MTEX

an open source texture analysis toolbox

Ralf Hielscher

TU Chemnitz, Germany

Belo Horizonte, 2015

What is MTEX?

1. a MATLAB toolbox for quantitative texture analysis
2. a scripting language
3. a tool for generating publication ready plots
4. large, well documented and exhaustively tested
5. free to use, to extend, to modify

```
>> startup_mtex

  MTEX 4.1.1 (show documentation)
  Import pole figure data
  Import EBSD data
  Import ODF data

  Uninstall MTEX

>> s = sin([0 30 60]*degree)
s =
         0    0.5000    0.8660
>>
```

What is MTEX?

1. a MATLAB toolbox for quantitative texture analysis
2. a scripting language
3. a tool for generating publication ready plots
4. large, well documented and exhaustively tested
5. free to use, to extend, to modify

why scripts?

- *reproducible results*
- *templates for common tasks*
- *extensively customizable*
- *batch processing of many data sets*
- *repeated calculations with different parameters*

What is MTEX?

1. a MATLAB toolbox for quantitative texture analysis
2. a scripting language
3. a tool for generating publication ready plots
4. large, well documented and exhaustively tested
5. free to use, to extend, to modify

```
ebsd = loadEBSD('mylonite.txt') % load data

grains = calcGrains(ebsd)       % reconstruct grains

[m,id] = max(grains.area)       % find largest grain

plot(grains(id))                % plot largest grain
```

What is MTEX?

1. a MATLAB toolbox for quantitative texture analysis
2. a scripting language
3. a tool for generating publication ready plots
4. large, well documented and exhaustively tested
5. free to use, to extend, to modify

```
ebsd = loadEBSD('mylonite.txt')
```

```
ebsd = EBSD (show methods, plot)

 Phase  Orientations      Mineral         Color  Symmetry
     1     3444 (28)     Andesina   light blue        -1
     2     3893 (31)       Quartz  light green      -3m1
     3     368 (2.9)      Biotite    light red     12/m1
     4     4781 (38)  Orthoclase         cyan     12/m1

 Properties: x, y, bc, mad
 Scan unit : um
```

What is MTEX?

1. a MATLAB toolbox for quantitative texture analysis
2. a scripting language
3. a tool for generating publication ready plots
4. large, well documented and exhaustively tested
5. free to use, to extend, to modify

```
ebsd = loadEBSD('mylonite.txt')
grains = calcGrains(ebsd)
```

```
grains = grain2d (show methods, plot)

 Phase   Grains   Pixels      Mineral   Symmetry
     1     1951     3444     Andesina         -1
     2      776     3893       Quartz       -3m1
     3      307      368      Biotite      12/m1
     4     1641     4781   Orthoclase      12/m1

 Properties: GOS, meanRotation
```

What is MTEX?

1. a MATLAB toolbox for quantitative texture analysis
2. a scripting language
3. a tool for generating publication ready plots
4. large, well documented and exhaustively tested
5. free to use, to extend, to modify

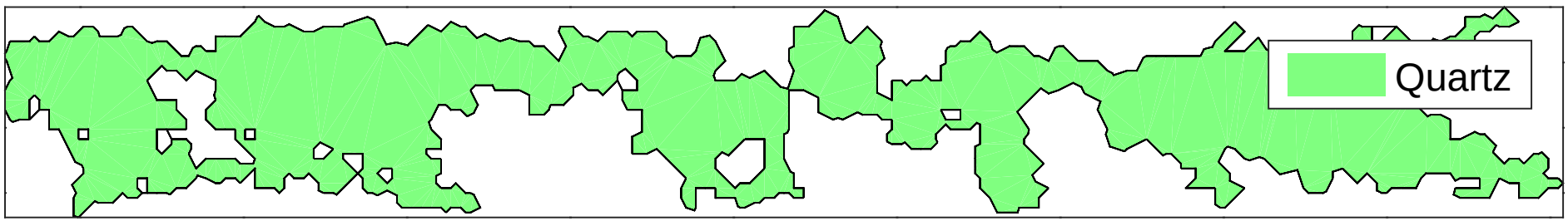
```
ebsd = loadEBSD('mylonite.txt')
grains = calcGrains(ebsd)
[m,id] = max(grains.area)
```

```
m =
   1.4985e+06
id =
        3369
```

What is MTEX?

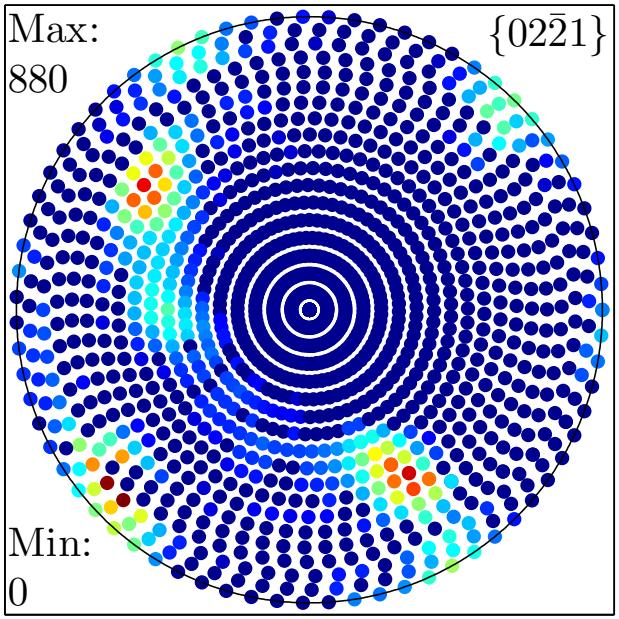
1. a MATLAB toolbox for quantitative texture analysis
2. a scripting language
3. a tool for generating publication ready plots
4. large, well documented and exhaustively tested
5. free to use, to extend, to modify

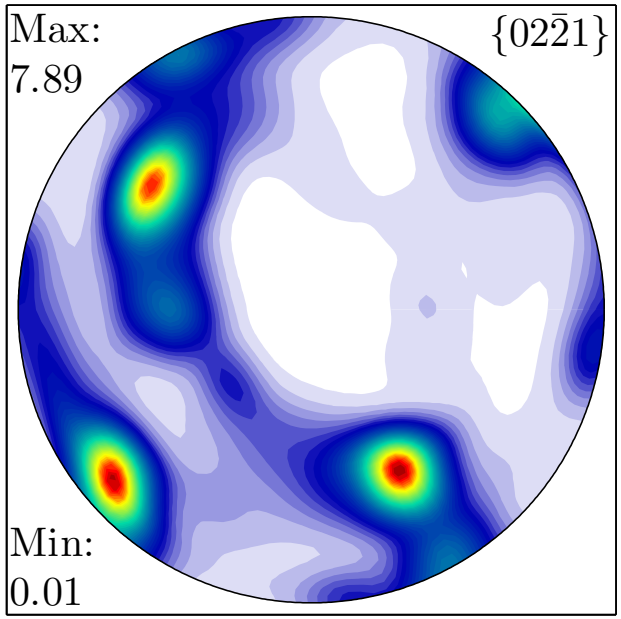
```
ebsd = loadEBSD('mylonite.txt')
grains = calcGrains(ebsd)
[m,id] = max(grains.area)
plot(grains(id))
```

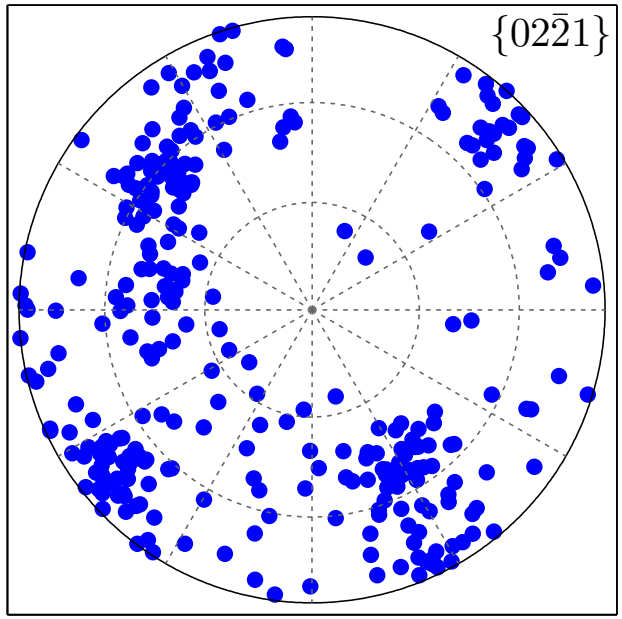


What is MTEX?

1. a MATLAB toolbox for quantitative texture analysis
2. a scripting language
3. a tool for generating publication ready plots
4. large, well documented and exhaustively tested
5. free to use, to extend, to modify







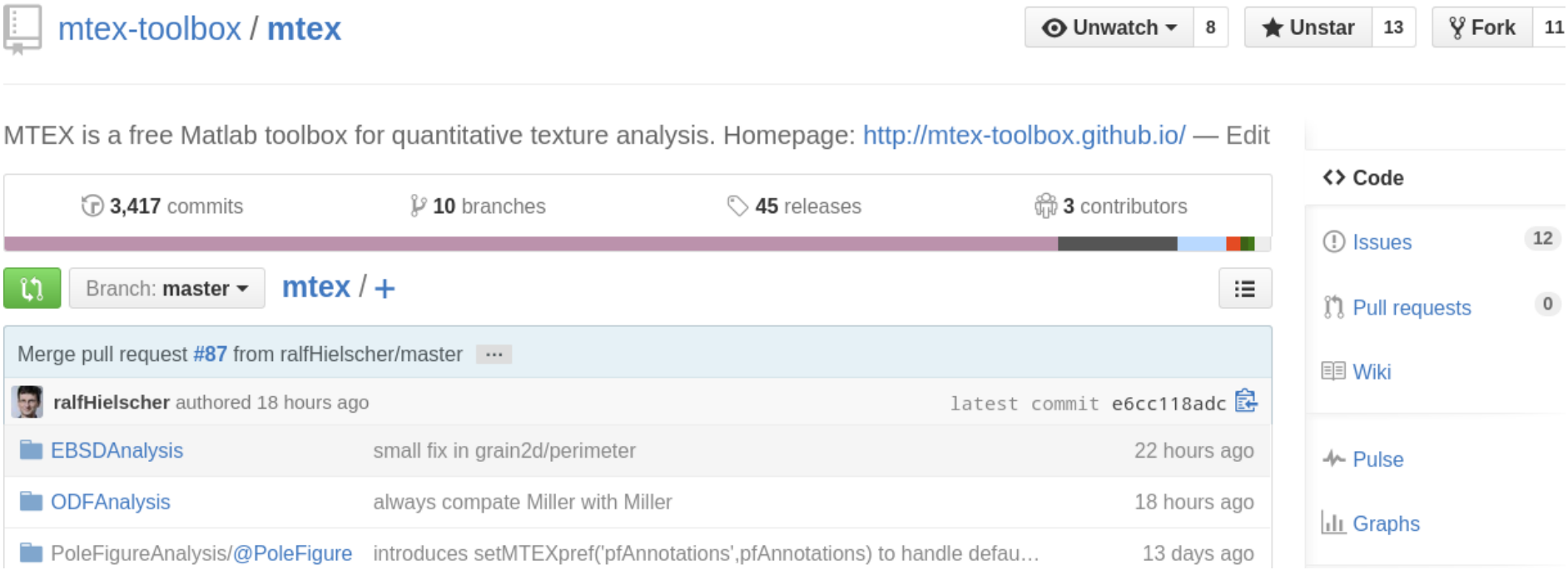
What is MTEX?

1. a MATLAB toolbox for quantitative texture analysis
2. a scripting language
3. a tool for generating publication ready plots
4. large, well documented and exhaustively tested
5. free to use, to extend, to modify

- *9 years of development*
- *1000 functions*
- *40 000 lines of code, 33 percent comment lines*
- *14 reference paper, about 500 citations*
- *1500 downloads per version*
- *1000 help pages*
- *compatible toolboxes: MSAT, PolyLx, CVA (Crystallographic Vorticity Axis analysis), Stabix, ebsdam*

What is MTEX?

1. a MATLAB toolbox for quantitative texture analysis
2. a scripting language
3. a tool for generating publication ready plots
4. large, well documented and exhaustively tested
5. free to use, to extend, to modify



Feature Overview

- crystal geometry
- odf modeling
- pole figure measurements
- individual orientation measurements
- elastic and plastic deformations

```
cs = loadCIF('Magnetite')
O1 = orientation('Euler',90*degree,45*degree,0,cs)
```

```
ori1 = orientation (show methods, plot)
  size: 1 x 1
  crystal symmetry : Magnetite (m-3m)
  specimen symmetry: 1

  Bunge Euler angles in degree
  phi1   Phi  phi2  Inv.
    90    45     0     0
```

Feature Overview

- crystal geometry
- odf modeling
- pole figure measurements
- individual orientation measurements
- elastic and plastic deformations

```
cs = loadCIF('Magnetite')
O1 = orientation('Euler',90*degree,45*degree,0,cs)
```

```
r = O1 * Miller(1,1,1,cs)
```

```
r = vector3d (show methods, plot)
  size: 1 x 1
    x         y         z
    0 0.119107 0.168443
```

Feature Overview

- crystal geometry
- odf modeling
- pole figure measurements
- individual orientation measurements
- elastic and plastic deformations

```
cs = loadCIF('Hematite')
O2 = orientation('Euler',180*degree,35*degree,0,cs)
MO = inv(O1) * O2
```

```
mori = misorientation (show methods, plot)
  size: 1 x 1
  crystal symmetry : Hematite (-3m1, X||a*, Y||b, Z||c)
  crystal symmetry : Magnetite (m-3m)

  Bunge Euler angles in degree
  phi1  Phi phi2 Inv.
   180   35    0    0
```

Feature Overview

- crystal geometry
- odf modeling
- pole figure measurements
- individual orientation measurements
- elastic and plastic deformations

```
odf = 0.8 * unimodalODF(O1) + 0.2 * uniformODF(cs)
```

```
odf = ODF (show methods, plot)
  crystal symmetry: Magnetite (m-3m)
  sample symmetry : triclinic

  Radially symmetric portion:
    kernel: de la Vallee Poussin, hw = 10
    center: (90, 45, 0)
    weight: 0.8

  Uniform portion:
    weight: 0.2
```

Feature Overview

- crystal geometry
- odf modeling
- pole figure measurements
- individual orientation measurements
- elastic and plastic deformations

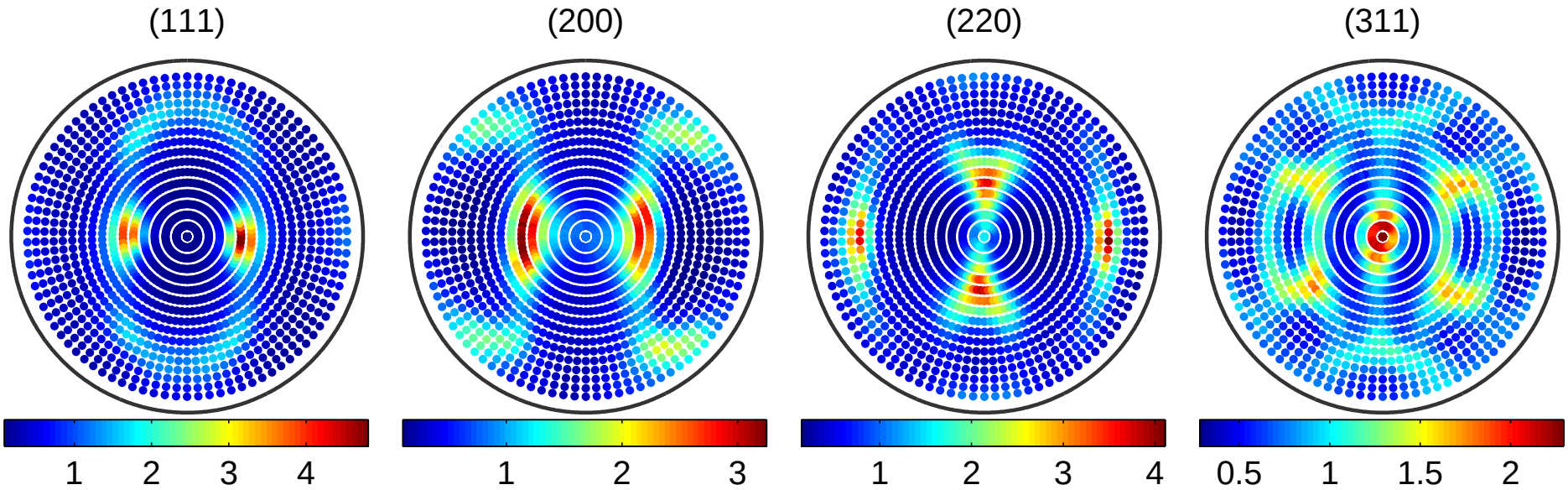
```
pf = loadPoleFigure('Queens_alu.plf')
```

```
pf = PoleFigure (show methods, plot)
  file name: Queens_alu.plf
  crystal symmetry : cubic (m-3m)
  specimen symmetry: orthorhombic

  h = {111}, r = 90x17 points
  h = {200}, r = 90x17 points
  h = {220}, r = 90x17 points
  h = {311}, r = 90x17 points
```

Feature Overview

- crystal geometry
- odf modeling
- pole figure measurements
- individual orientation measurements
- elastic and plastic deformations

```
pf = loadPoleFigure('Queens_alu.plf')
plot(pf)
```



Feature Overview

- crystal geometry
- odf modeling
- pole figure measurements
- individual orientation measurements
- elastic and plastic deformations

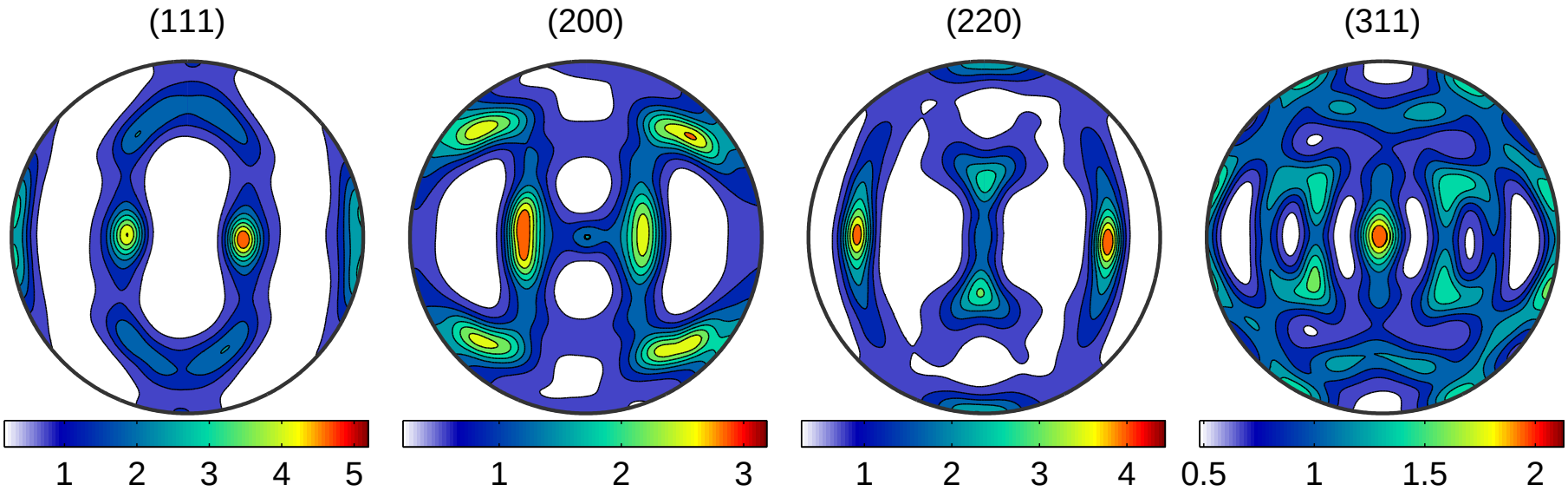
```
odf = calcODF(pf)
```

```
odf = ODF (show methods, plot)
  file name: Queens_alu.plf
  crystal symmetry: cubic
  sample symmetry : orthorhombic

  Uniform portion:
    weight: 0.14633

  Radially symmetric portion:
    kernel: de la Vallee Poussin, hw = 4
    center: 2444 orientations, resolution: 3.9
    weight: 0.85367
```

Feature Overview

- crystal geometry
- odf modeling
- pole figure measurements
- individual orientation measurements
- elastic and plastic deformations

```
odf = calcODF(pf)
plotPDF(odf,pf.h)
```



Feature Overview

- crystal geometry
- odf modeling
- pole figure measurements
- individual orientation measurements
- elastic and plastic deformations

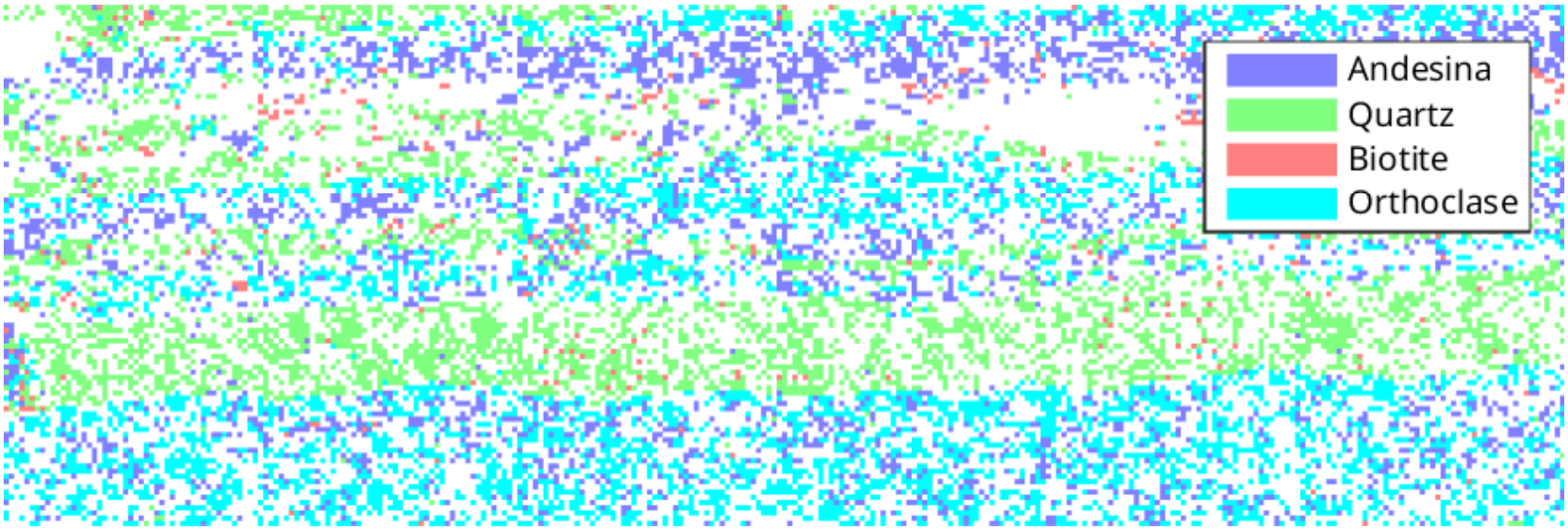
```
ebsd = loadEBSD('mylonite.txt')
```

```
ebsd = EBSD (show methods, plot)
  file name: mylonite.txt
  Properties: x, y
  Phase Orientations     Mineral Symmetry Crystal reference frame
      1          3444   Andesina       -1                X||a*, Z||c
      2          3893     Quartz      -3m      X||a*, Y||b,  Z||c*
      3           368    Biotite      2/m      X||a*, Y||b*, Z||c
      4          4781 Orthoclase      2/m      X||a*, Y||b*, Z||c
```

Feature Overview

- crystal geometry
- odf modeling
- pole figure measurements
- individual orientation measurements
- elastic and plastic deformations

```
ebsd = loadEBSD('mylonite.txt')
plot(ebsd)
```



Feature Overview

- crystal geometry
- odf modeling
- pole figure measurements
- individual orientation measurements
- elastic and plastic deformations

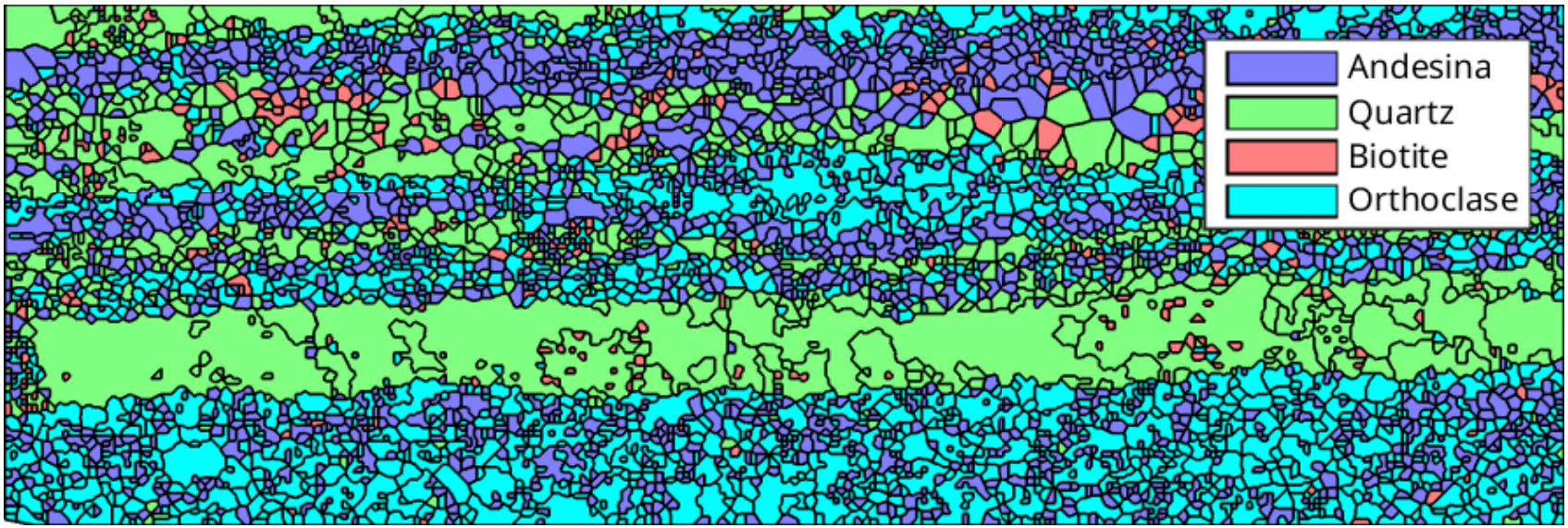
```
grains = calcGrains(ebsd)
```

```
grains = grain2d (show methods, plot)

  Phase  Grains     Mineral  Symmetry  Cryst. reference frame
      1    1951    Andesina        -1              X||a*, Z||c
      2     776      Quartz       -3m    X||a*, Y||b,  Z||c*
      3     305     Biotite       2/m    X||a*, Y||b*, Z||c
      4    1641  Orthoclase       2/m    X||a*, Y||b*, Z||c
```

Feature Overview

- crystal geometry
- odf modeling
- pole figure measurements
- individual orientation measurements
- elastic and plastic deformations

```
grains = calcGrains(ebsd)
plot(grains)
```



Feature Overview

- crystal geometry
- odf modeling
- pole figure measurements
- individual orientation measurements
- elastic and plastic deformations

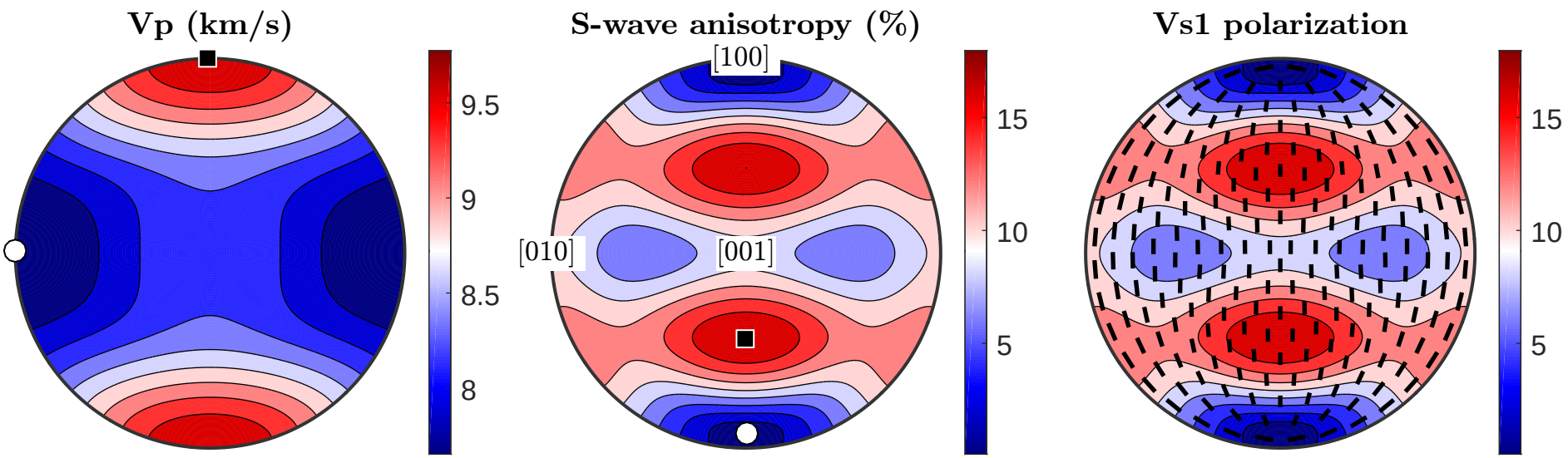
```
cs = symmetry('Olivin');
C  = loadTensor('Olivine1997PC.GPa',cs,...
     'propertyname','elastic stiffness','unit','Pa')
```

```
C = elastic stiffness tensor
  unit          : Pa
  mineral       : Olivin (mmm)

 320.5   68.2   71.6      0      0      0
  68.2  196.5   76.8      0      0      0
  71.6   76.8  233.5      0      0      0
     0      0      0     64      0      0
     0      0      0      0     77      0
     0      0      0      0      0   78.7
```

Feature Overview

- crystal geometry
- odf modeling
- pole figure measurements
- individual orientation measurements
- elastic and plastic deformations

SeismicsMultiplot



Design Principle - Everything is a List

```
[grains,ebsd.grainId,ebsd.mis2mean] = calcGrains(ebsd)
```

```
grains = grain2d (show methods, plot)

 Phase   Grains   Pixels      Mineral   Symmetry   Cryst. ref. frame
     1     1951     3444     Andesina         -1          X||a*, Z||c
     2      776     3893       Quartz       -3m1   X||a*, Y||b, Z||c
     3      307      368      Biotite      12/m1   X||a, Y||b*, Z||c
     4     1641     4781   Orthoclase      12/m1   X||a*, Y||b, Z||c

 Properties: GOS, meanRotation
```

Design Principle - Everything is a List

```
[grains,ebsd.grainId,ebsd.mis2mean]=calcGrains(ebsd)
```

```
grains(3369)
```

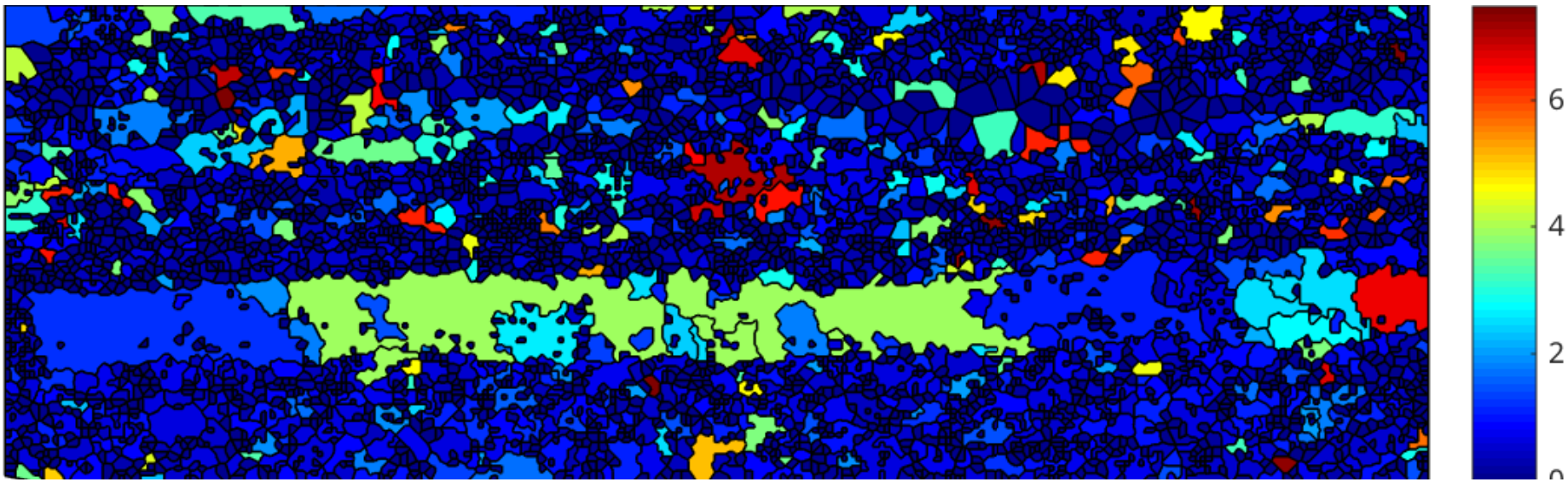
```
ans = grain2d (show methods, plot)

 Phase   Grains   Pixels   Mineral   Symmetry   Cryst. ref. frame
     2        1      700    Quartz       -3m1   X||a*, Y||b, Z||c

   Id   Phase   Pixels          GOS   phi1    Phi   phi2
 3369       2      700   0.0623375      1    119     54
```

Design Principle - Everything is a List

```
[grains,ebsd.grainId,ebsd.mis2mean]=calcGrains(ebsd)
```

```
grains(3369)
```

```
for i = 1:length(grains)
  m2m = ebsd(grains(i)).mis2mean;
  GOS(i) = sqrt(mean(mori.angle.^2));
end
plot(grains, GOS ./ degree)
```



Design Principle - Everything is a List

```
[grains,ebsd.grainId,ebsd.mis2mean]=calcGrains(ebsd)
```

```
grains(3369)
```

```
for i = 1:length(grains)
  m2m = ebsd(grains(i)).mis2mean;
  GOS(i) = sqrt(mean(mori.angle.^2));
end
plot(grains, GOS ./ degree)
```

```
grains(GOS > 5*degree)
```

```
ans = grain2d (show methods, plot)

 Phase  Grains  Pixels      Mineral  Symmetry  Cryst. ref. frame
     1      13      63     Andesina        -1         X||a*, Z||c
     2      15     121       Quartz      -3m1  X||a*, Y||b, Z||c
     4       8      76  Orthoclase     12/m1  X||a*, Y||b, Z||c

 Properties: GOS, meanRotation
```

Lists of Boundary Segments

```
ebsd = loadEBSD('CSL.txt')
```

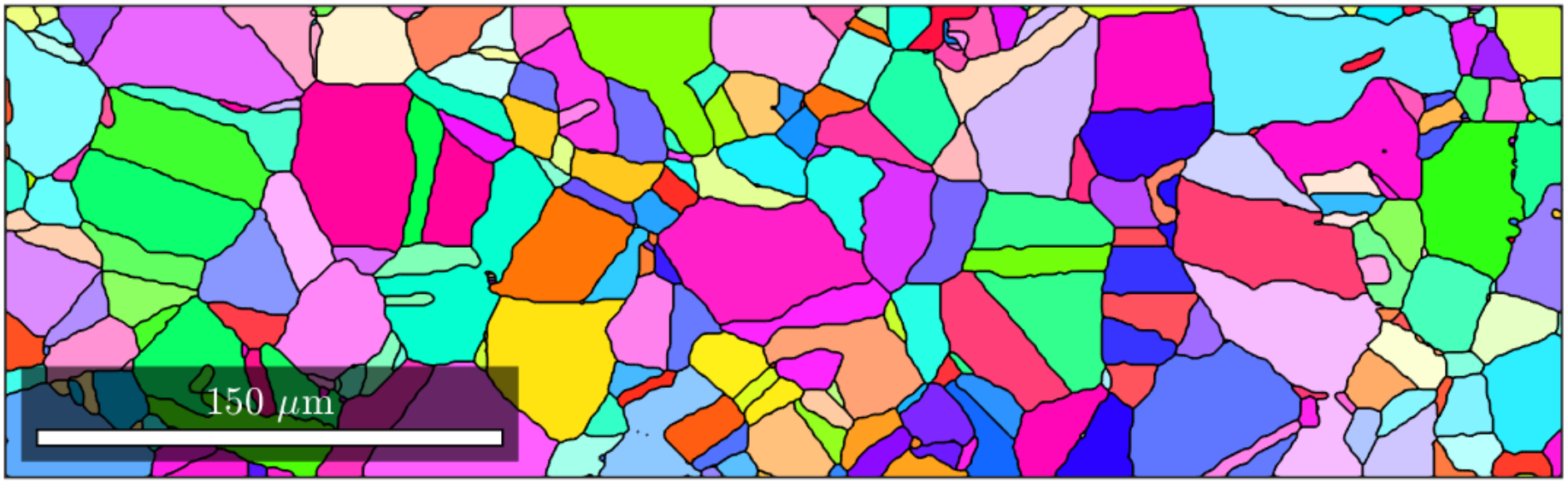
```
ebsd = EBSD (show methods, plot)

 Phase   Orientations   Mineral   Symmetry
     1   154107 (100)      iron       m-3m

 Properties: ci, error, iq, x, y
 Scan unit : um
```

Lists of Boundary Segments

```
ebsd = loadEBSD('CSL.txt')
```

```
[grains,ebsd.grainId] = calcGrains(ebsd)
```

```
grains = grain2d (show methods, plot)

 Phase   Grains   Pixels   Mineral   Symmetry
    -1      465    75651      iron       m-3m

 Properties: GOS, meanRotation
```



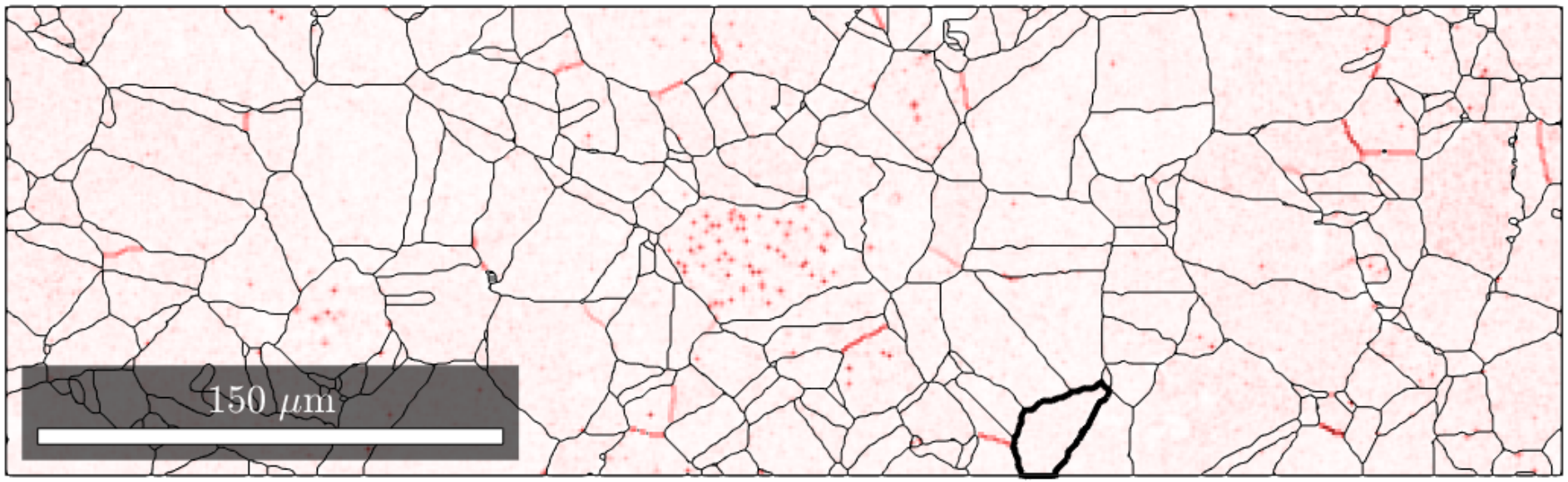
Lists of Boundary Segments

```
ebsd = loadEBSD('CSL.txt')
```

```
[grains,ebsd.grainId] = calcGrains(ebsd)
```

```
grains(90).boundary
```

```
ans = grainBoundary (show methods, plot)

 Segments     mineral 1   mineral 2
       11   not indexed        iron
      111          iron        iron
```



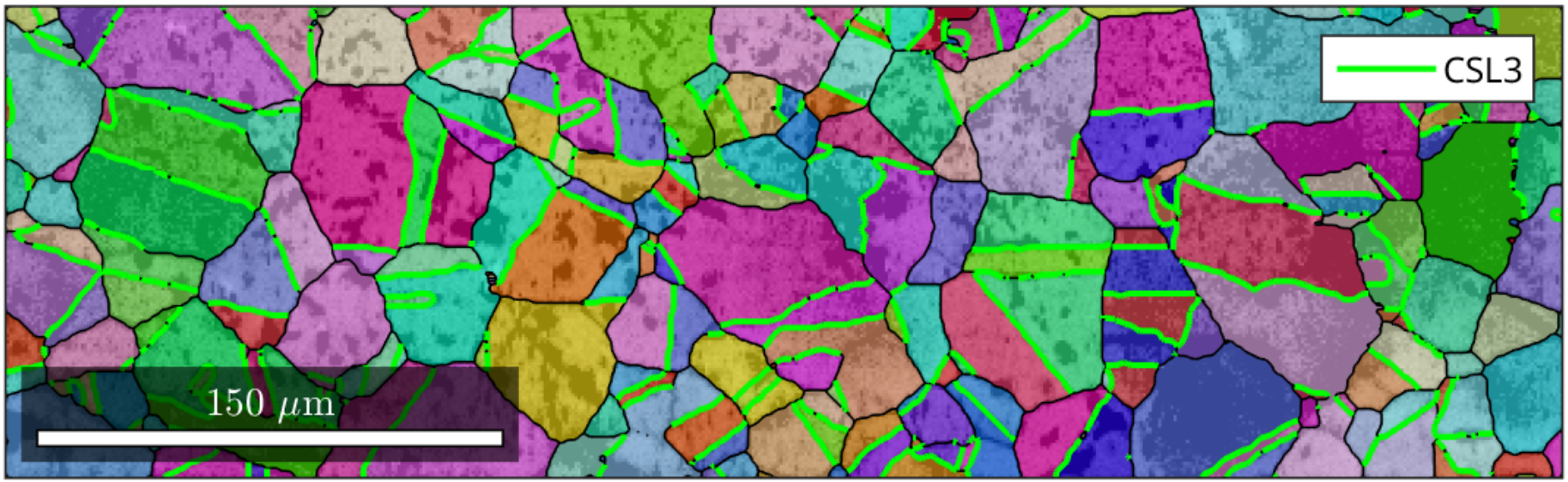
Lists of Boundary Segments

```
ebsd = loadEBSD('CSL.txt')
```

```
[grains,ebsd.grainId] = calcGrains(ebsd)
```

```
gB = grains.boundary('indexed');
gB3 = gB(angle(gB.misorientation,CSL(3)) < 3*degree)
```

```
gB3 = grainBoundary (show methods, plot)

 Segments  mineral 1  mineral 2
     4152        iron        iron
```



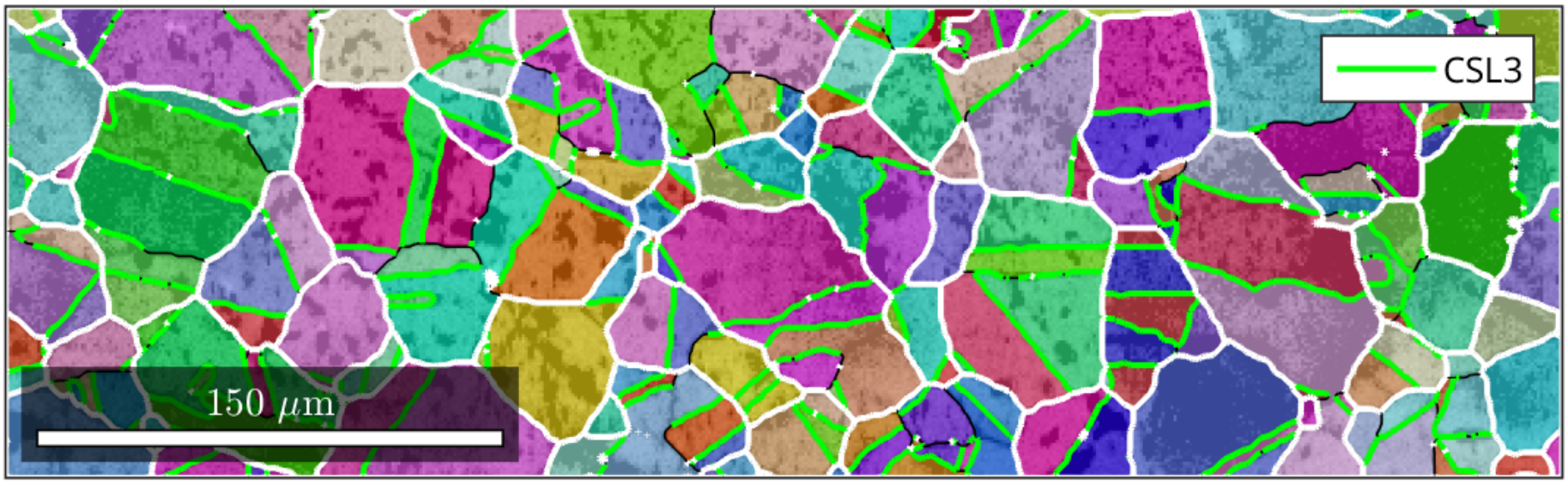
Lists of Boundary Segments

```
ebsd = loadEBSD('CSL.txt')
```

```
[grains,ebsd.grainId] = calcGrains(ebsd)
```

```
gB = grains.boundary('indexed');
gB3 = gB(angle(gB.misorientation,CSL(3)) < 3*degree)
```

```
mergedGrains = merge(grains,gB3);
```



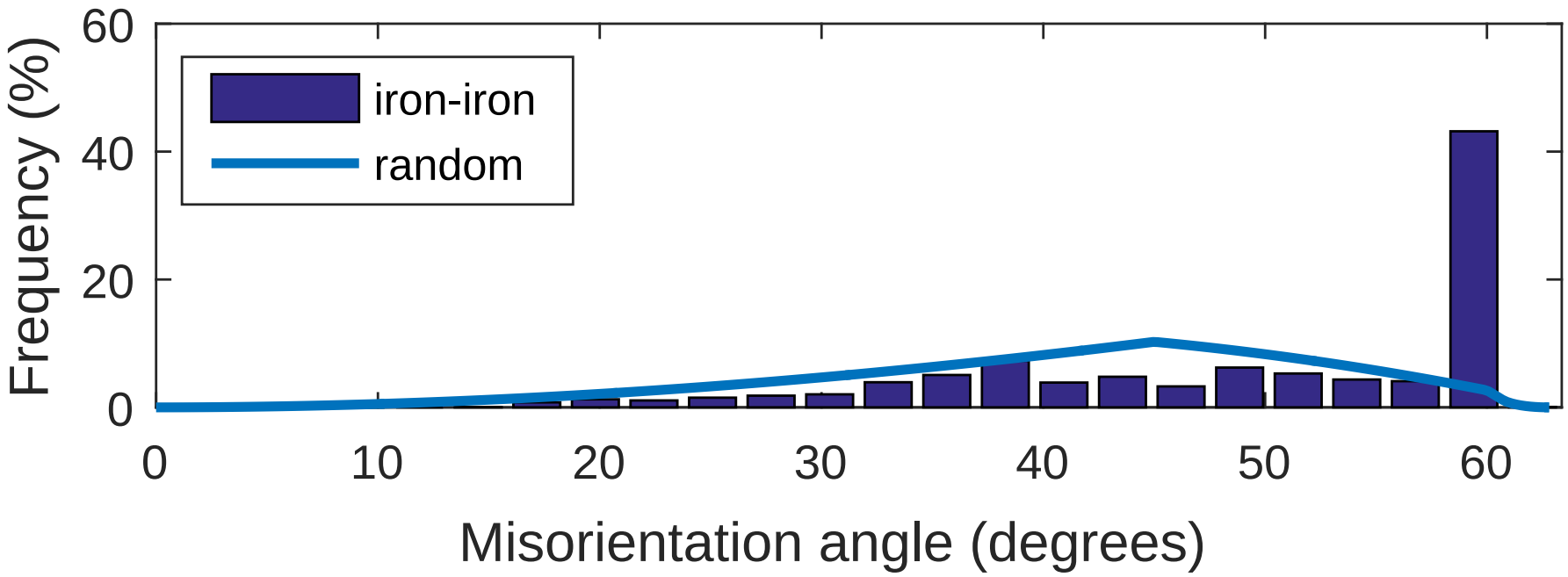
Lists of Boundary Segments

```
ebsd = loadEBSD('CSL.txt')
```

```
[grains,ebsd.grainId] = calcGrains(ebsd)
```

```
gB = grains.boundary('indexed');
gB3 = gB(angle(gB.misorientation,CSL(3)) < 3*degree)
```

```
plotAngleDistribution(gB.misorientation)
```



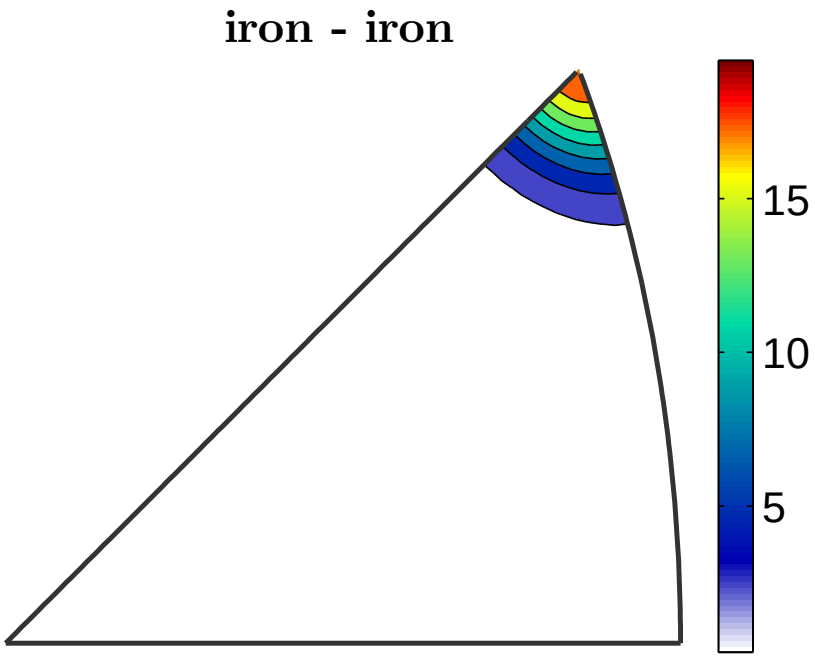
Lists of Boundary Segments

```
ebsd = loadEBSD('CSL.txt')
```

```
[grains,ebsd.grainId] = calcGrains(ebsd)
```

```
gB = grains.boundary('indexed');
gB3 = gB(angle(gB.misorientation,CSL(3)) < 3*degree)
```

```
plotAxisDistribution(gB.misorientation,'contourf')
```



Lists of Boundary Segments

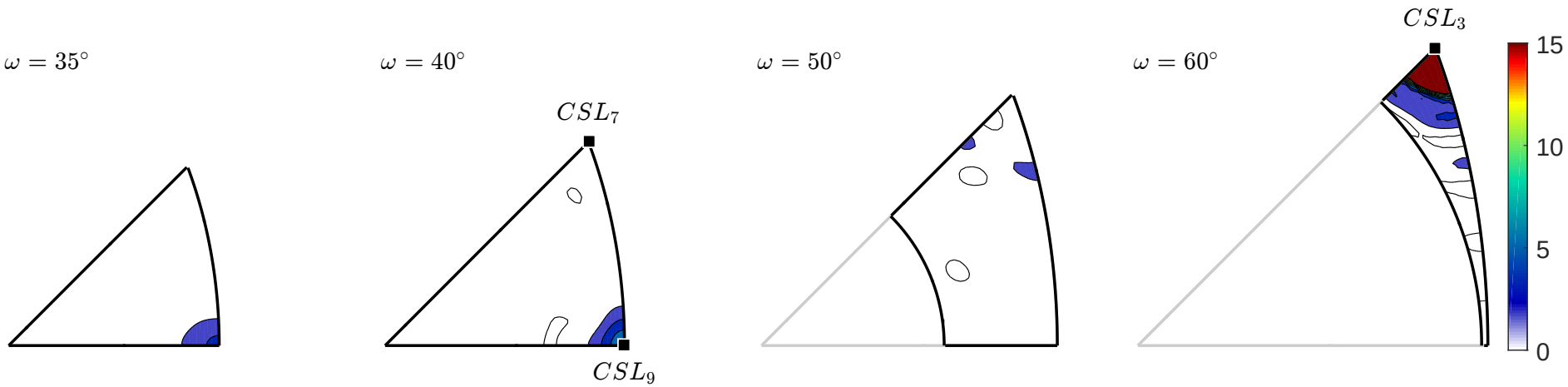
```
ebsd = loadEBSD('CSL.txt')
```

```
[grains,ebsd.grainId] = calcGrains(ebsd)
```

```
gB = grains.boundary('indexed');
gB3 = gB(angle(gB.misorientation,CSL(3)) < 3*degree)
```

```
mdf = calcMDF(gB.misorientation)
```

```
mdf = MDF (show methods, plot)
  crystal symmetry : iron (m-3m)
  crystal symmetry : iron (m-3m)

  Radially symmetric portion:
    kernel: de la Vallee Poussin, halfwidth 2.5
    center: 3436 orientations, resolution: 1.2
```



Lists of Boundary Segments

```
ebsd = loadEBSD('CSL.txt')
```

```
[grains,ebsd.grainId] = calcGrains(ebsd)
```

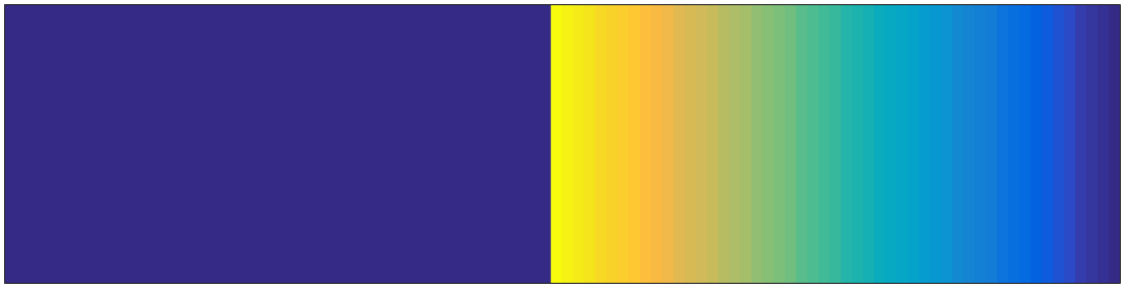
```
gB = grains.boundary('indexed');
gB3 = gB(angle(gB.misorientation,CSL(3)) < 3*degree)
```

```
mdf = calcMDF(gB.misorientation)
```

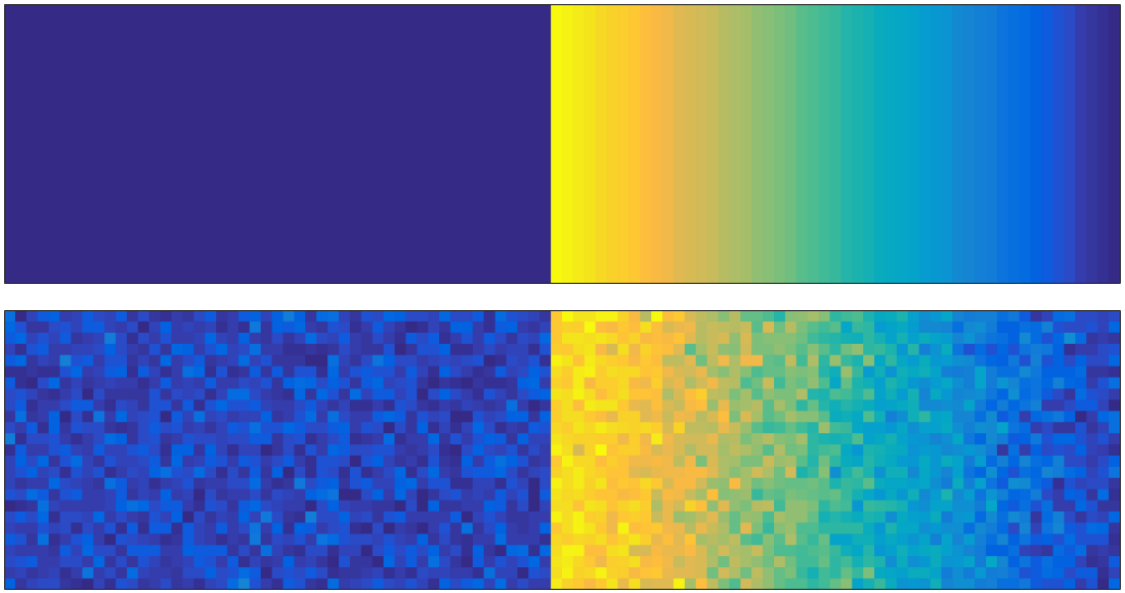
```
mori = calcModes(mdf,2)
```

```
mori = misorientation (show methods, plot)
  size: 1 x 2
  crystal symmetry : iron (m-3m)
  crystal symmetry : iron (m-3m)

  Bunge Euler angles in degree
     phi1      Phi     phi2     Inv.
  62.7843 48.0359 333.939        0
  103.129 26.8018  284.91        0
```

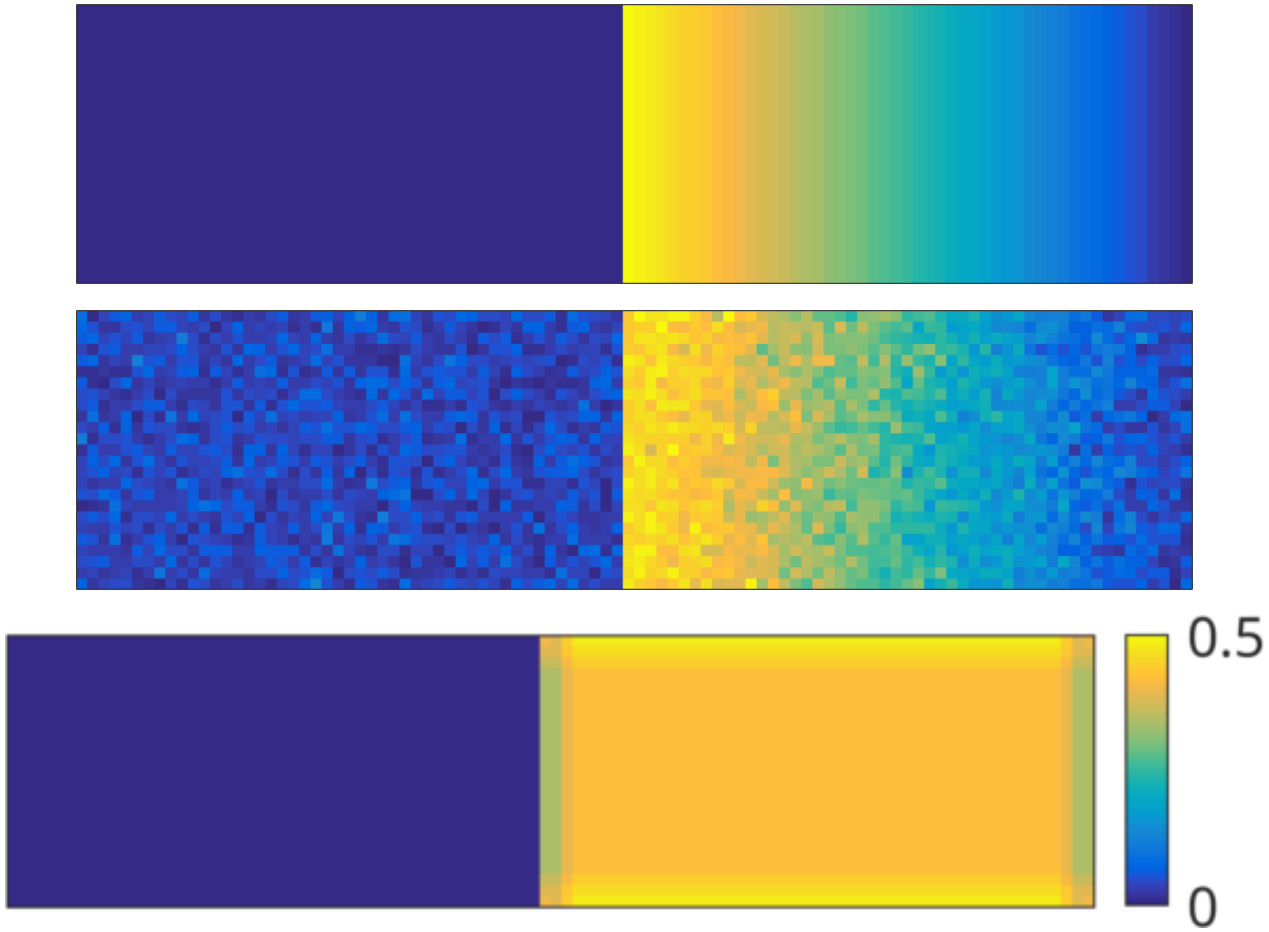
Noisy Orientation Data



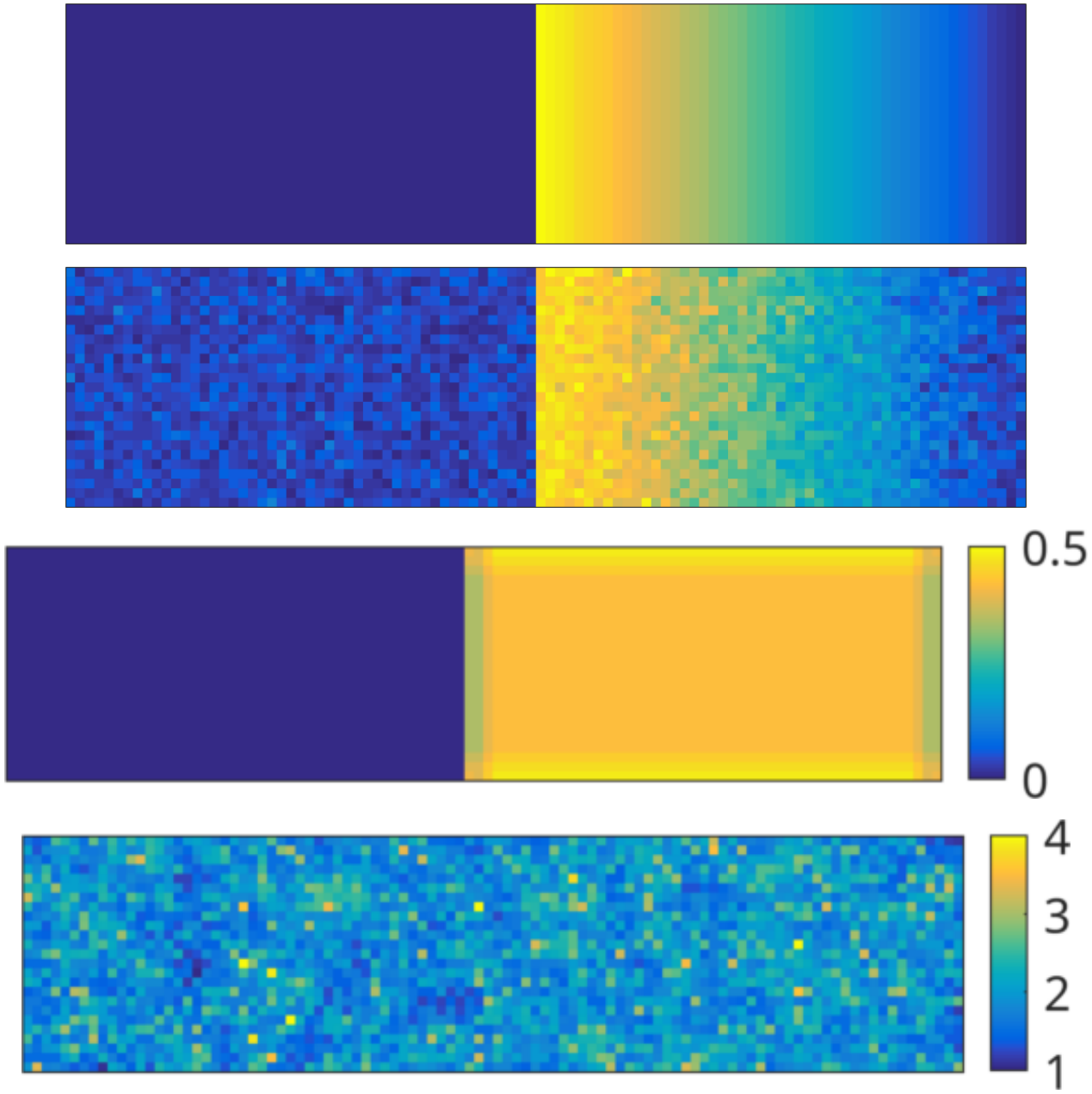
Noisy Orientation Data



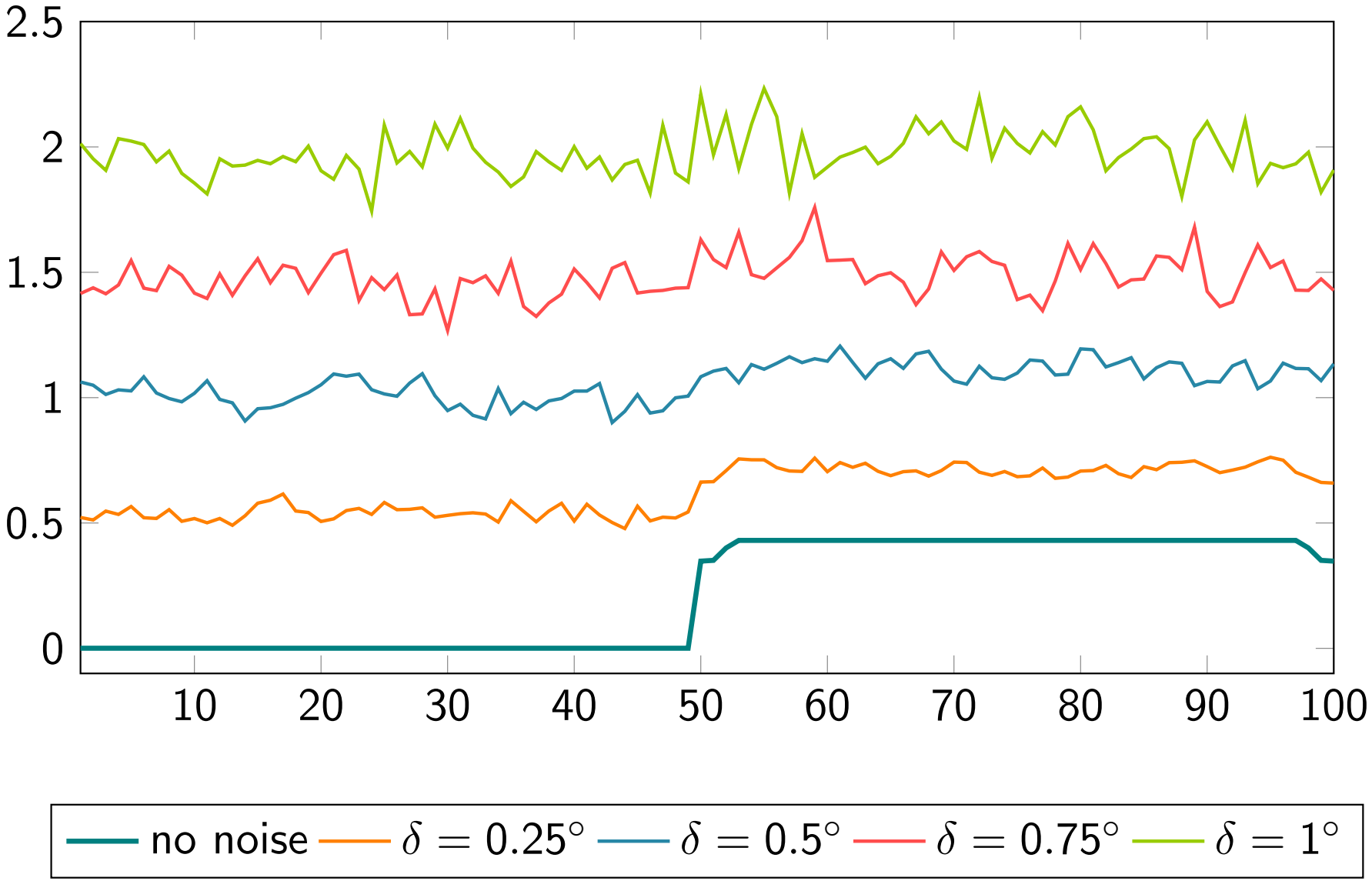
Noisy Orientation Data



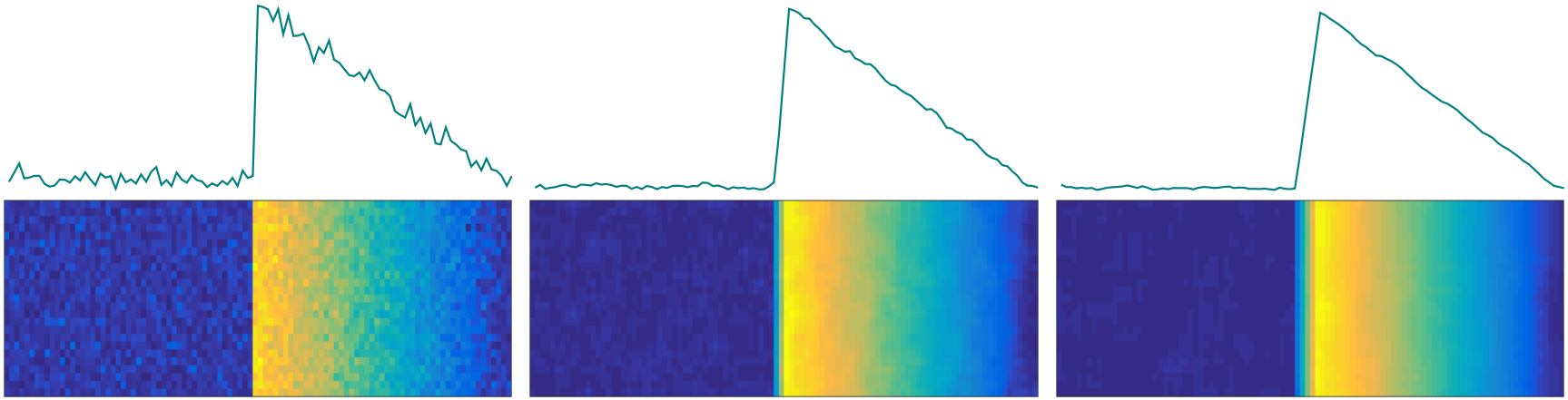
Noisy Orientation Data



KAM

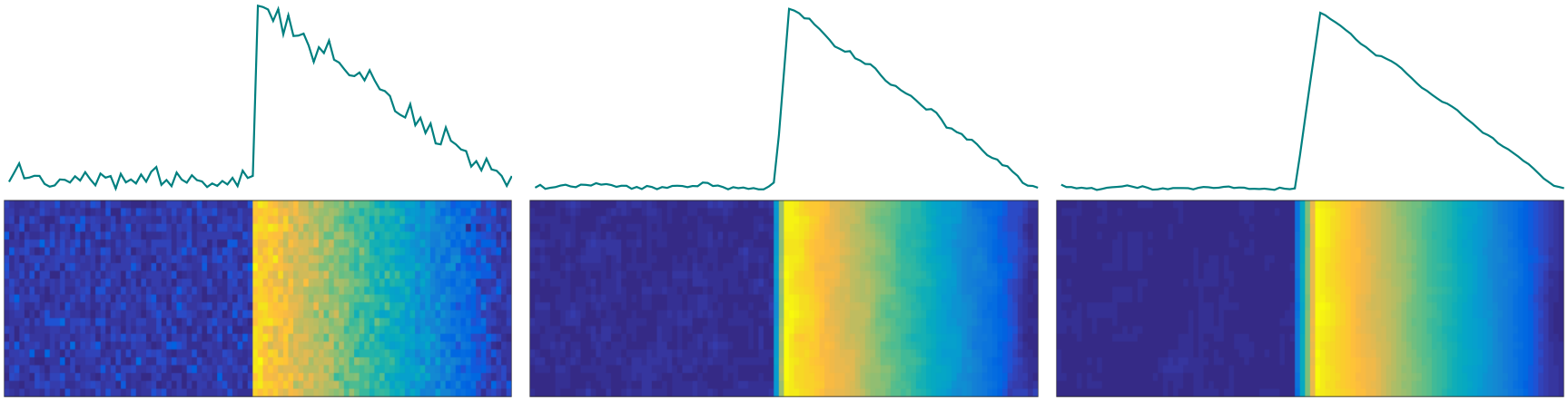

2.5
2
1.5
1
0.5
0
10
20
30
40
50
60
70
80
90
100
no noise
$\delta = 0.25°$
$\delta = 0.5°$
$\delta = 0.75°$
$\delta = 1°$

Basic Denoising Techniques

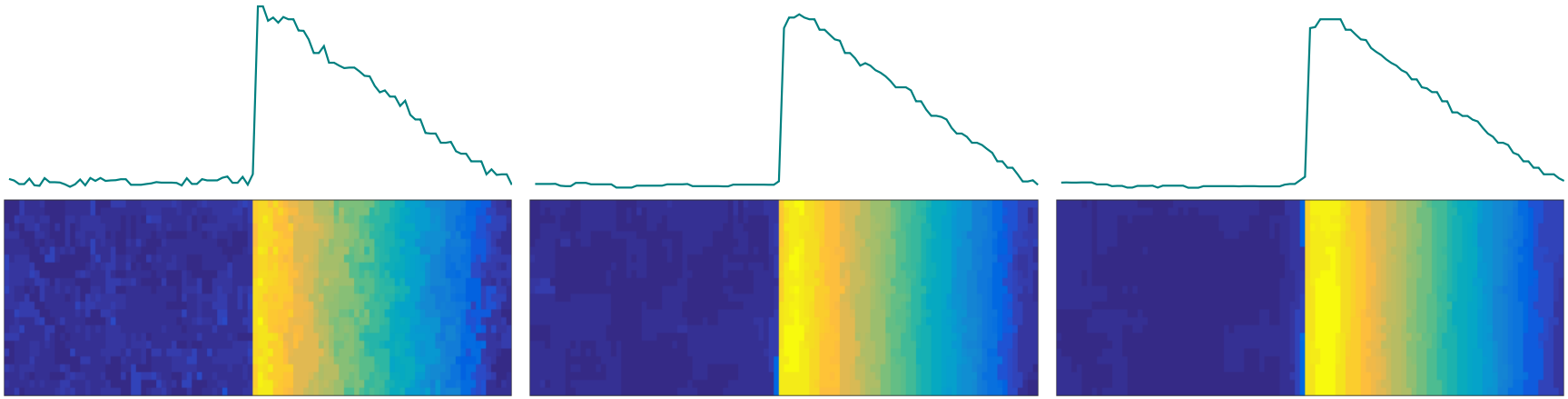


noisy data, mean filter: first and second neighbor

Basic Denoising Techniques

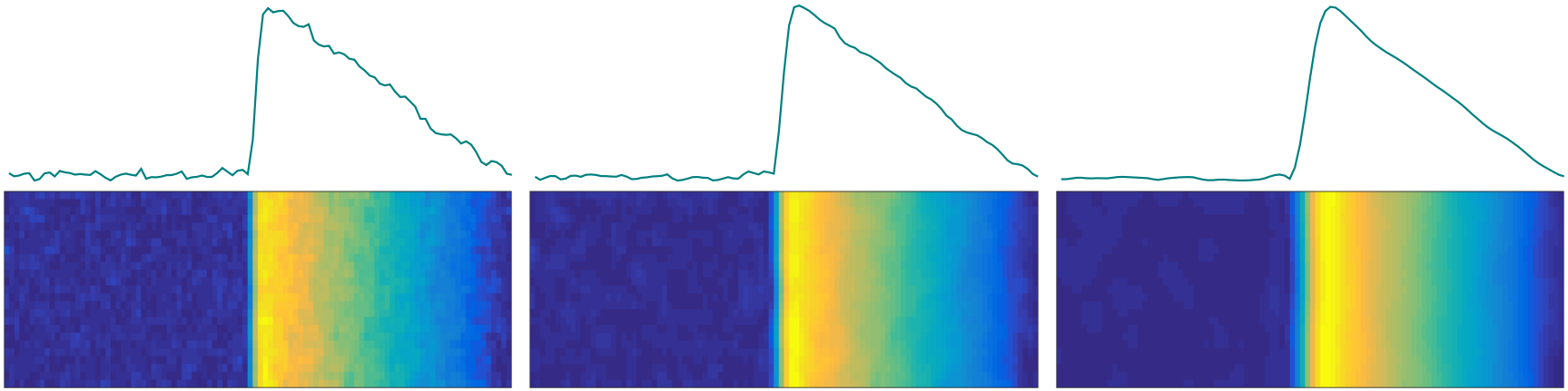


noisy data, mean filter: first and second neighbor



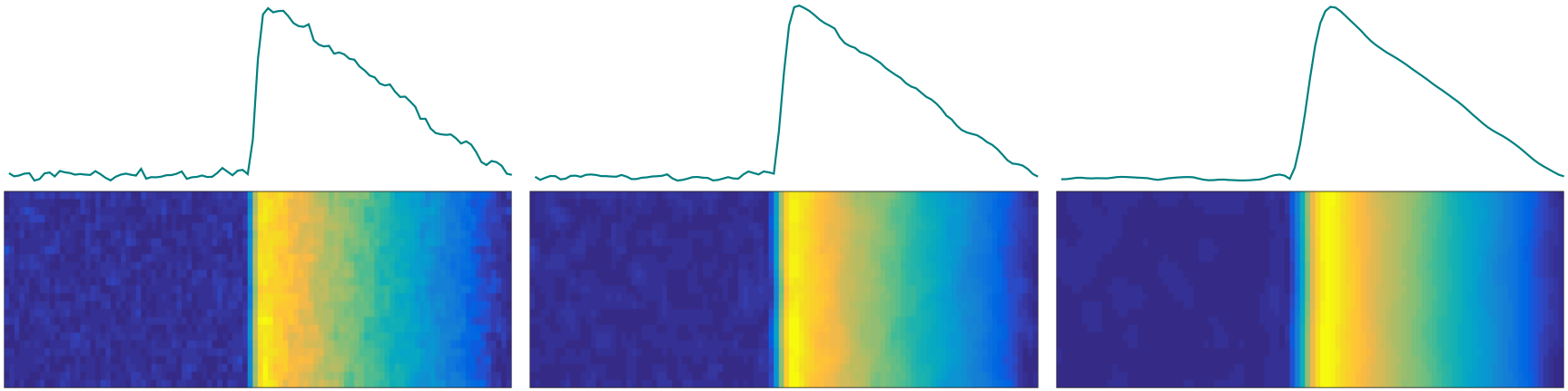
median filter: first, third, and fifth neighbor

Advanced Denoising Techniques

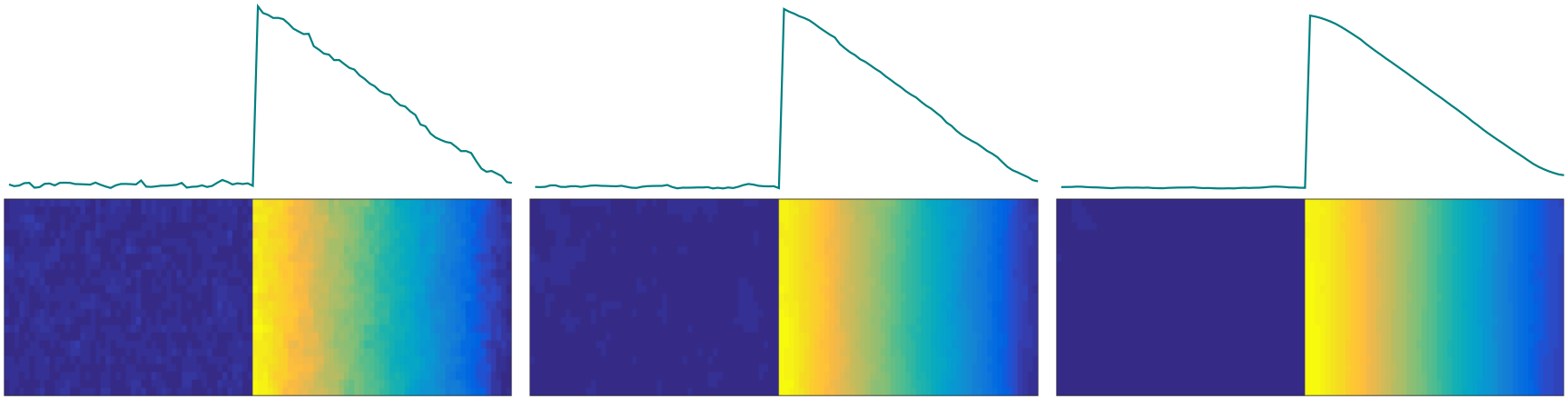


smoothing spline filter with $\alpha = 0.1$, $\alpha = 0.58$, and $\alpha = 5$

Advanced Denoising Techniques

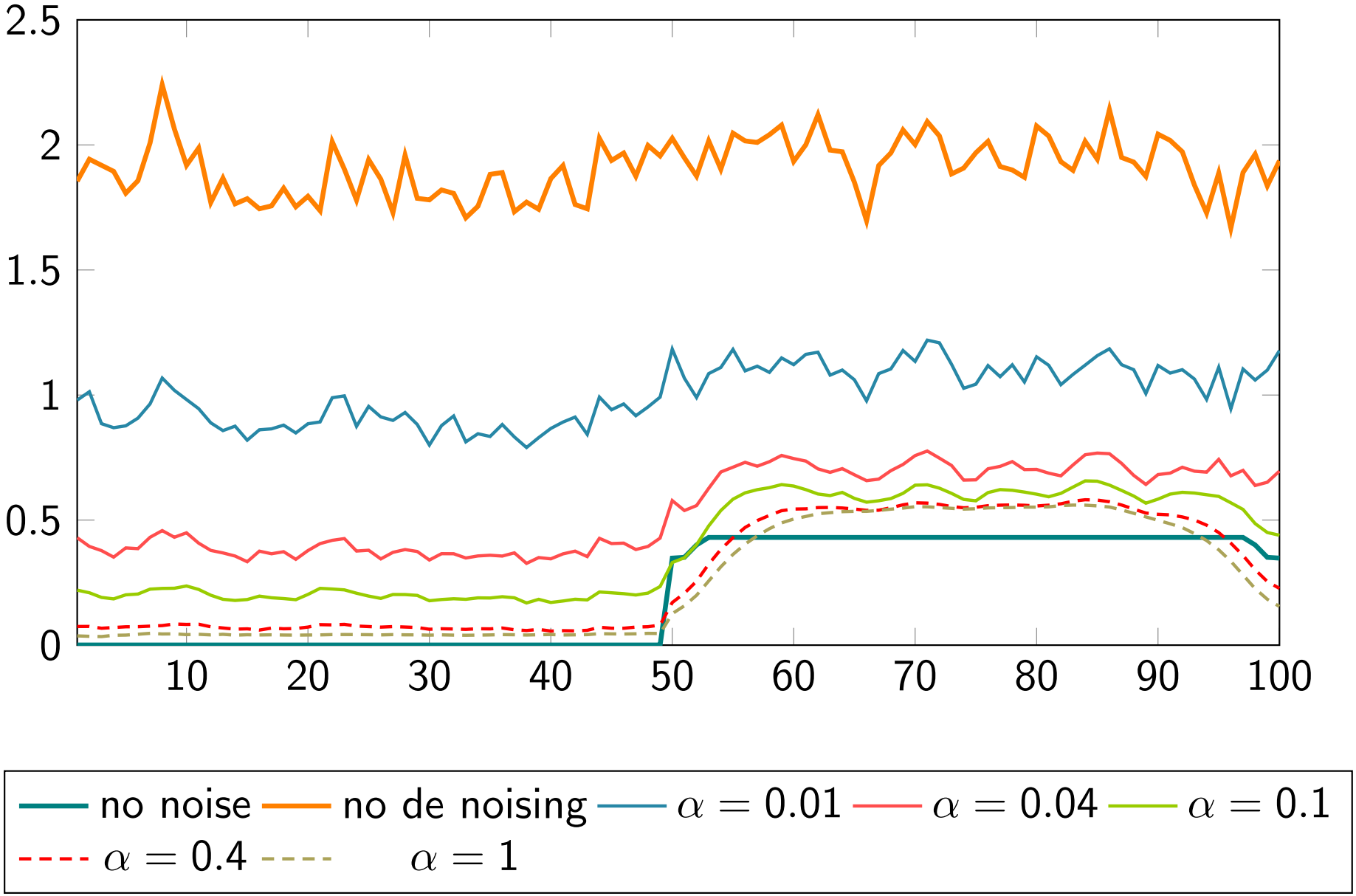


smoothing spline filter with $\alpha = 0.1$, $\alpha = 0.58$, and $\alpha = 5$

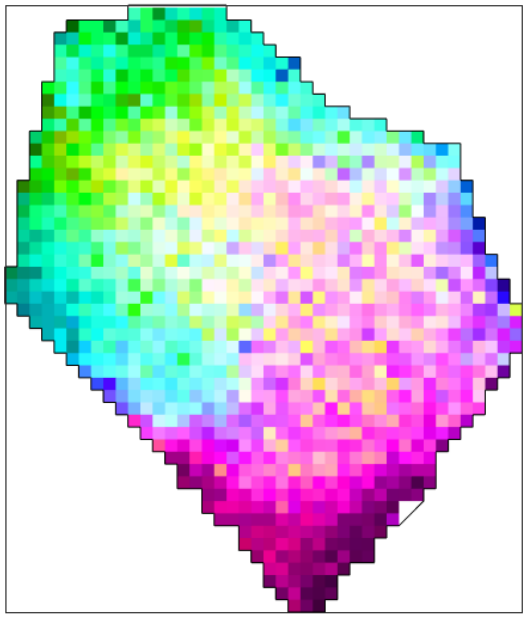


half quadratic filter with $\alpha = 0.025$, $\alpha = 0.1$, and $\alpha = 0.5$

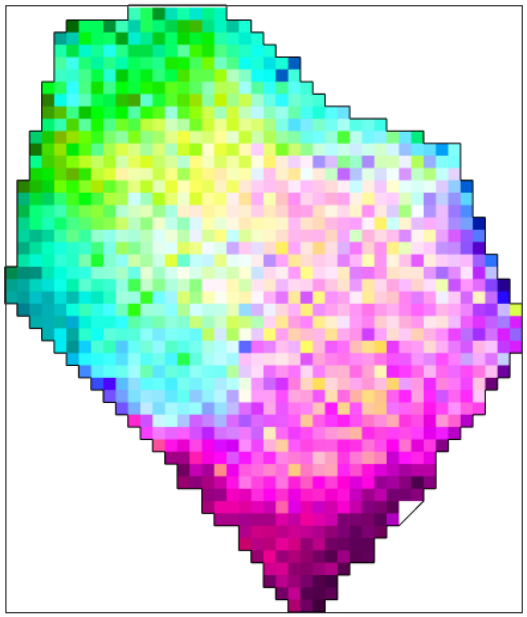
KAM of Denoised Data

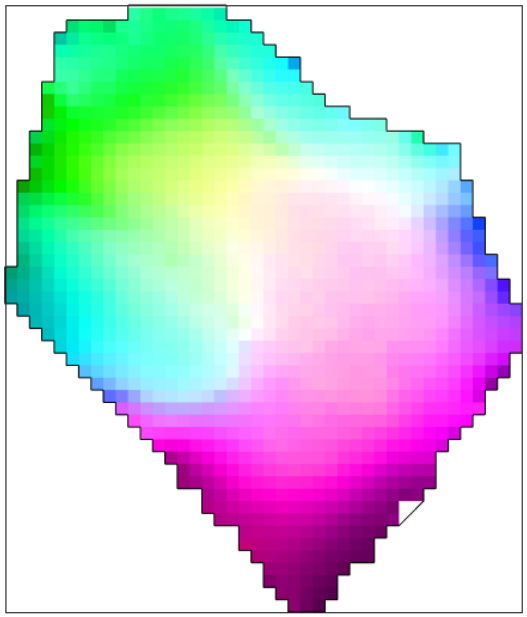


A deformed grain

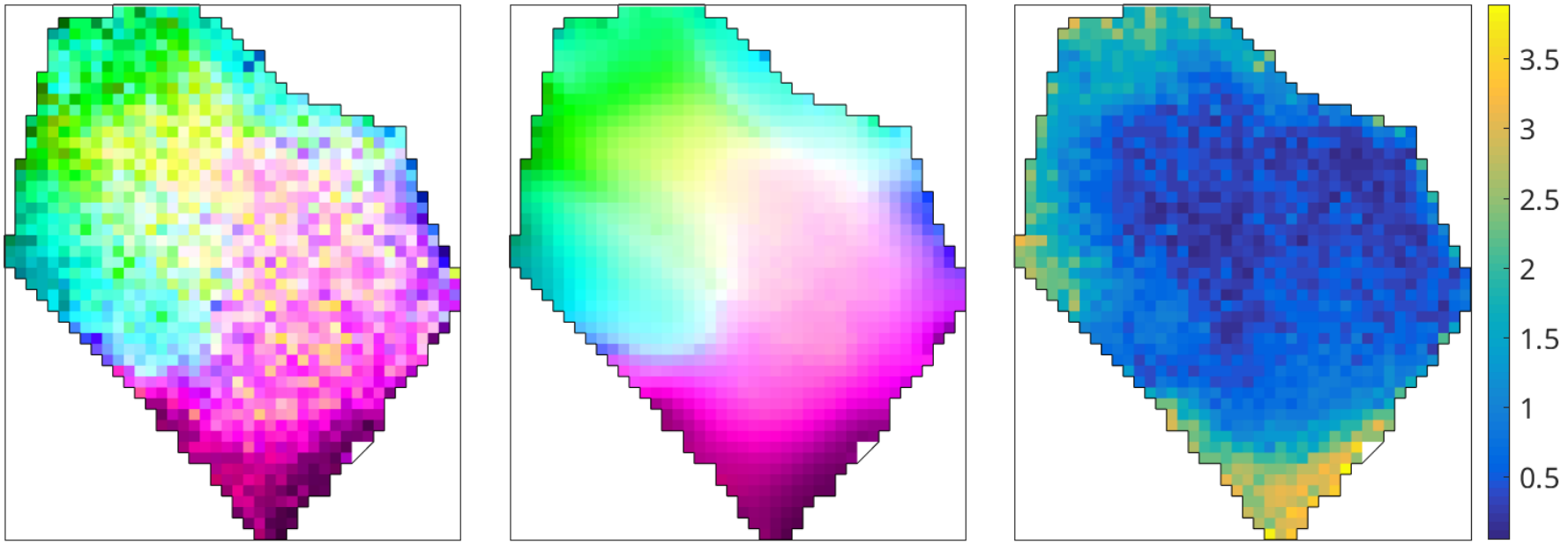


A deformed grain

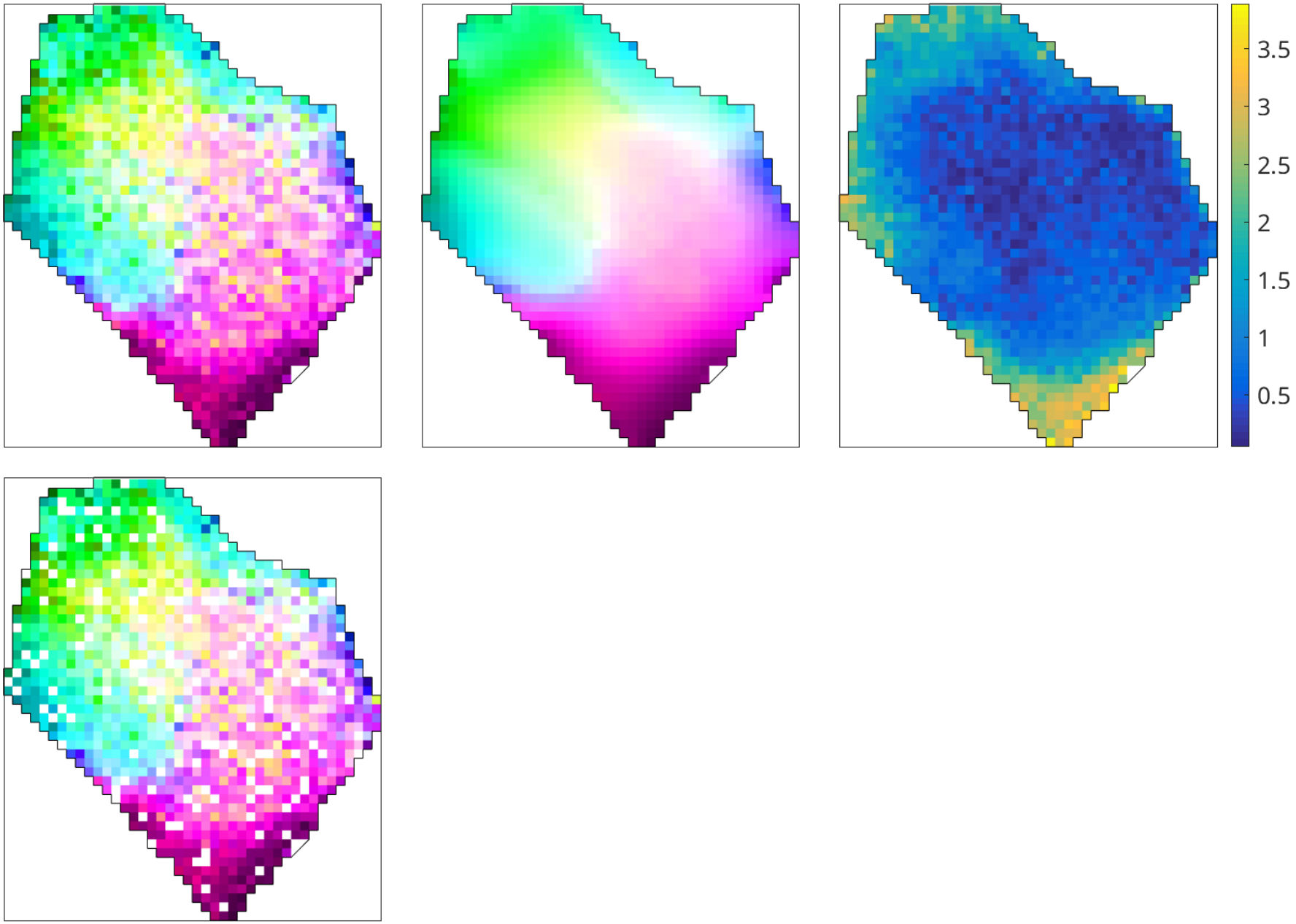




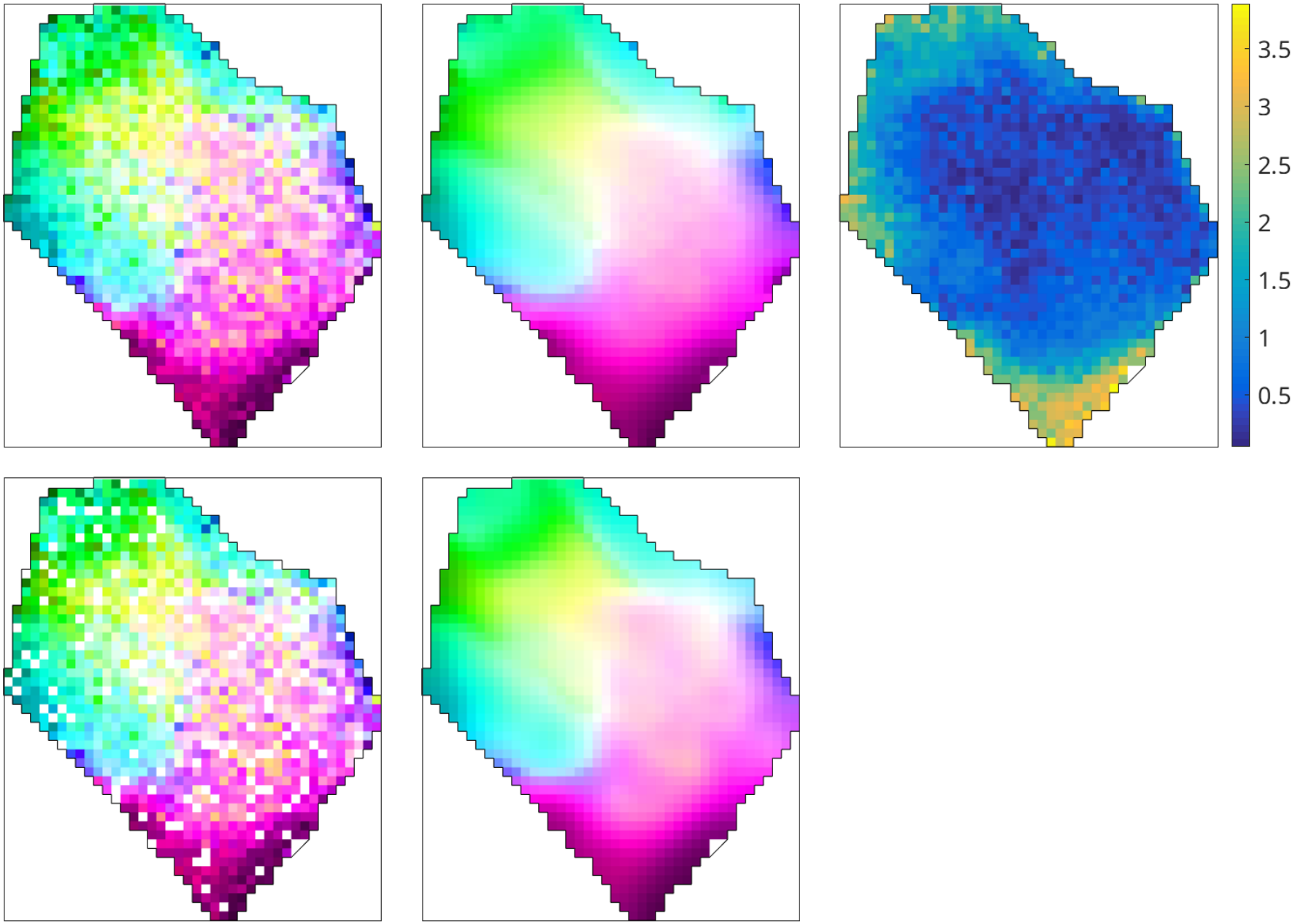
A deformed grain



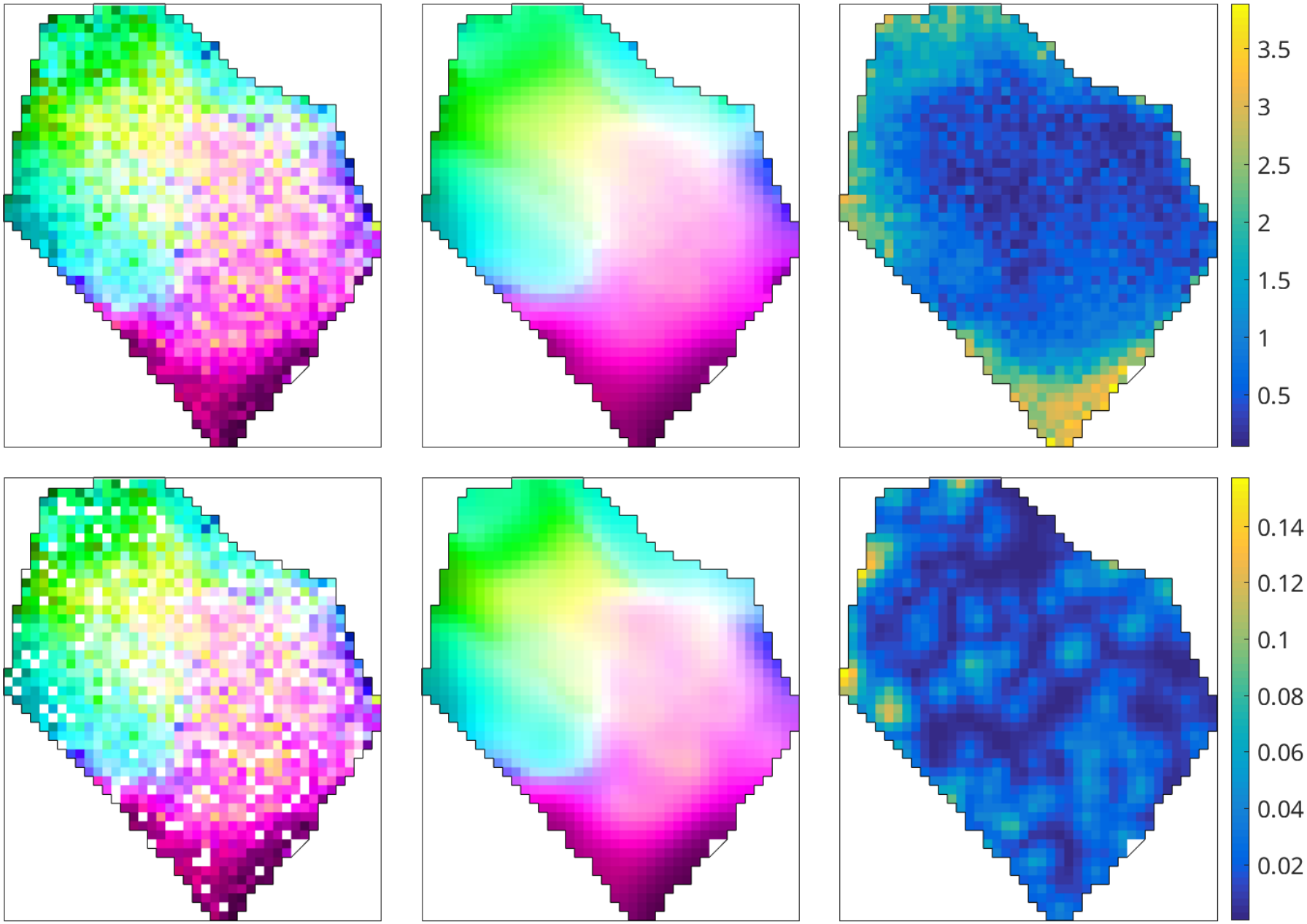
A deformed grain



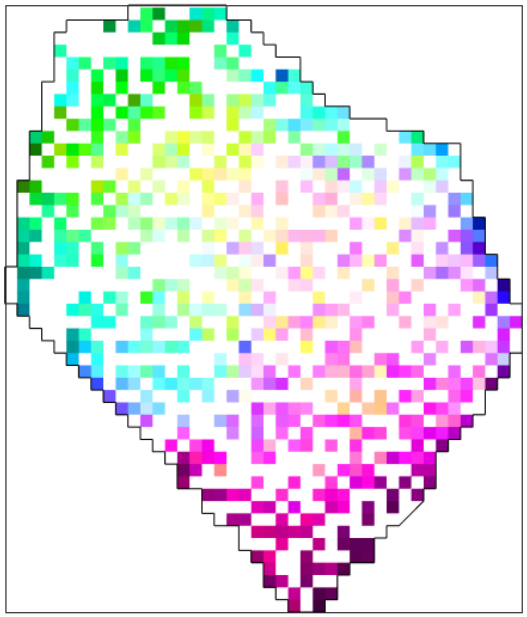
A deformed grain



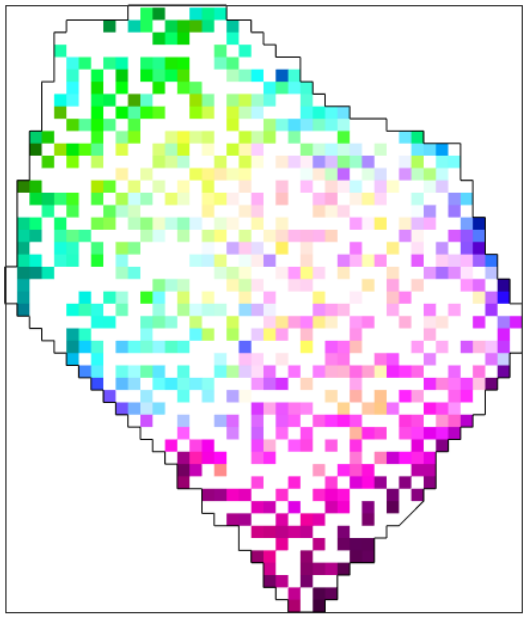
A deformed grain

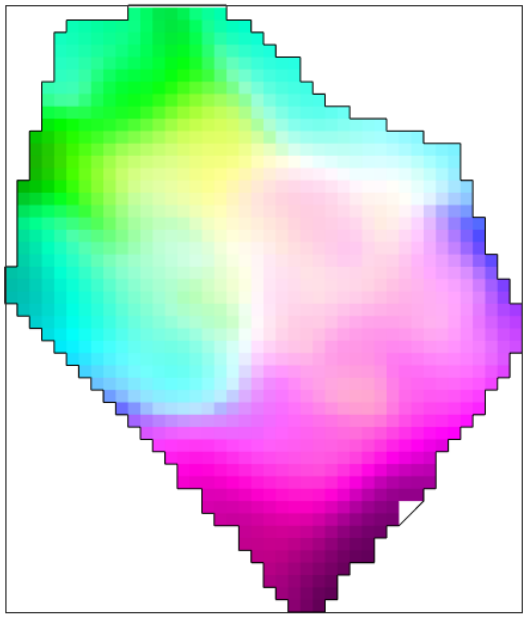


Not Indexed Data

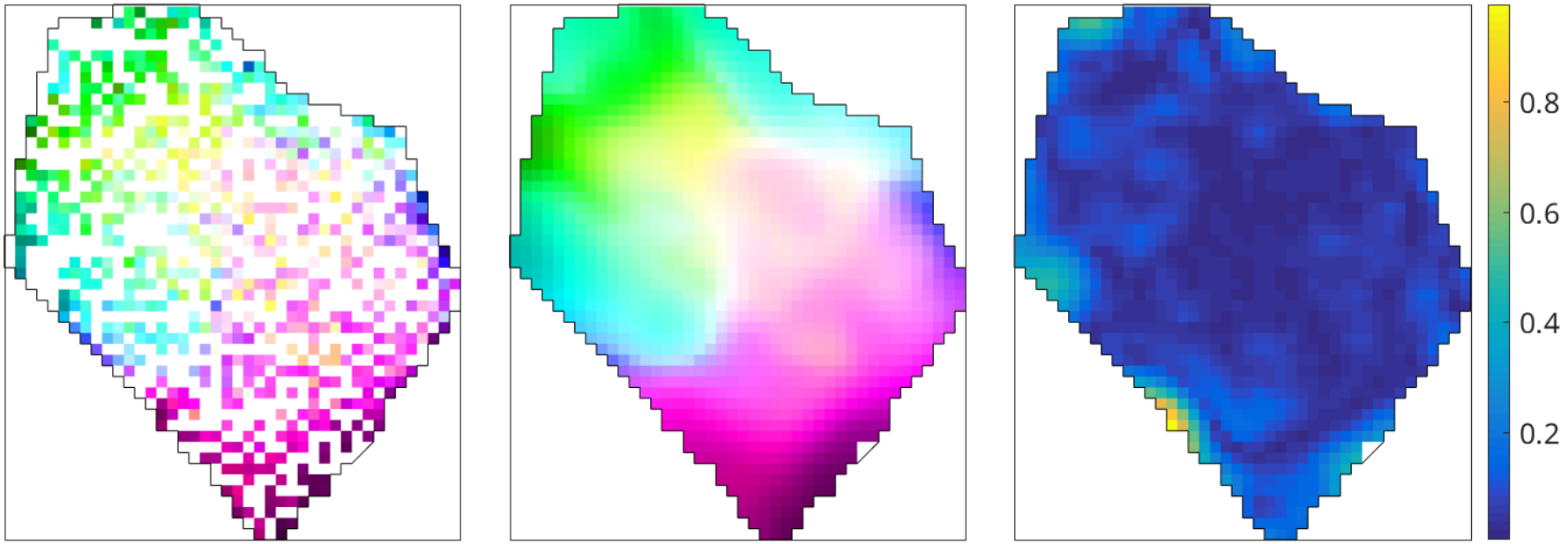


Not Indexed Data

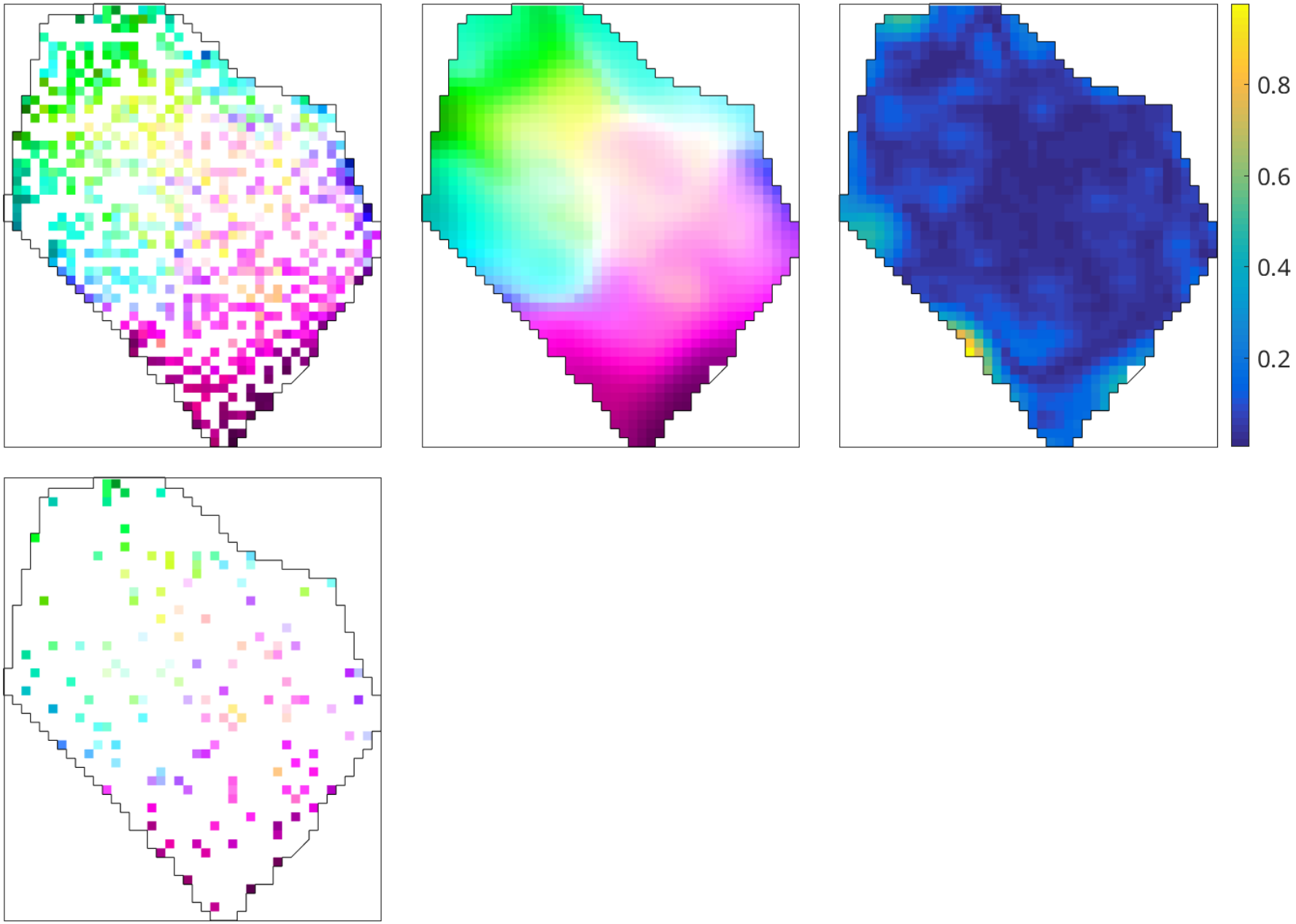




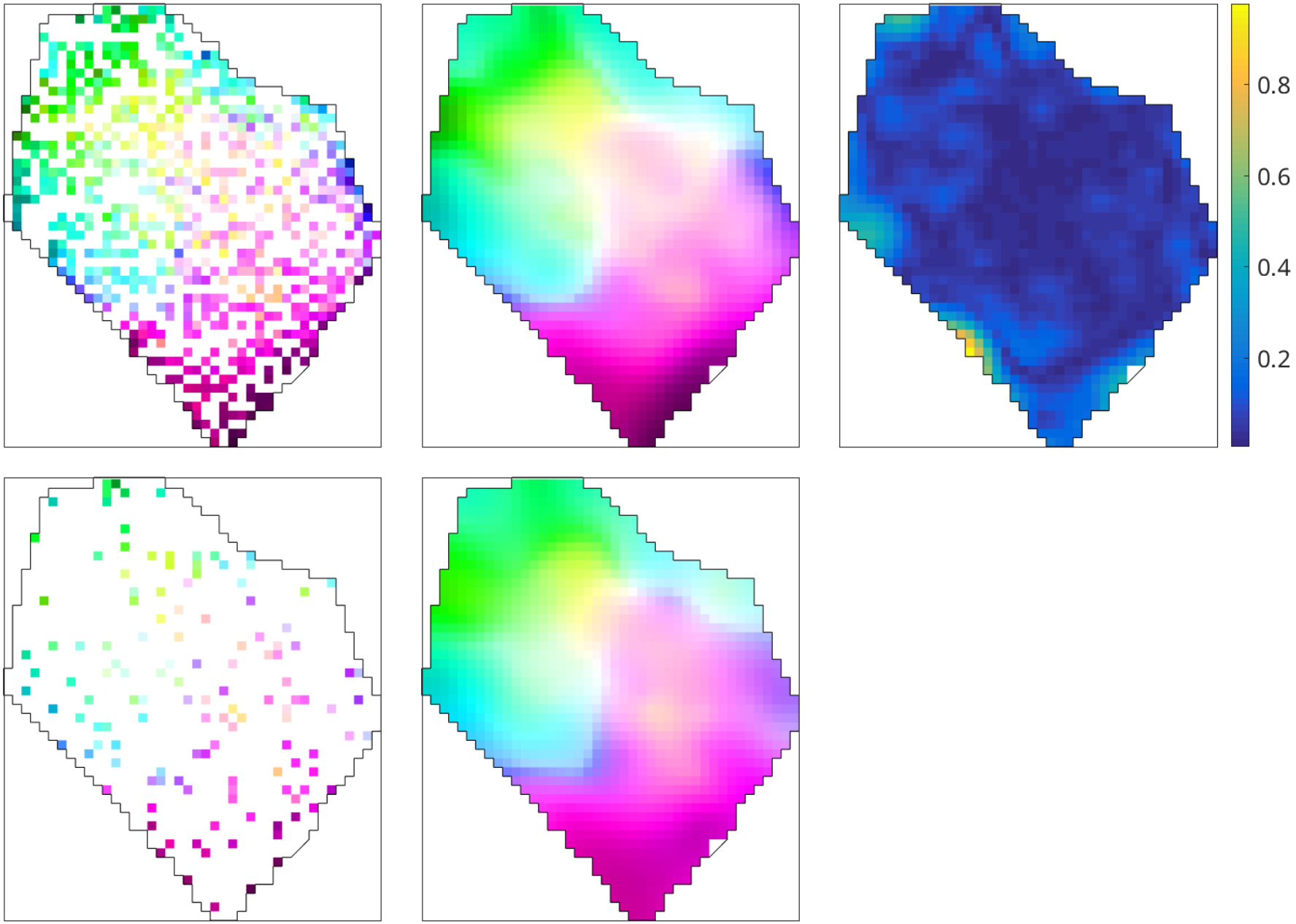
Not Indexed Data



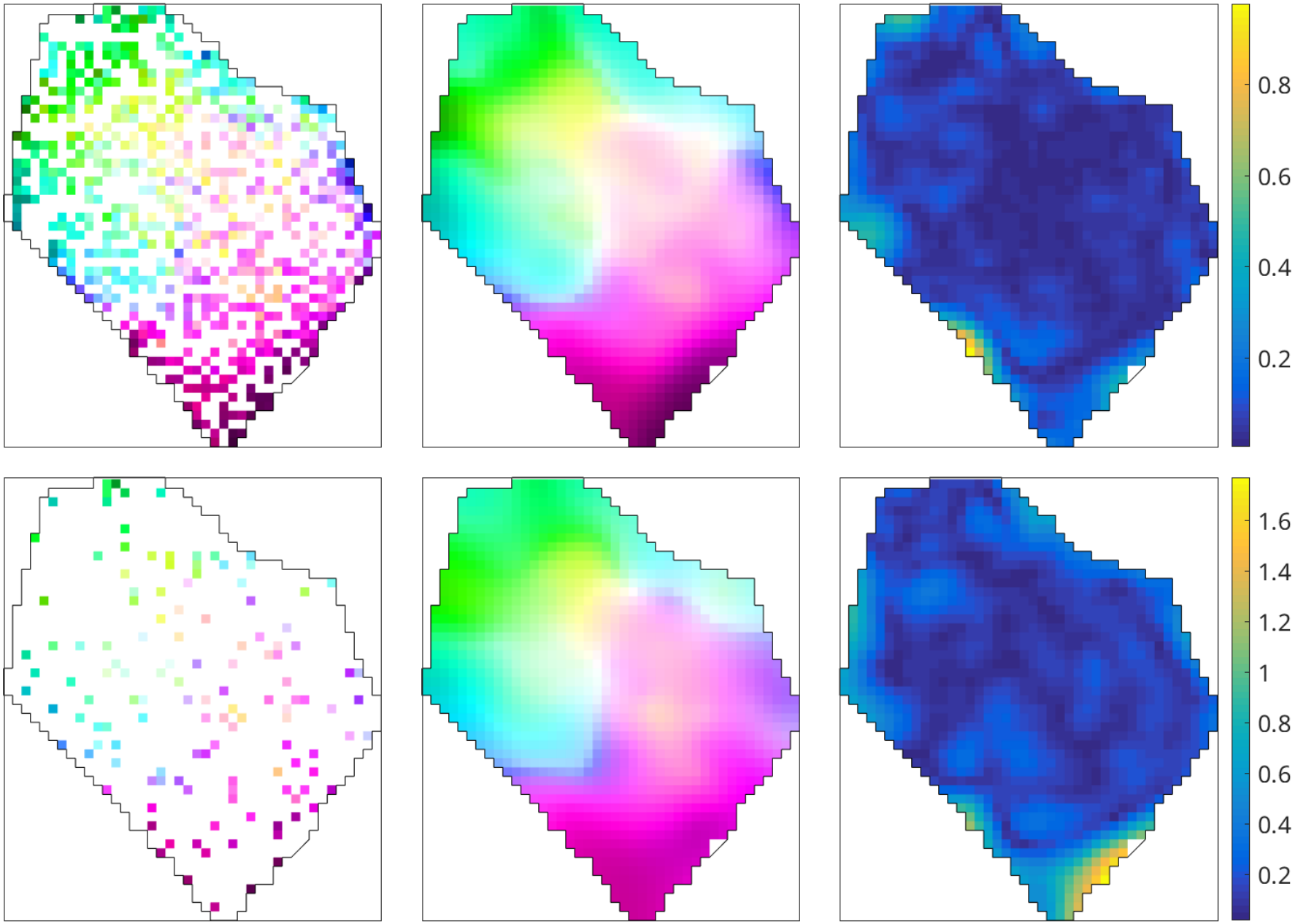
Not Indexed Data



Not Indexed Data



Not Indexed Data



Chemnitz MTEX Workshop 2016

Scope: bring together old and new MTEX users to share their ideas and applications

Schedule:
- MTEX beginners tutorial
- new developments in MTEX
- step by step introduction to specific use cases by invited speakers
- recent applications of MTEX

Date: February 2015

Location: Chemnitz University of Technology, Germany

Organizer: R. Hielscher