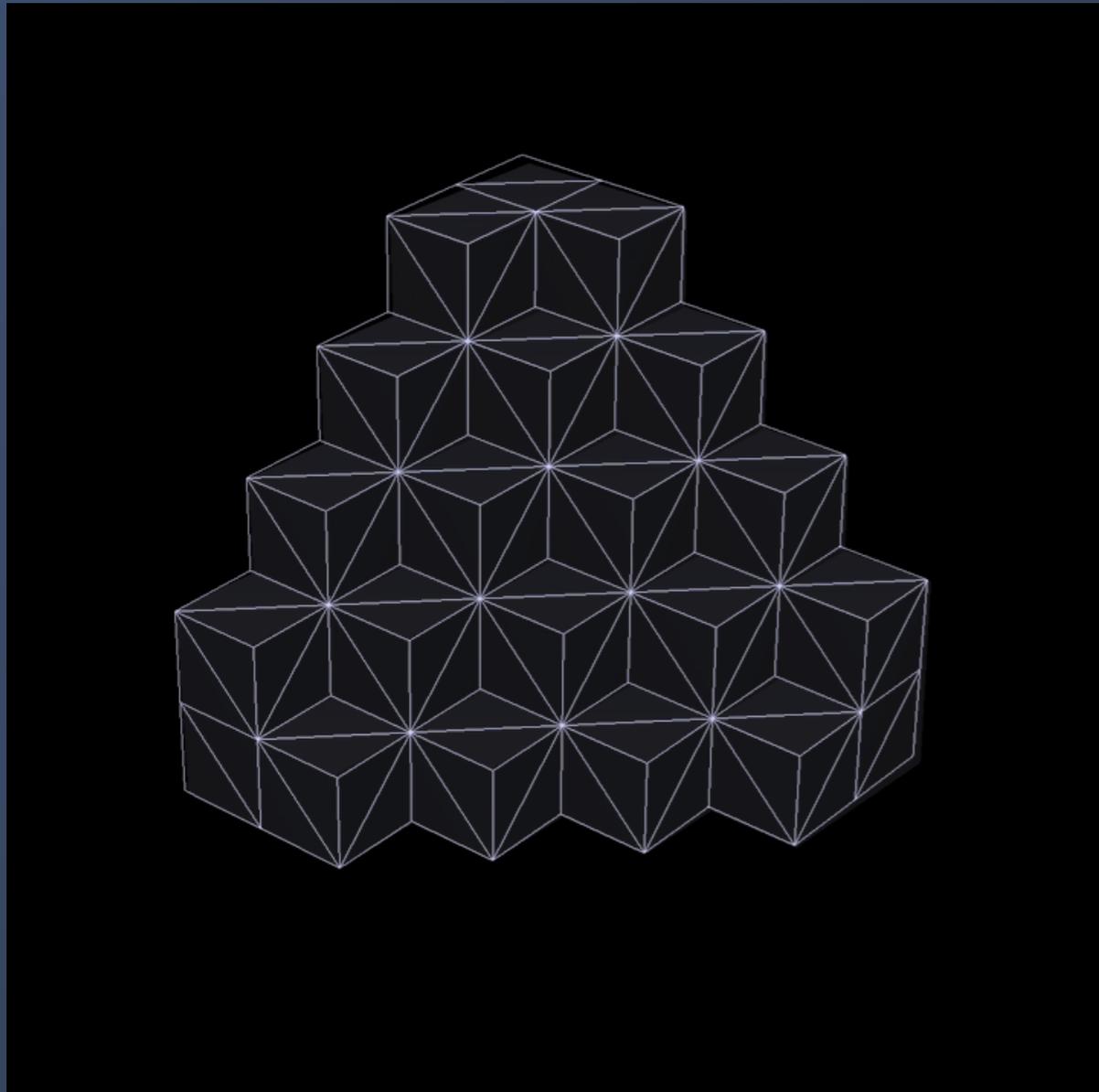
For a mix of polygon shapes:

$$p = [3, 5, 13, 6, 4, 13, 5, 7, 9, 3, 9, 7, 2\dots]$$

Same data structures used in most 3D vision and video games  $\rightarrow$  a lot of different resources for data visualization .



# Surface meshing



Not trivial in 3D structures to make surface mesh.

Two phases → Marching cubes algorithm. Only a small subset of possible arrangements for the types of interfaces encountered. Can accommodate smooth transition by examining the grayscale values on each side of the interface.

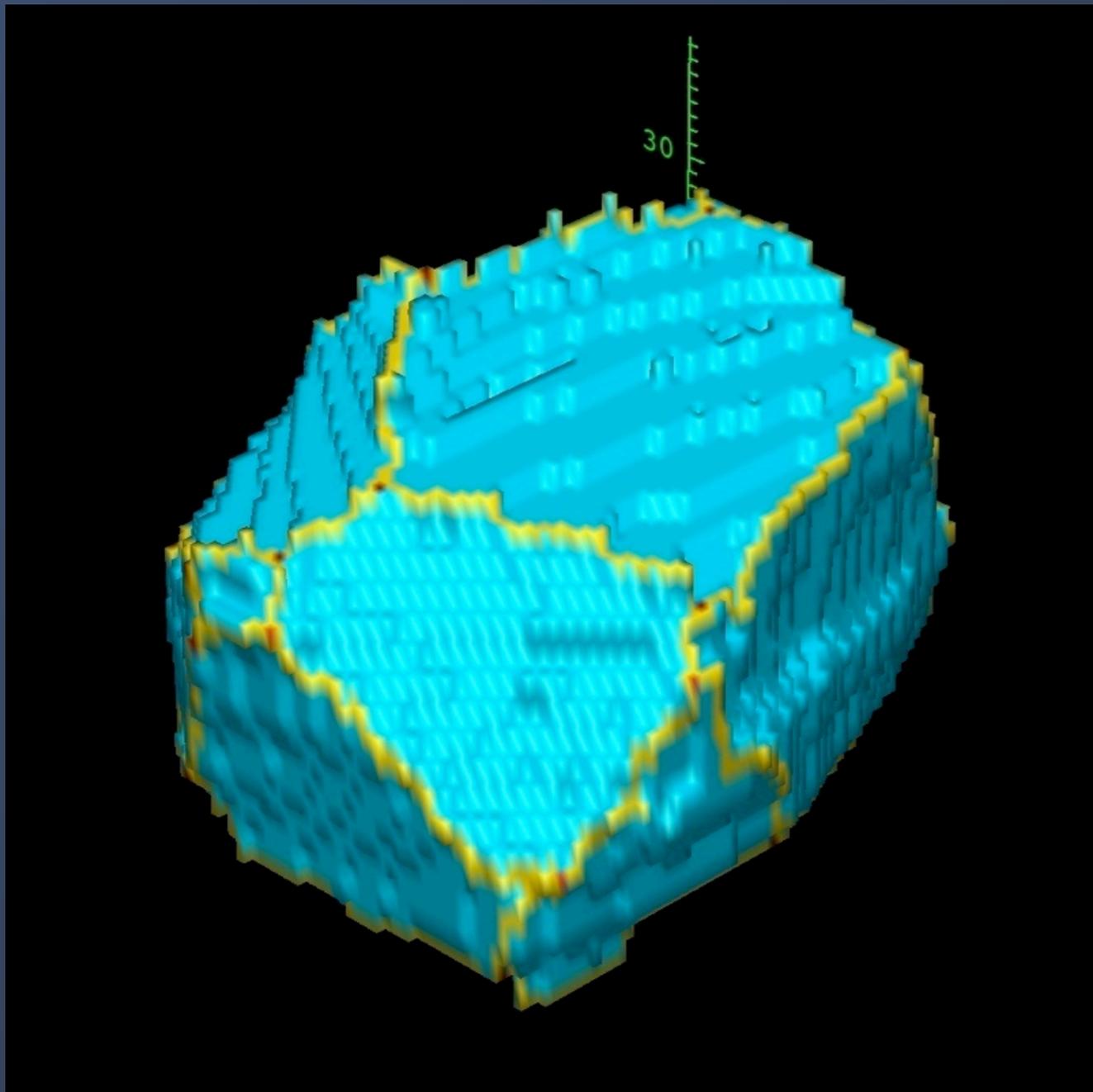
Multiple phases is much more complex.

Easy but incomplete: assign two triangles to the outside of each voxel face on the interface. Does not account for narrow objects well.

More complex: Multi materials marching cubes (Z. Wu and J. M. Sullivan Jr, “Multiple material marching cubes algorithm,” International Journal for Numerical Methods in Engineering, vol. 58, no. 2, pp. 189–207, 2003).  
(Found in DREAM.3D)



# Removing Voxel Artifacts



Regardless of surface mesh conversion method, with serial sections where voxel size is rectangular:  $z > x, y$ , we end up with stair case structure.

Remove pixel edges by a curvature based smoothing:

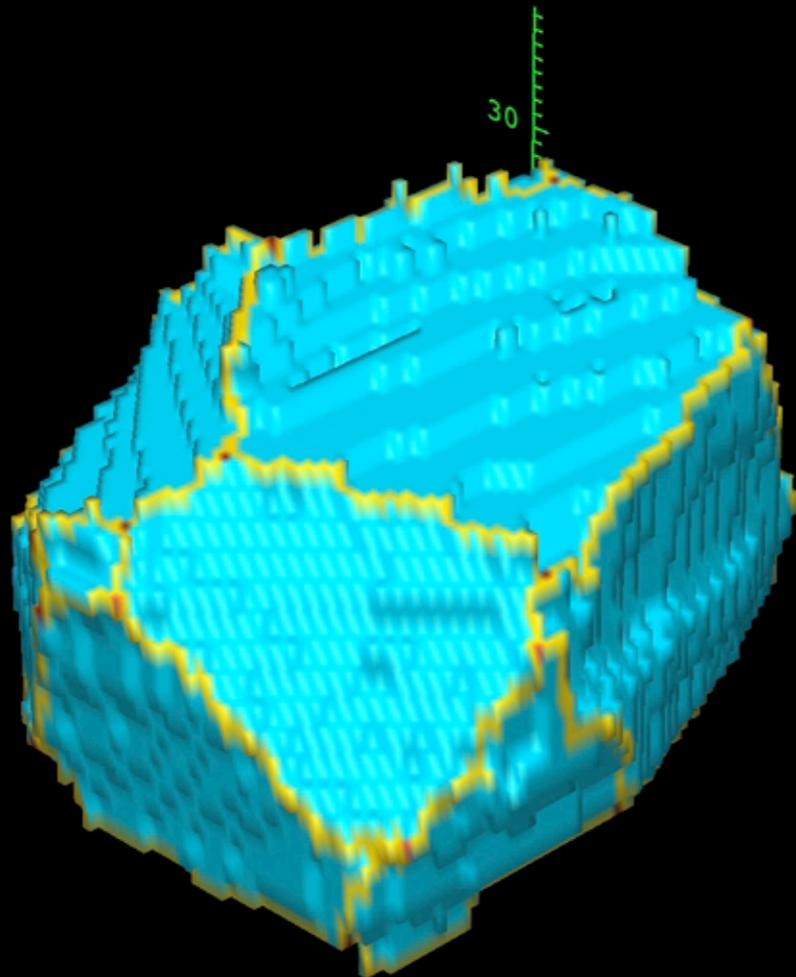
$$v_i^{\text{smooth}} = v_i + \frac{1}{M} \sum_{j=0}^M \lambda_i (v_j - v_i)$$

Where  $j$  loops over the  $M$  connected vertices of vertex  $i$

Iterate this until the desired result is obtained.



# Removing Voxel Artifacts



Remove voxel steps by a curvature based smoothing:

$$v_i^{\text{smooth}} = v_i + \frac{1}{M} \sum_{j=0}^M \lambda_i (v_j - v_i)$$

Where  $j$  loops over the  $M$  connected vertices of vertex  $I$

Iterate this until the desired result is obtained.

Here I set:

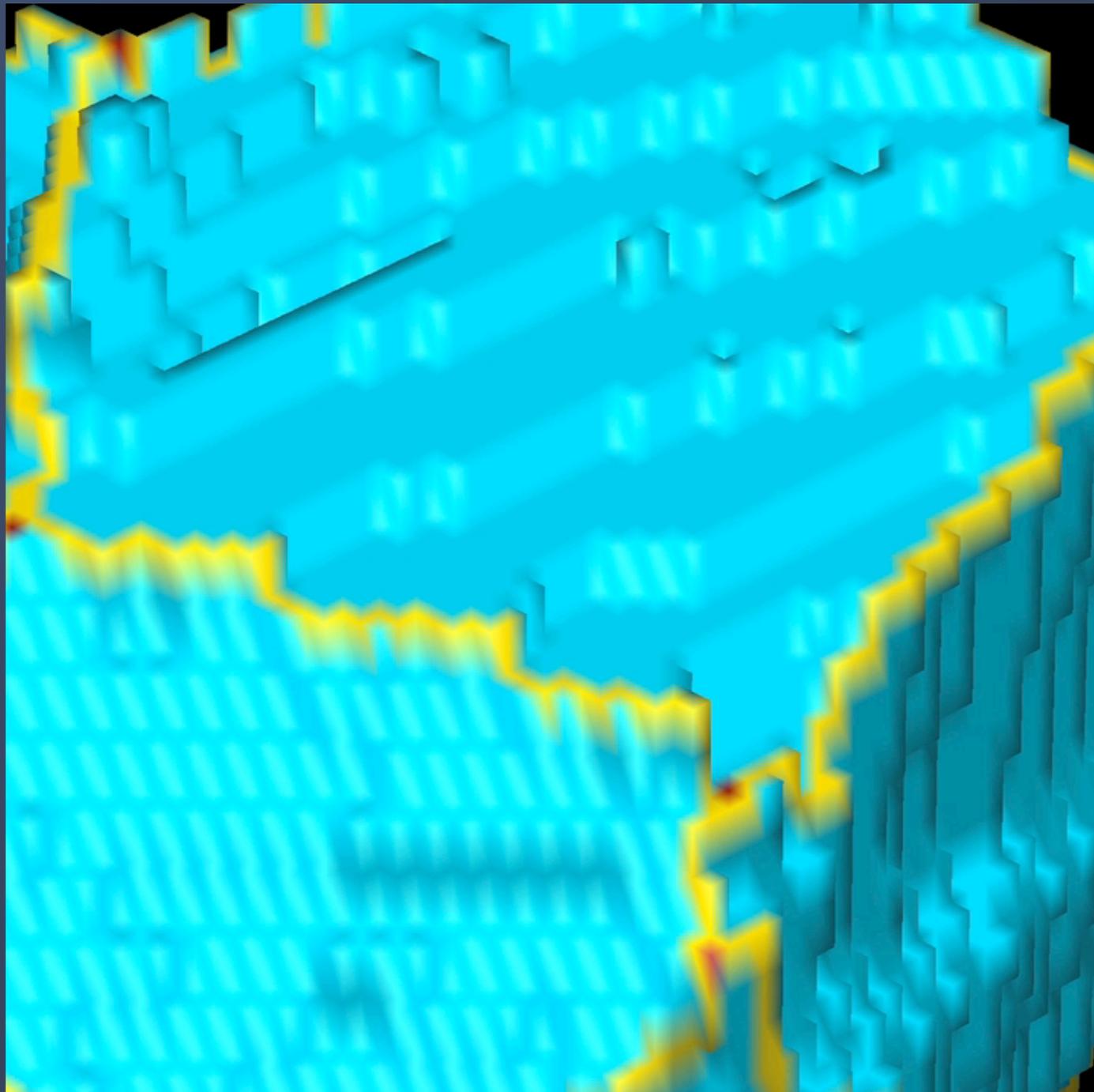
$$\lambda_{\text{faces}} = 0.5$$

$$\lambda_{\text{triple lines, quad points}} = 0.01$$

Iterations = 100



# Removing Voxel Artifacts



Remove voxel steps by a curvature based smoothing:

$$v_i^{\text{smooth}} = v_i + \frac{1}{M} \sum_{j=0}^M \lambda_i (v_j - v_i)$$

Where  $j$  loops over the  $M$  connected vertices of vertex  $I$

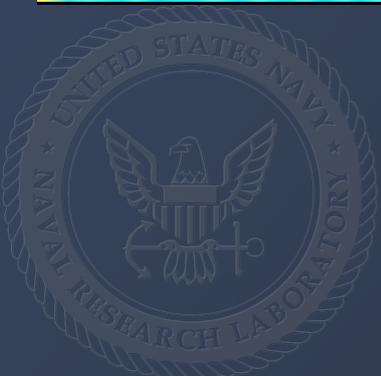
Iterate this until the desired result is obtained.

Here I set:

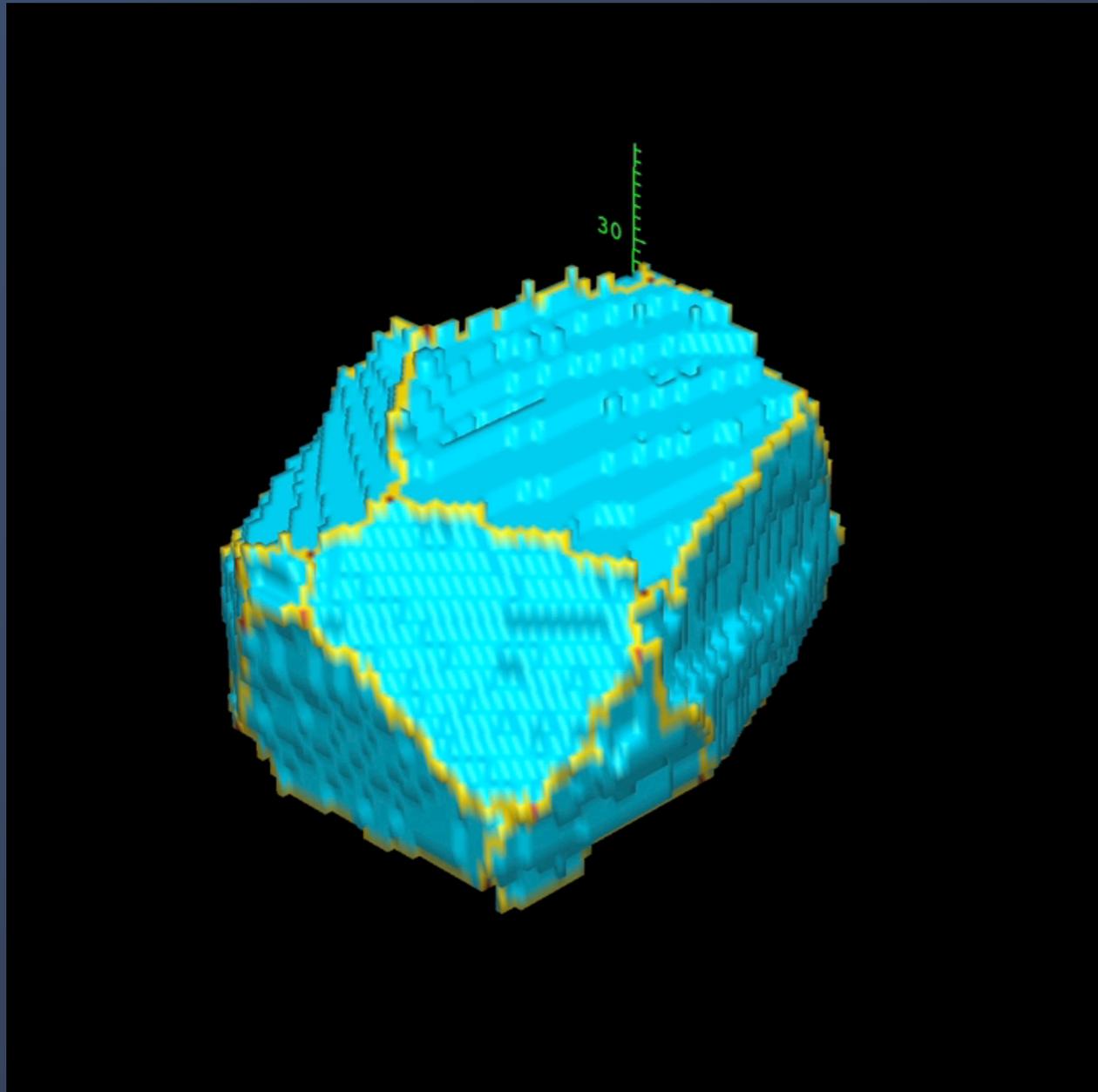
$$\lambda_{\text{faces}} = 0.5$$

$$\lambda_{\text{triple lines, quad points}} = 0.01$$

Iterations = 100



# Removing Voxel Artifacts



First isolate the triple lines

$$\lambda_{\text{triline}} = 0.1$$

Iterations = 50

Full mesh:

$$\lambda_{\text{faces}} = 0.5$$

+

10% alternating random faces = 0

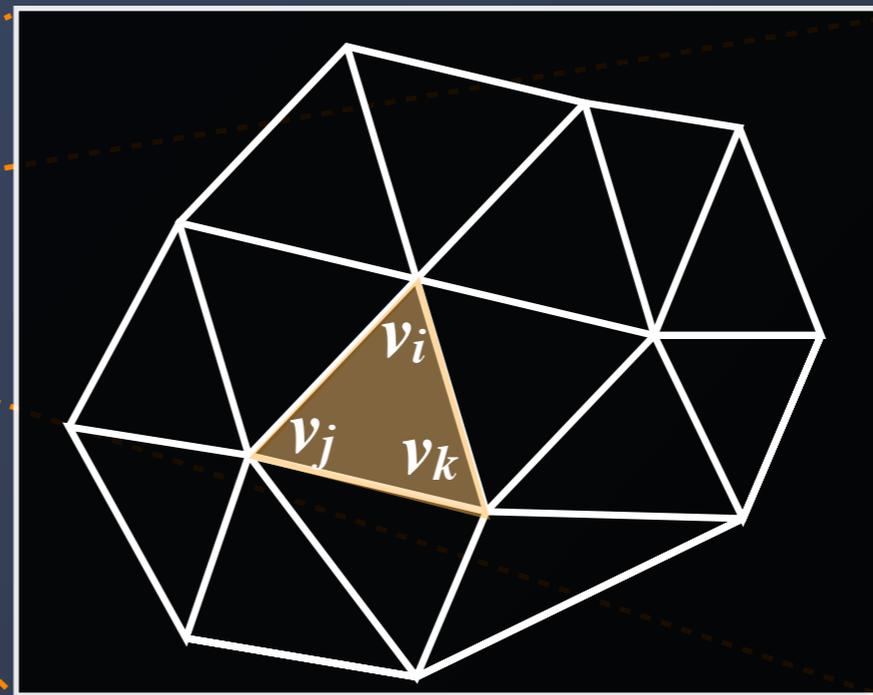
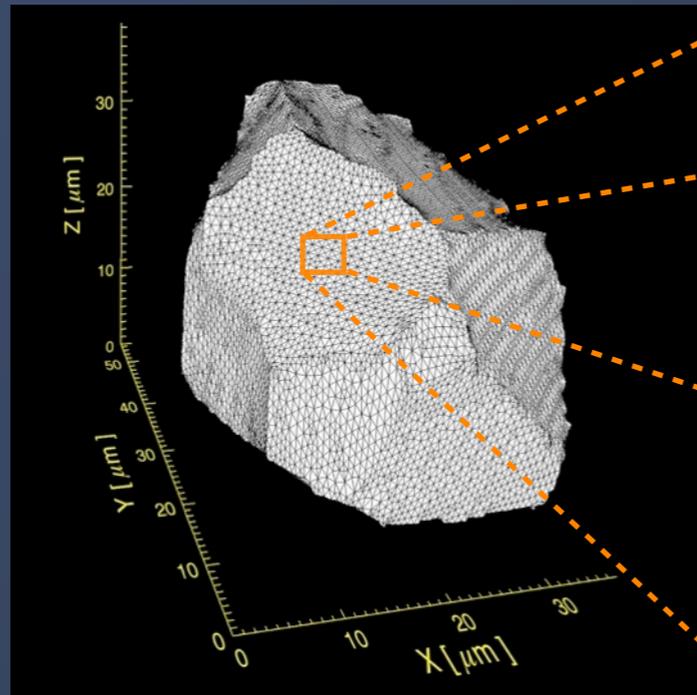
$$\lambda_{\text{triple lines, quad points}} = 0.001$$

Iterations = 50

The random points slow the large scale movement of the interfaces without introducing a regular length scale.



# Determination of Interface Normal



N = 0 ring

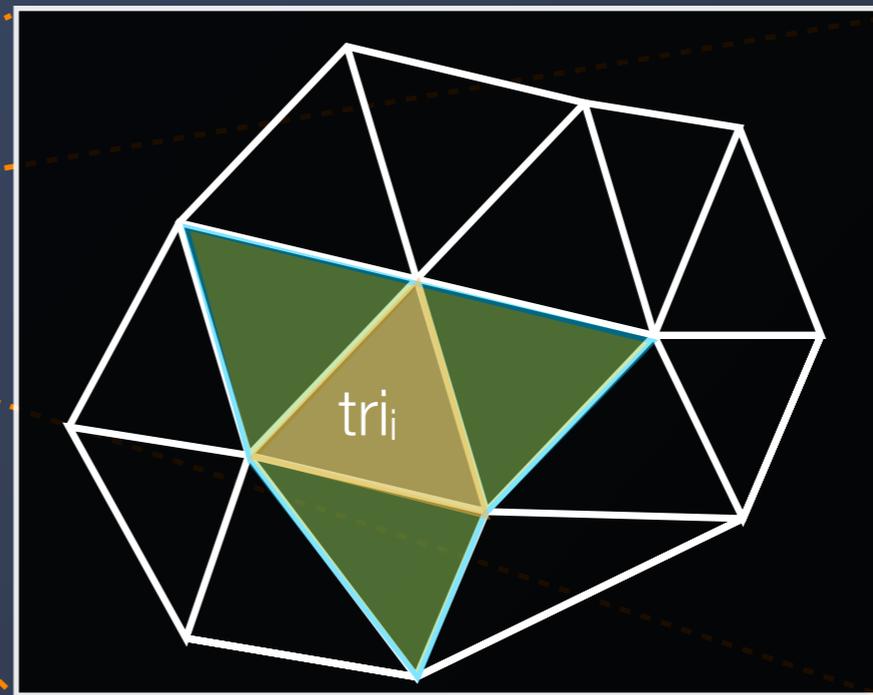
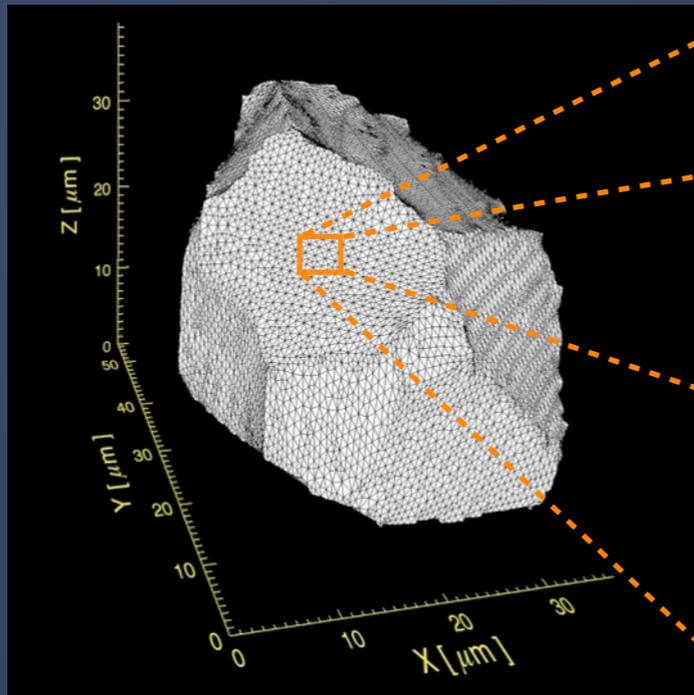
From the properties of a cross product:

$$\mathcal{A}_{\text{tri}} = \frac{1}{2} |(v_j - v_i) \times (v_k - v_i)|$$

$$\mathbf{n}_{\text{tri}} = \frac{1}{2\mathcal{A}_{\text{tri}}} (v_j - v_i) \times (v_k - v_i)$$



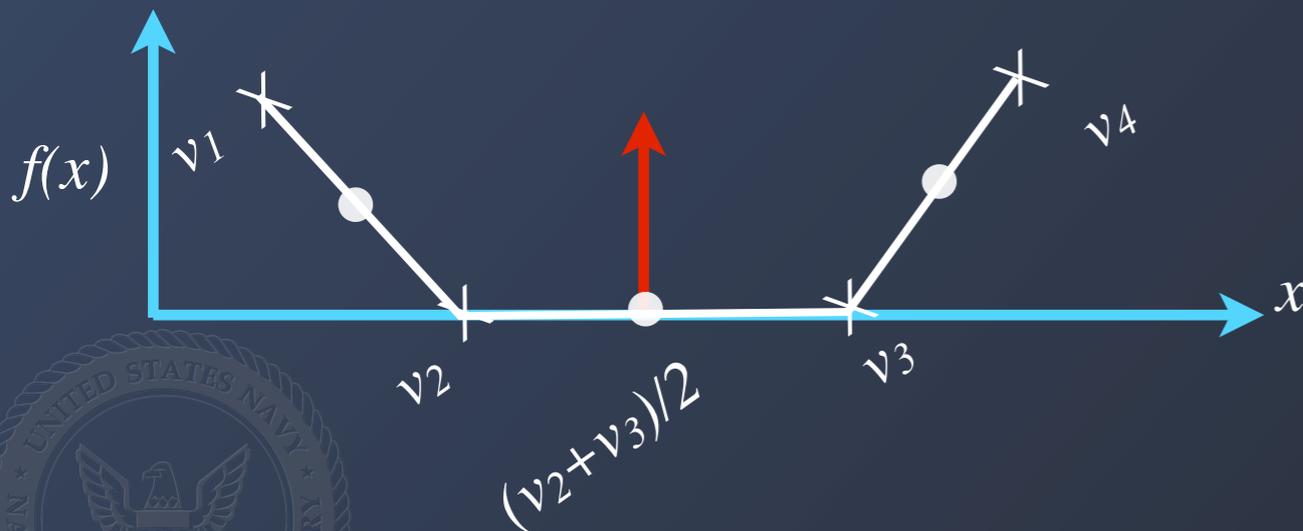
# Determination of Interface Curvature



N = 1 ring

For each triangle set up a local coordinate system, with +z along the triangle normal. Collect the N-ring nearest neighbor triangles and normals, and express their locations in the local coordinate system. 2D (line segments):

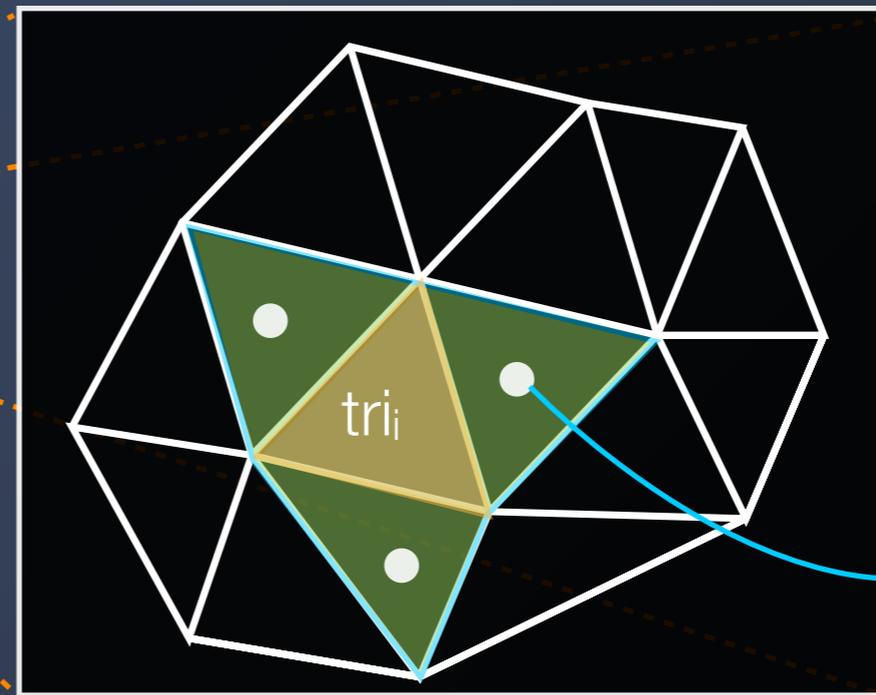
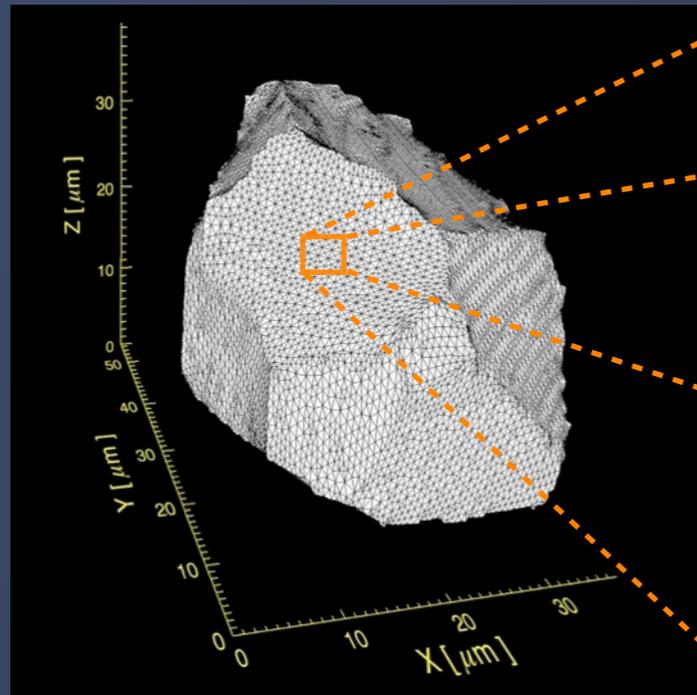
Least squares fit of the “elevation” of the neighboring segments with a function that assumes constant curvature.



$$f(x) = \frac{A}{2}x^2$$



# Determination of Interface Curvature



N = 1 ring

$$l = (v_1 + v_2 + v_3)/3$$

The curvature is determined by fitting the positions of the N ring neighborhood around each vertex point to a cubic surface (Goldfeather & Interrante 2004):

$$f(x, y) = \frac{A}{2}x^2 + Bxy + \frac{C}{2}y^2$$

The curvature tensor is then given by:

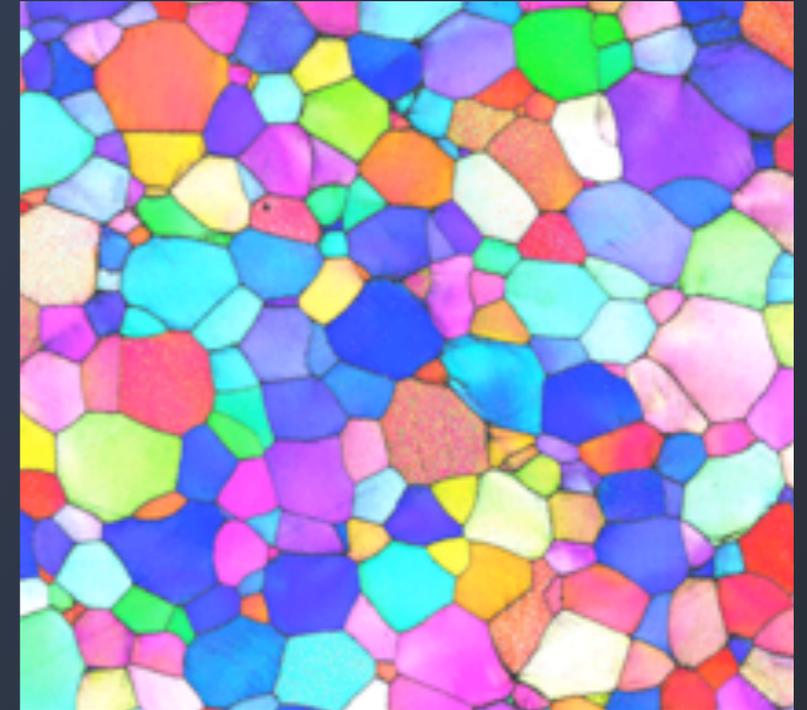
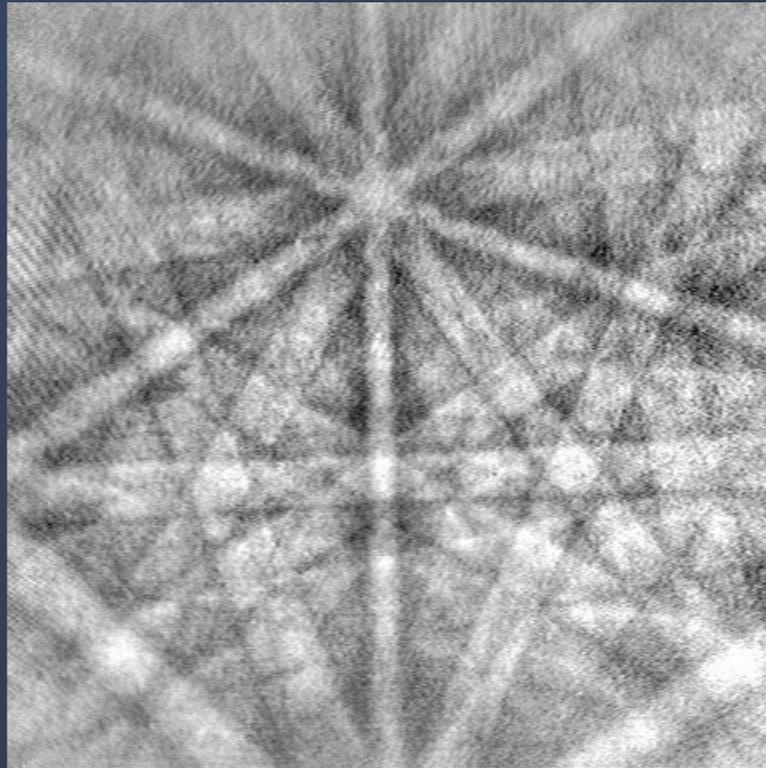
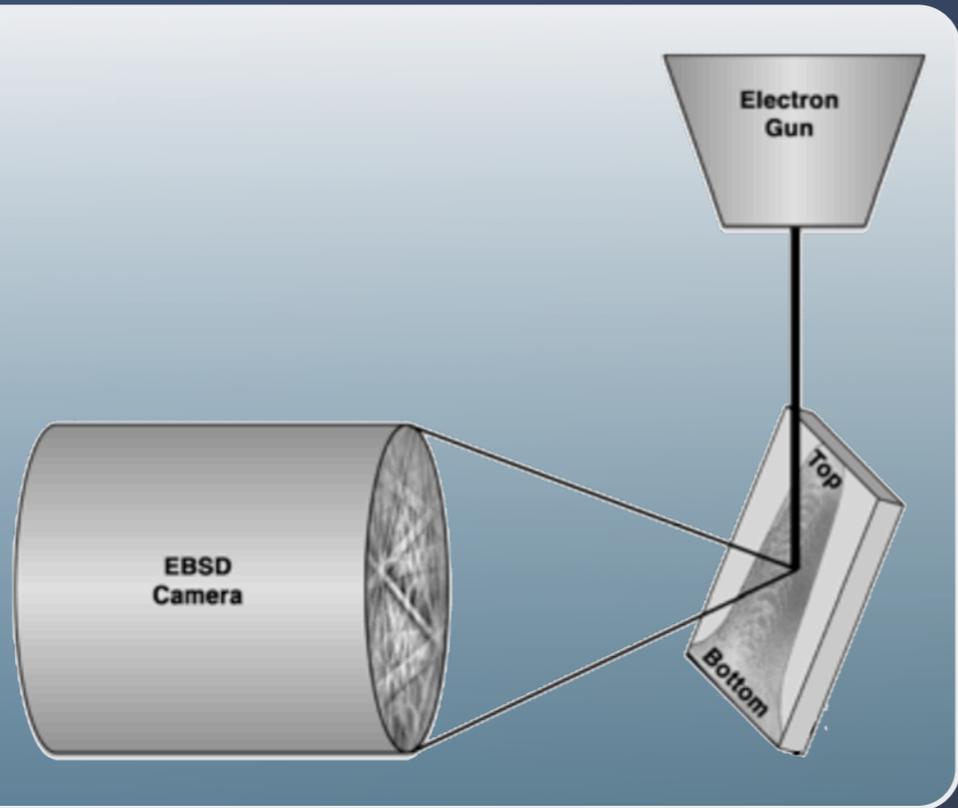
$$W = \begin{pmatrix} A & B \\ B & C \end{pmatrix}$$

This eigenvalue/vector problem is then solved for the principle curvatures  $\kappa_1$  and  $\kappa_2$

This can be improved by fitting not just triangle locations, but also triangle normals (3rd degree polynomial fit)

This method is considerable slower than others, but also more accurate. We use a N=3 ring neighborhood not allowing the neighborhood to transverse a triple line.

# EBSD - One slide introduction



The electron beam impinges on the sample surface, with some orientations satisfying a Bragg condition, producing a Kichuchi pattern.

A Kichuchi pattern is detected and automatically indexed at each point on the surface, producing an image based on crystal orientation.

The newest systems are capable of achieving  $> 400$  points/sec.

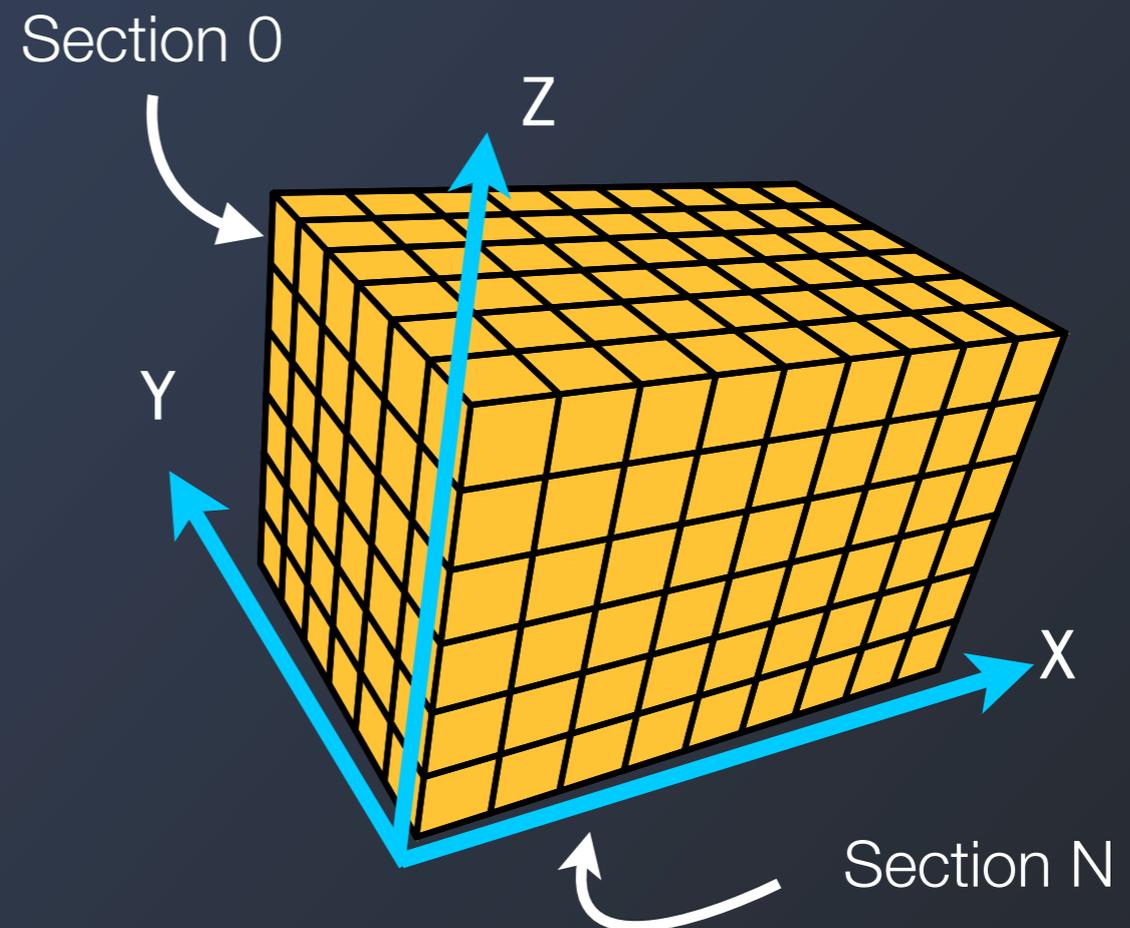
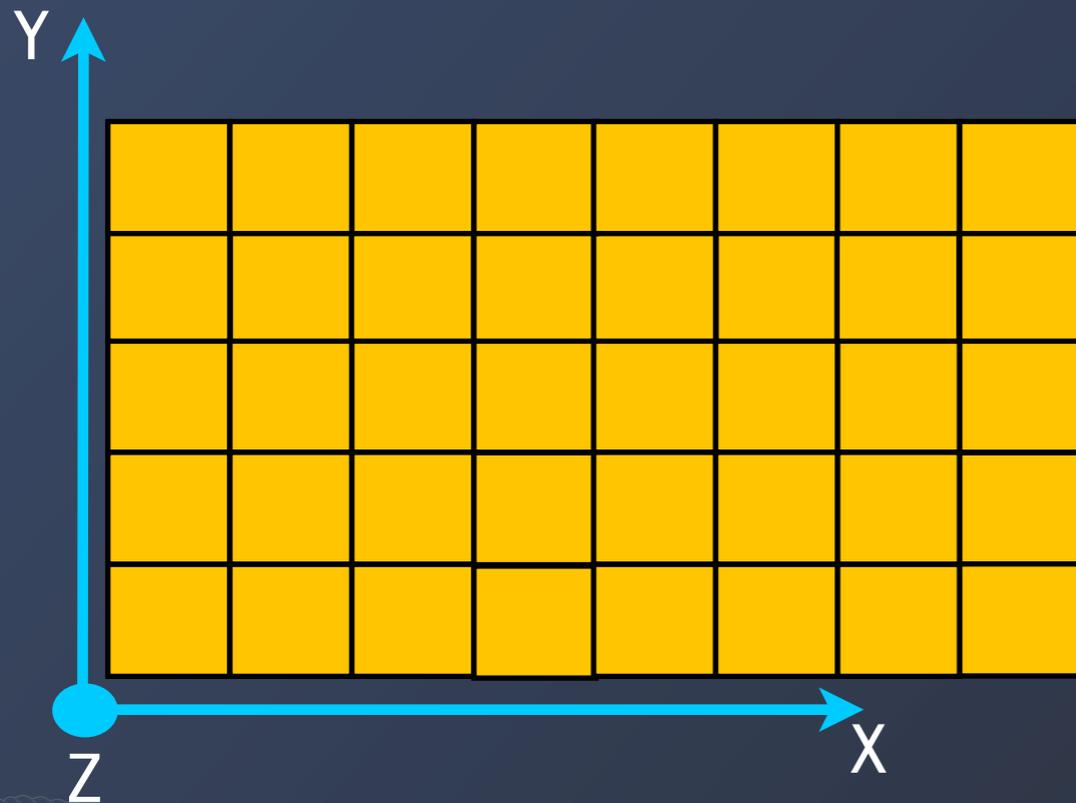


# Image coordinate systems

# Coordinate Systems

“Intuitive” coordinate system for serial section materials analysis:

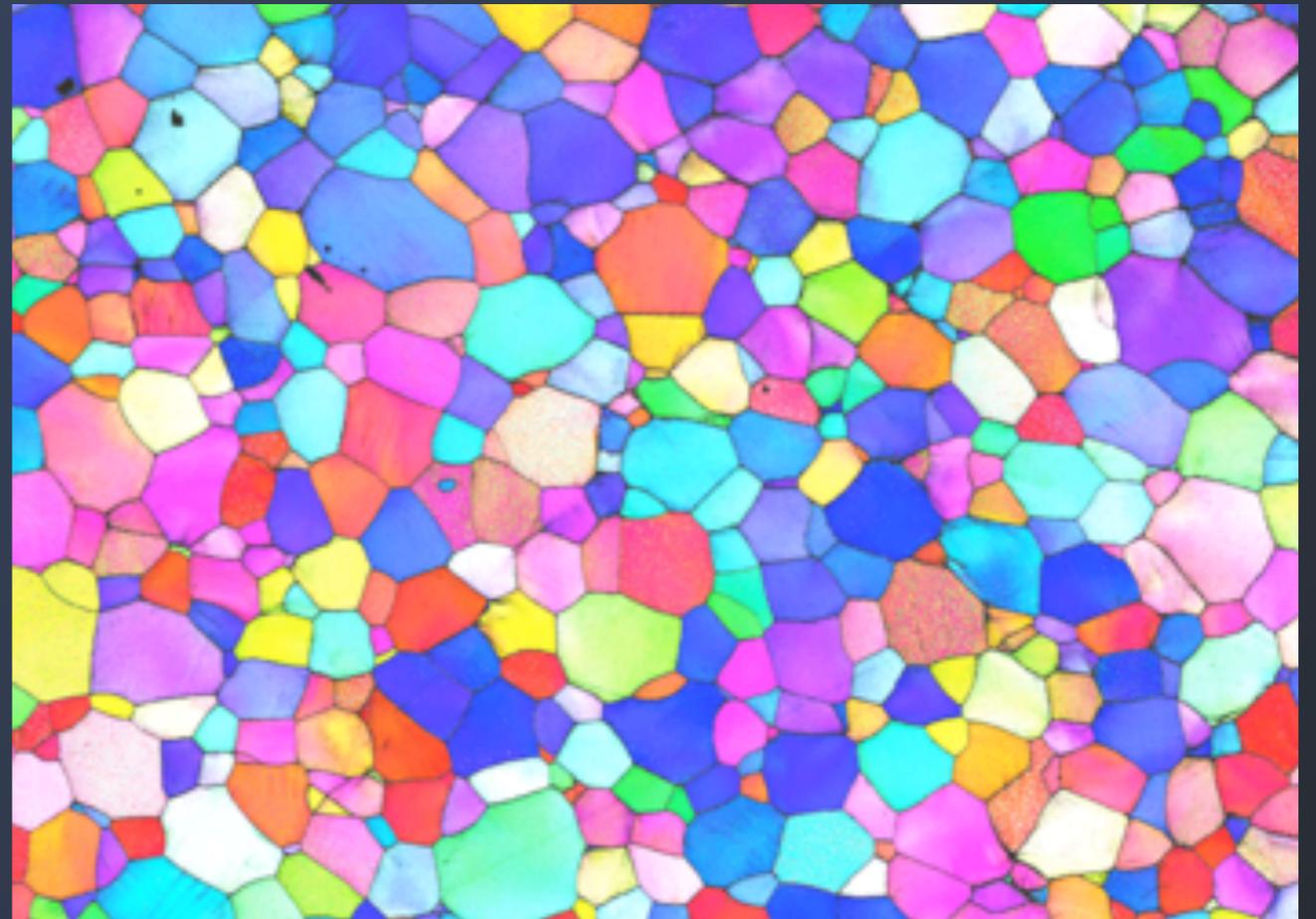
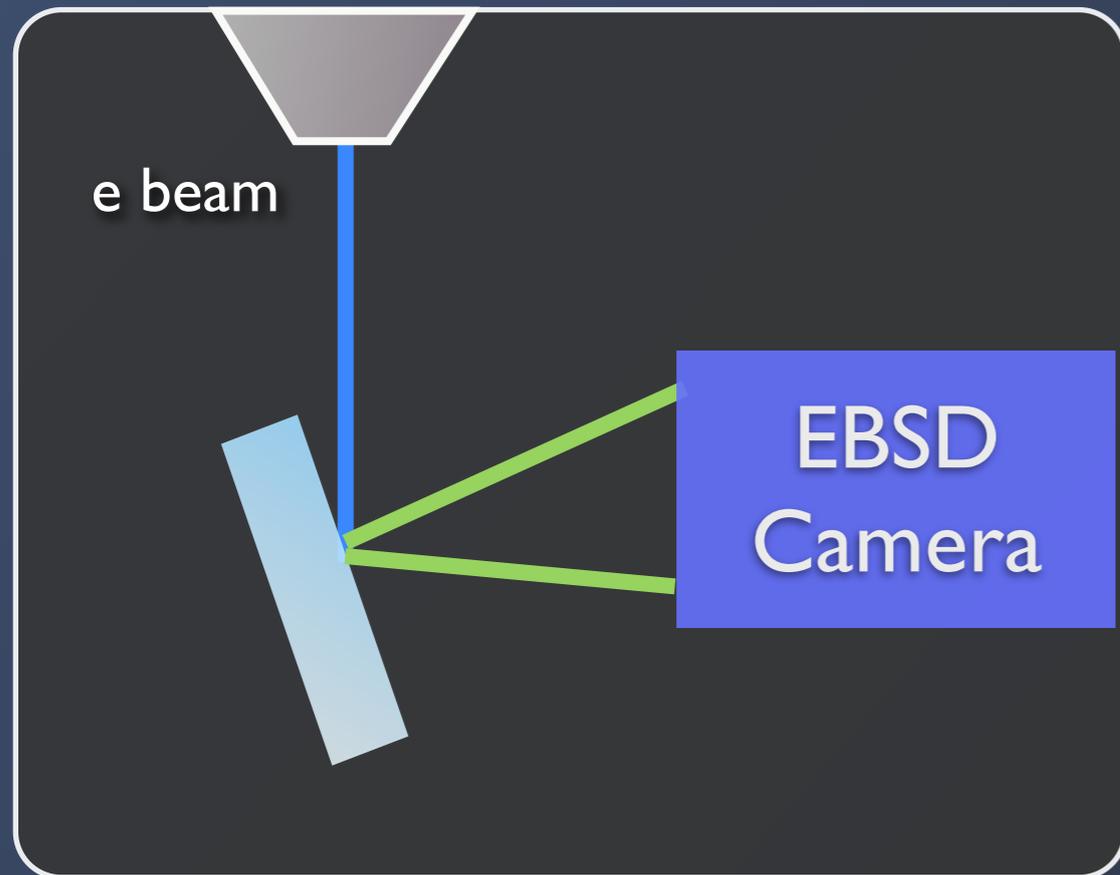
- x,y,z are orthogonal and positive.
- x,y describe a section
- z points up
- when stored in computer memory, x increases fastest, y, then z



\*\*Check your optical microscope that it creates the image you expect:  
Many inverted optical microscopes produce a flipped image



# TSL\* EBSD Coordinate Systems

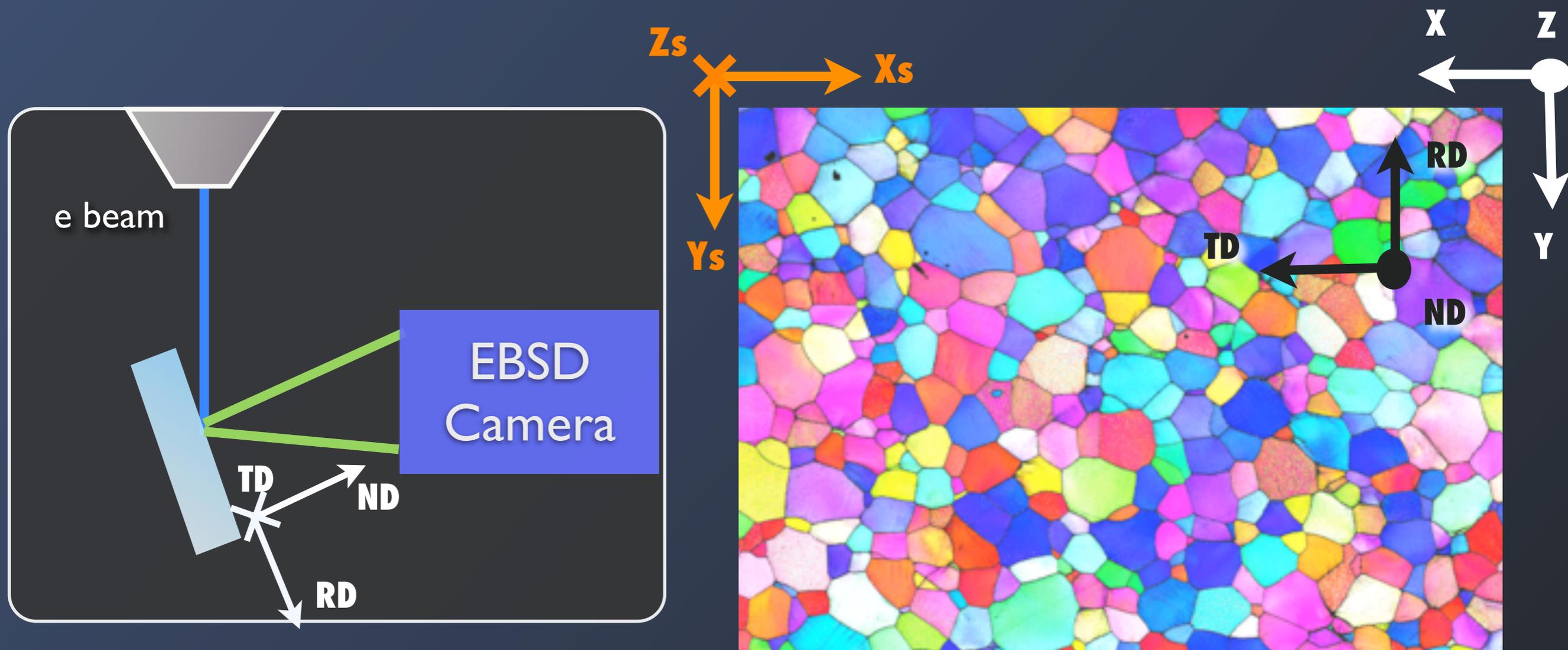


TSL/OIM (as well as Oxford/HKL) uses two different reference coordinate systems that does not align with the “intuitive” system.

\* Applies to .ang files from OIM version  $\leq 6$ ; TEAM unknown. This does not apply to within OIM software v6 or later.



# TSL\* EBSD Coordinate Systems



TSL/OIM (as well as Oxford/HKL) uses two different reference coordinate systems that does not align with the “intuitive” system.

\* Applies to .ang files from OIM version  $\leq 6$ ; TEAM unknown. This does not apply to within OIM software v6 or later.



# TSL EBSD Coordinate Systems



1.) Relabel  $X_s$  to increase going to the left,  $Z_s$  to coming out of the screen.

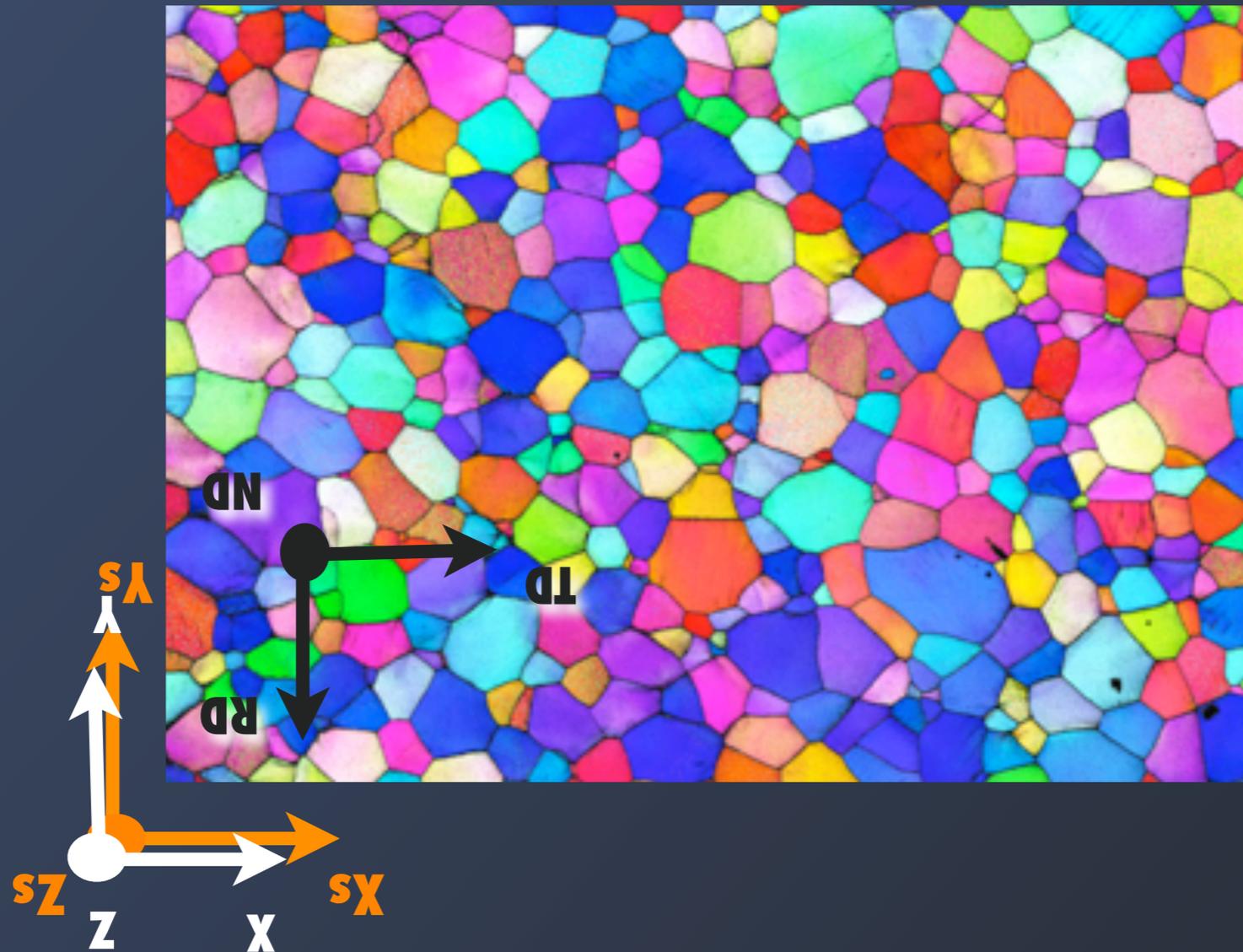
(Rotate  $X_s$ - $Y_s$ - $Z_s$  coordinate system about  $Y_s$ )

1a.) Make the origin so that all points are positive

2.) Place this in the “intuitive” system.



# TSL EBSD Coordinate Systems

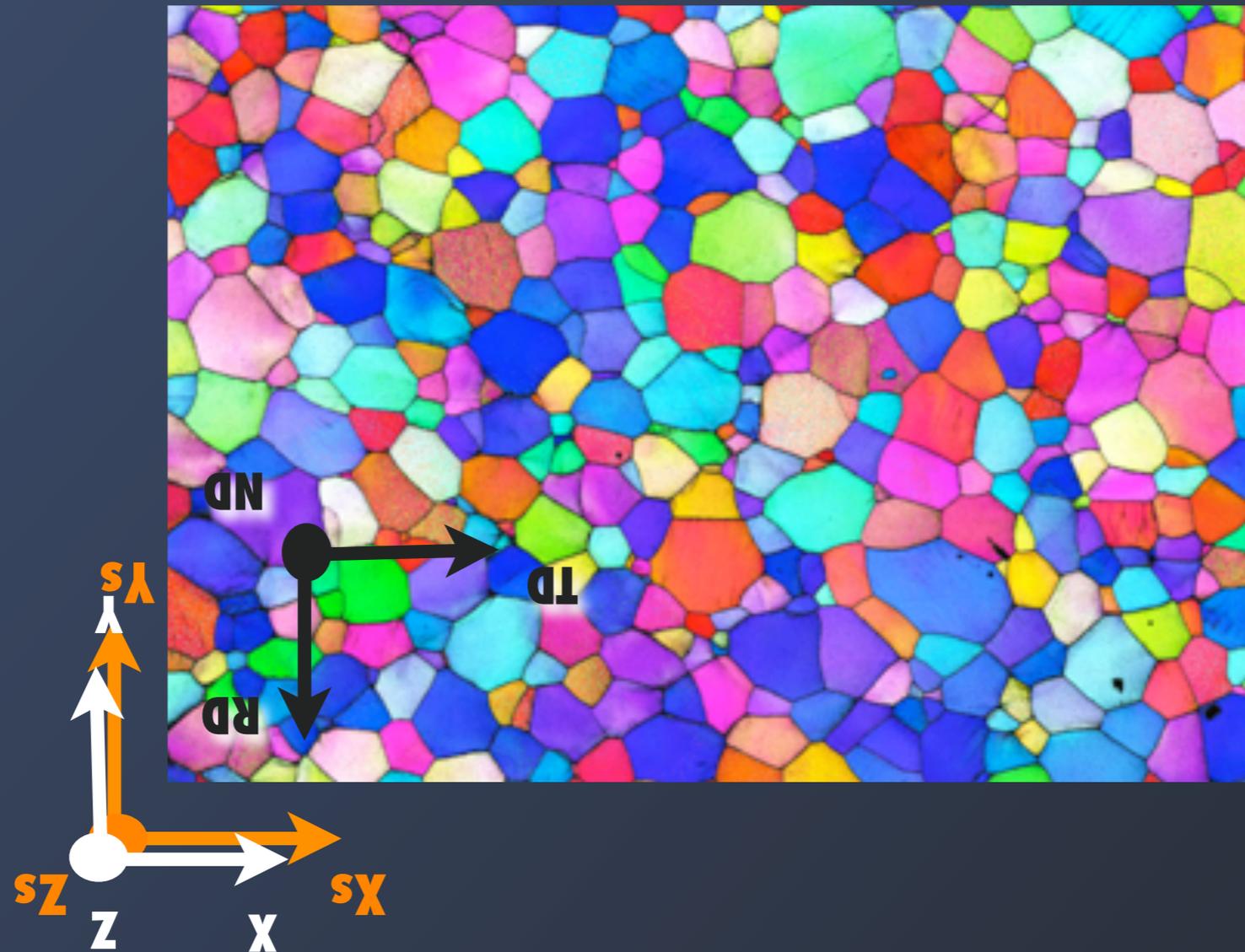


1.) Relabel Xs to increase going to the left, Zs to coming out of the screen.  
(Rotate Xs-Ys-Zs coordinate system about Ys)

1a.) Make the origin so that all points are positive

2.) Place this in the “intuitive” system.

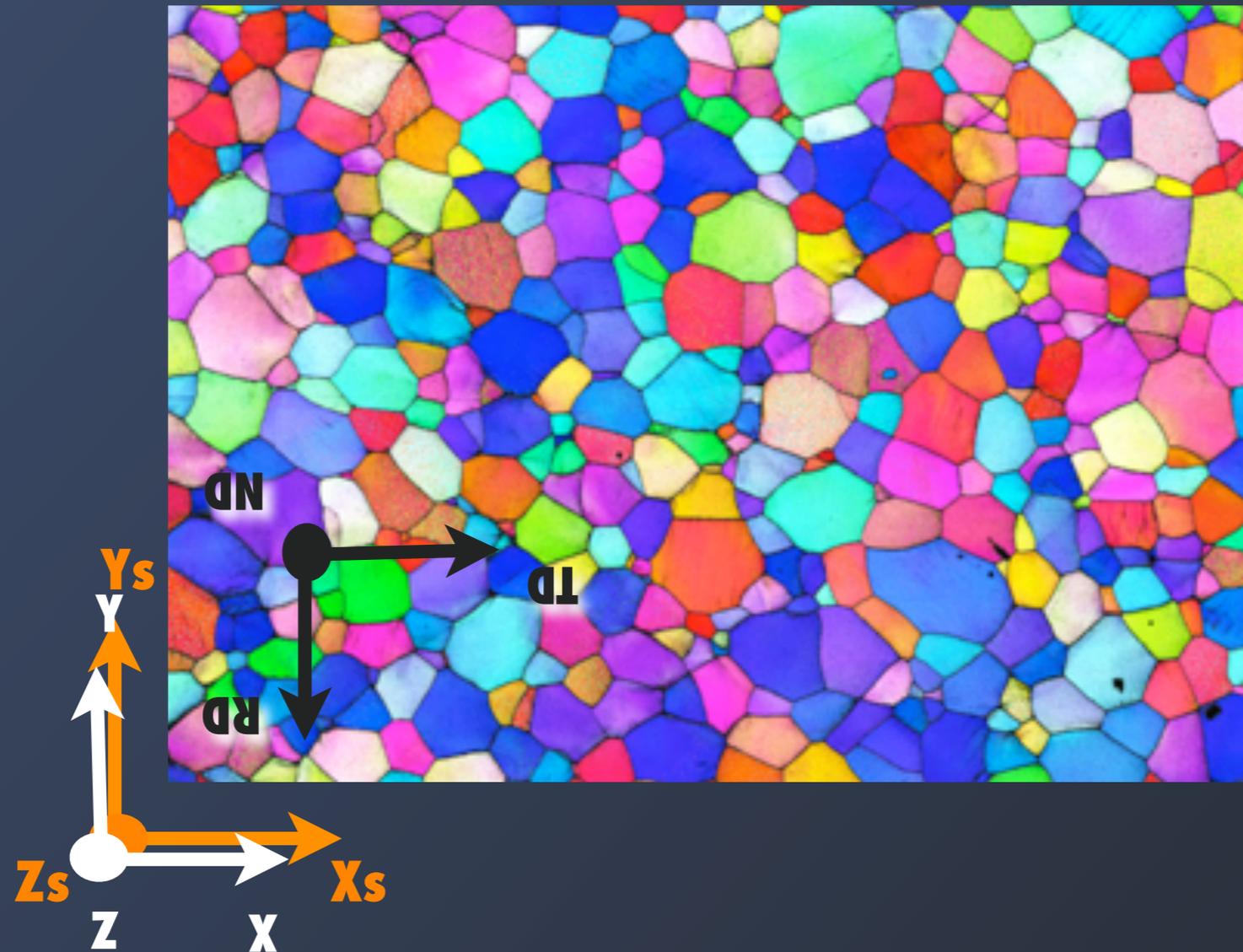
# TSL EBSD Coordinate Systems



- 1.) Transform the scanned data points to line up with the sample coordinate
  - 1a.) Relabel Xs to increase going to the left, Zs to coming out of the screen.  
(Rotate Xs-Ys-Zs coordinate system about Ys)
  - 1b.) Make the origin so that all points are positive

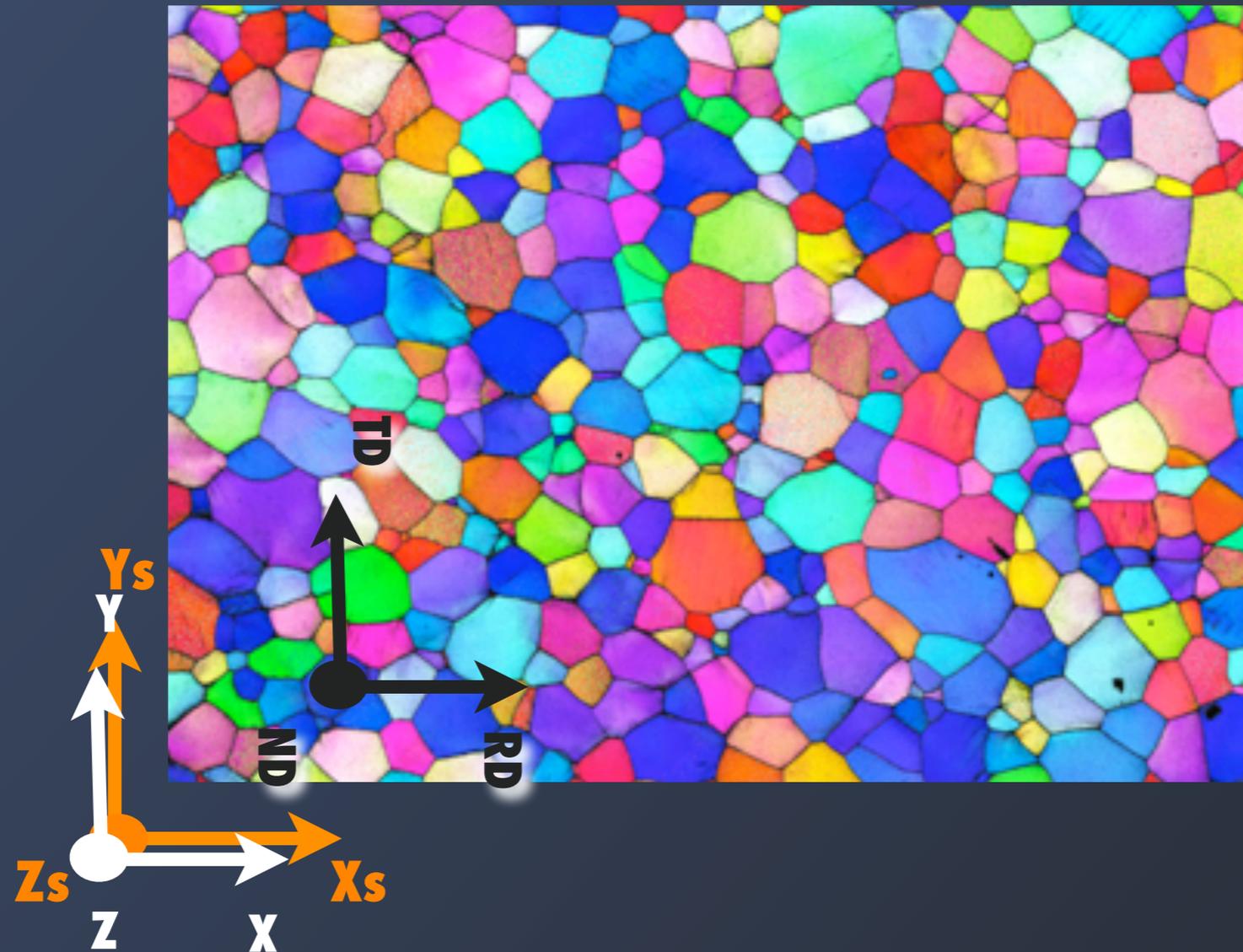


# TSL EBSD Coordinate Systems



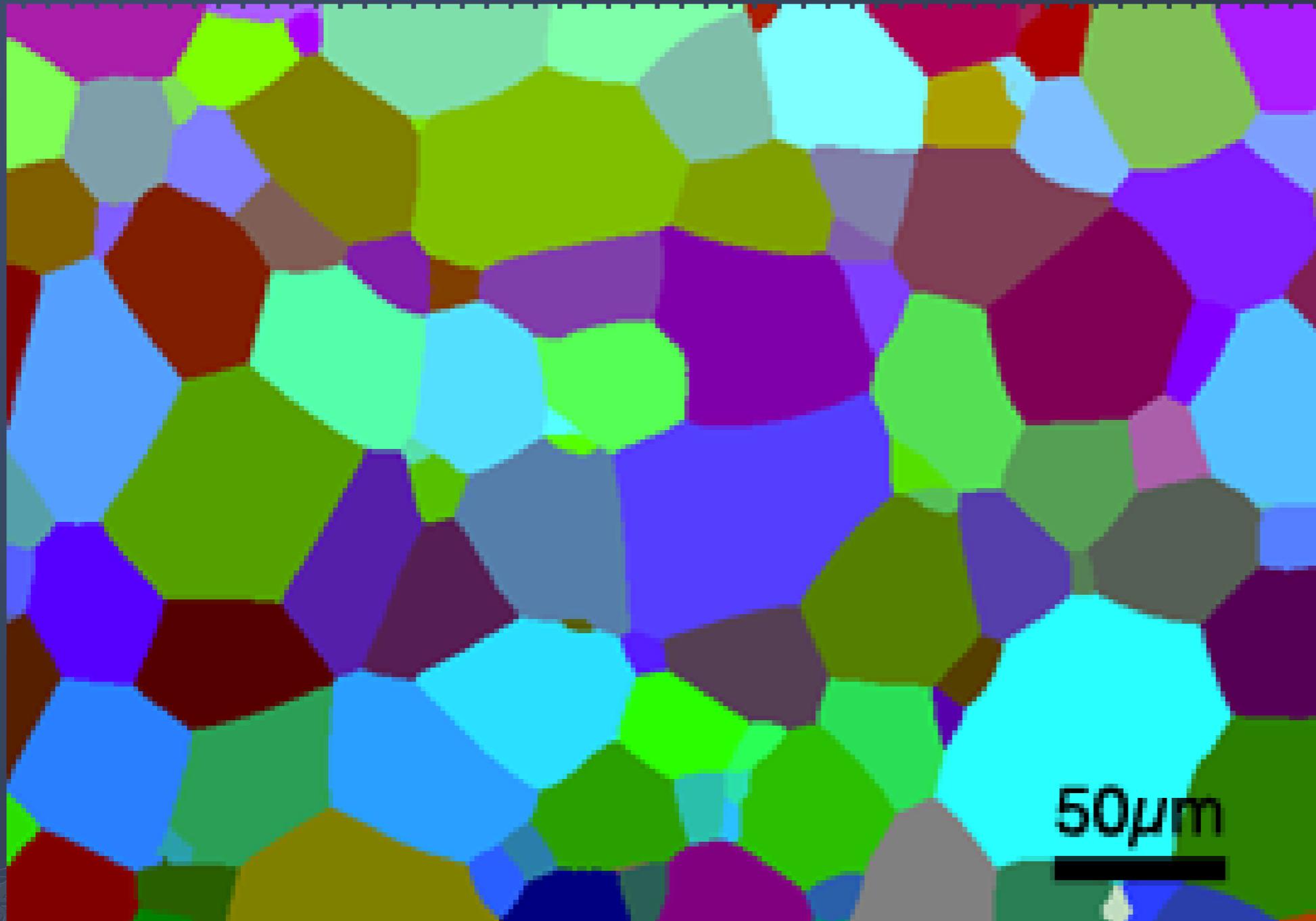
- 1.) Transform the scanned data points to line up with the sample coordinate
  - 1a.) Relabel Xs to increase going to the left, Zs to coming out of the screen.  
(Rotate Xs-Ys-Zs coordinate system about Ys)
  - 1b.) Make the origin so that all points are positive
- 2.) Bring the crystal reference in coincidence with the image coordinate by rotating RD-TD-ND around ND by  $\pi/2 \rightarrow \phi_1 = \phi_1 - \pi/2$

# TSL EBSD Coordinate Systems



- 1.) Transform the scanned data points to line up with the sample coordinate
  - 1a.) Relabel Xs to increase going to the left, Zs to coming out of the screen.  
(Rotate Xs-Ys-Zs coordinate system about Ys)
  - 1b.) Make the origin so that all points are positive
- 2.) Bring the crystal reference in coincidence with the image coordinate  
by rotating RD-TD-ND around ND by  $\pi/2 \rightarrow \phi_1 = \phi_1 - \pi/2$

# Misalignment of EBSD data



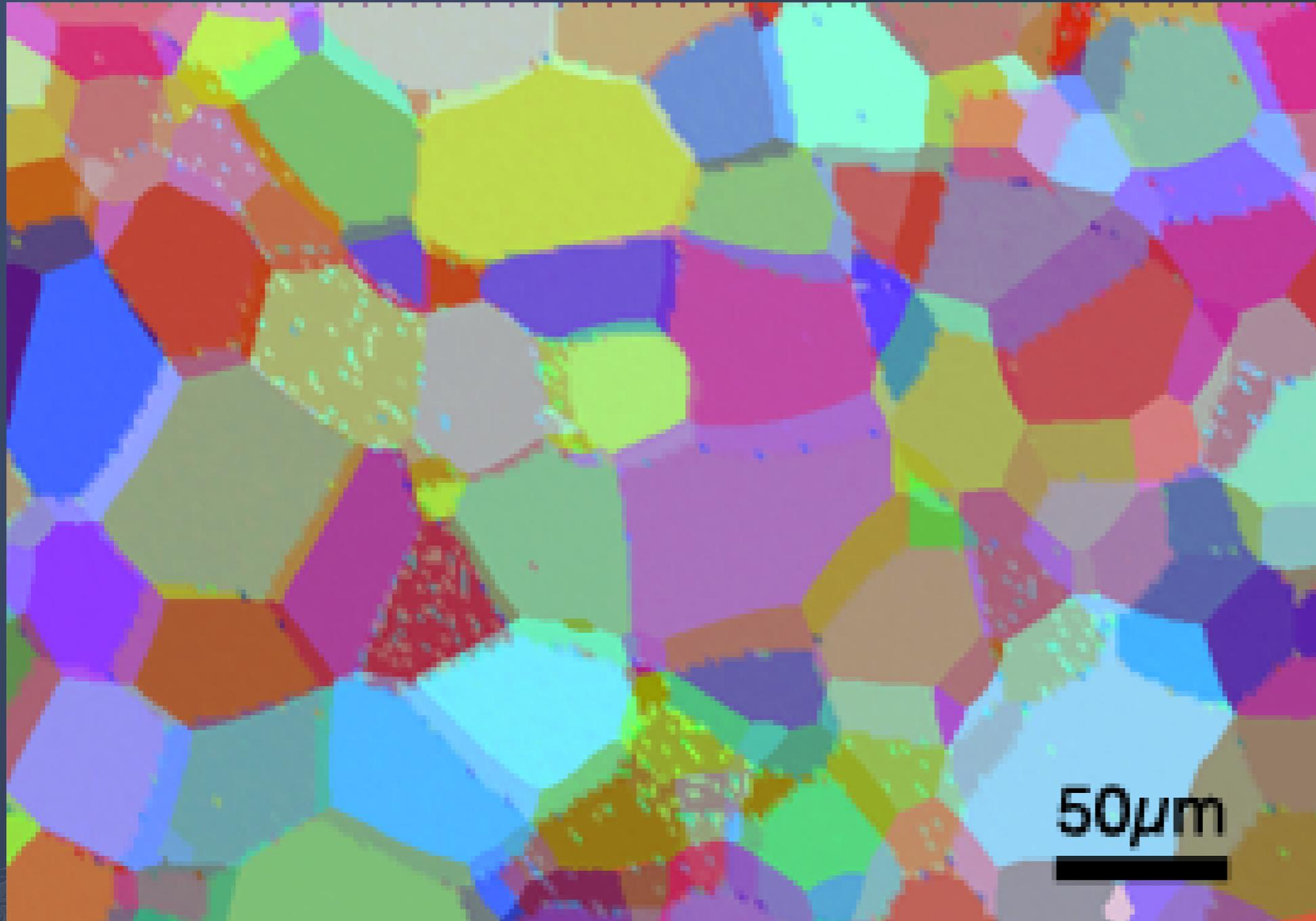
EBSD scans taken every 10th section using a TSL system on a Leo 1550 SEM

Step size  $2\mu\text{m}$   
Total Scan area  $\sim 1000 \times 800\mu\text{m}$

Distortions in the SEM made alignment more difficult.



# Misalignment of EBSD data



EBSD scans taken every 10th section using a TSL system on a Leo 1550 SEM

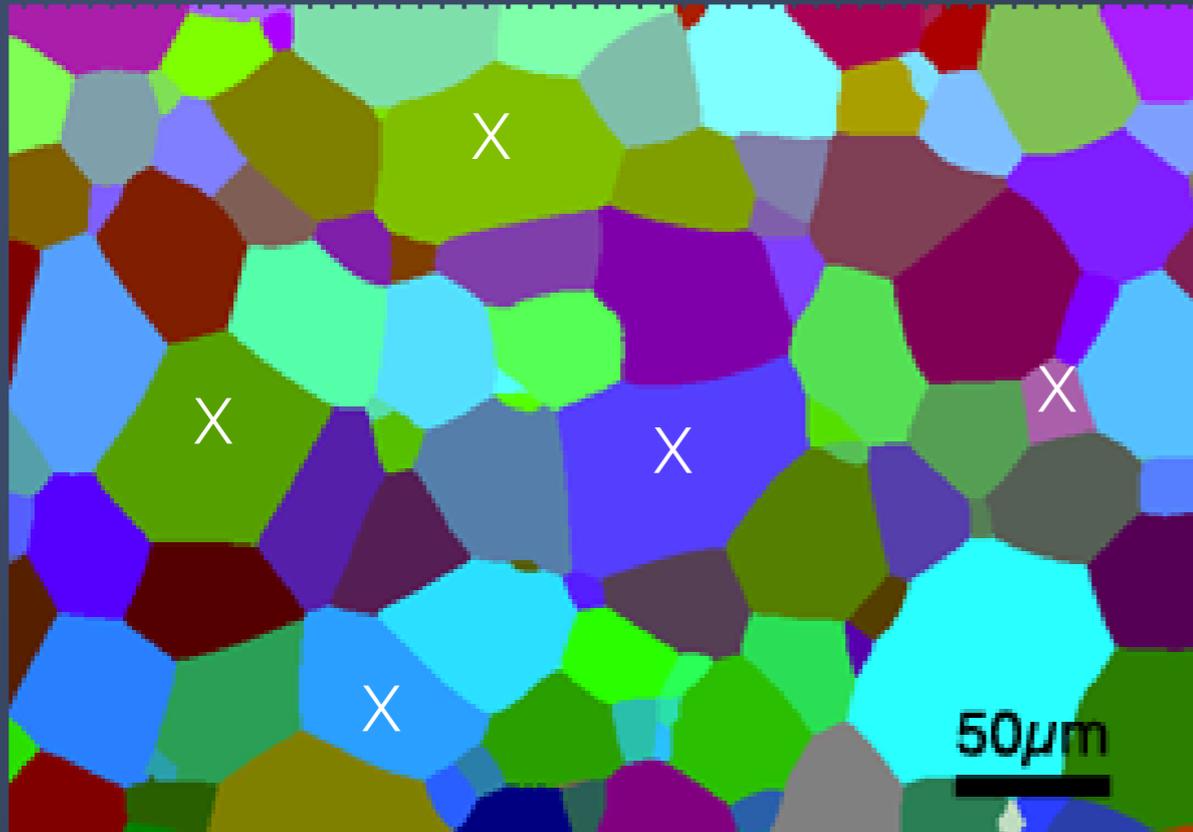
Step size 2µm  
Total Scan area ~1000 x 800µm

Distortions in the SEM made alignment more difficult.

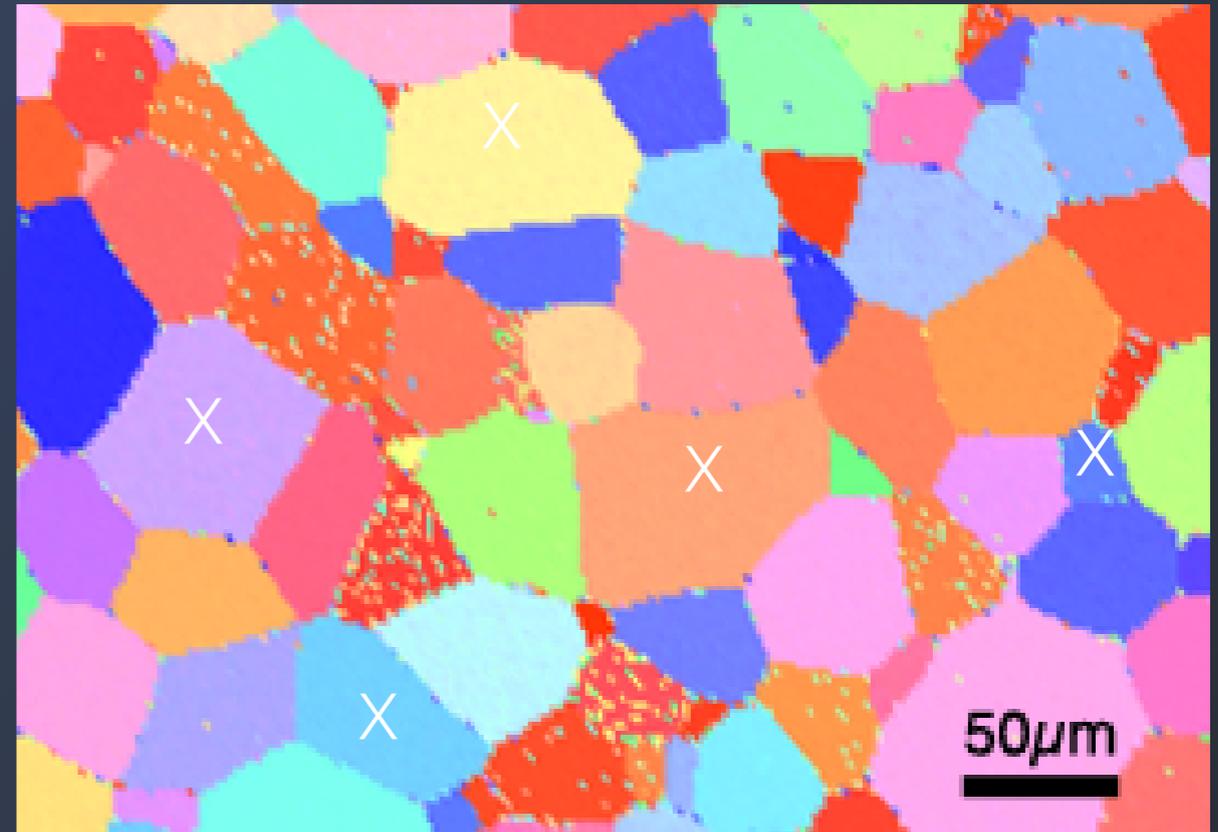


# Geometric EBSD Alignment

Segmented Optical Image



EBSD Grain Map

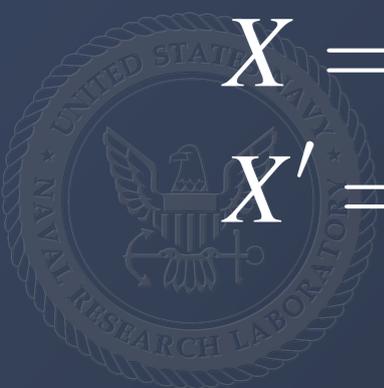


Alignment of the images was achieved by comparing the center of area for several grains in the segmented optical image and the EBSD grain map.  $T$  is determined through a linear algebra least squares fit.

$$X = [x_i, y_i, 1]$$
$$X' = [x'_i, y'_i, 1]$$

$$X' = XT$$

$$T = \begin{bmatrix} a & b & x_t \\ c & d & y_t \\ 0 & 0 & 1 \end{bmatrix}$$

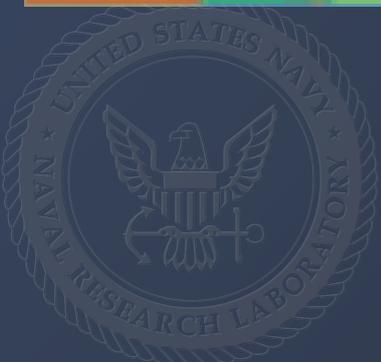
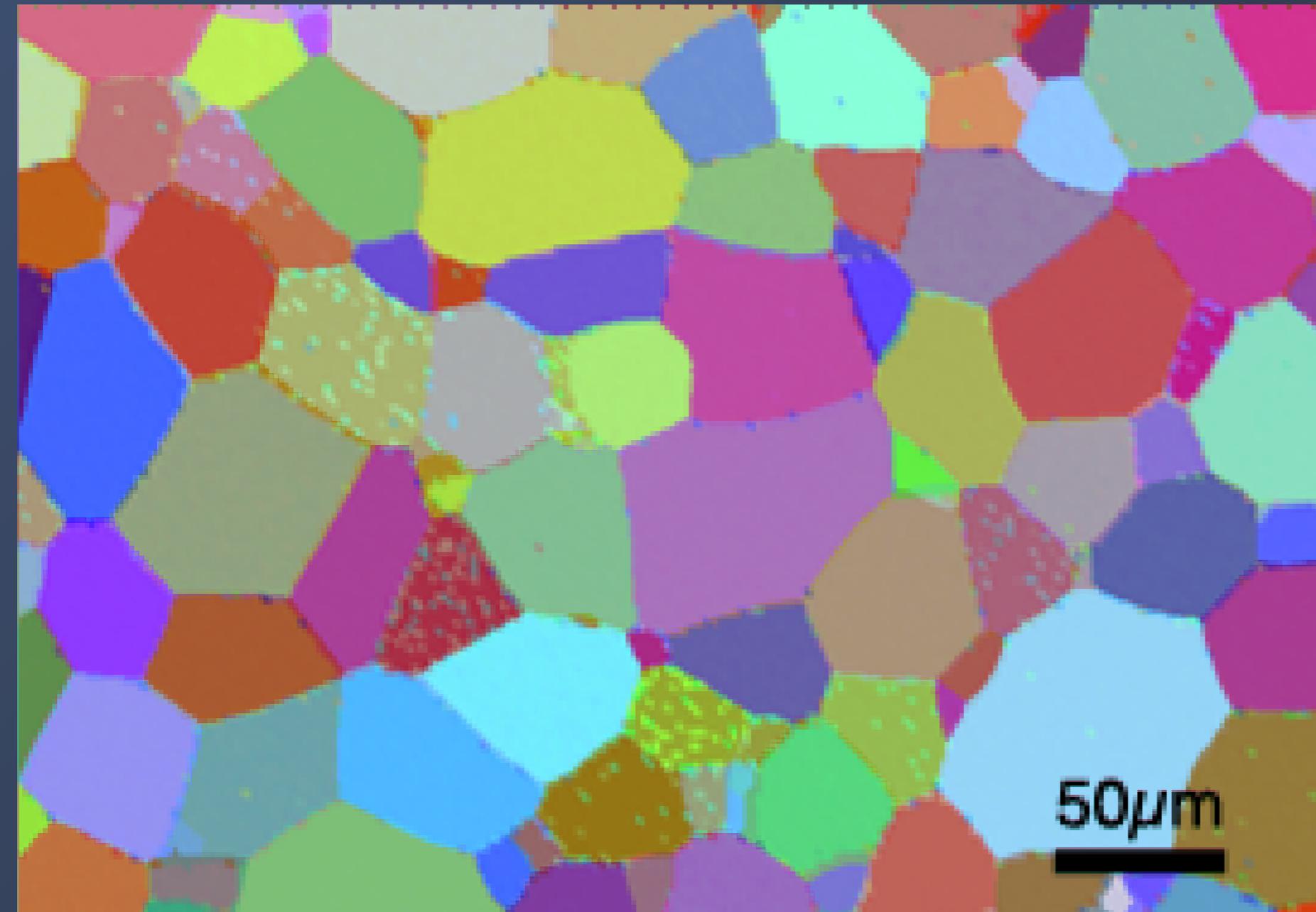


# EBSD Alignment

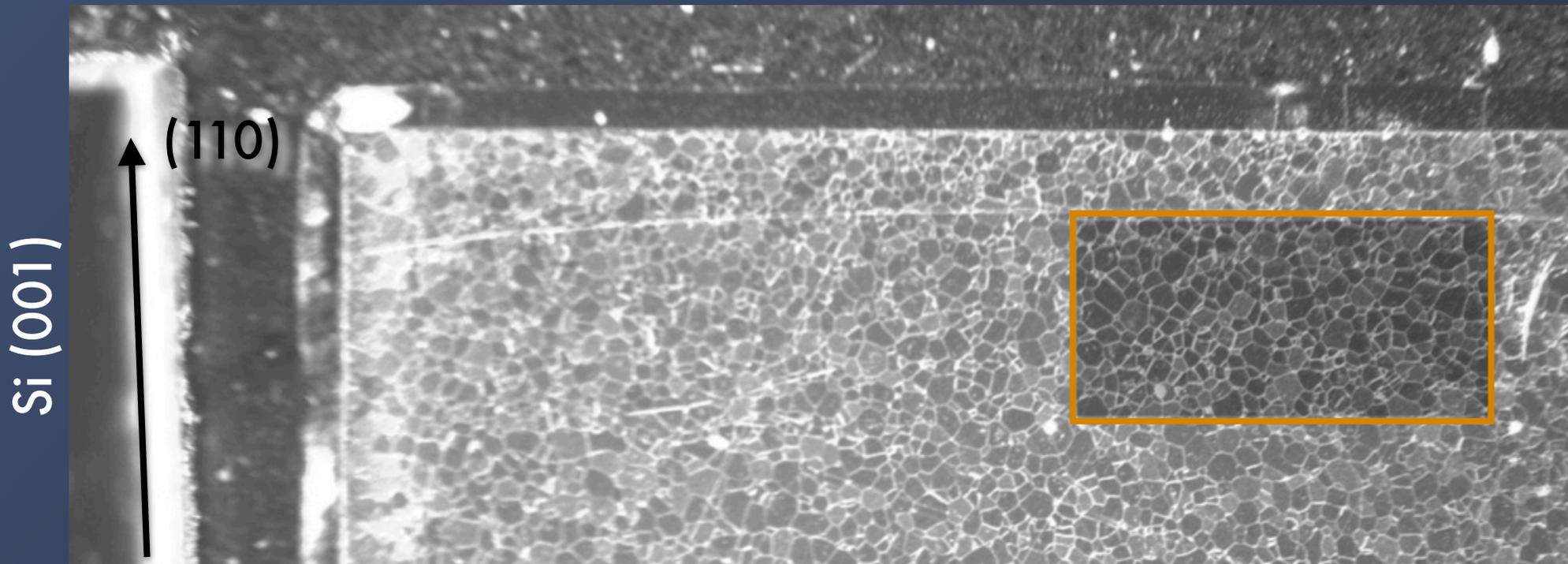
Using the grain center of areas for the two sections (optical/EBSD) an affine transform was determined that aligned the EBSD data with the optical images.

All the EBSD data was aligned to a Reference Crystal.

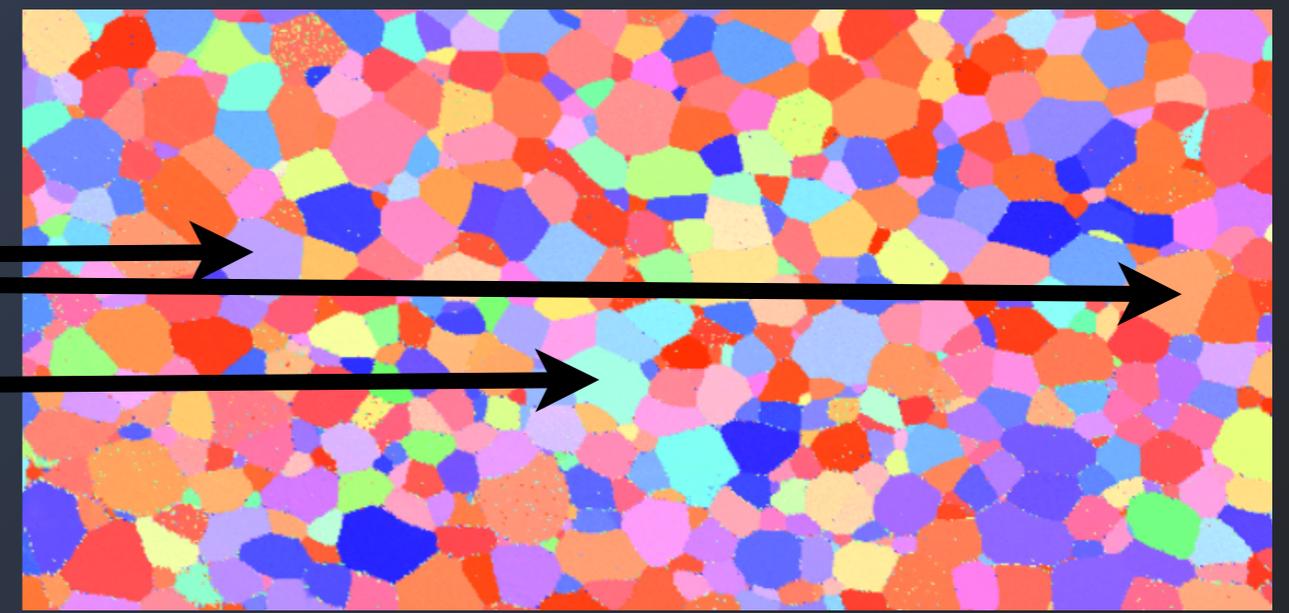
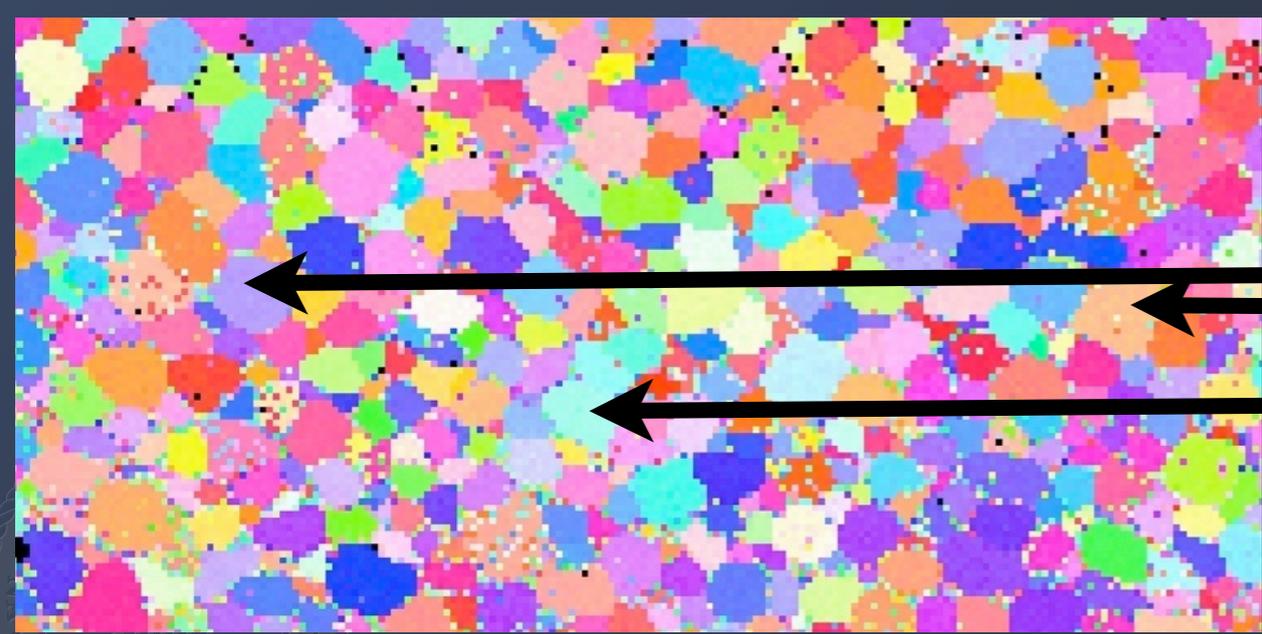
A single average orientation was assigned to each grain.



# EBSD Crystal Realignment



A Si reference crystal was mounted on the sample after the last section.



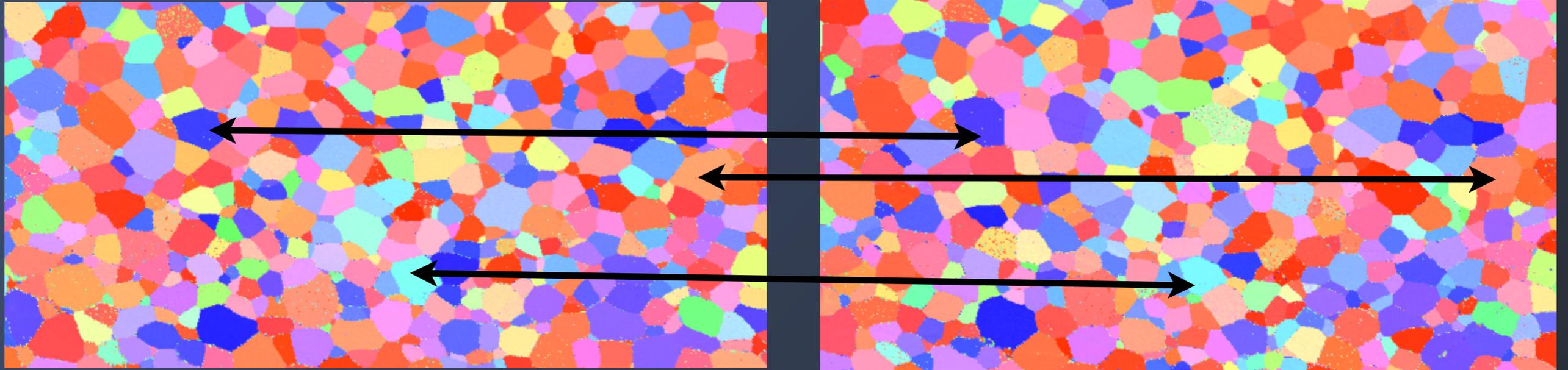
Section 0

Section 0 - High Res

# Crystallographic Alignment

Section 0

Section 10



The crystal orientation on each section were corrected by calculating the image rotation that minimized the misorientation of the grains that are present in both sections.

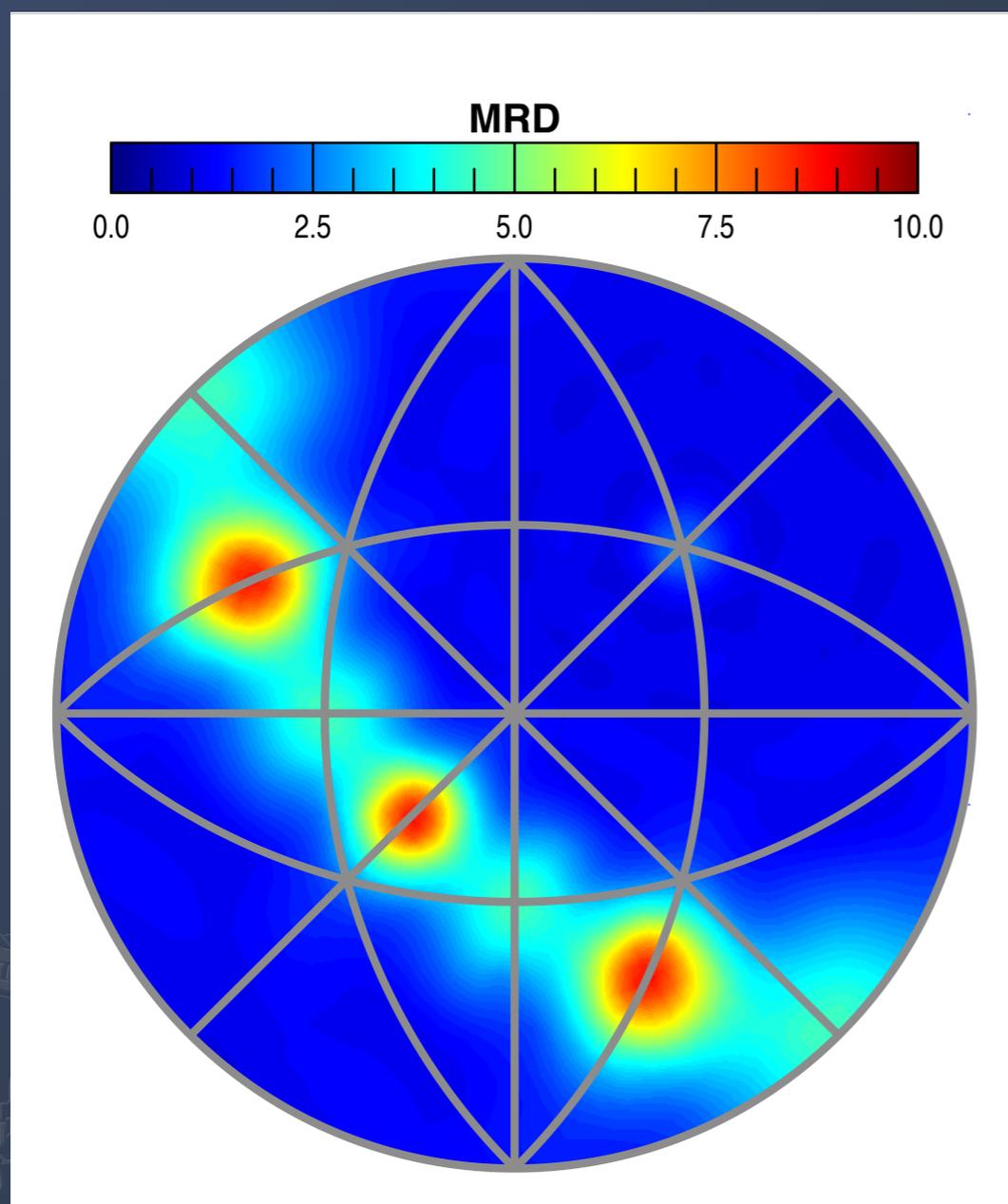
The resulting spread in the orientation is approximately 0.5 degrees, thus an average orientation for each grain by averaging the data for all the EBSD sections.



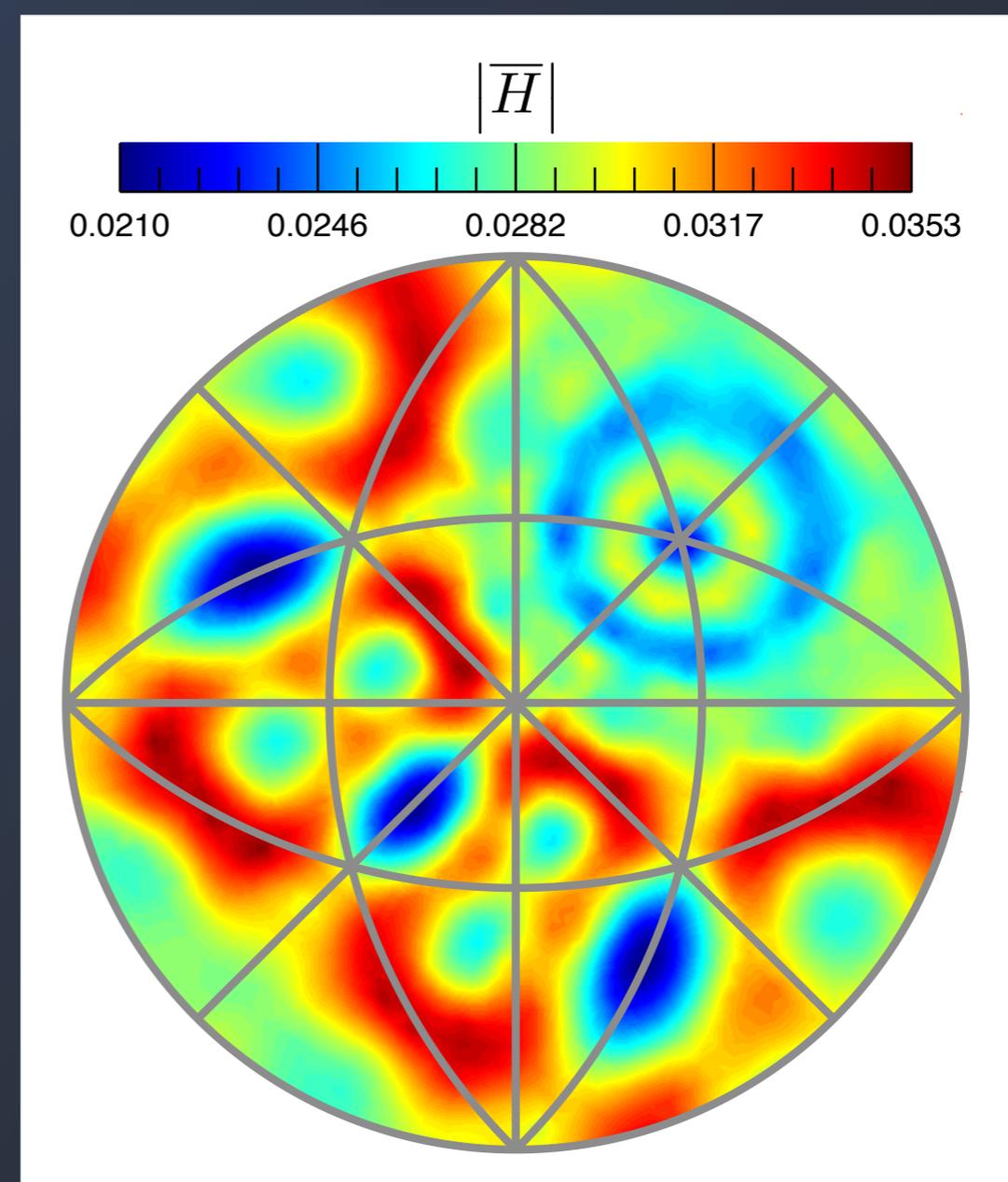
# Interfaces types of $\Sigma 3$ boundaries

Extract all of the grain boundaries that have a misorientation of  $60^\circ$  @  $[111]$ .  
Plot the population of the different types of interfaces, and the magnitude of the interfacial curvature:

Surface area

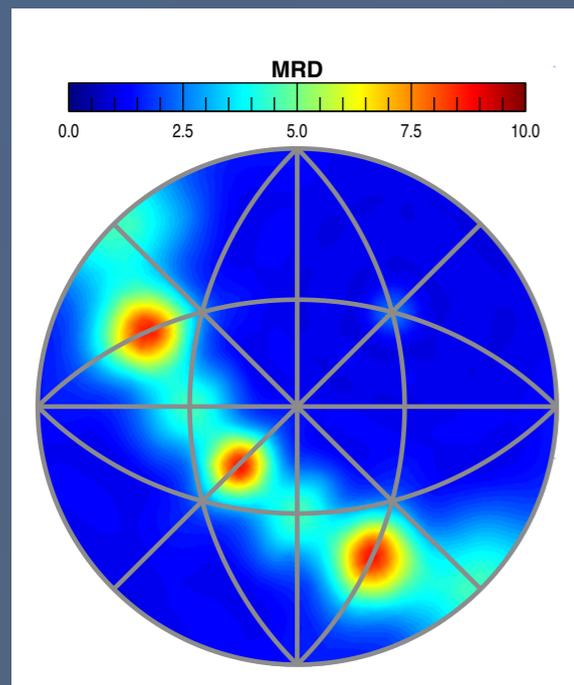


Curvature

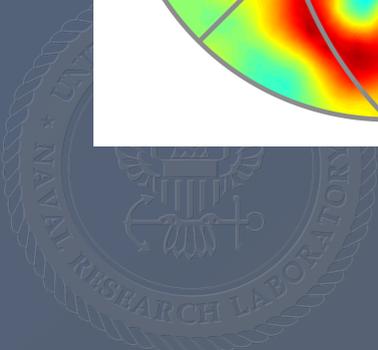
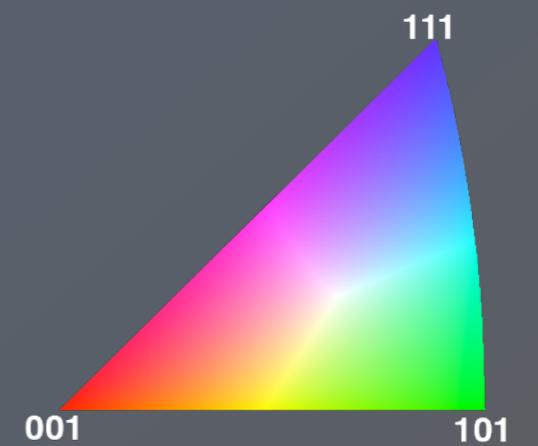
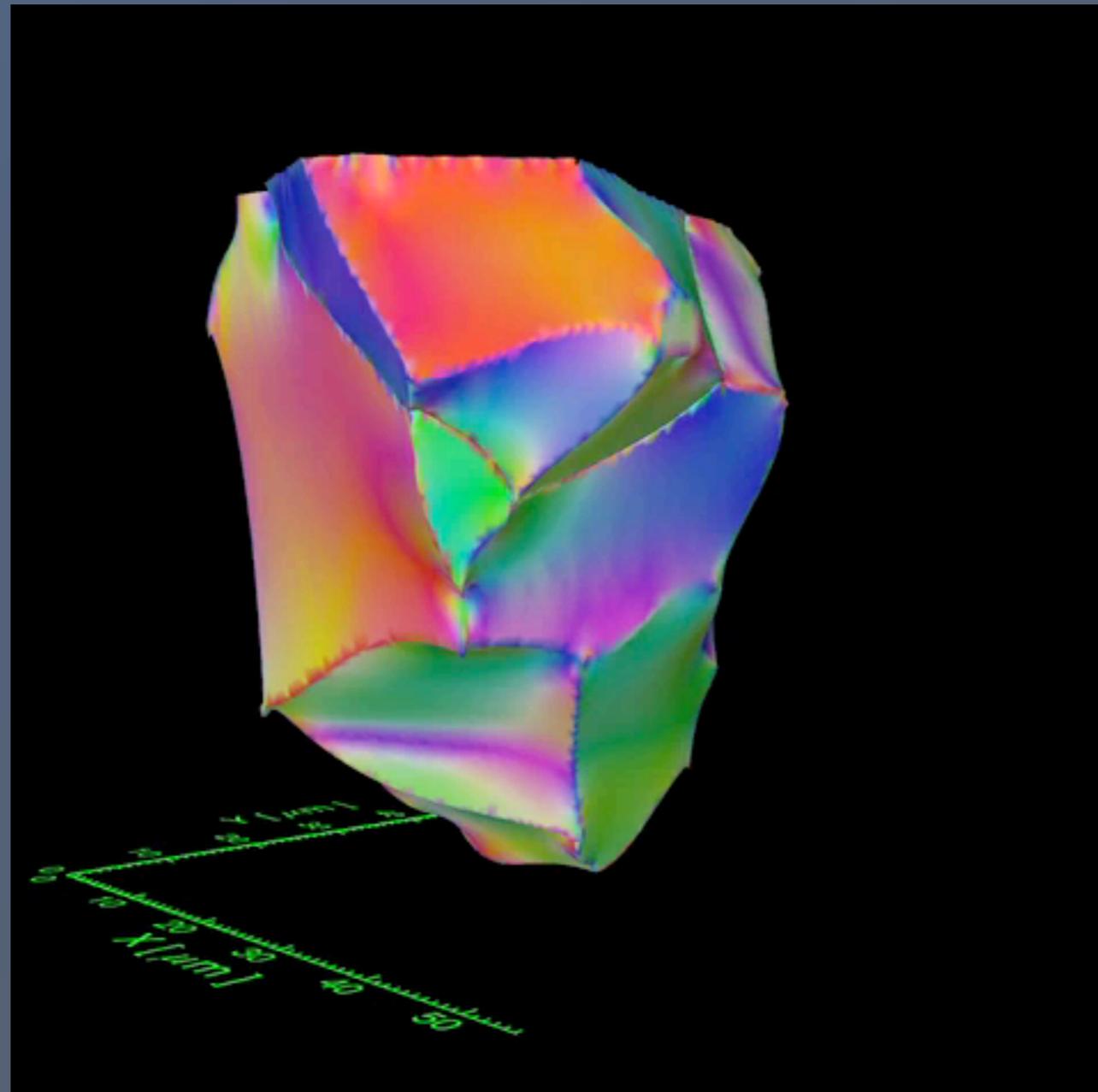
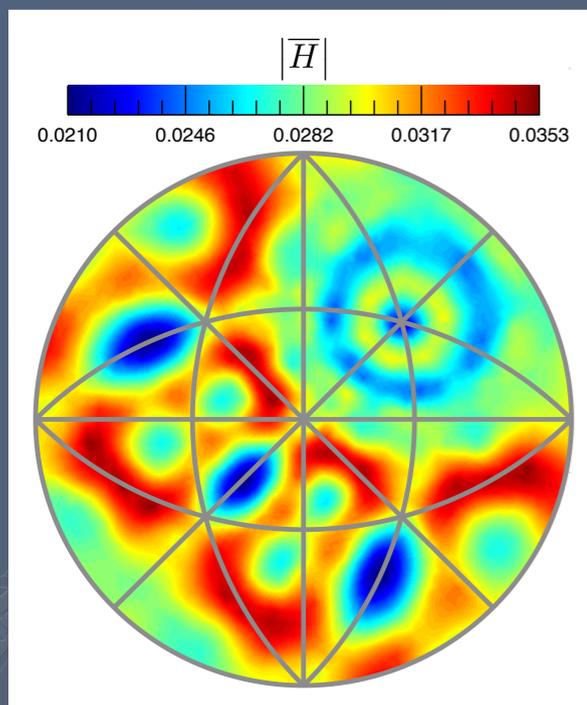


# $\Sigma 3$ Boundaries

Surface area

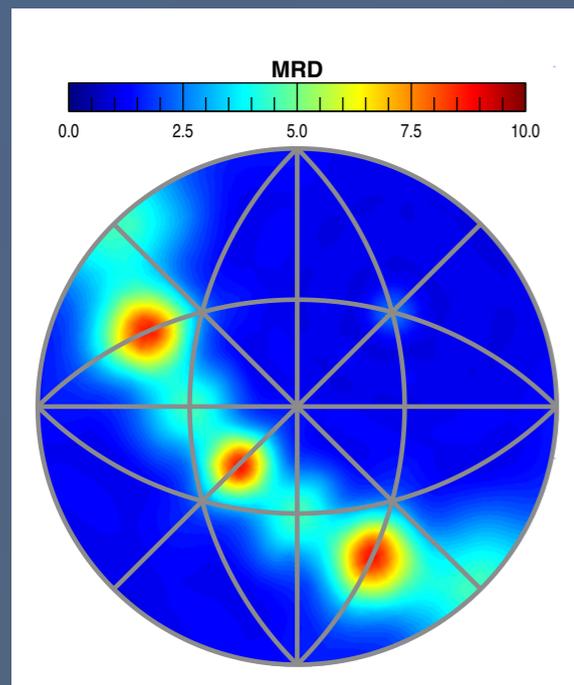


Curvature

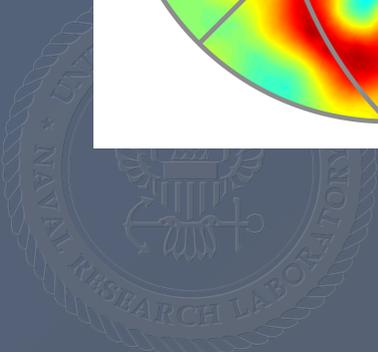
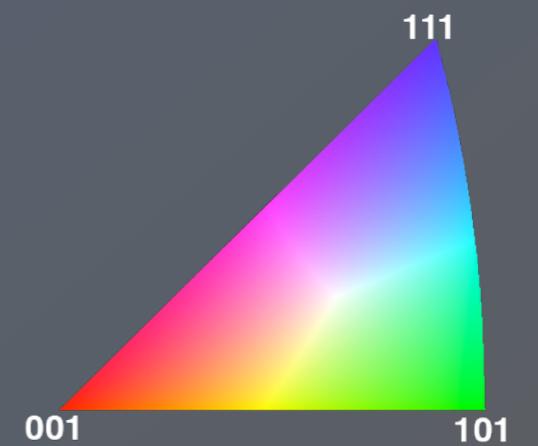
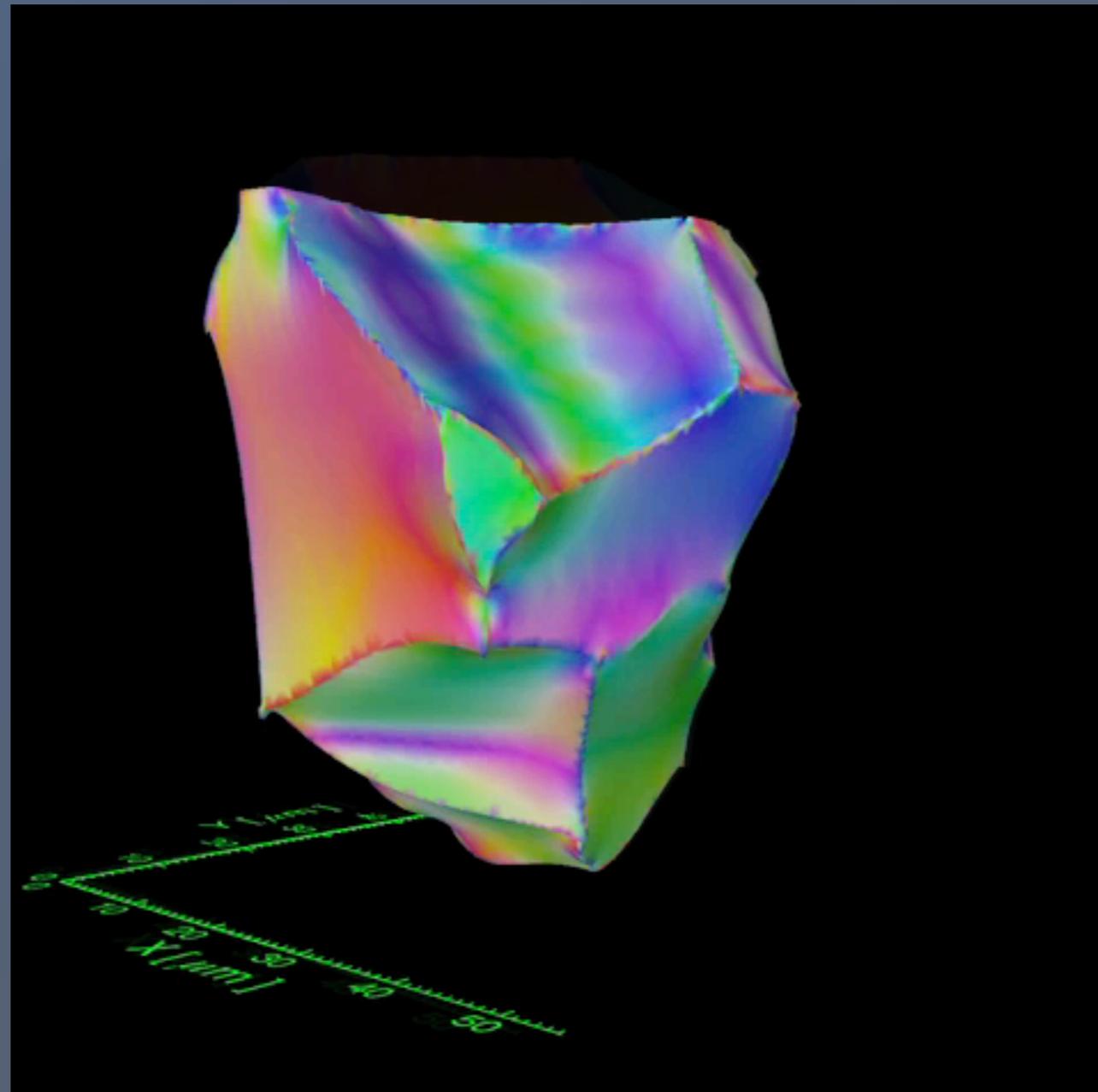
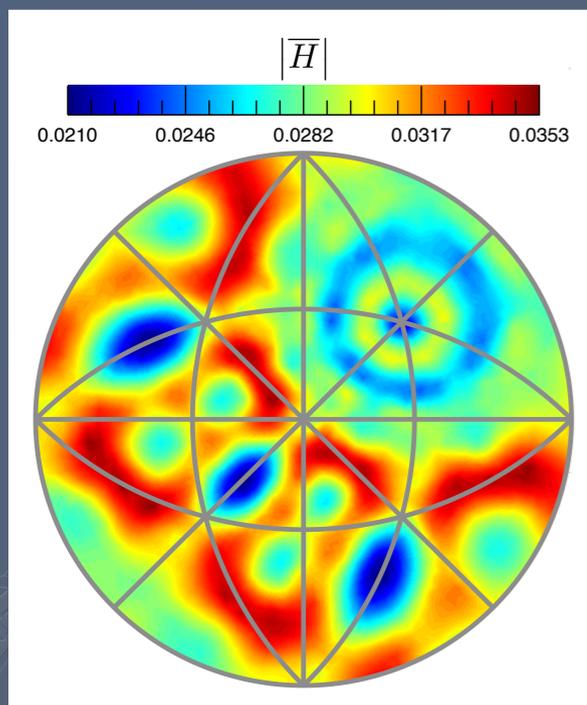


# $\Sigma 3$ Boundaries

Surface area

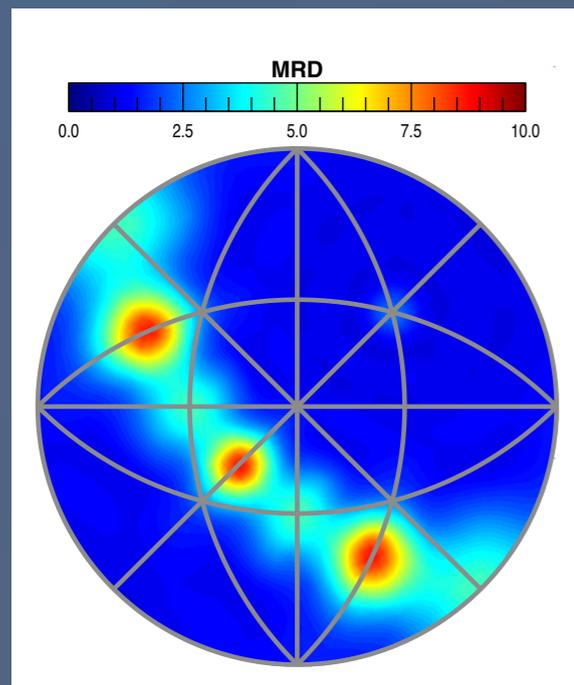


Curvature

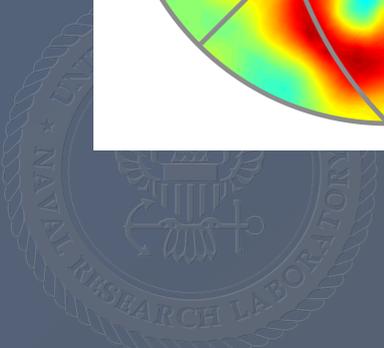
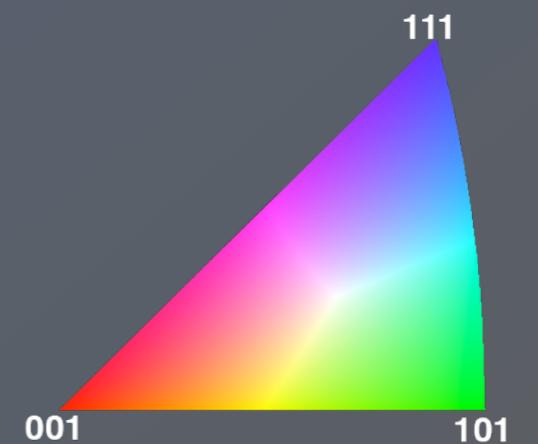
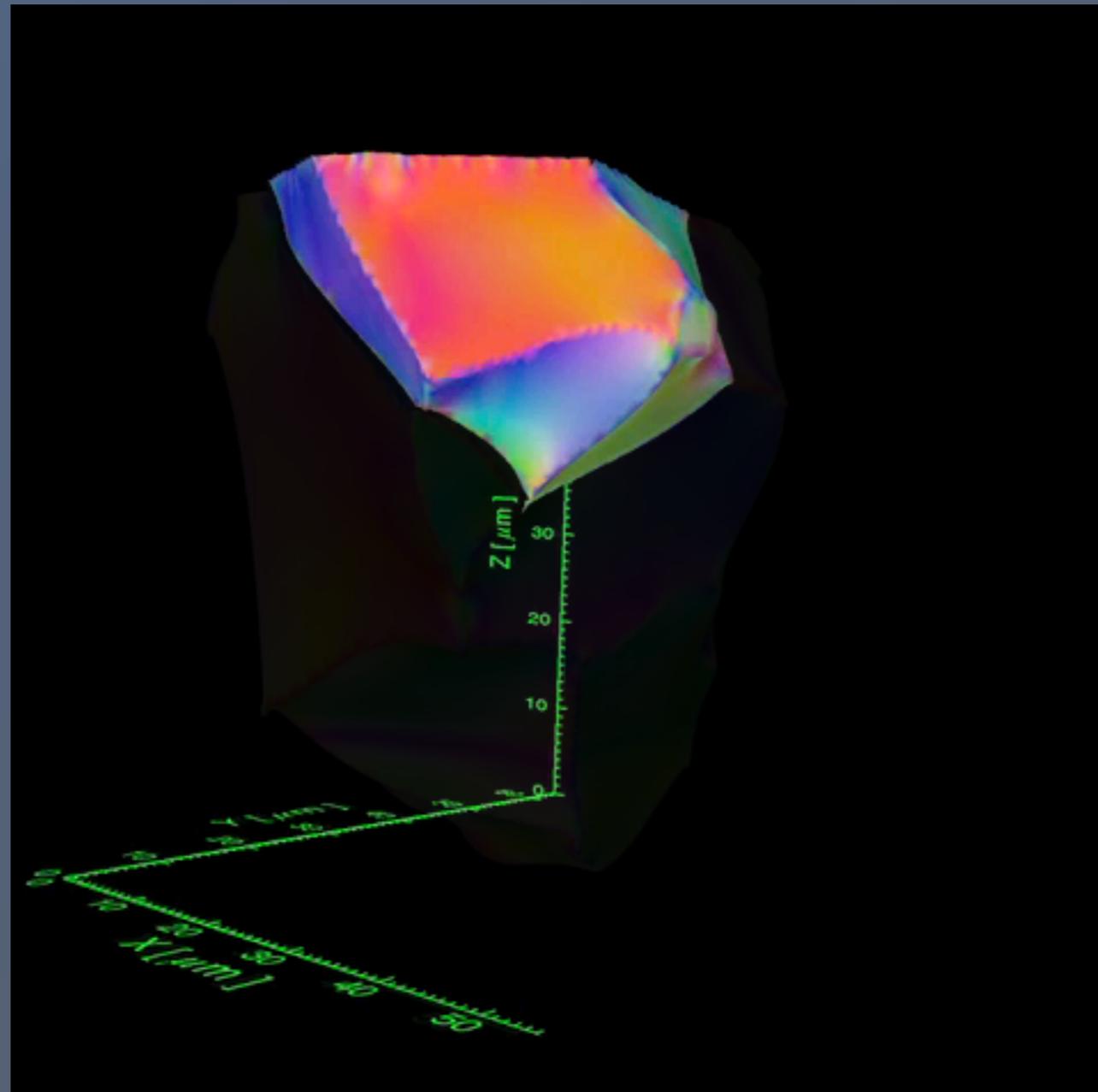
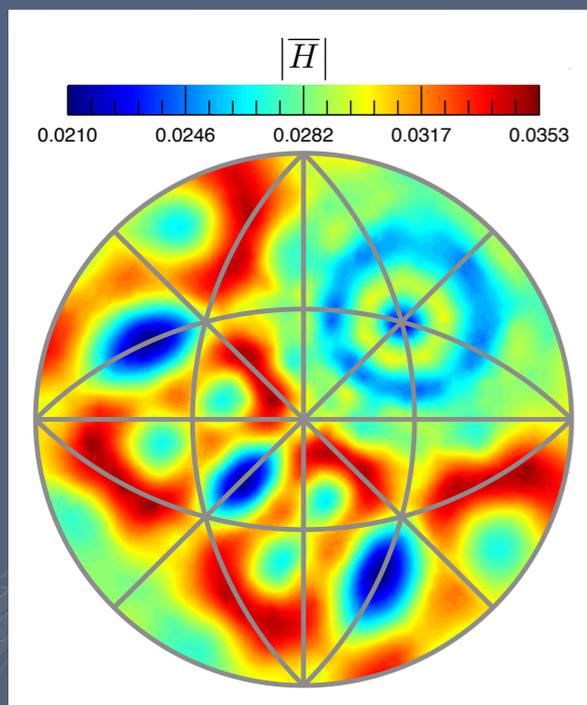


# $\Sigma 3$ Boundaries

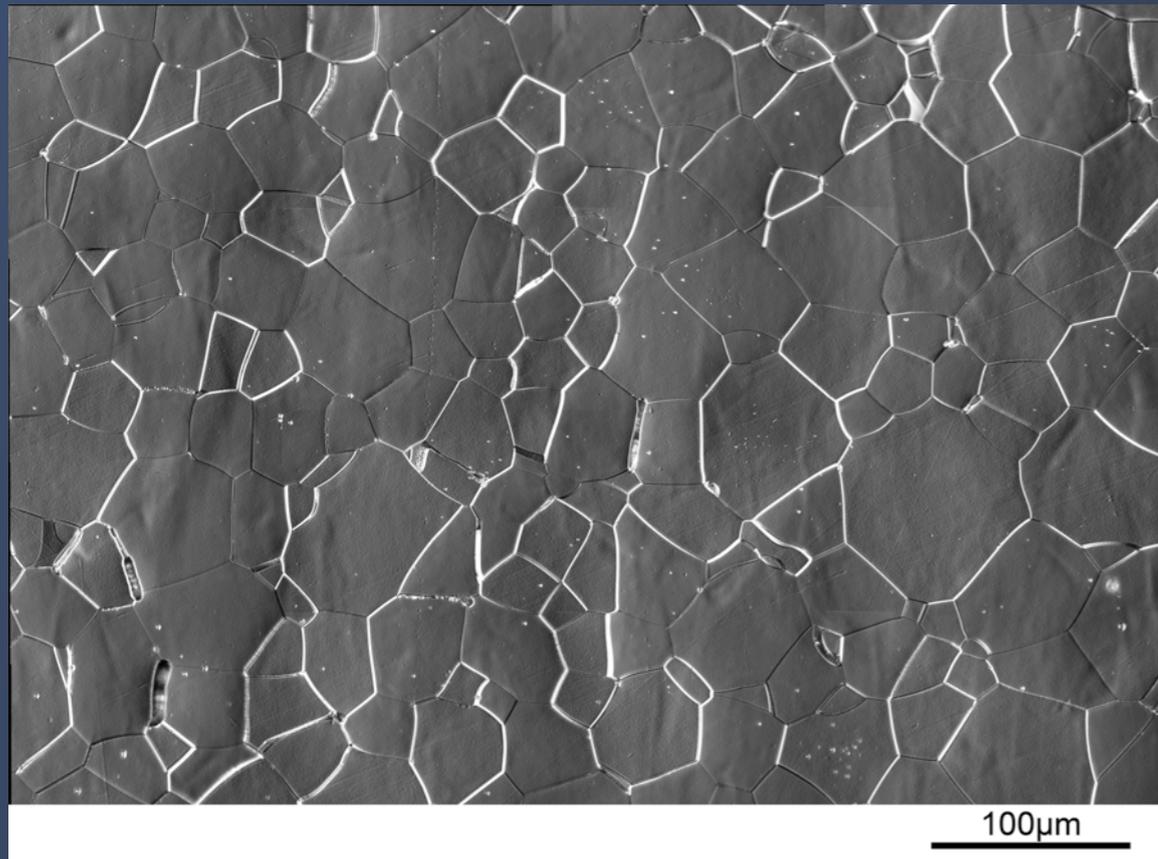
Surface area



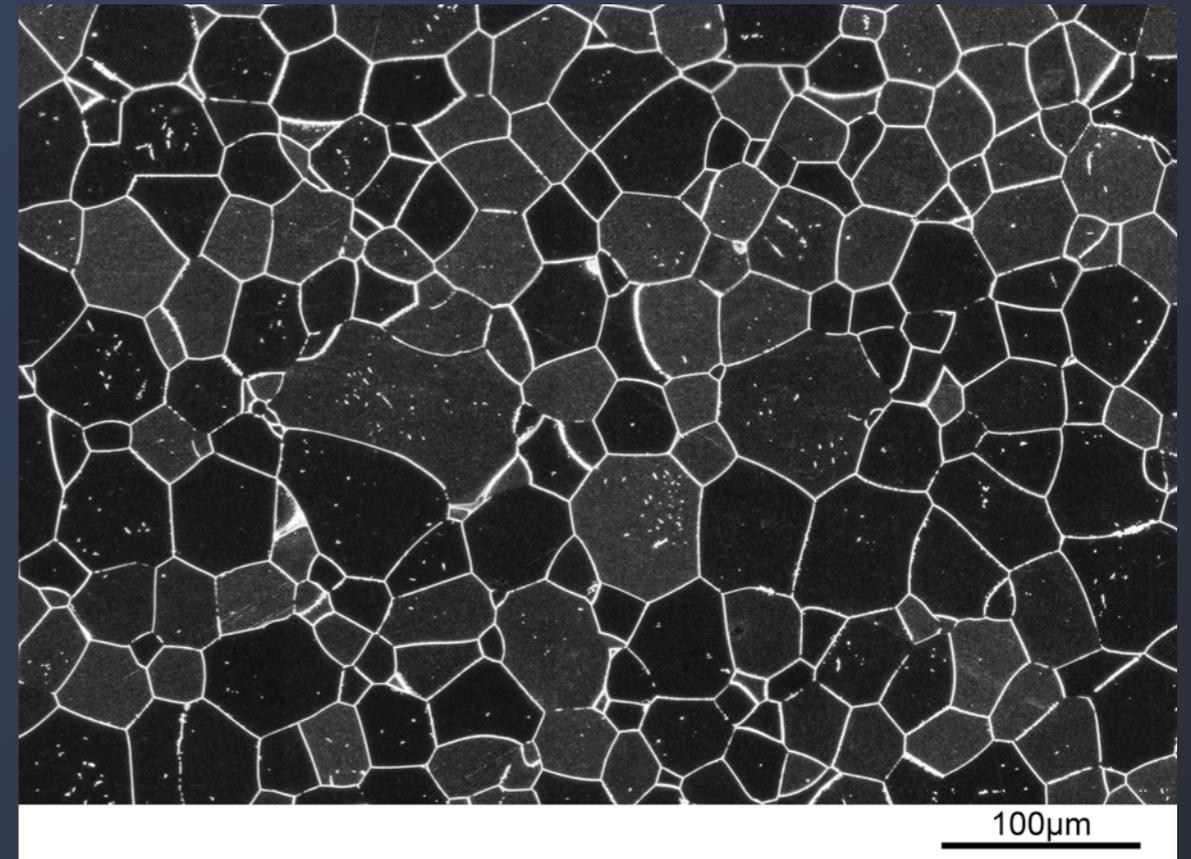
Curvature



# The slow step in 3D analysis is nearly always image processing



As quenched Ti-21S: No  $\alpha$ -precipitation,  $\beta$ -grains. Etched, imaged under Newmarski differential contrast.



$\alpha$ -Ti was precipitated out on to decorate the GB - not a true single phase material. Our alloy was chosen because of the ability to make easy to segment image.



# The slow step in 3D analysis is nearly always image processing

Solving an image processing problem more often than not only applies to a narrow set of alloys; new alloys often require completely starting over. This is very different than most image segmentation problems.

We need imaging solutions that provide clear, consistent image contrast across a wide variety of materials systems

The contrast mechanism should be based on properties that we know provide important materials information.

(Backscattered images, foreshattered images, EDS, EBSD)

As an example EBSD for a TRUE single phase material...



# Combining more than one signal

## Optical Microscopy serial sections:

- + Really fast (Pixel density is high)
- + Relatively distortion free over very large length scales.
- +/- Sample preparation is universal, and many times easy.
- +/- Resolution/field of view is sufficient for many materials problems.
- In many materials doesn't really provide measurements that are of interest for materials.

We want grain boundary interface properties, but grain boundaries are a function of their crystallography.

Electron Backscattered Diffraction (EBSD) provides spatial mapping of crystallographic orientations.

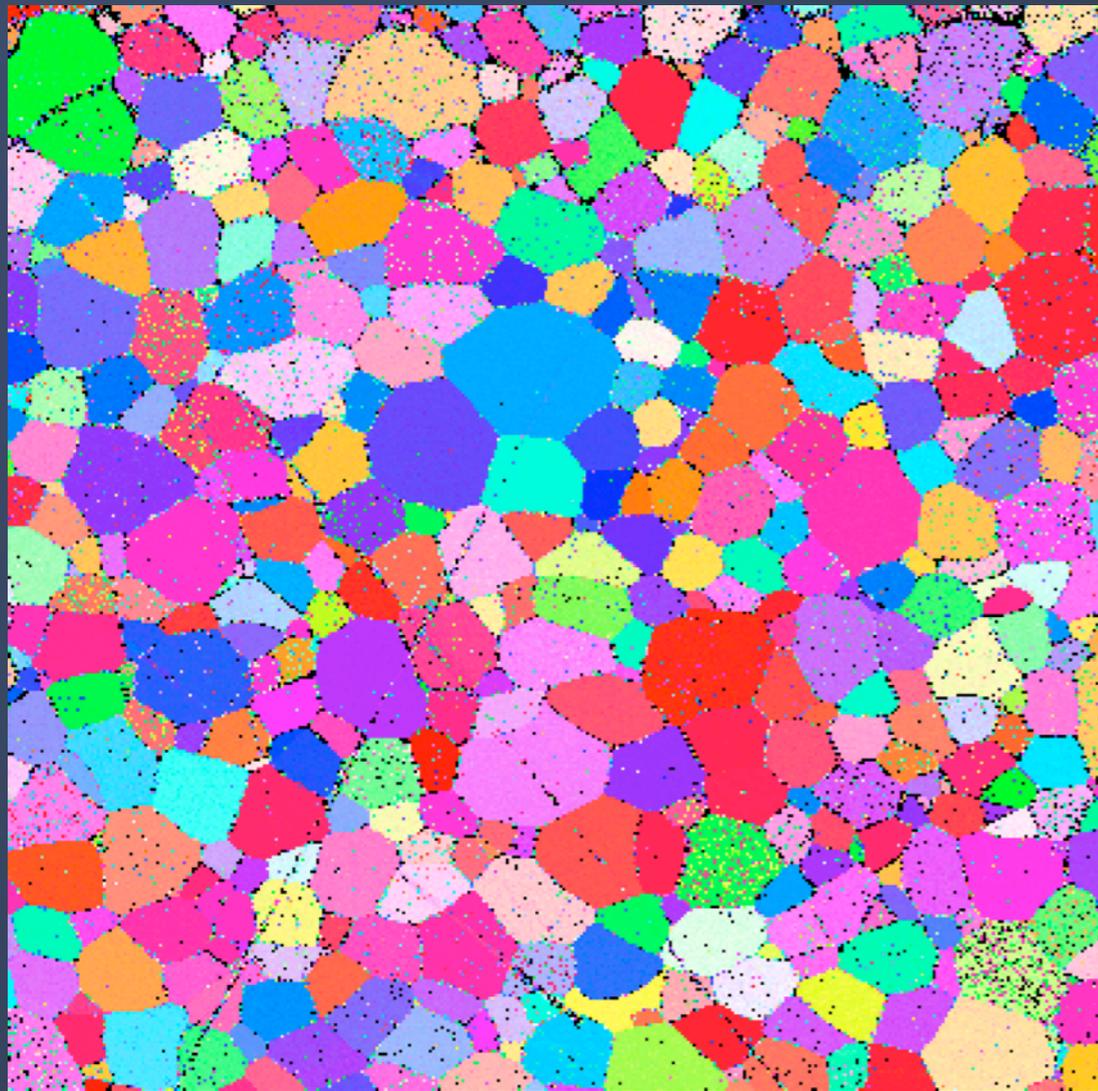
## EBSD/EDS (BSE,

- + Provides “contrast” based on real physics of the material, especially information that materials scientist are interested in.
- + Resolution/field of view is good for many materials problems.
- Fairly slow (or pixel density is low)
- Often contains distortions, sometimes significant.
- Sample preparation is not easy and far from universal.



# EBSD as an *imaging* method

2004



250 $\mu$ m

Step Size 2.0 micron  
Collected in 3.6Hrs

2010



250 $\mu$ m

Step Size 0.5 micron  
Collected in 3Hrs



# EBSD as an *imaging* method

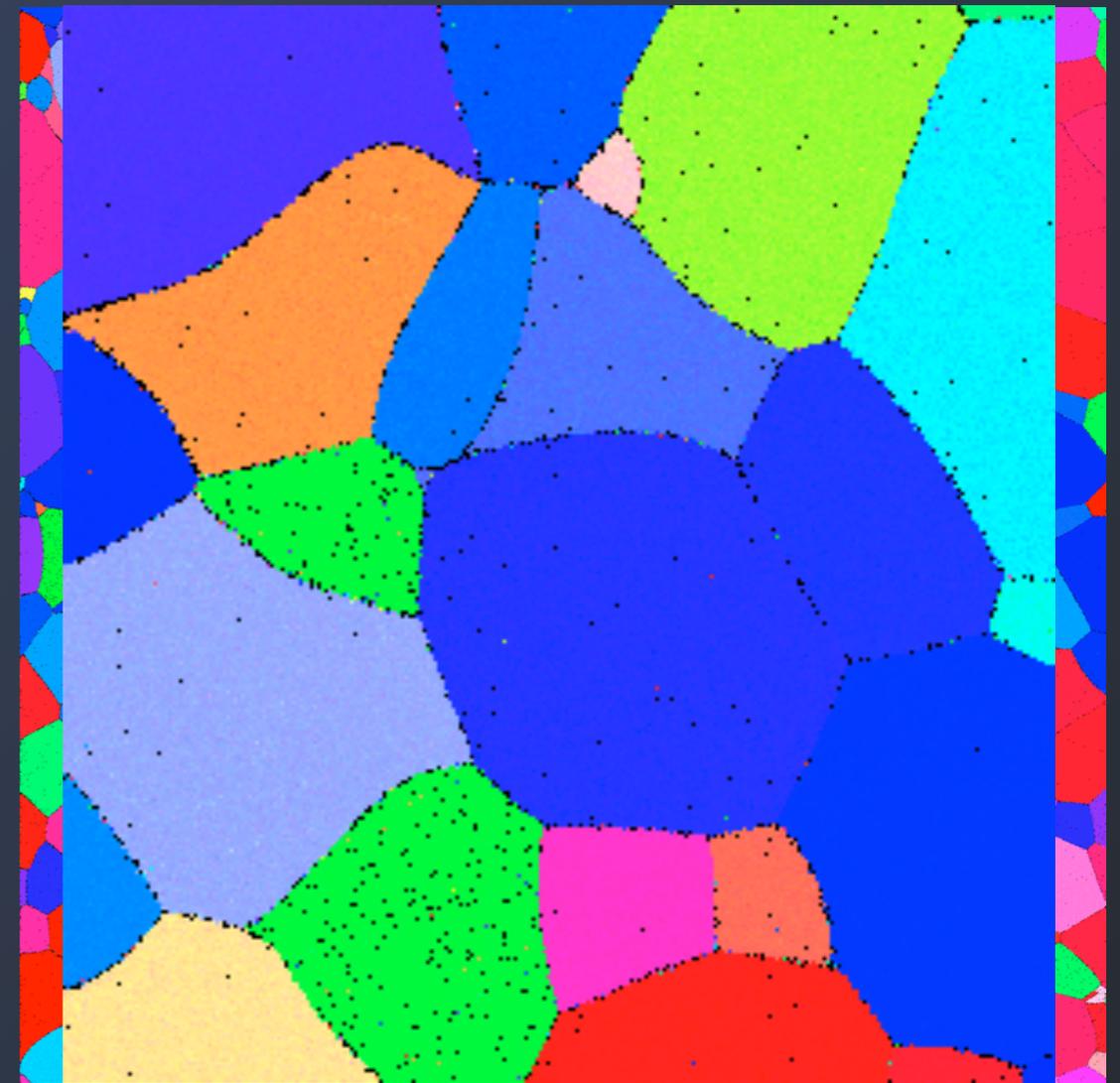
2004



250µm

Step Size 2.0 micron  
Collected in 3.6Hrs

2010



250µm

Step Size 0.5 micron  
Collected in 3Hrs

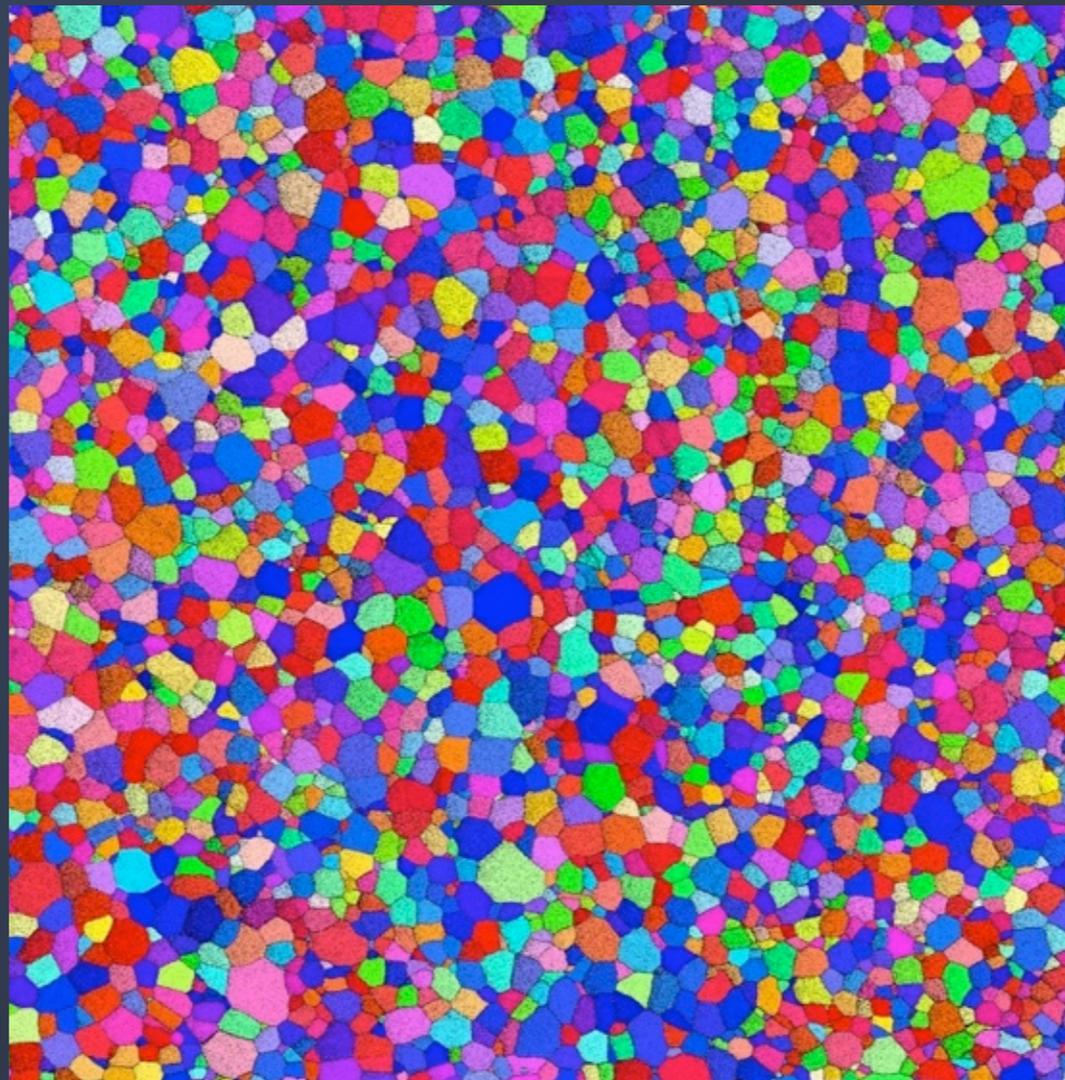


# EBSD Combo Scans

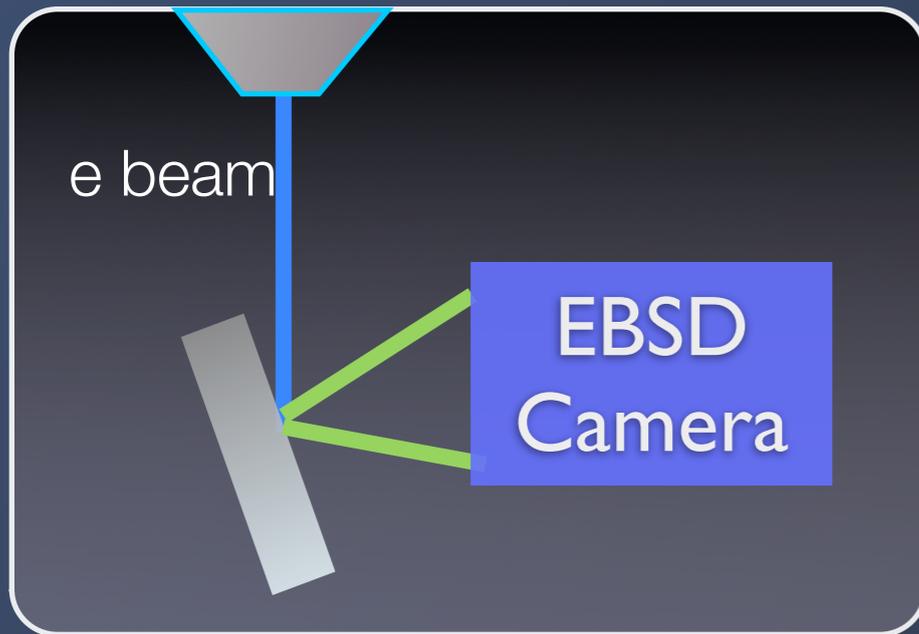
“Combo Scans” are used to collect either larger areas than can be collected at low magnification or collect large areas at a high resolution.

This can work if you use high precession stages/software, and SEM images are distortion free

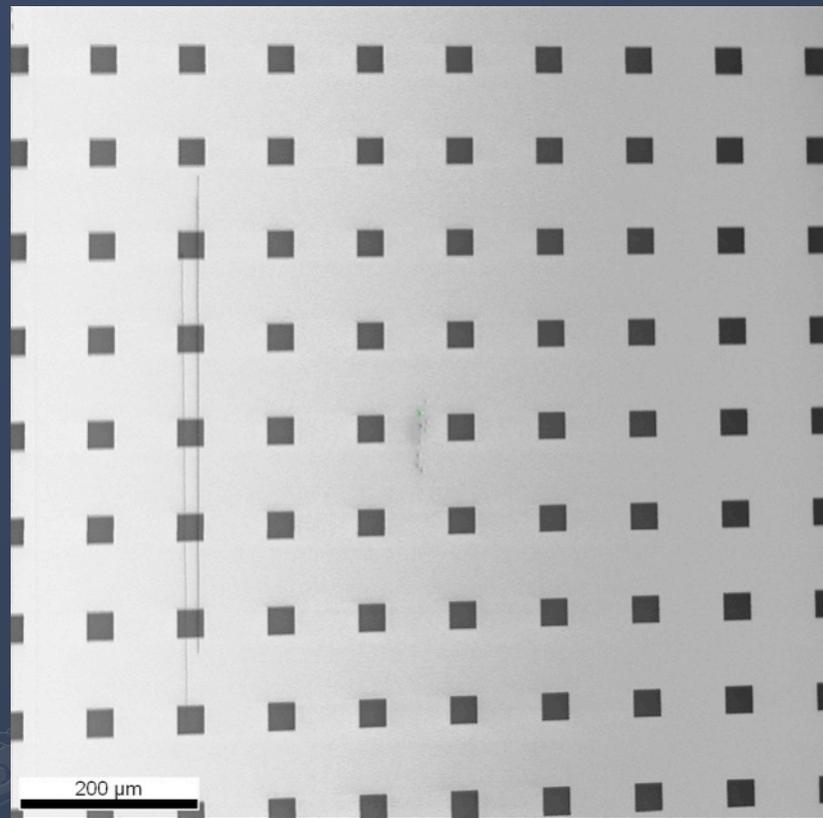
A. L. Pilchak, A. R. Shiveley, P. A. Shade, J. S. Tiley, and D. L. Ballard, J Microsc-Oxford, vol. 248, pp. 172, 2012.  
A. R. Shiveley, P. A. Shade, A. L. Pilchak, J. S. Tiley, and R. Kerns, J Microsc-Oxford, vol. 244, no. 2, pp. 181, 2011.



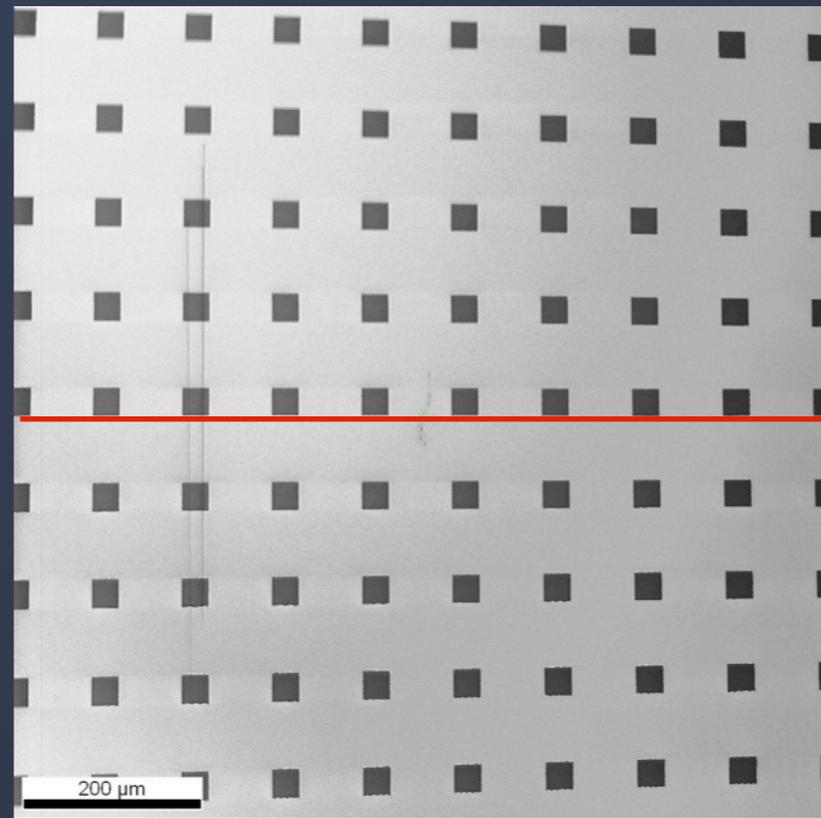
# Image Distortions in EBSD



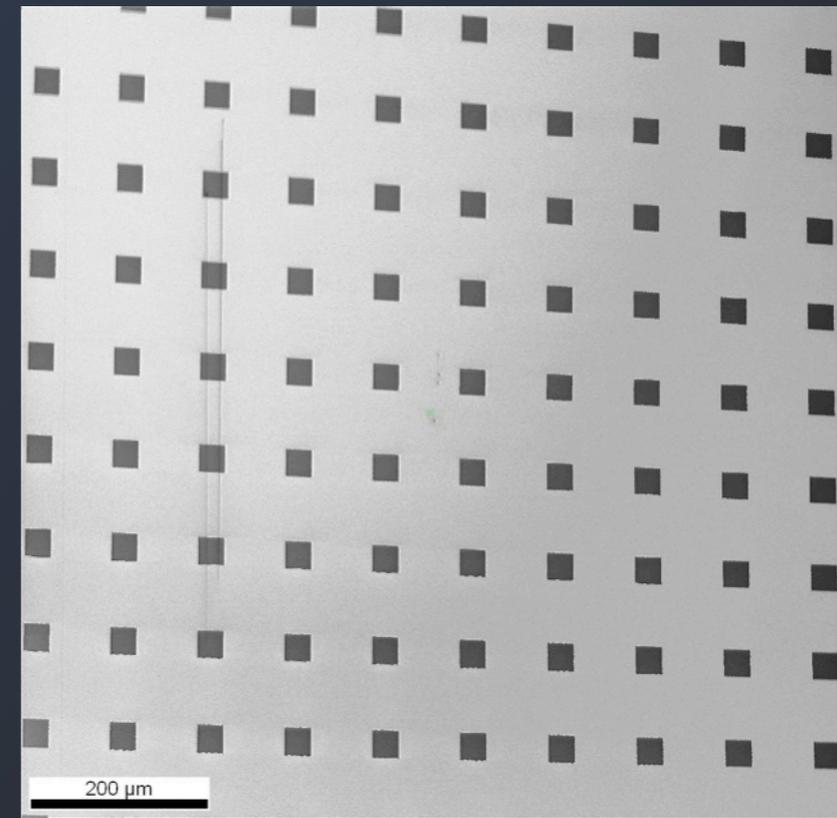
The sample is tilted at 70° tilt, an imaging condition at which SEM manufacturers do not expect a distortion free image.



100x, 70.0° Tilt,  
18.0mm WD



100x, 70.0° Tilt,  
16.0mm WD



100x, 70.0° Tilt,  
14.0mm WD

# EBSD Image Distortions

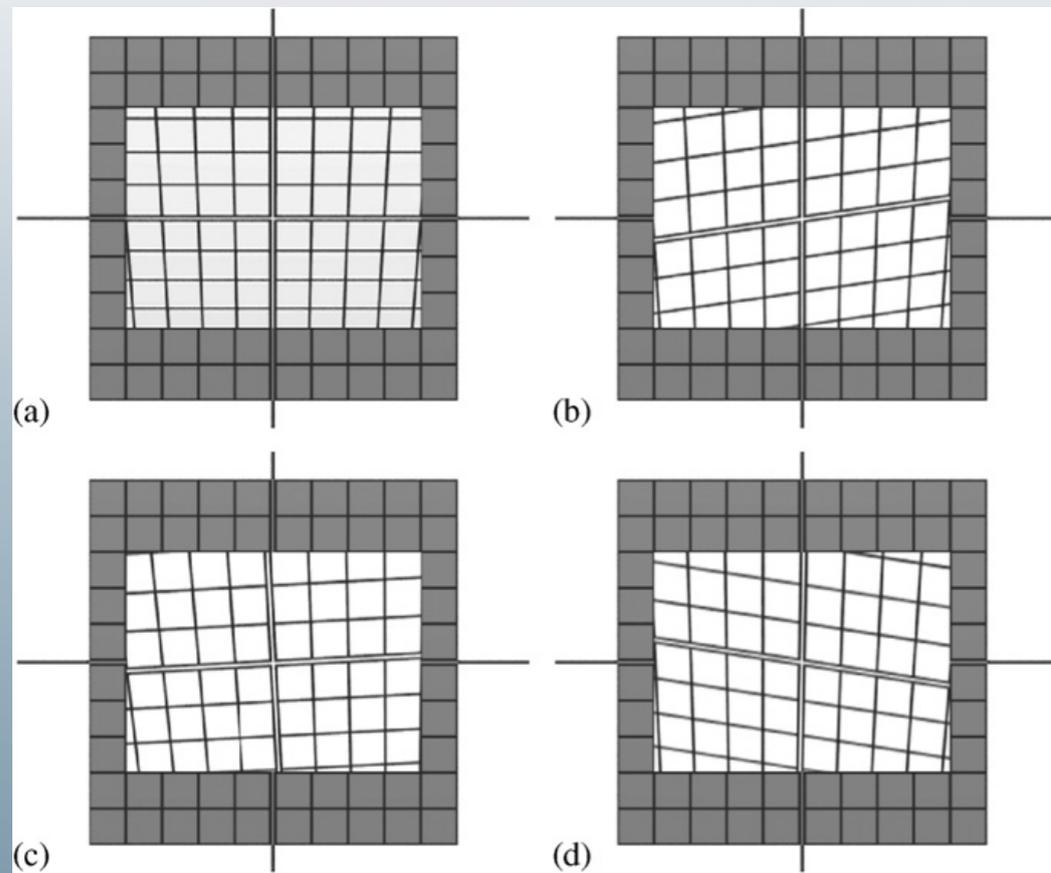


Fig. 10. Simulations of the influence of different, additional sample misalignments on the image distortion: (a)  $3^\circ$ -tilt around X1; (b)  $3^\circ$ -tilt around Y1; (c)  $3^\circ$ -rotation around Z0. Furthermore, the influence of scan rotation is given: (d)  $3^\circ$ -scan rotation. The white grid represent the SEM screen image corrected for a  $70^\circ$  tilted sample. The grey images behind show the untilted sample which is used as reference image.

The image distortions are not consistent due to variations in sample placement that cause distortions unrelated to the particular machine and will differ every time a sample is placed in the SEM.

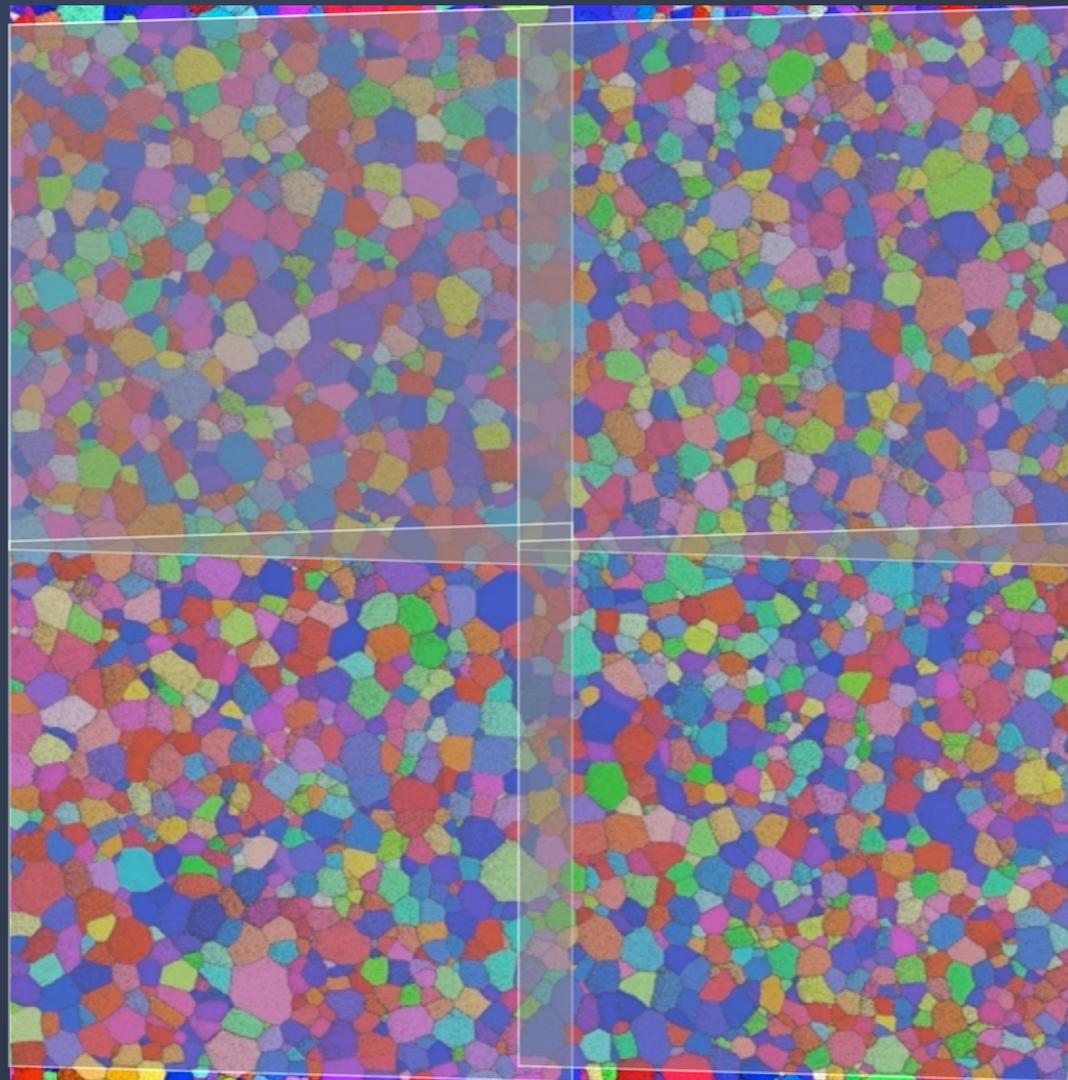
For more information see:  
“Image distortions in SEM and their influences on EBSD measurements” Ultramicroscopy, Vol 107. p172 (2007)

This results in very strict tolerances for the user in placing and aligning the sample, which are going to be very difficult to detect at the time of collection.

# EBSD Image Distortions: Combo Scans

If the scan area is distorted, then the overlapping areas will not scan the correct geometry.

Even a single field of view does not have the correct geometry.



# EBSD Image Distortions: Combo Scans

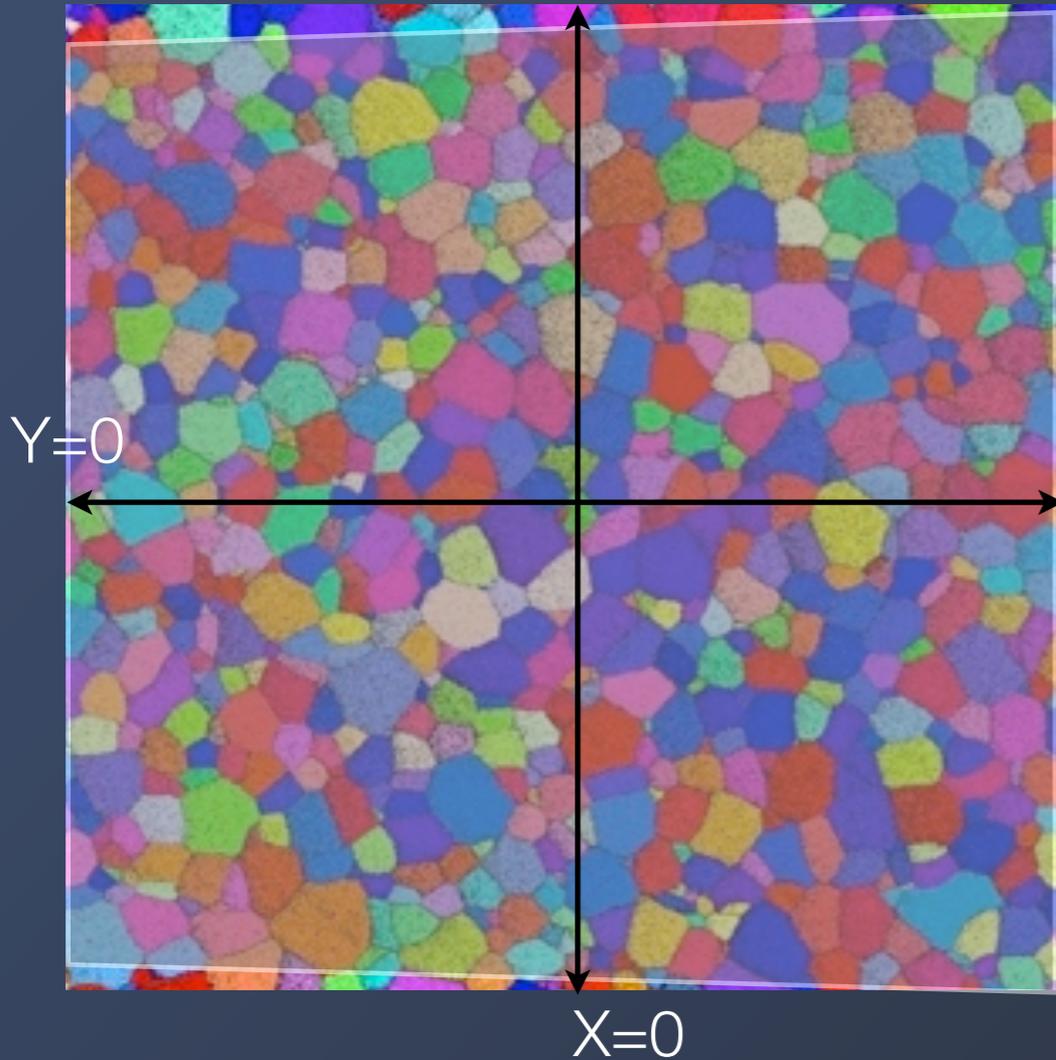


2 x 2 Combo Scan of a 2205 Duplex Stainless Steel. A 1mm laser cut grid outlines the area of interest.

The images have been translated to align the center of the sample.

The edges of the seams do not match.

# Use the duplicate data in overlaps to correct the distortions



-Assume that each tile has the same distortion.

-Assume that the center of each tile will remain unchanged.

-Assume that the distortion can be described by a 2D-polynomial (experience shows that a 3rd degree polynomial is sufficient)

$$x' = \sum_{i,j=1}^3 a_{ij} x^i y^j \quad y' = \sum_{i,j=1}^3 b_{ij} x^i y^j$$

- Only user inputs is the approximate fraction of overlap between the images.

- The degree of overlap between will be maximized for an undistorted image

- Evaluate the amount of overlap through the maximum value of the normalized cross-correlation.

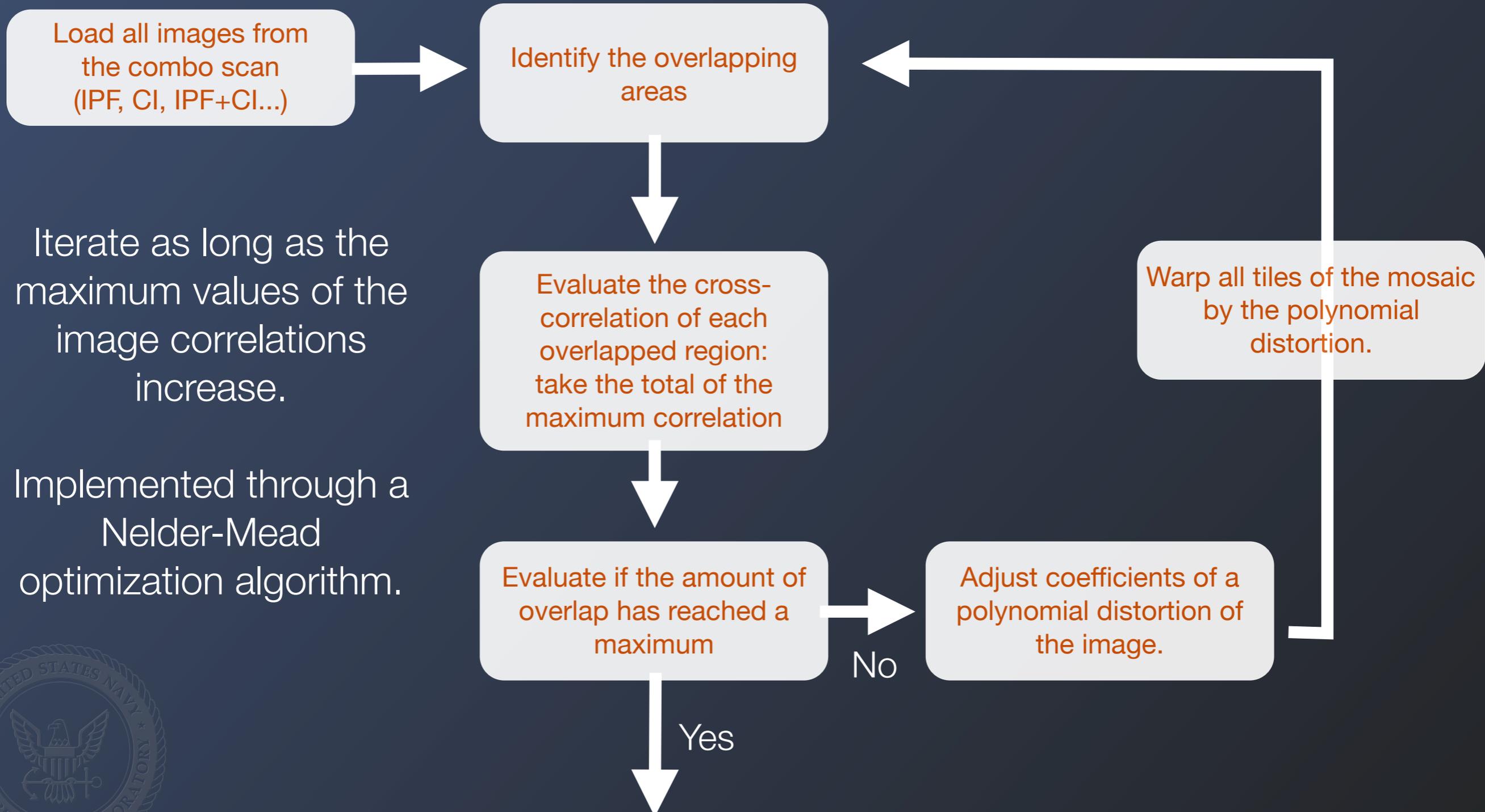
$$I = \max(\text{tile1} \star \text{tile2})$$



# EBSD Mosaic Stitching Algorithm

Key information:

- 1.) We know that the geometries mapped within these regions are the same.
- 2.) We know that the distortion of the raster is the same for every scan.



Evaluate if the amount of overlap has reached a maximum

Distort each tile according to the polynomial  
- Nearest neighbor interpolation

Stitch tiles into unified dataset:  
-Translations determined by cross-correlations of the EBSD images. Optional affine distortion is applied to perfect seams.  
- Data cut at the midline of the overlaps.  
- Write data back out to standard EBSD format.

### Problems:

- The algorithm is *slow*, 6x6 tiles with 600 x 600 pixel/tiles takes 40min on 2.9GHz 16CPU machine
- Not guaranteed to maintain scale.
- Does not necessarily remove all distortions - shear of the image may still be present.
- The solution is not unique.
- Currently the algorithm is “lossy”, poor assumption for a low number of data points (prioritizes matching shapes over maintaining original data)

All of these would be mitigated by lower initial distortions, and well calibrated measurements.

# EBSD Mosaic Stitching Algorithm: Results



1K x 1K image with 1 micron step. 2 X 2 montage.

The location of the seams is nearly imperceptible, only noticeable from small differences in the measured crystal orientation.

# EBSD Mosaic Stitching Algorithm: Results

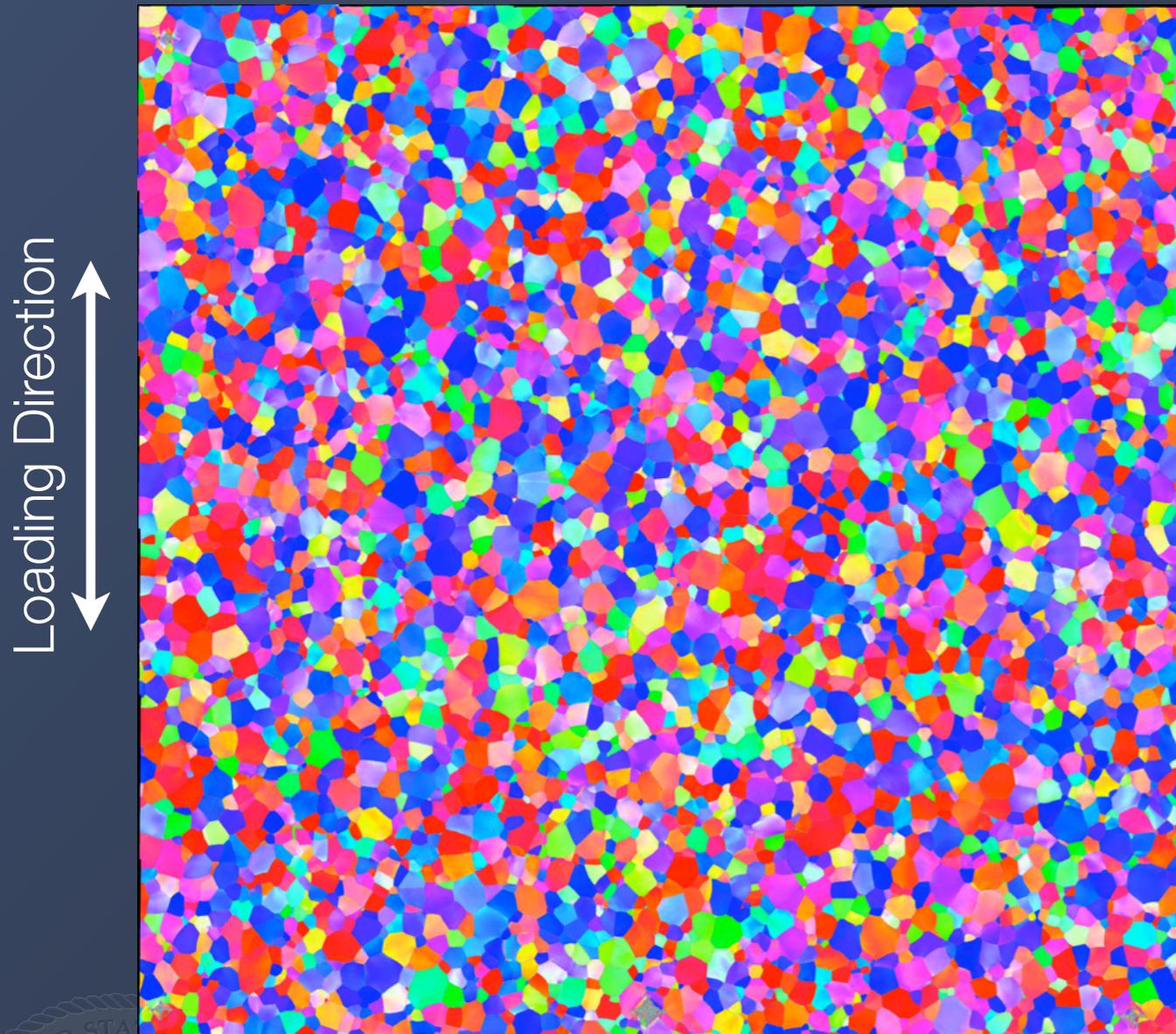


1K x 1K image with 1 micron step. 2 X 2 montage.

The location of the seams is nearly imperceptible, only noticeable from small differences in the measured crystal orientation.

# Ti-21S Serial Sections

Sample of Ti-21S pulled in uniaxial tension to 4% strain.



Eight indents placed around perimeter of the analysis area. Typically 1 indent replaced every-other section.

Mechanically polish material away: Allied Multi-Prep; 1 $\mu$ m diamond (10min) + 0.04 $\mu$ m silica (20min). Removal 1.4 $\mu$ m per section.

EDAX/TSL Combo EBSD scan with a 0.6  $\mu$ m step, over a 2.1 x 2.1 mm area (6 x 6 mosaic tiles, 15% overlap). 8x8 pattern binning, 290points/s. 15 hr scan.

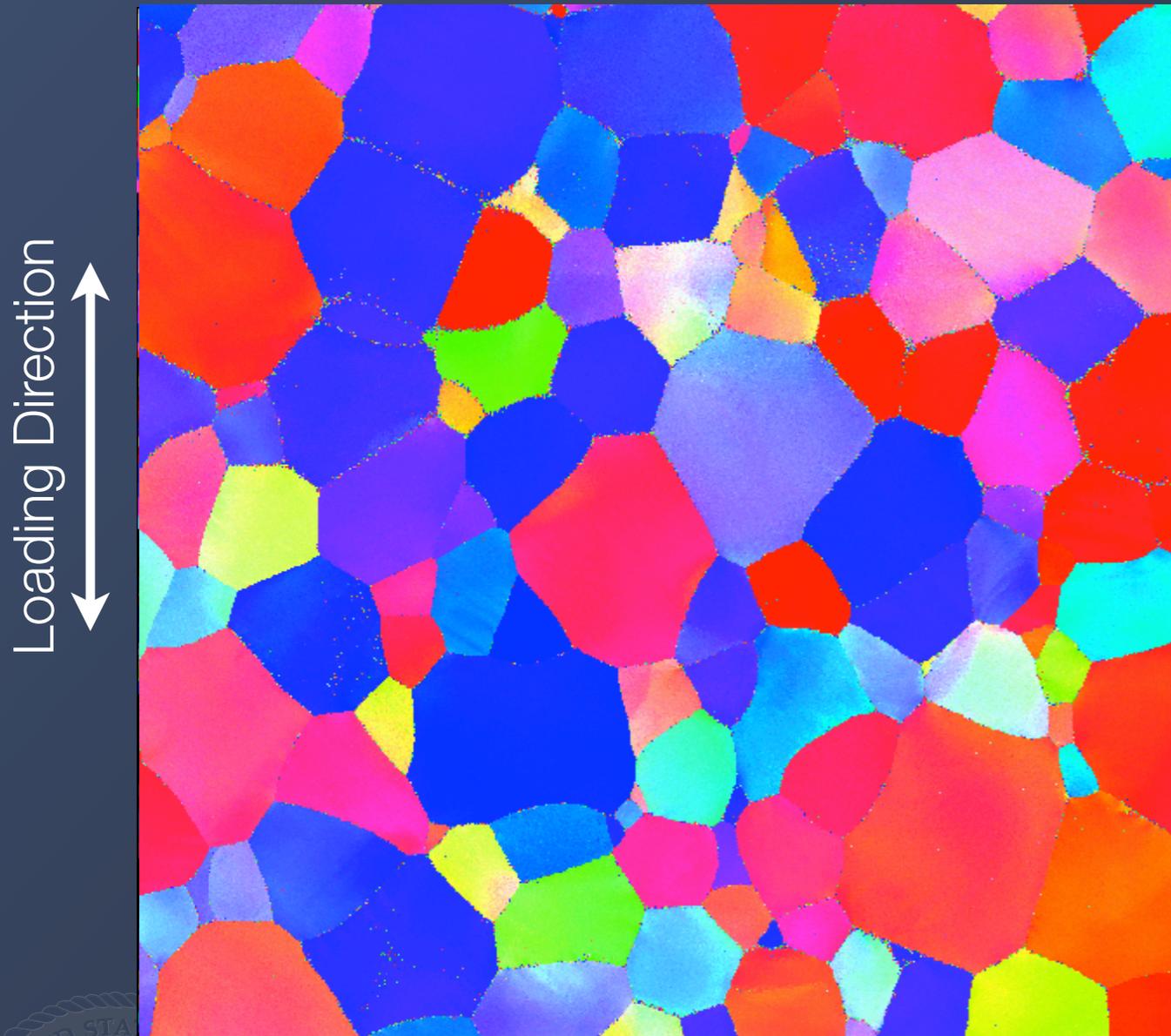
Entire user interaction with sample 1.5-2Hours (depends on SEM alignments)

To date 5 sections have been collected...



# Ti-21S Serial Sections

Sample of Ti-21S pulled in uniaxial tension to 4% strain.



Eight indents placed around perimeter of the analysis area. Typically 1 indent replaced every-other section.

Mechanically polish material away: Allied Multi-Prep; 1 $\mu$ m diamond (10min) + 0.04 $\mu$ m silica (20min). Removal 1.4 $\mu$ m per section.

EDAX/TSL Combo EBSD scan with a 0.6  $\mu$ m step, over a 2.1 x 2.1 mm area (6 x 6 mosaic tiles, 15% overlap). 8x8 pattern binning, 290points/s. 15 hr scan.

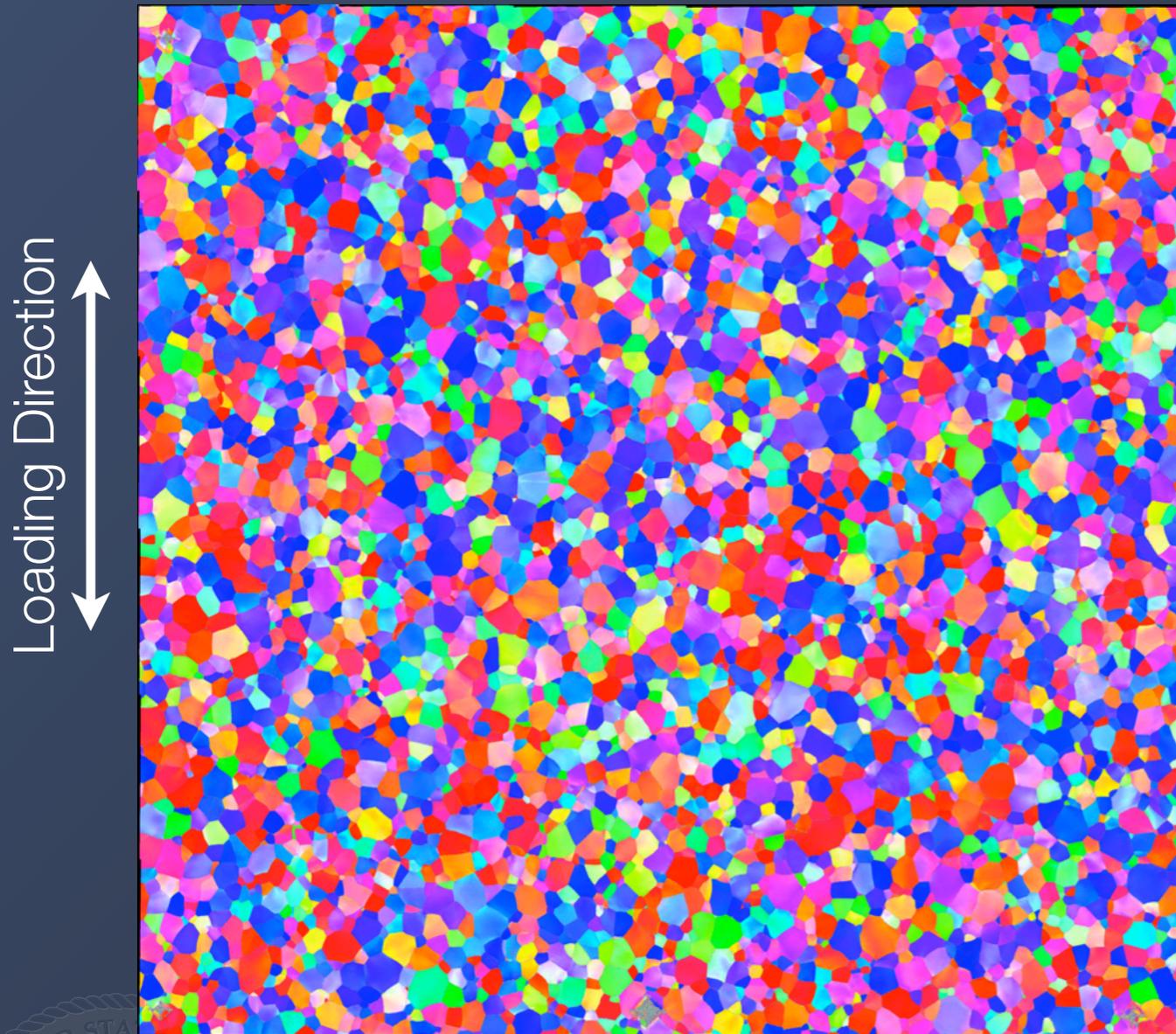
Entire user interaction with sample 1.5-2Hours (depends on SEM alignments)

To date 5 sections have been collected...



# Ti-21S Serial Sections

Sample of Ti-21S pulled in uniaxial tension to 4% strain.



Eight indents placed around perimeter of the analysis area. Typically 1 indent replaced every-other section.

Mechanically polish material away: Allied Multi-Prep; 1 $\mu$ m diamond (10min) + 0.04 $\mu$ m silica (20min). Removal 1.4 $\mu$ m per section.

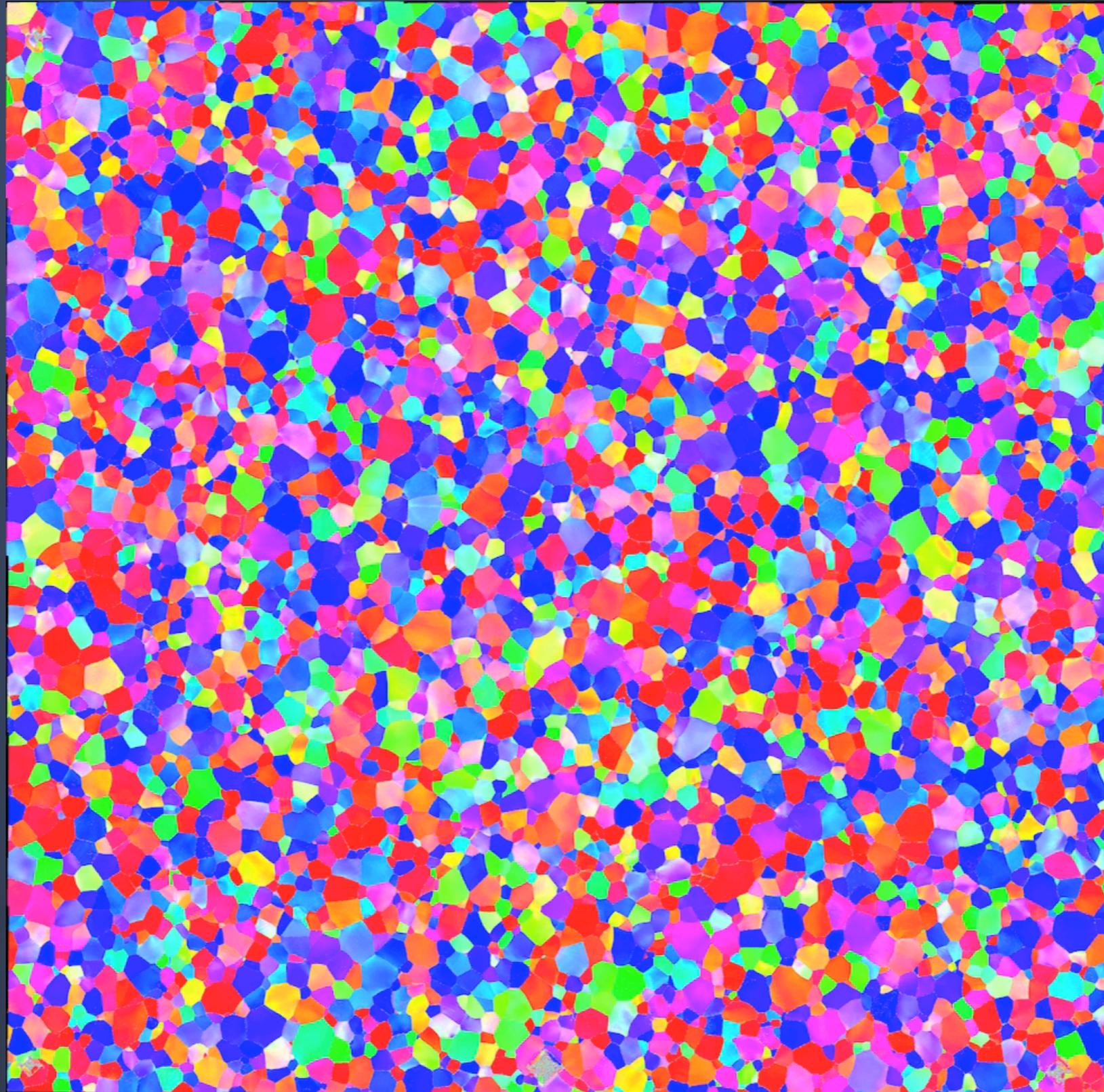
EDAX/TSL Combo EBSD scan with a 0.6  $\mu$ m step, over a 2.1 x 2.1 mm area (6 x 6 mosaic tiles, 15% overlap). 8x8 pattern binning, 290points/s. 15 hr scan.

Entire user interaction with sample 1.5-2Hours (depends on SEM alignments)

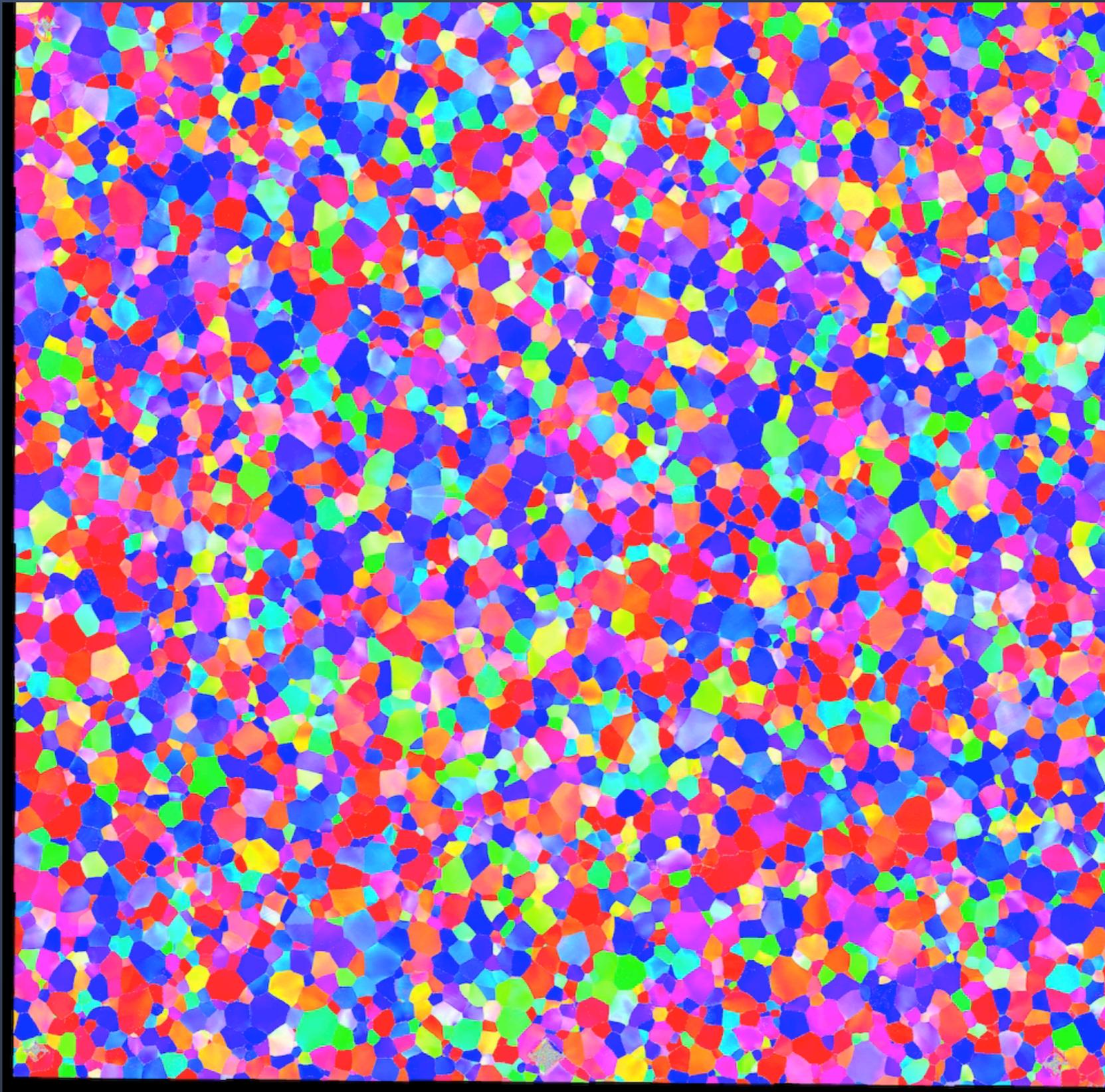
To date 5 sections have been collected...



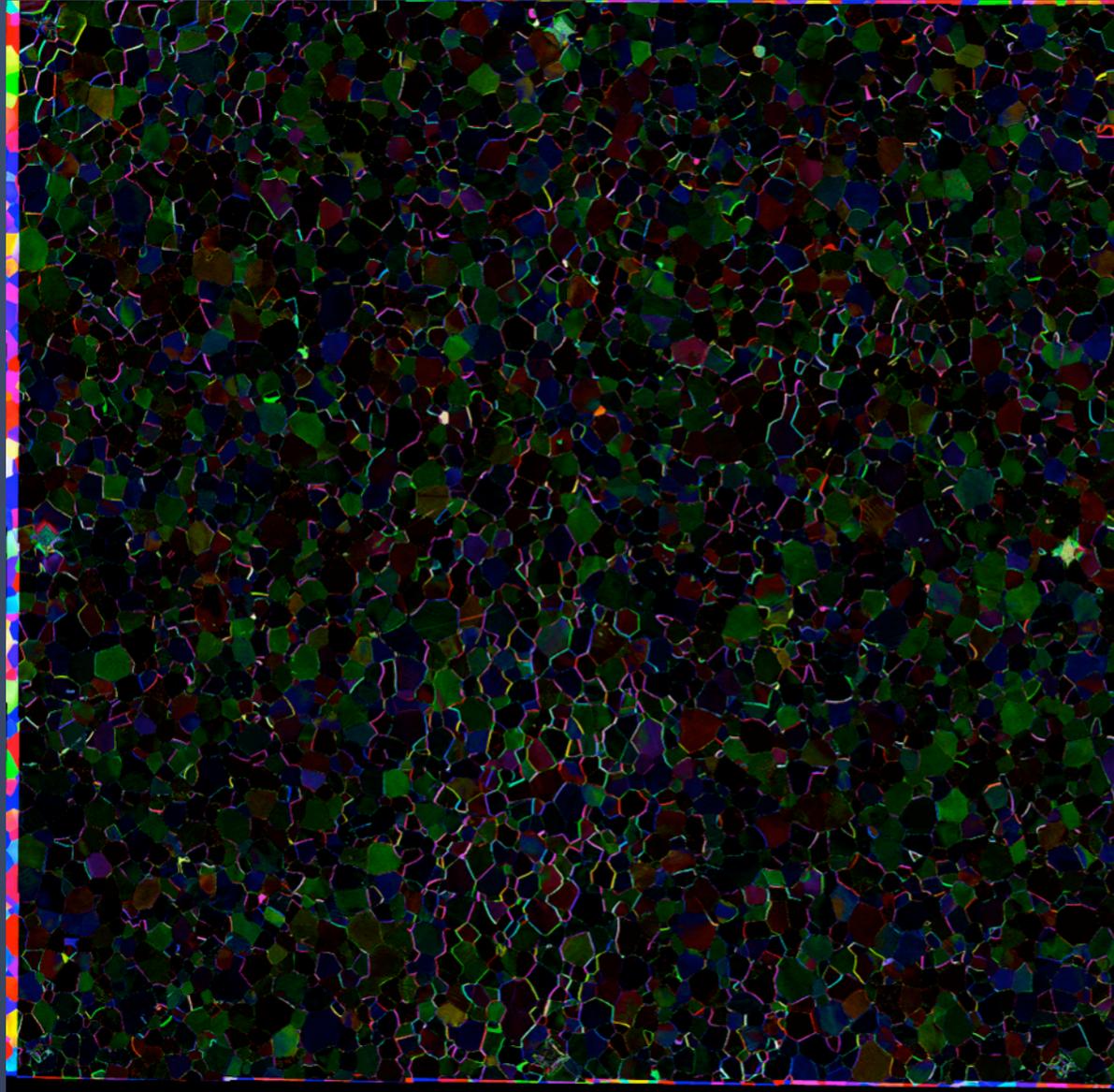
# Section-to-section alignment



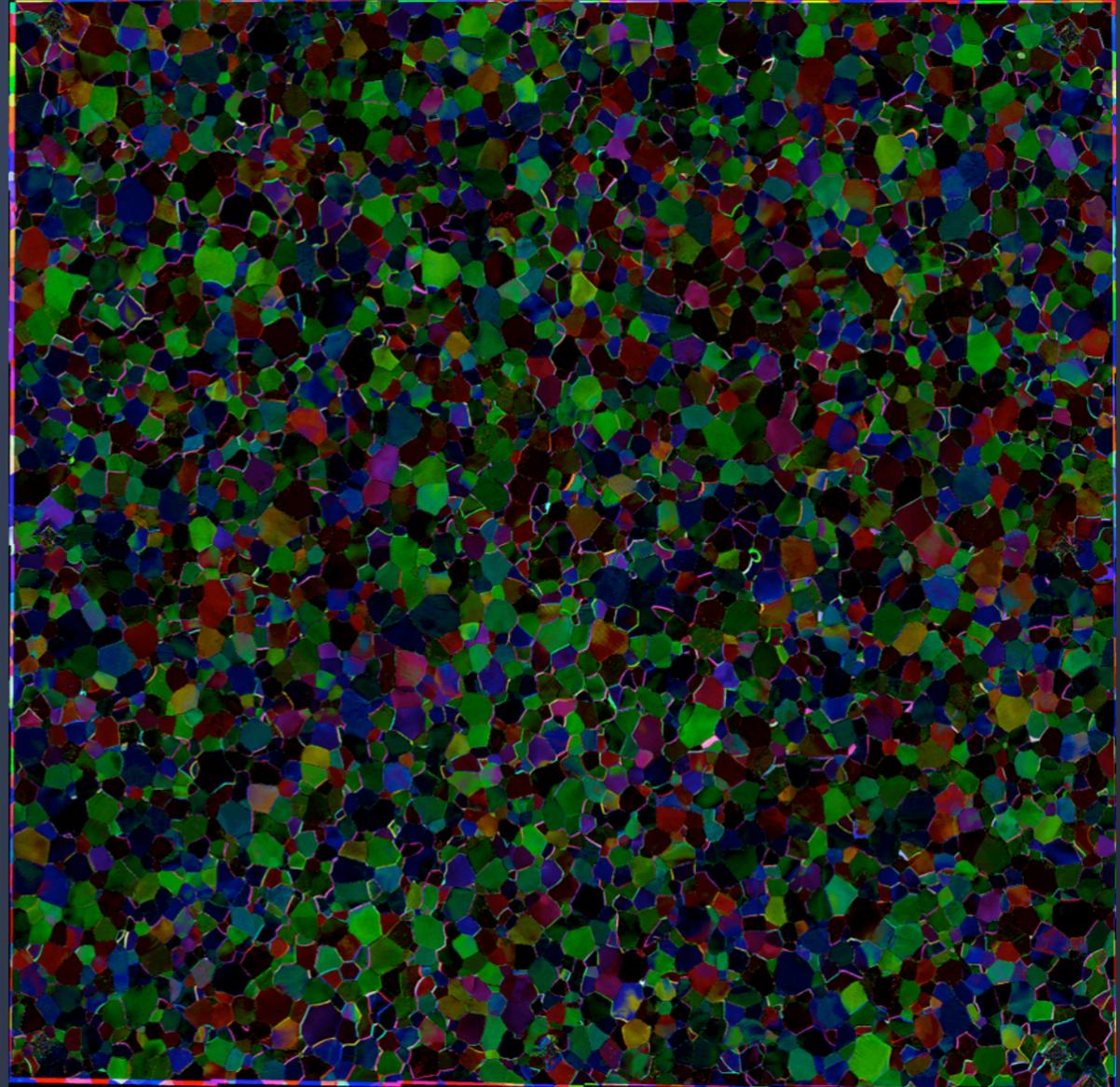
# Section-to-section alignment



# Section IPF differences



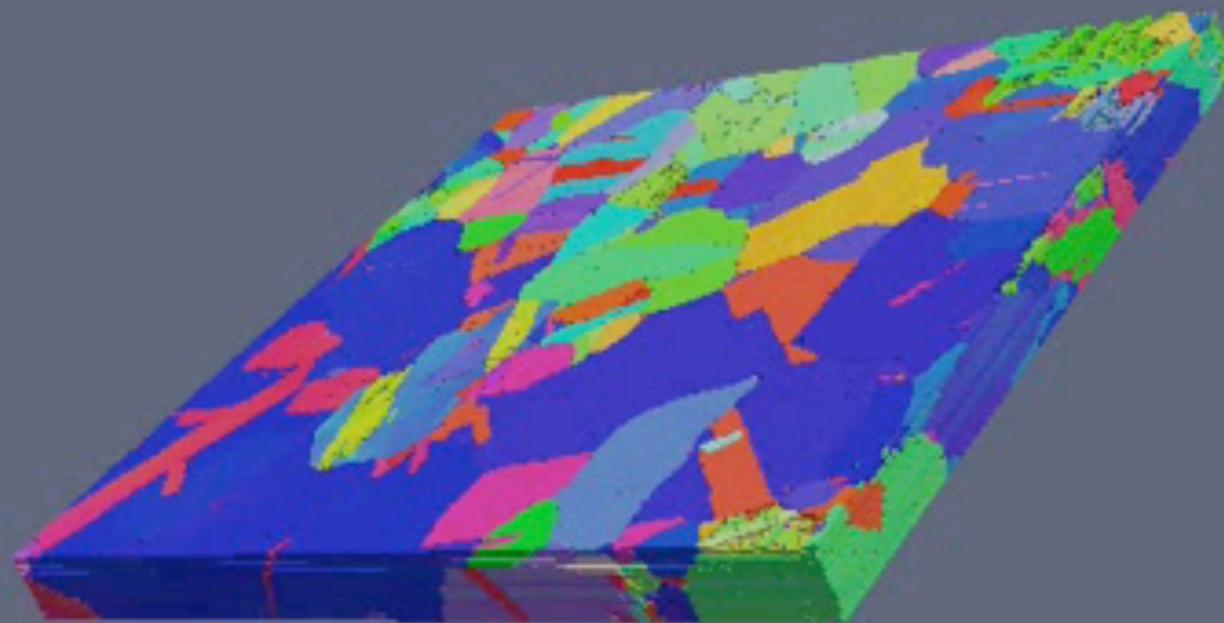
Non-uniques of stitching solutions



Orientation differences of the positioning of the sample in the SEM during EBSD collection.



# 316 Stainless steel (Alexis Lewis, Amanda Levinson)



1/16 of total volume, 500 x 500 x 20 microns

Reconstructed with NRL IDL programs and DREAM.3D



# Lessons learned

New EBSD acquisition hardware and software have made large area EBSD sectioning feasible.

The advantage of EBSD images is that it greatly reduces the image segmentation problems for serial sectioning, and reduces what problems are left to more general problems that can be applied across alloy types, and material classes.

This begins to make up for the significant investment in data acquisition, that is largely automated.

Early results point out the need for better repeatability of sample mounting with in the SEM (or at least the ability to measure the differences).

A dedicated SEM makes the process easier, less problems with beam alignments and stability.



# What can we do better...

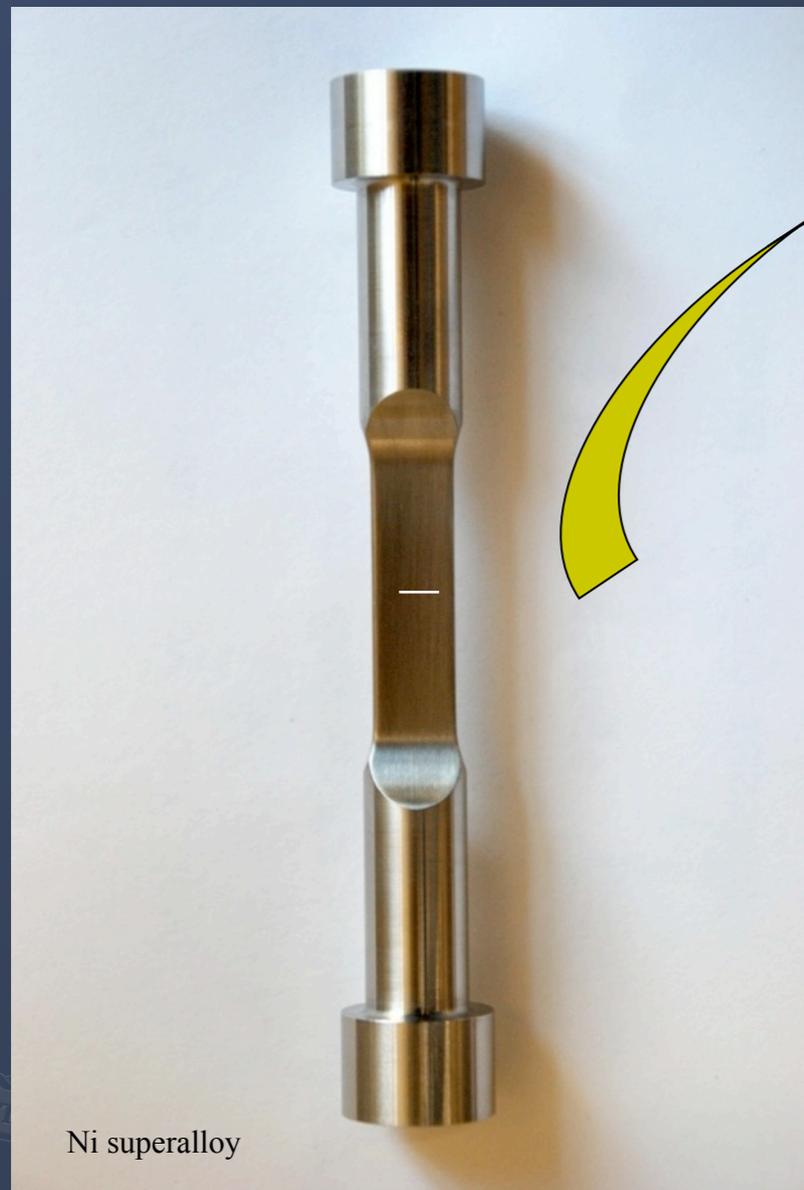
What can be done to make near zero-segmentation images practical solution for serial sectioning?

- Faster image collection. Currently EBSD systems can collect  $\approx 900$  points/s. Is this the top?
- Automation. The sectioning sample preparation does not need human interaction.
  - Automated polishing (Robomet) + sample delivery system to an SEM (Robots!). (See M. Uchic at AFRL, soon NRL).
  - Move sample removal into the SEM (need a much larger removal rates than a FIB) Femtosecond laser! (See T. Pollack and M. Echlin UCSB), Xe-Ion sources.
  - Automate electron imaging.
- Data fusion. Combining lower resolution images with backscattered,fore scattered, EDS, ...
  - Need better solutions than manual alignment. For instance...

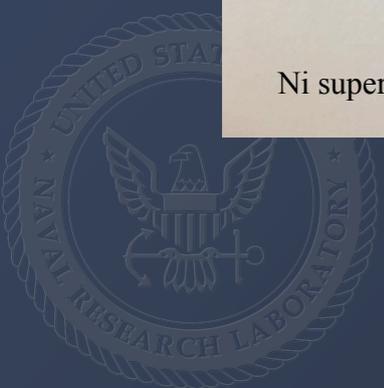
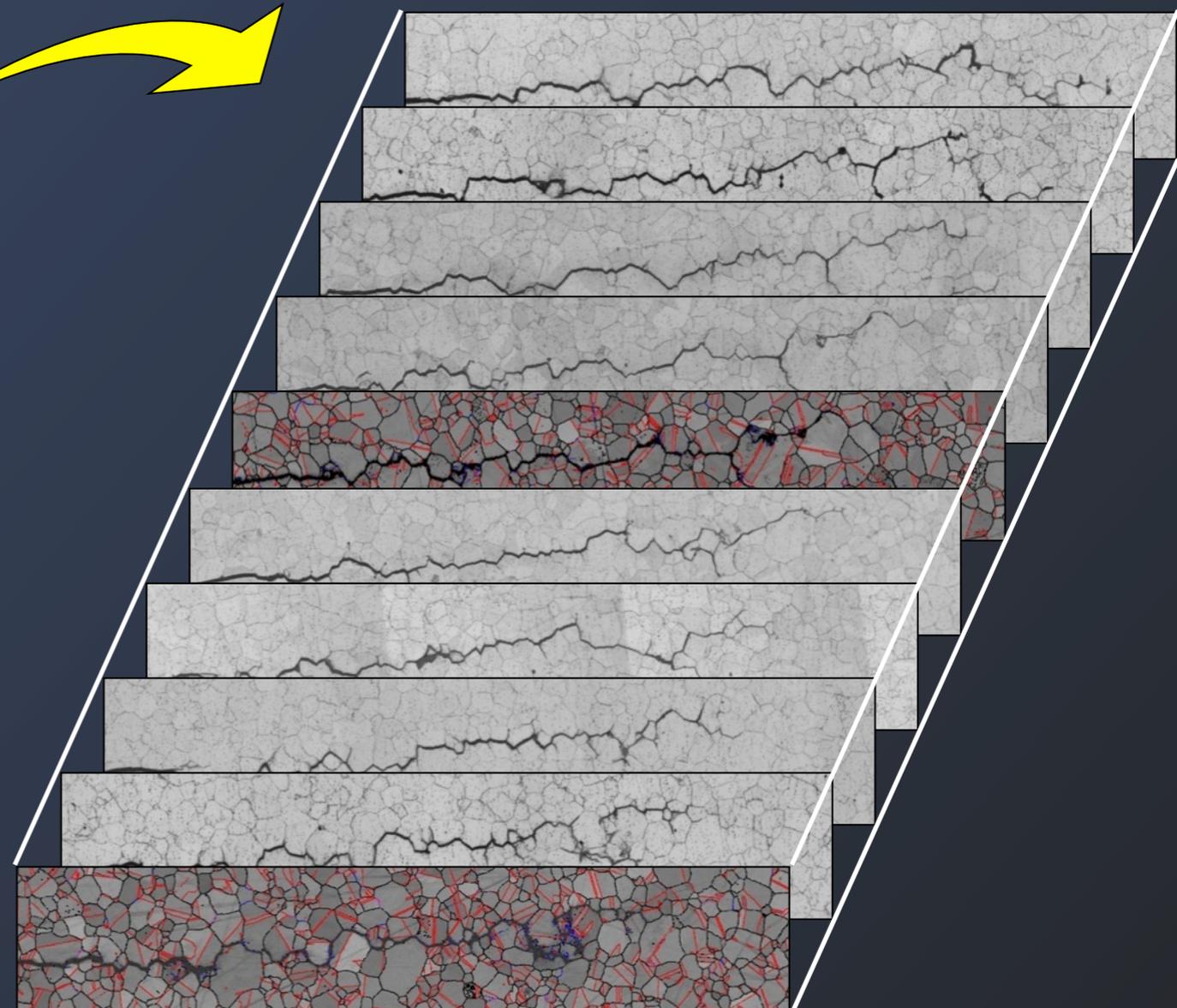


# Examination of Fatigue Cracks in Ni-based superalloy (A. Deal GE Global Research)

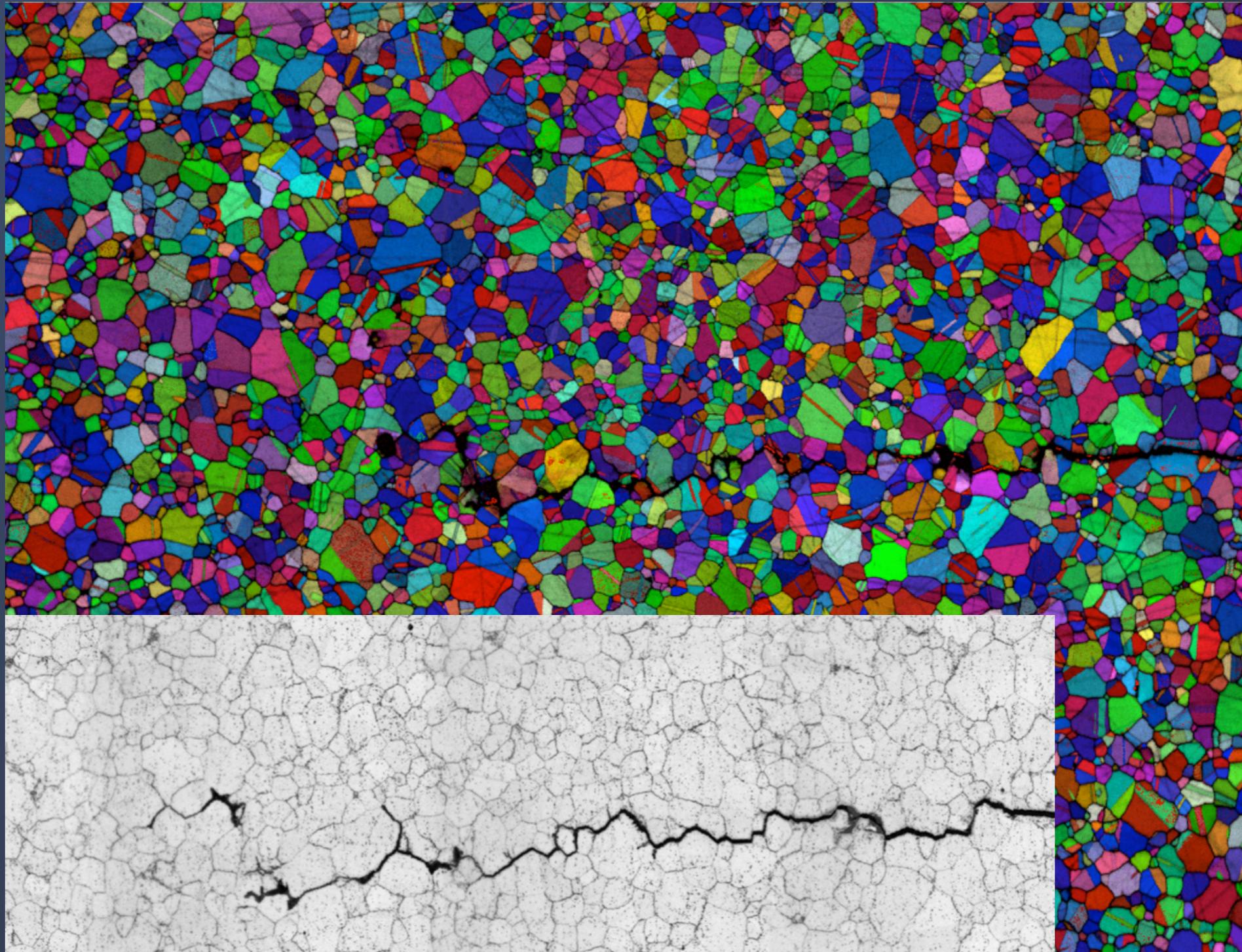
KB Bar Fatigue Test



3D EBSD/Optical Analysis – exploded view of a few cross-sectional slices



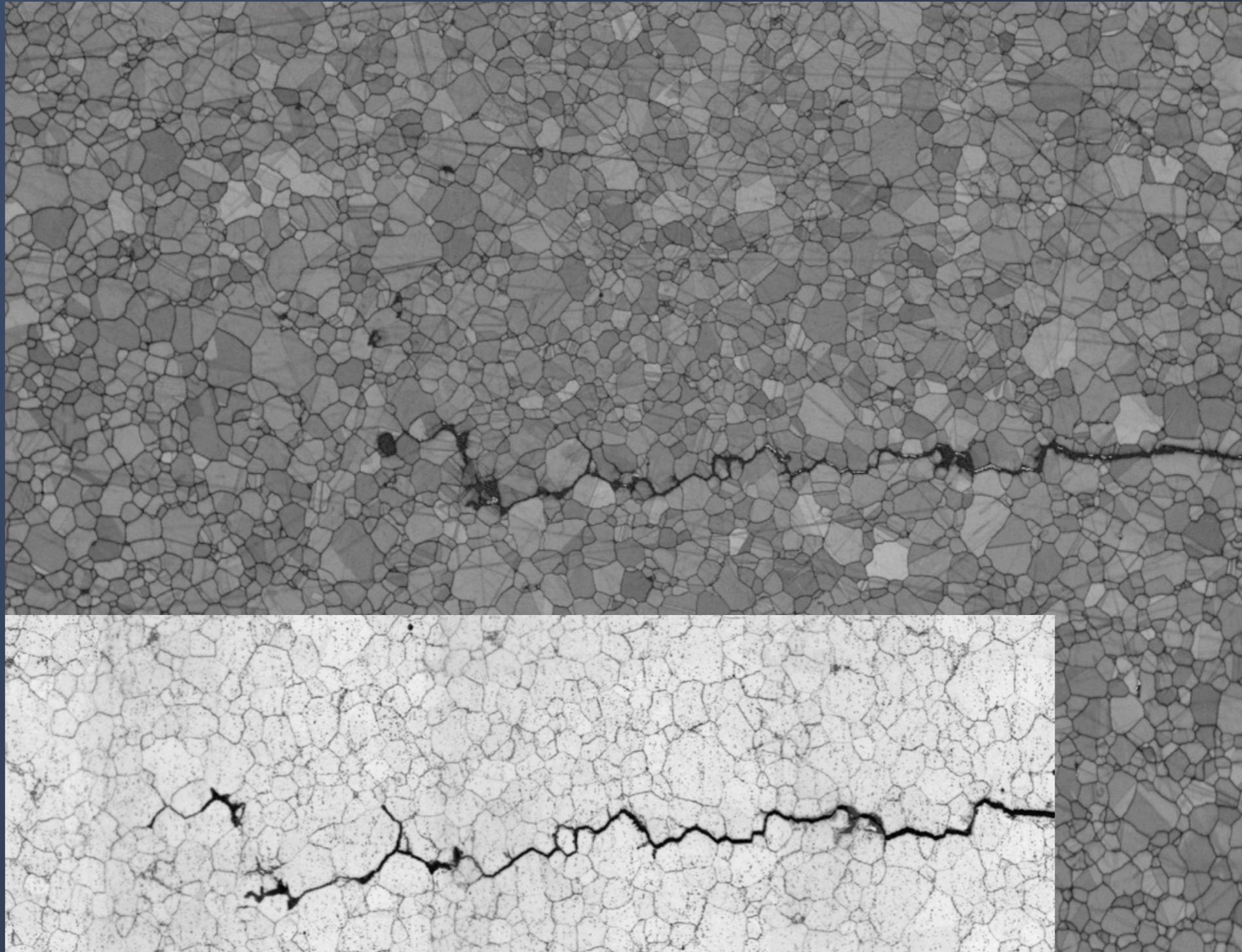
# Image matching



Match images using Image Entropy and Mutual Information



# Image matching



Match images using Image Entropy and Mutual Information



# Image Entropy

*E.B. Gulsoy J.P. Simmons and M. DeGraef; Scripta Mat. 2009:p381*

Image entropy is a measure of the amount of information in an image.

The entropy of one image is given by:

$$H = - \sum_{i=0}^{255} p_i \ln p_i$$

The entropy of “mixing” the brightness of pixels of brightness values.  
 $p_i$  is the probability of finding a pixel of intensity  $i \rightarrow$  histogram of the image.

The joint entropy of two images is given by:

$$H(\text{im}_1, \text{im}_2) = - \sum_{i,j=0}^{255} p_{i,j} \ln p_{i,j}$$

$p_{i,j}$  is the probability of finding a pixel of intensity  $j$  in image2, given the probability of finding the probability of having a pixel with intensity  $i$  in image1  $\rightarrow$  joint histogram of the image.

The joint entropy is a measure of the amount of additional information that having a second provides.



# Joint Histogram

Image1

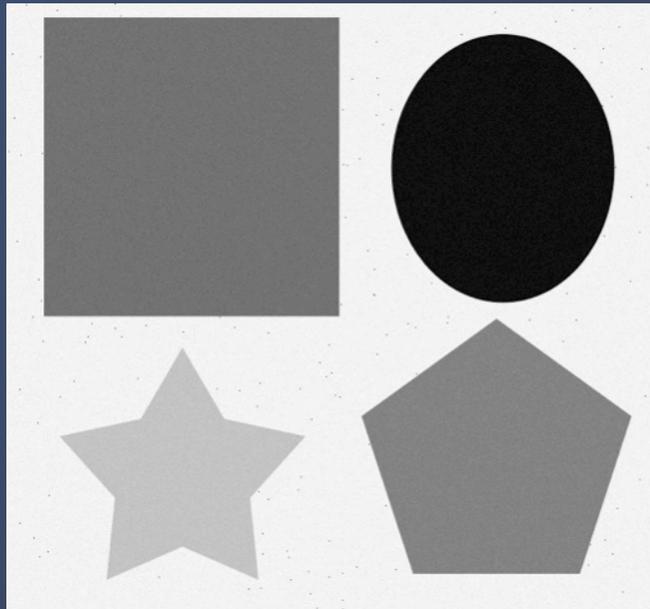
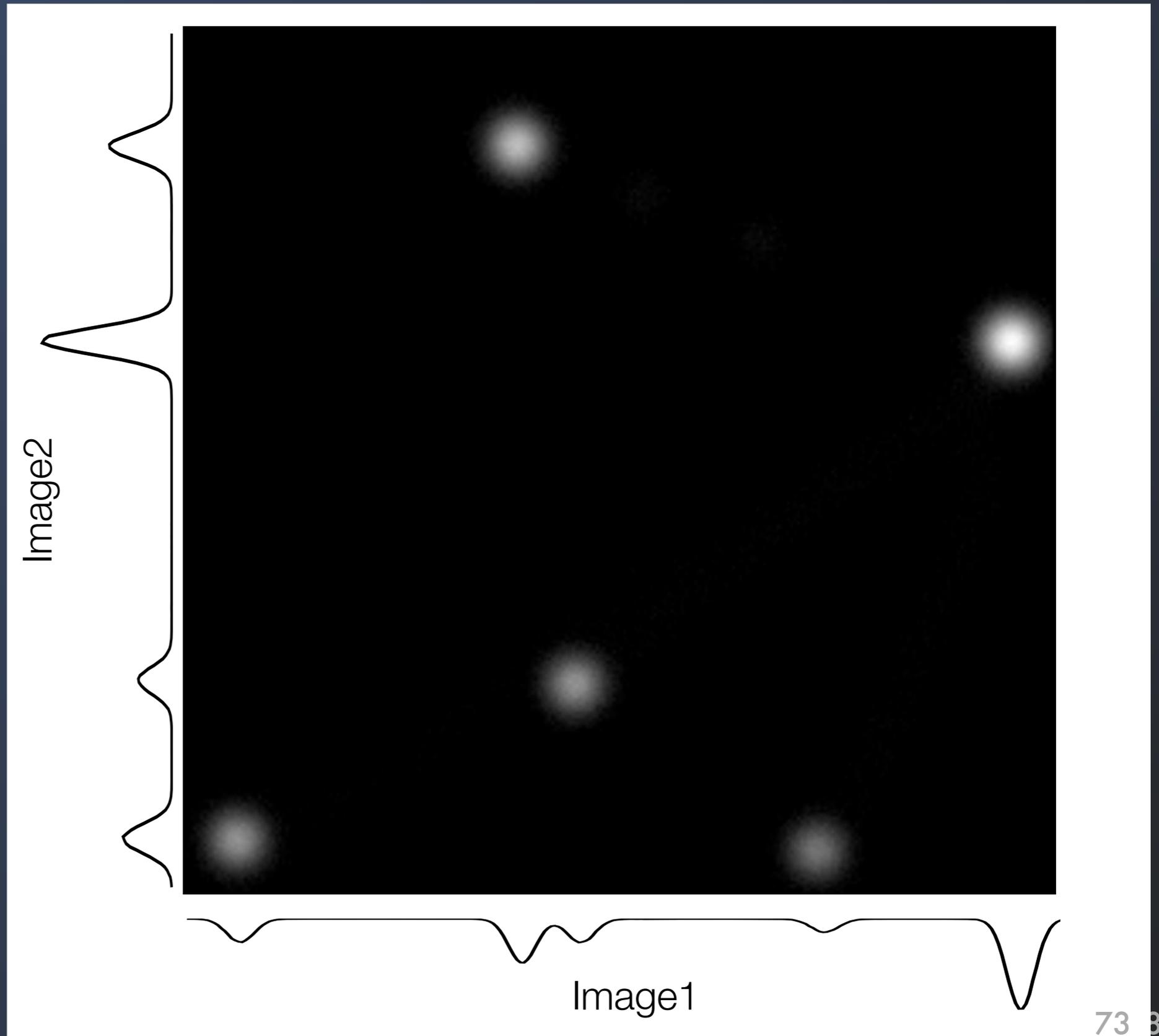
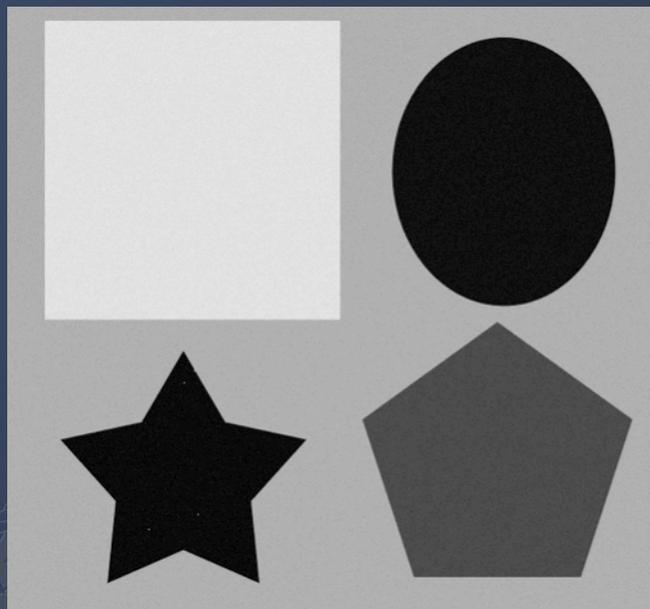


Image2



# Mutual Information

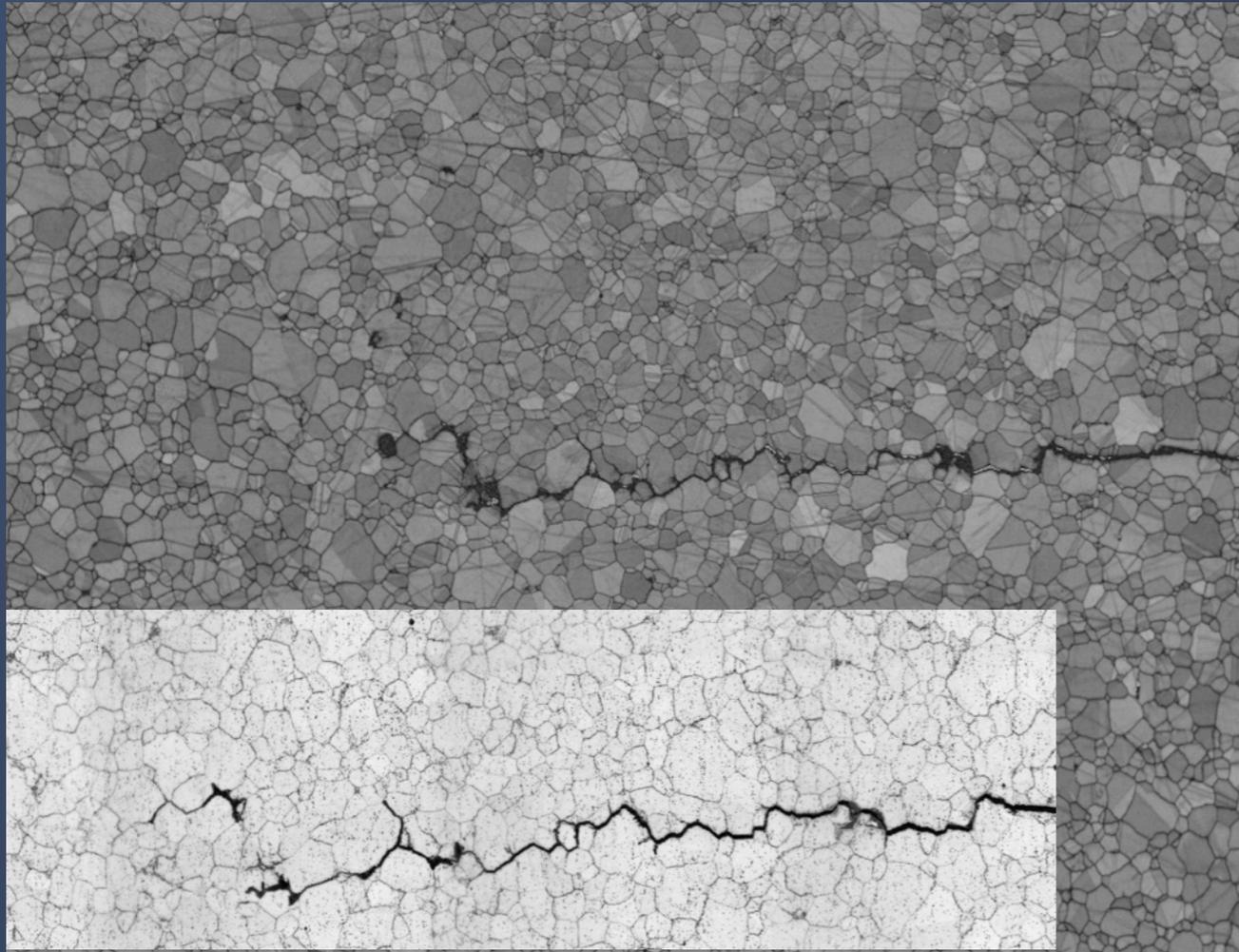
The mutual information measures the degree of shared information between two (or more) images.

$$MI = H(im_1) + H(im_2) - H(im_1, im_2)$$

Thus the mutual information provides a figure of merit as to the degree of overlapping information between two images, even if those images do have the same brightness values.



# Image matching using MI



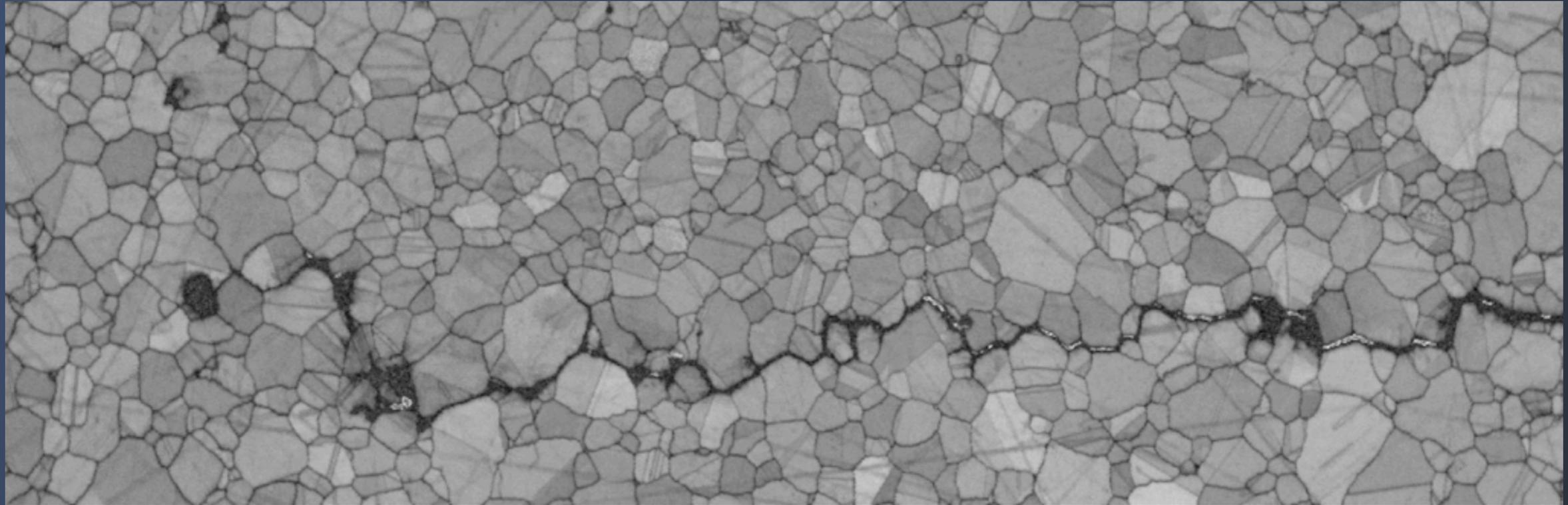
We use a  $N$  degree polynomial warping field to distort the EBSD image so that the Mutual Information between the two images is maximized.

This is done using a Nelder & Mead optimization (AKA Amoeba minimization, or downhill simplex method).

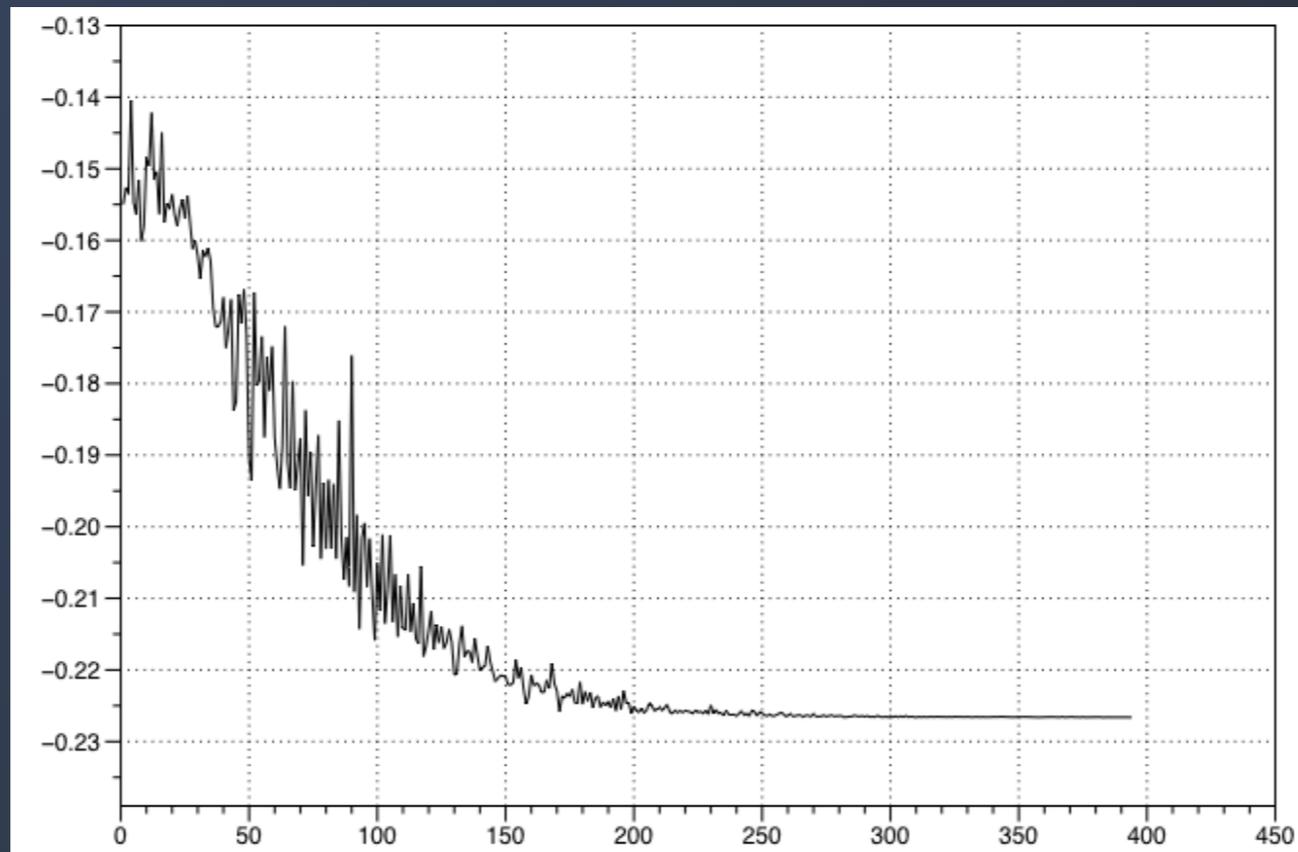
Provide an initial guess for the parameters:  
image translations and scaling changes (must be good guesses)  
rotations appear to be fairly robust up to  $5^\circ$



# Image matching using MI



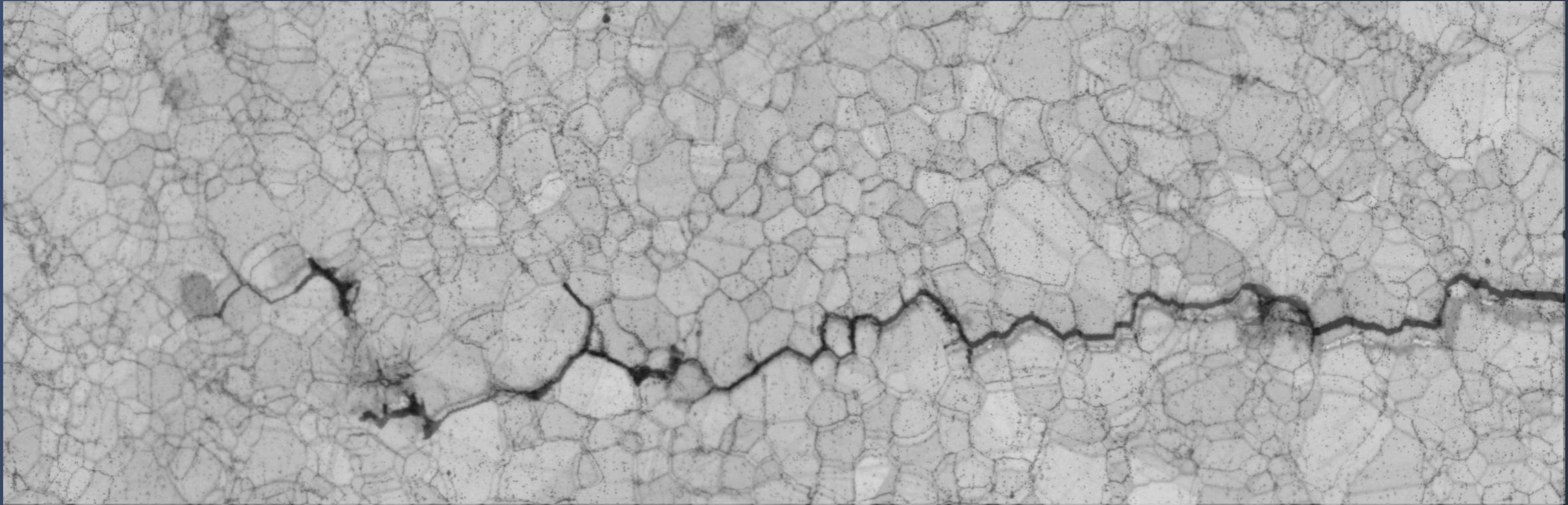
-(Mutual Information)



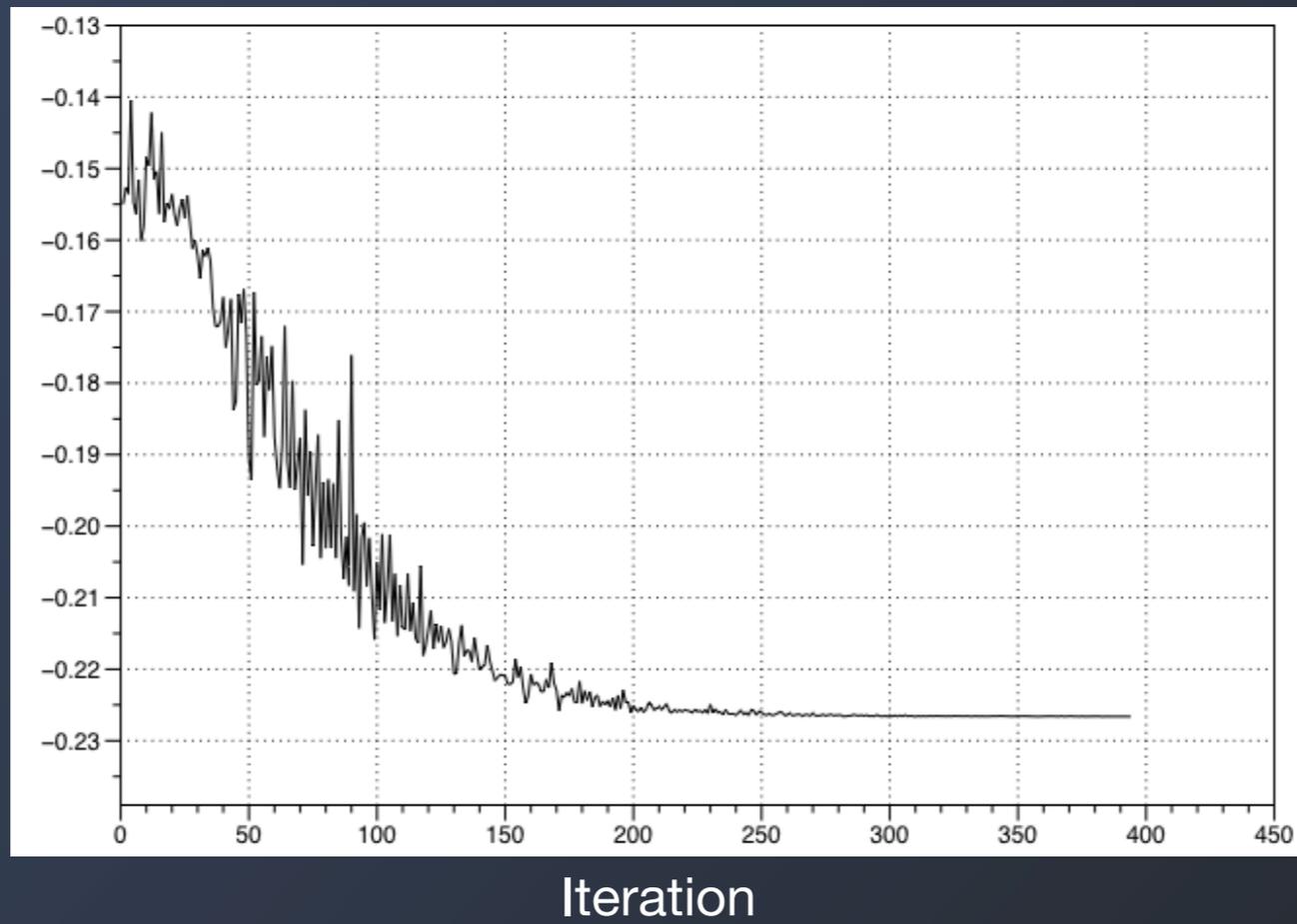
Iteration



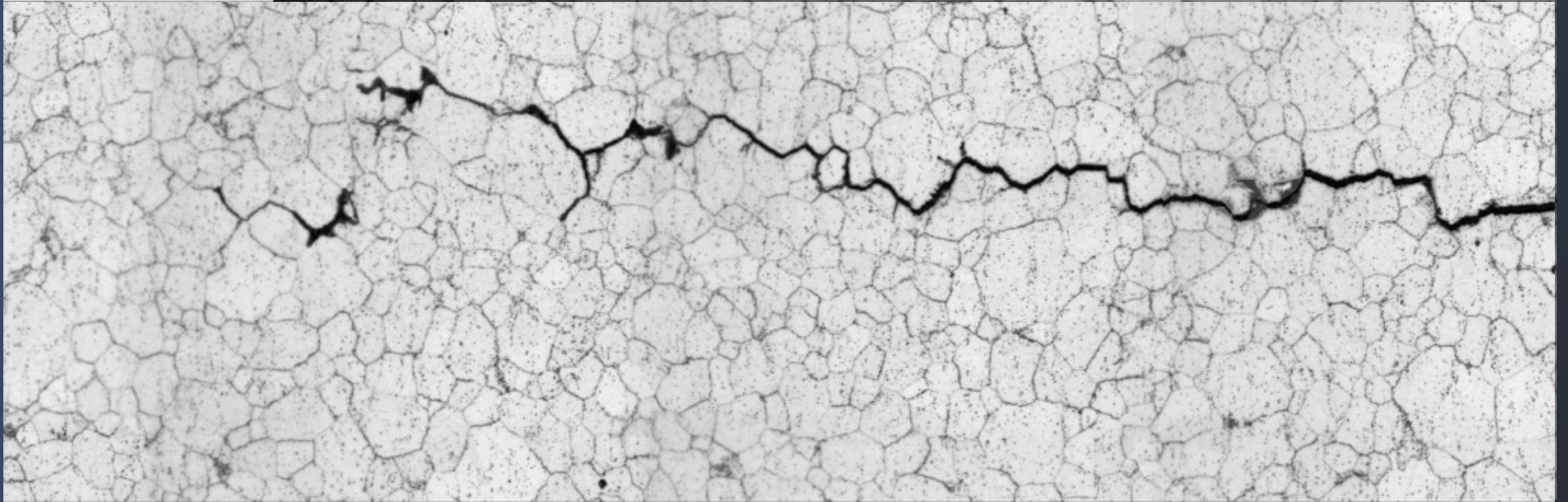
# Image matching using MI



-(Mutual Information)



# Automatic Pre-segmentation Matching

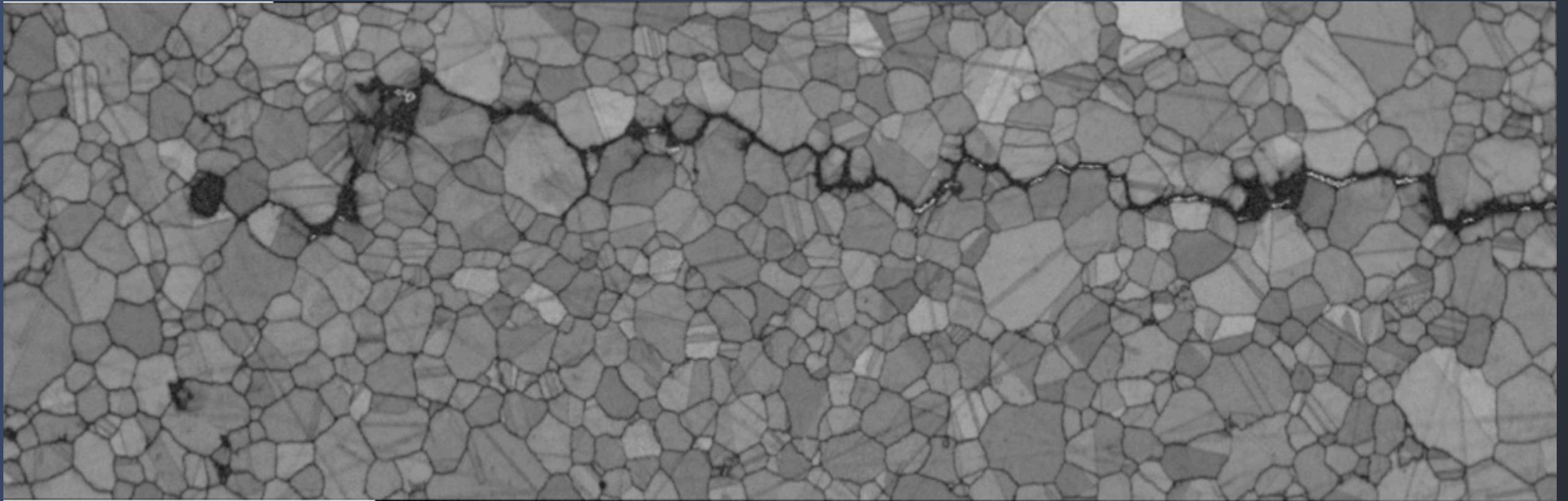


Now can use the EBSD images to help segment the optical/SEM images leveraging the full utility of multiple imaging methods.

Here we used the higher resolution optical to segment the crack, then used the EBSD to include crystallography at the crack.



# Automatic Pre-segmentation Matching

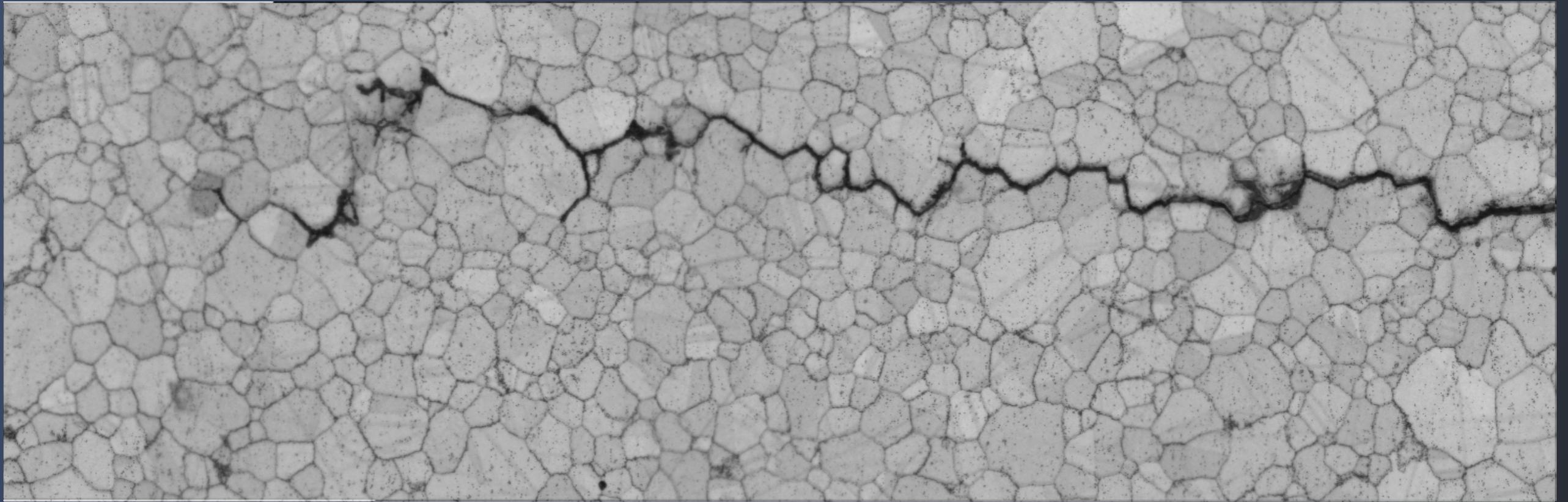


Now can use the EBSD images to help segment the optical/SEM images leveraging the full utility of multiple imaging methods.

Here we used the higher resolution optical to segment the crack, then used the EBSD to include crystallography at the crack.



# Automatic Pre-segmentation Matching

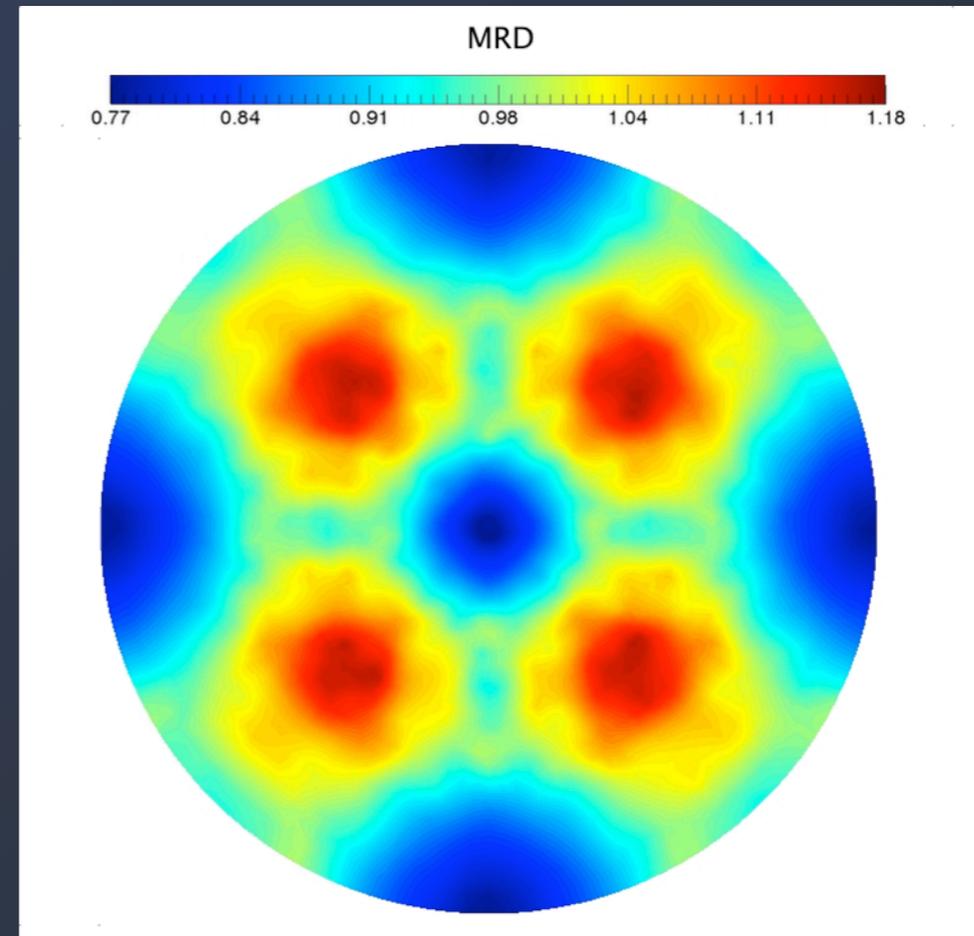
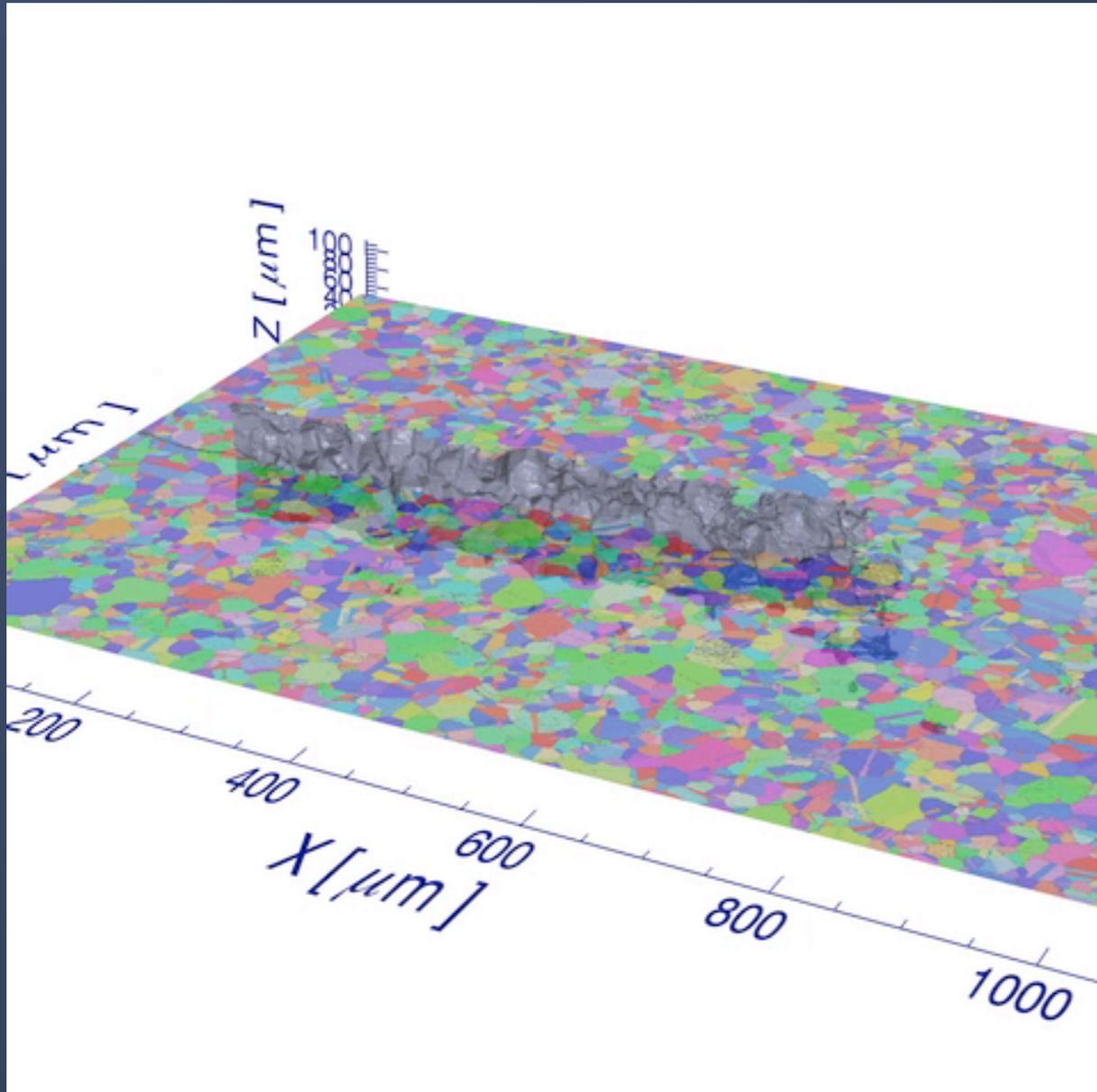


Now can use the EBSD images to help segment the optical/SEM images leveraging the full utility of multiple imaging methods.

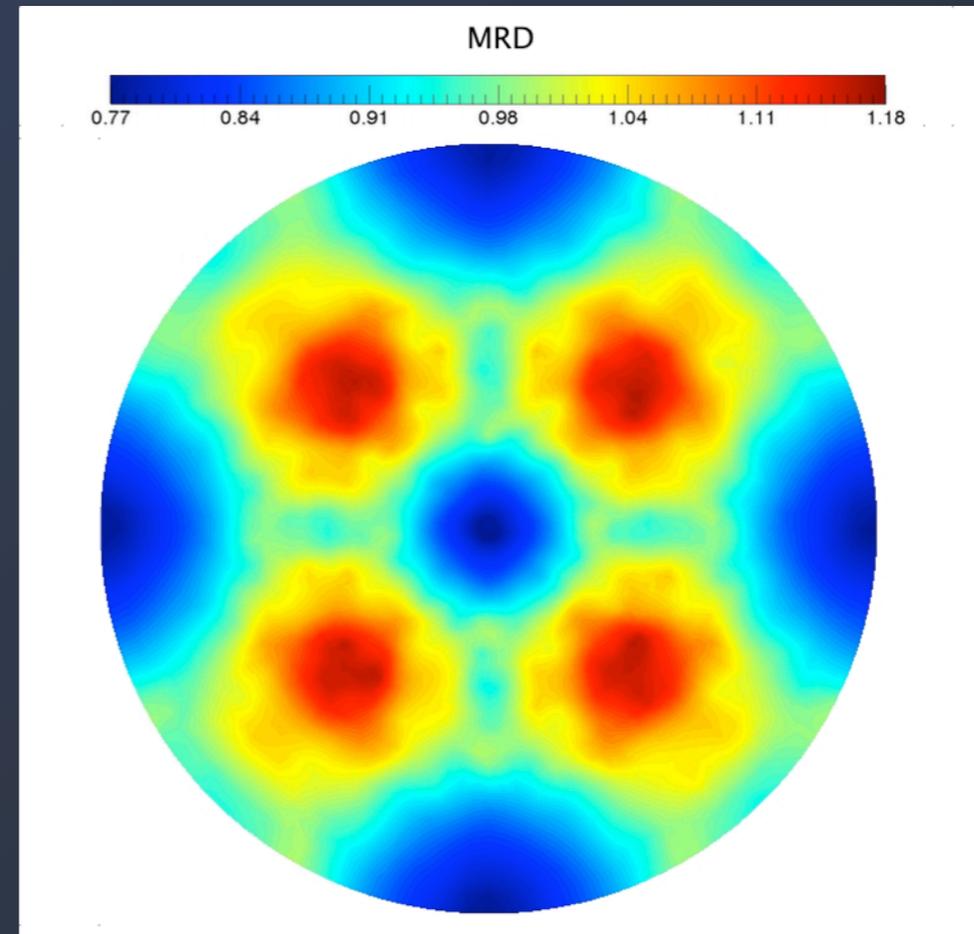
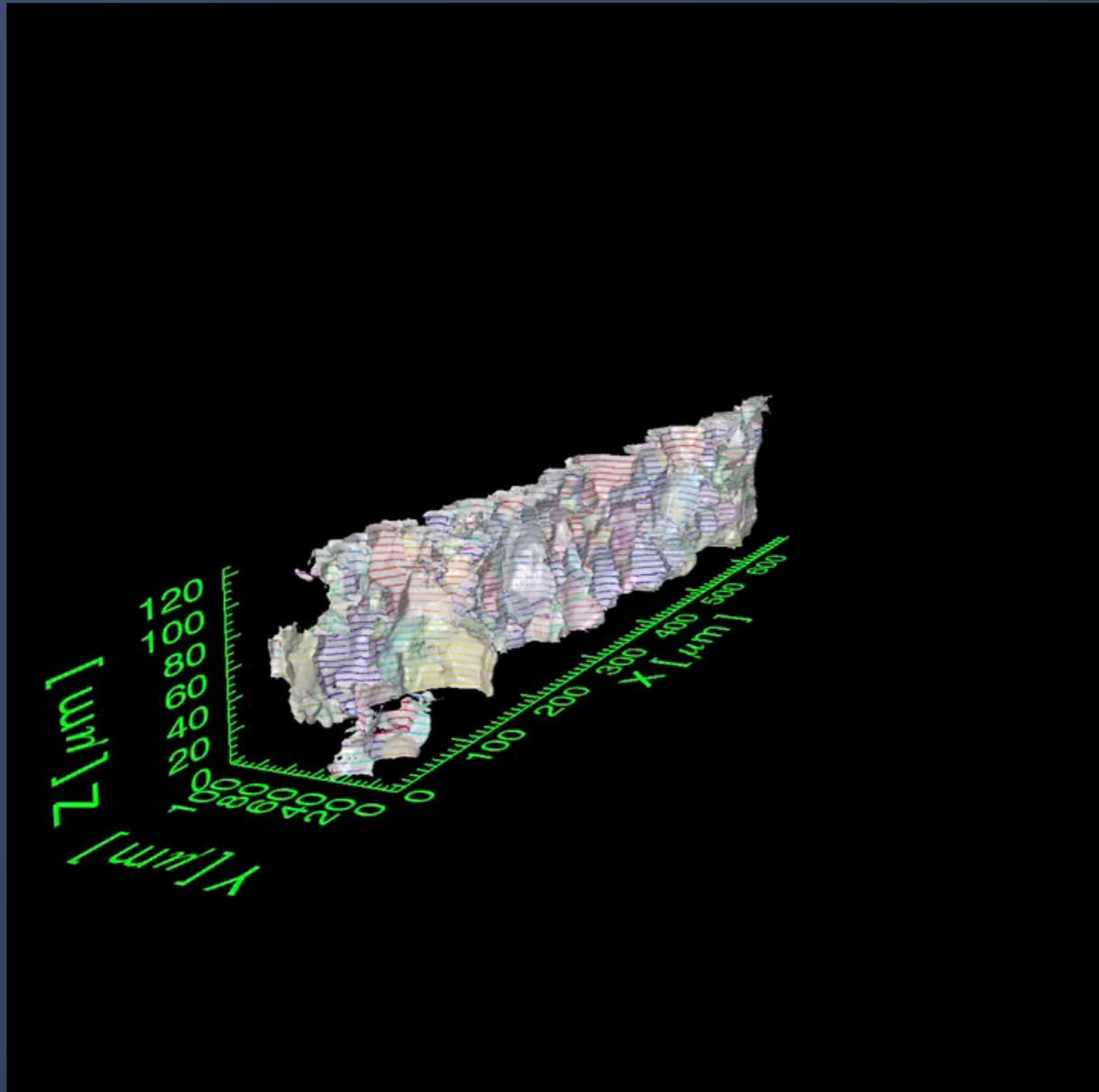
Here we used the higher resolution optical to segment the crack, then used the EBSD to include crystallography at the crack.



# Crack interface normal distribution



# Crack interface normal distribution



Questions?