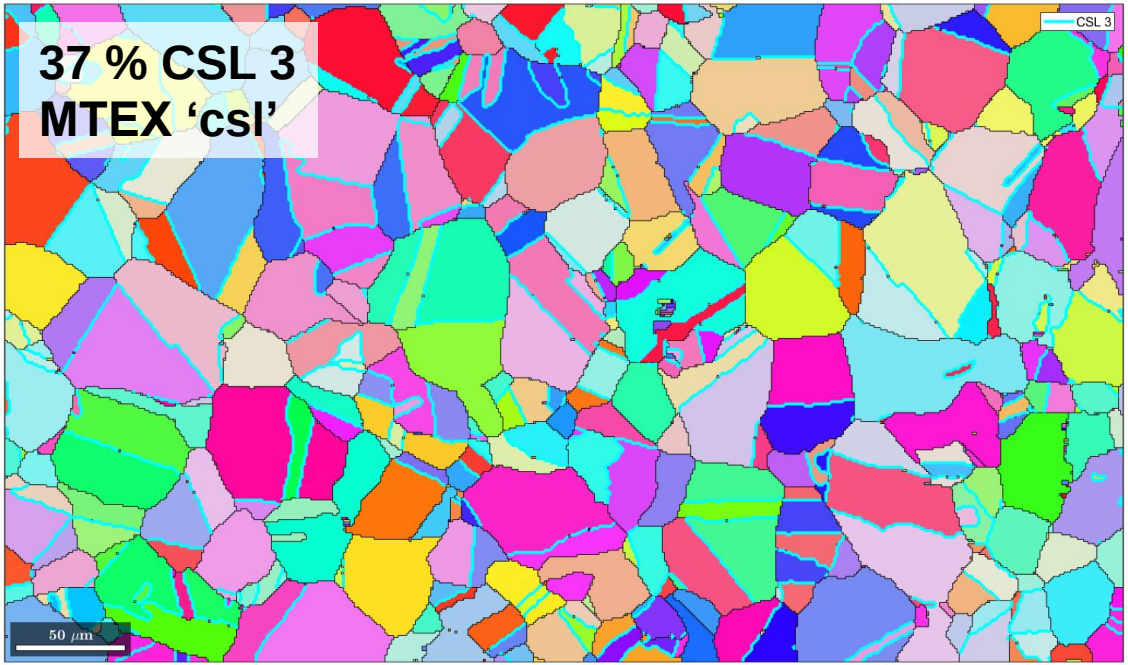
Annealing twins in reconstructed austenite

Tuomo Nyyssönen

Senior researcher, tuomo.nyyssonen@swerim.se

CSL 3 annealing twins

- Found in austenitic metals and alloys
- CSL 3 corresponds to
  - 60° around {111}
- At least one boundary plane in an annealing twin parallel with the CSL plane
- Grain boundaries faceted and angular in shape



Annealing twins in low-carbon steels

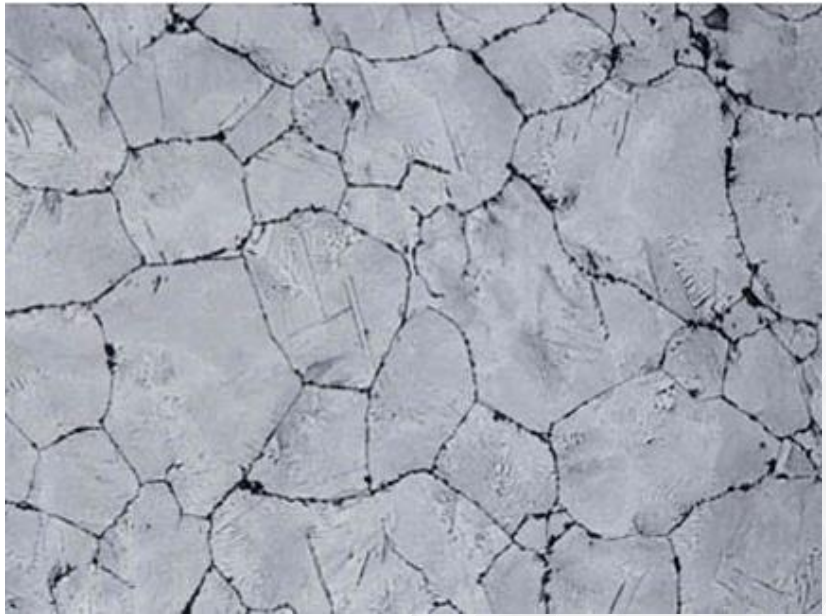
- At a high temperature, austenite undergoes a nucleation and growth process before the prior austenite structure is finished
- During growth, CSL 3 type twin boundaries nucleate at grain boundaries during the boundary migration steps

- Annealing twins in austenite…
  - Affect the nucleation and grain growth of austenite
  - Affect mechanical properties
- Necessary to characterize, just as important as (if not more than) other grain boundaries

Difficulties with finding PAG annealing twins

- Typically not visible in LOM, even when etching for PAG boundaries
  - Etching techniques based on segregation of impurities to high-energy grain boundaries
  - Twins have a high coherency and thus very low energy!

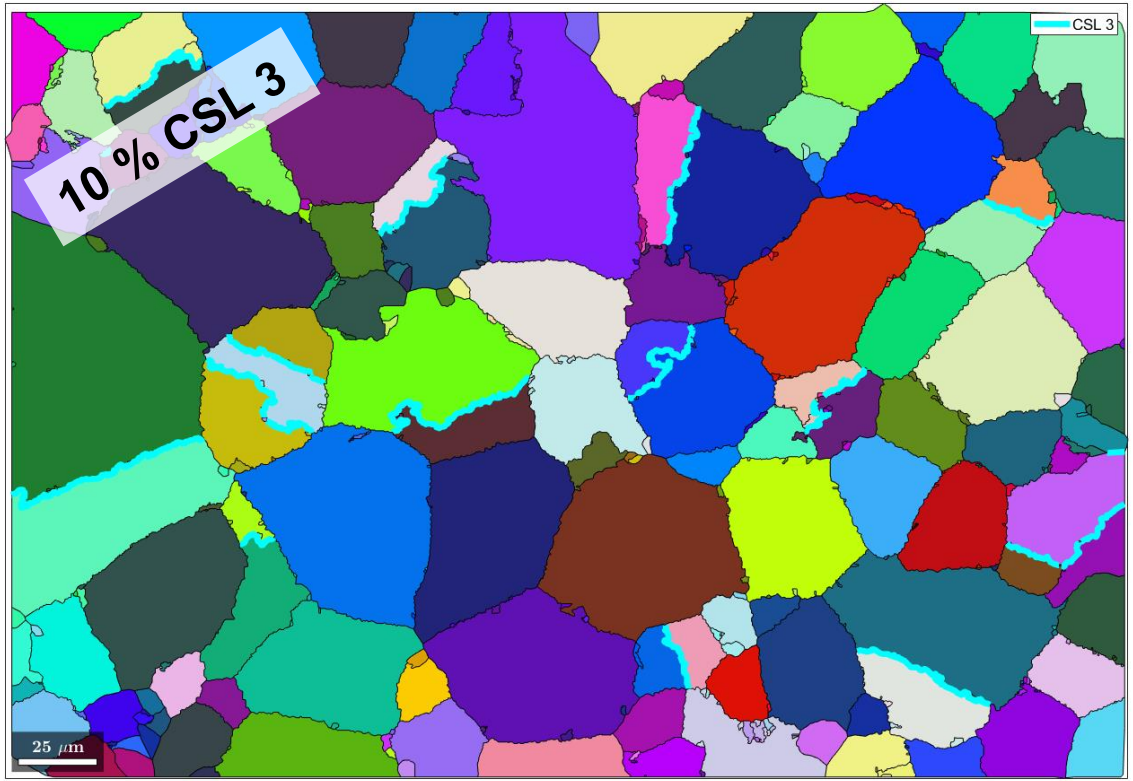
- What about **PAG reconstruction with EBSD**?



Source: Vander Voort, http://www.georgevandervoort.com/revealing-prior-austenite-grain-boundaries-in-heat-treated-steels-article/

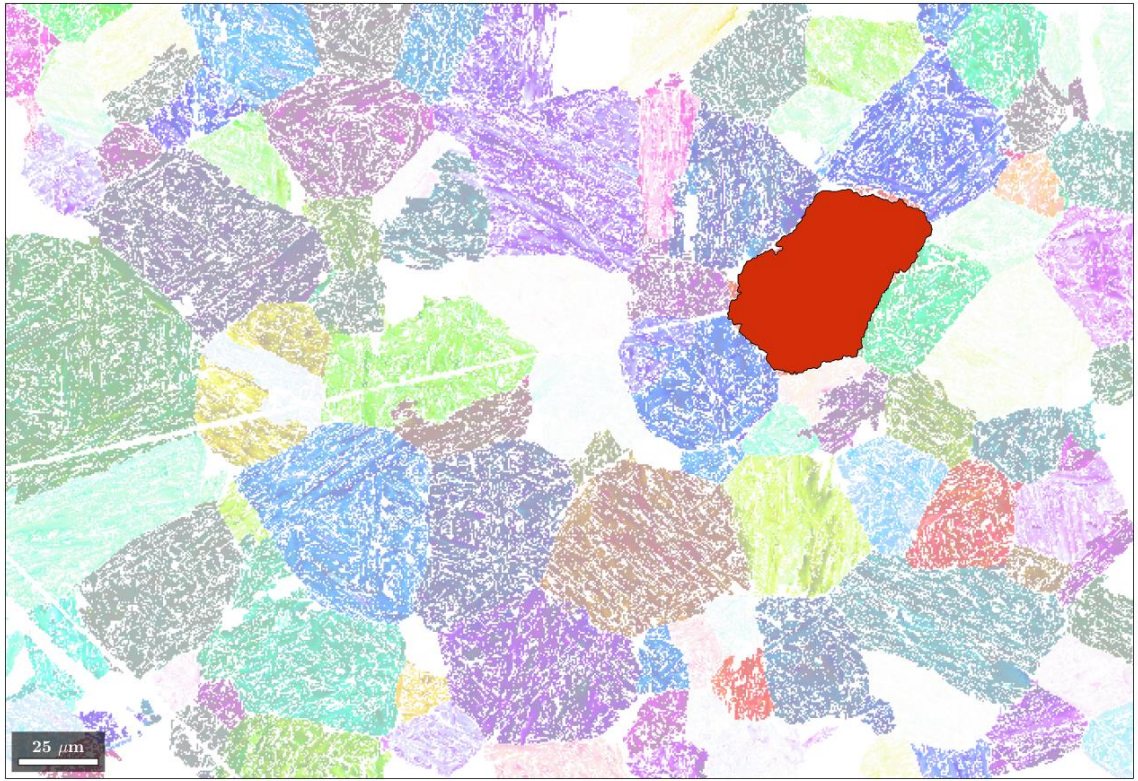
Example: MCL route

```
% load the data
mtexdata martensite

% grain reconstruction
[grains,ebsd.grainId] =
calcGrains(ebsd('indexed'),'angle',3*degree);
ebsd(grains(grains.grainSize < 3)).rotations =
quaternion.nan;
ebsd(grains(grains.grainSize < 3)).phase = 0;
ebsd(grains(grains.grainSize < 3)).grainId = 0;
[grains,ebsd.grainId] =
calcGrains(ebsd('indexed'),'angle',3*degree);
%%
job = parentGrainReconstructor(ebsd,grains) % define job
job.calcParent2Child; % get representative OR

%%
[fit,c2cPairs] = job.calcGBFit;
[gB,pairId] =
job.grains.boundary.selectByGrainId(c2cPairs);
plot(gB, fit(pairId));
mtexColorMap white2black

%%
job.calcGraph
job.clusterGraph
job.calcParentFromGraph
job.revert(job.grains.fit/degree > 5)
job.revert(job.grains.clusterSize < 15)

for k = 1:3 % do this three times
    job.calcGBVotes('noC2C');
    job.calcParentFromVote('minFit',7.5*degree)
end
```



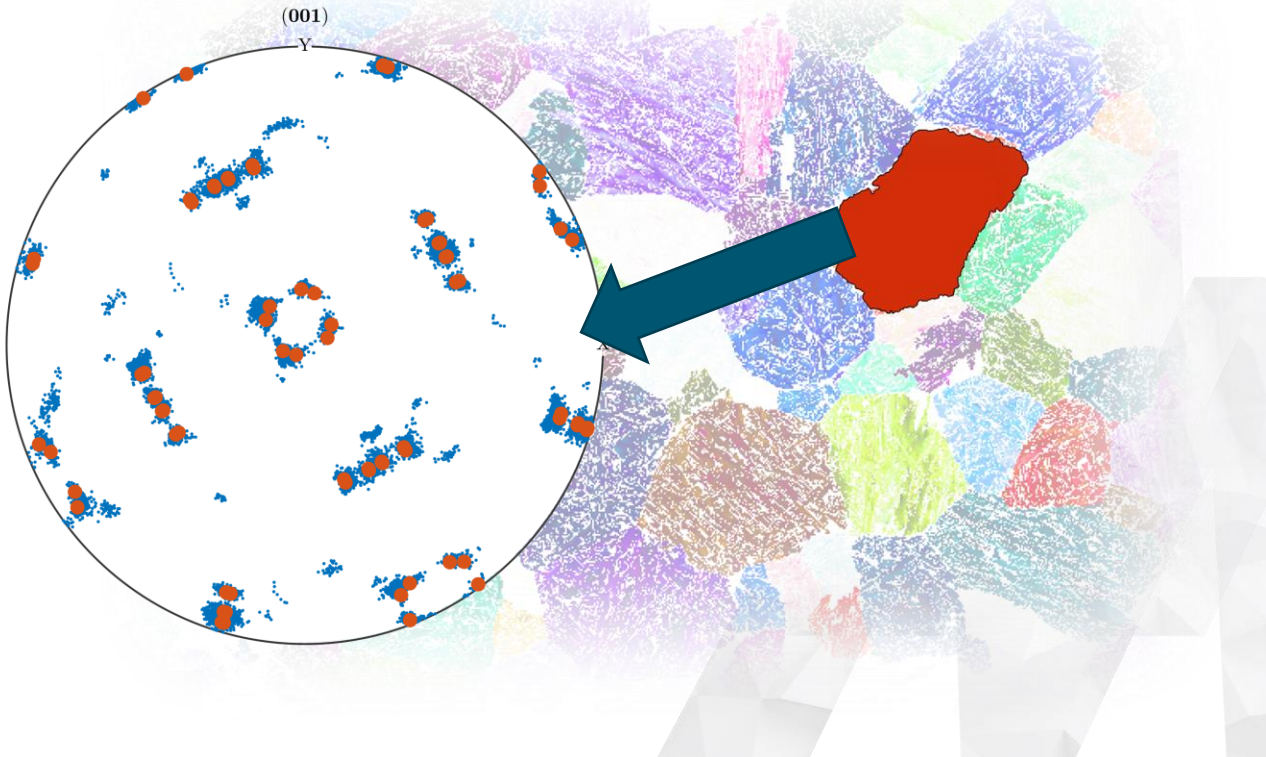
Were all the twins really indexed?

- Some twin boundaries have been identified
- Some suspicious reconstructions, especially at grain boundaries…
- Let’s pick a grain and investigate



Were all twins really indexed?

- It would appear that some orientations fall way outside the calculated variants for the determined prior orientation!



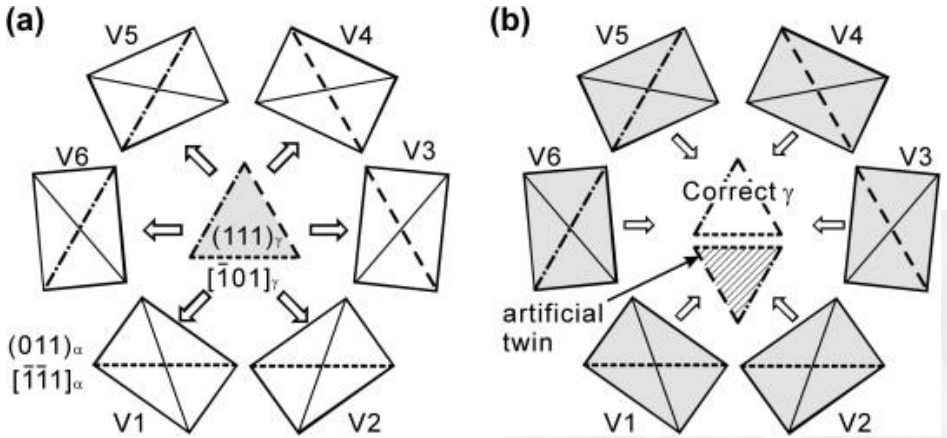
What's the problem?

- K-S orientation relationship:

  $(111)_\gamma \parallel (011)_\alpha$

  $[10\text{-}1]_\gamma \parallel [11\text{-}1]_\alpha$

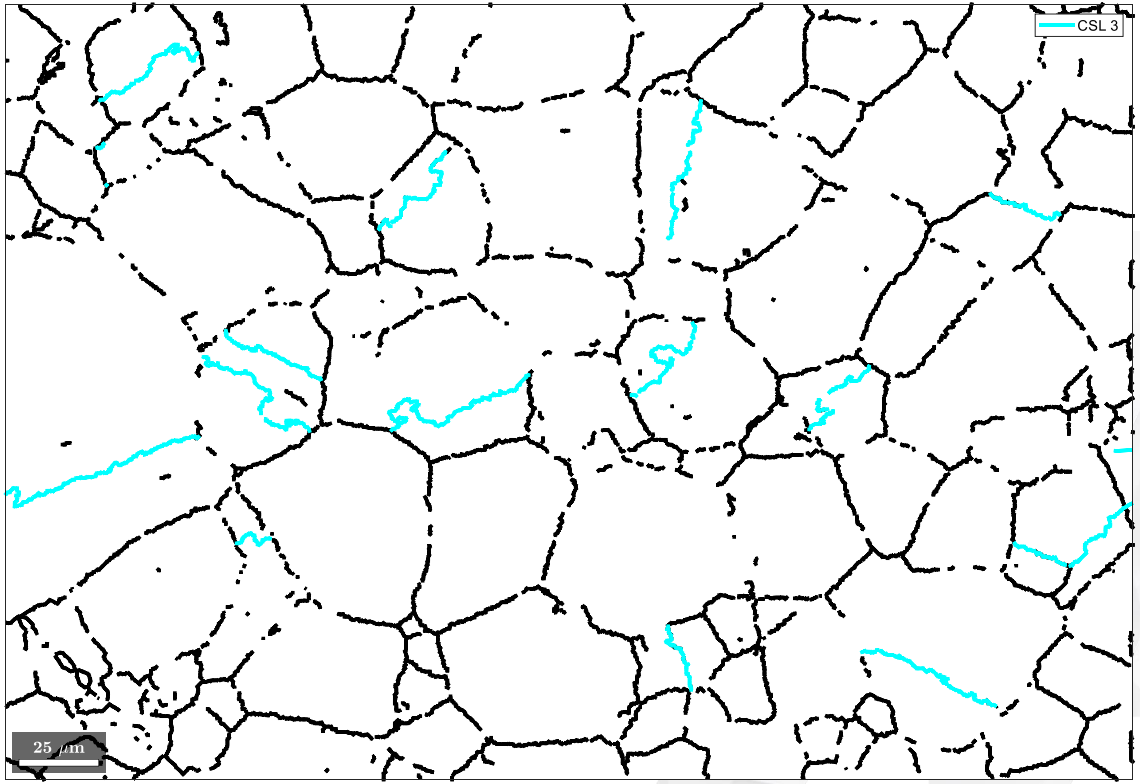
- Laths form on the $(111)_\gamma$
- Annealing twins share a (111) plane
  - Variants forming on this particular plane will be identical
- **Annealing twins basically share a packet, assuming K-S OR!**



G. Miyamoto et al., Acta Materialia, Volume 58, Issue 19, 2010, https://doi.org/10.1016/j.actamat.2010.08.001.

How does this relate to MCL clustering?

- Variant formation typically aims to accommodate plastic deformation during transformation
- To achieve this, variants from the same packet tend form next to each other
  - This also occurs across annealing twin boundaries
- **Result: twin boundaries cannot be characterized for MCL!**



Fortunately, MTEX has a lot of tools available.

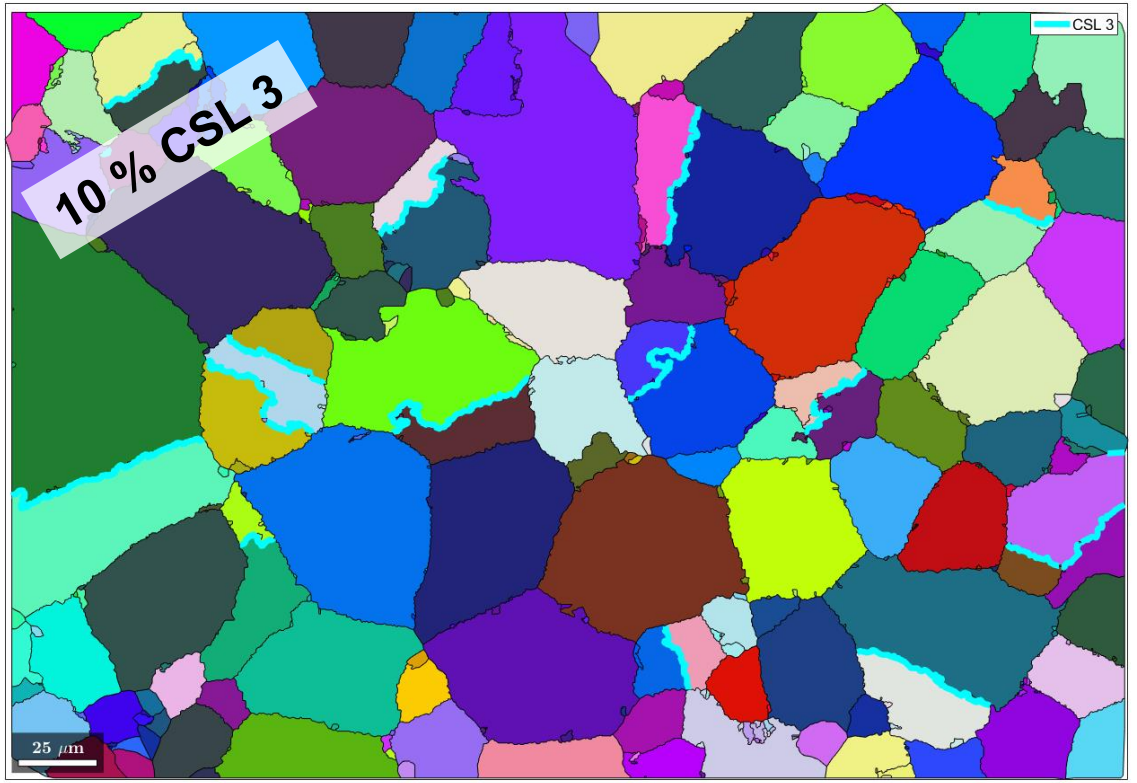
- The confidence in a correct prior austenite orientation increases when several neighbors can agree on a common solution

- Kerner + growth approach:
  1. Determine prior orientations that agree with 4 neighbors
  2. Use the reliable orientations as kernels, which are then grown into full austenite grains

Ranger, C. *et al. Metall Mater Trans A* **49,** 4521–4535 (2018). https://doi.org/10.1007/s11661-018-4825-7

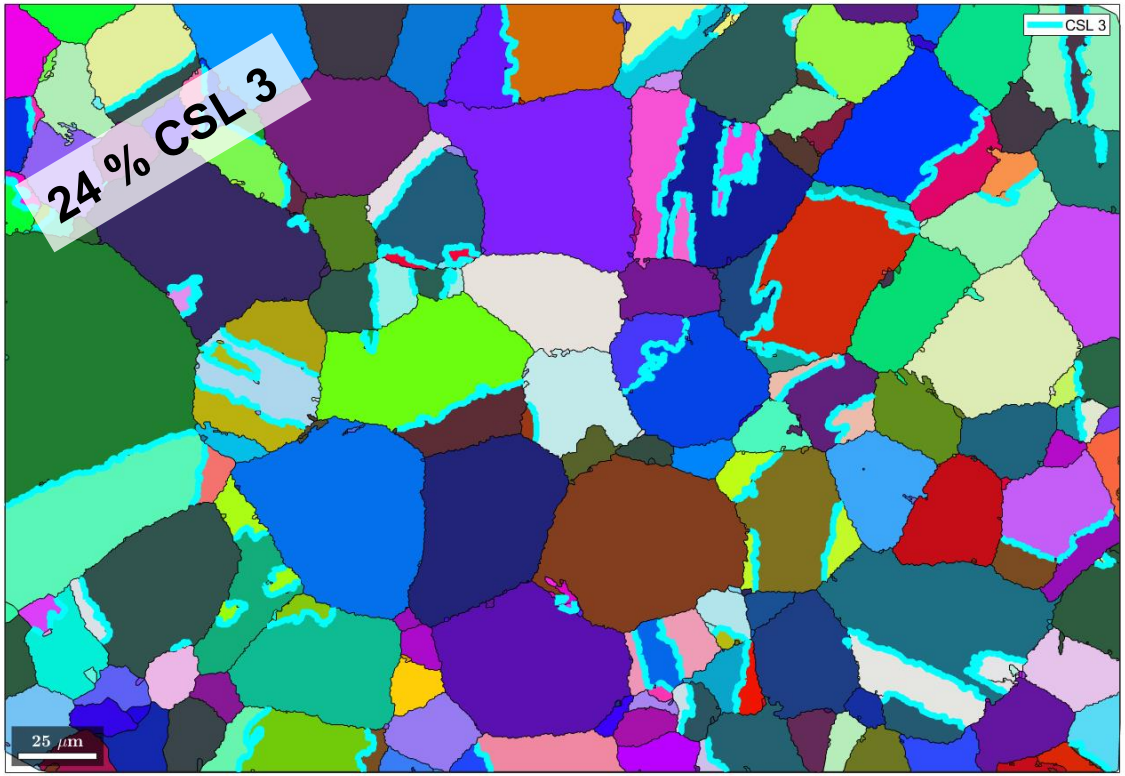
MCL route...

```
% load the data
mtexdata martensite

% grain reconstruction
[grains,ebsd.grainId] =
calcGrains(ebsd('indexed'),'angle',3*degree);
ebsd(grains(grains.grainSize < 3)).rotations =
quaternion.nan;
ebsd(grains(grains.grainSize < 3)).phase = 0;
ebsd(grains(grains.grainSize < 3)).grainId = 0;
[grains,ebsd.grainId] =
calcGrains(ebsd('indexed'),'angle',3*degree);
%%
job = parentGrainReconstructor(ebsd,grains) % define job
job.calcParent2Child; % get representative OR

%%
[fit,c2cPairs] = job.calcGBFit;
[gB,pairId] =
job.grains.boundary.selectByGrainId(c2cPairs);
plot(gB, fit(pairId));
mtexColorMap white2black

%%
job.calcGraph
job.clusterGraph
job.calcParentFromGraph
job.revert(job.grains.fit/degree > 5)
job.revert(job.grains.clusterSize < 15)

for k = 1:3 % do this three times
    job.calcGBVotes('noC2C');
    job.calcParentFromVote('minFit',7.5*degree)
end
```



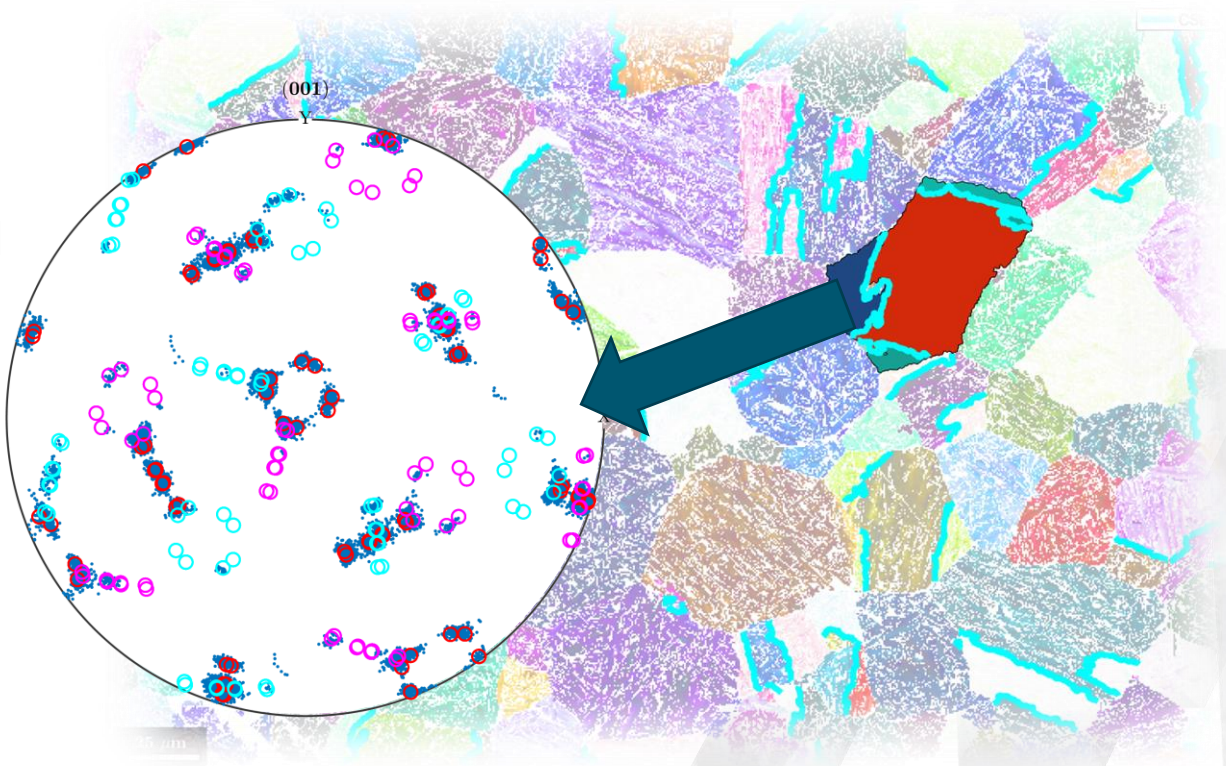
Kernel + growth approach (Ranger et al.)

```
% load the data
mtexdata martensite

% grain reconstruction
[grains,ebsd.grainId] =
calcGrains(ebsd('indexed'),'angle',3*degree);
ebsd(grains(grains.grainSize < 3)).rotations =
quaternion.nan;
ebsd(grains(grains.grainSize < 3)).phase = 0;
ebsd(grains(grains.grainSize < 3)).grainId = 0;
[grains,ebsd.grainId] =
calcGrains(ebsd('indexed'),'angle',3*degree);
%%
job = parentGrainReconstructor(ebsd,grains) %
define job
job.calcParent2Child; % get representative OR

%%
job.calcGBVotes % compute votes from child grain
boundaries
job.calcParentFromVote('minVotes',4)

%%
for l = 1:2
job.calcGBVotes('noC2C') % compute votes ONLY from
parent-child boundaries;
job.calcParentFromVote('minFit',5*degree) %compute
parent orientations from votes
end
```

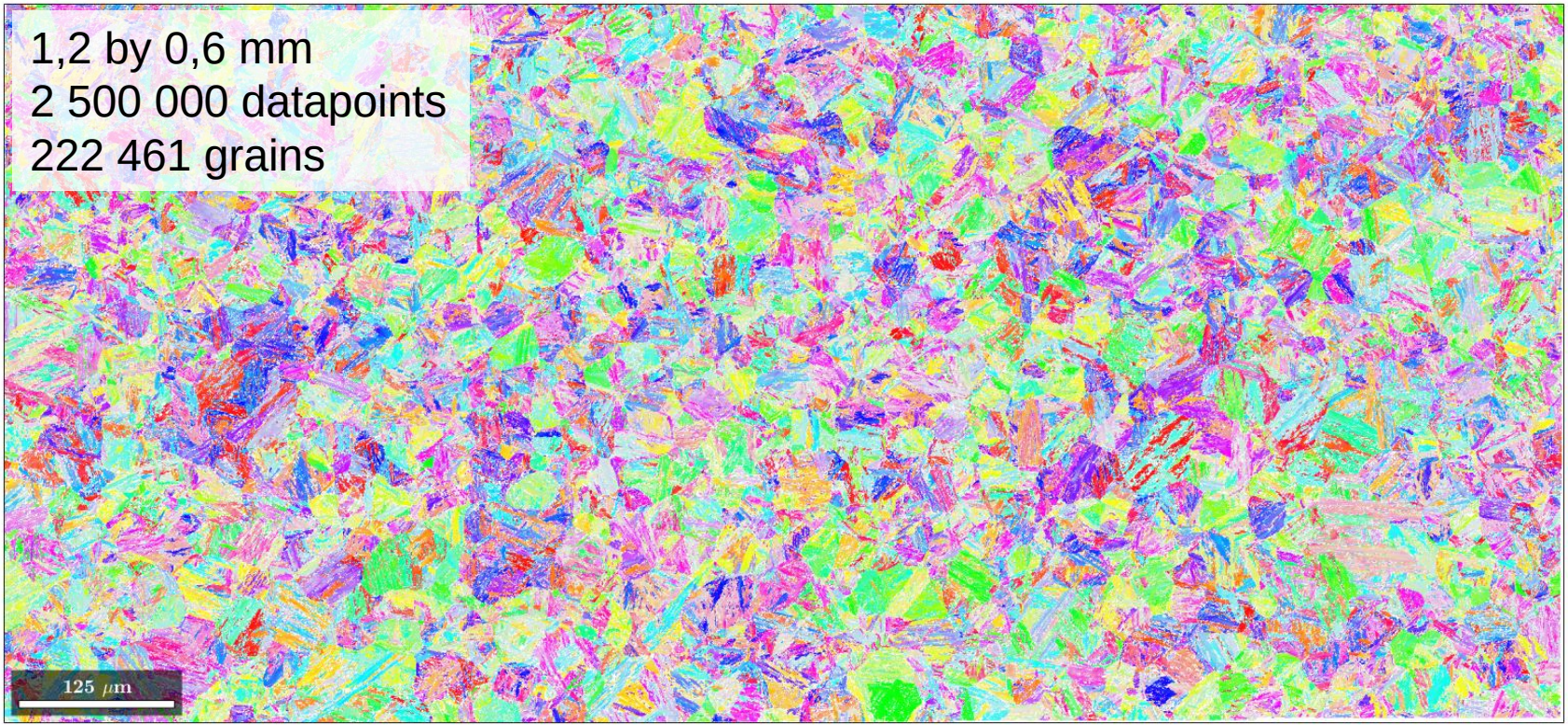


Were all twins really indexed?

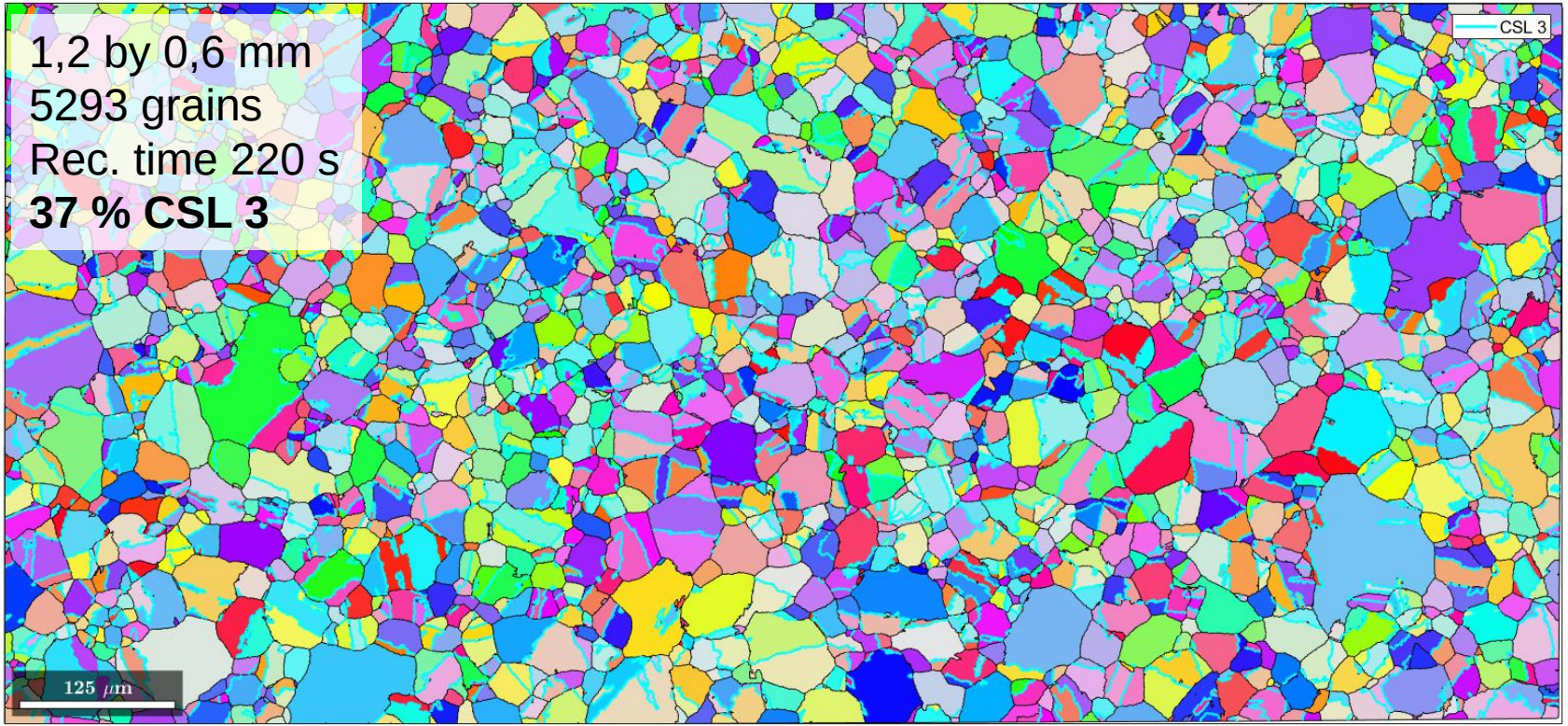
- Three twins present and identified!
- Some orientations still unaccounted for…



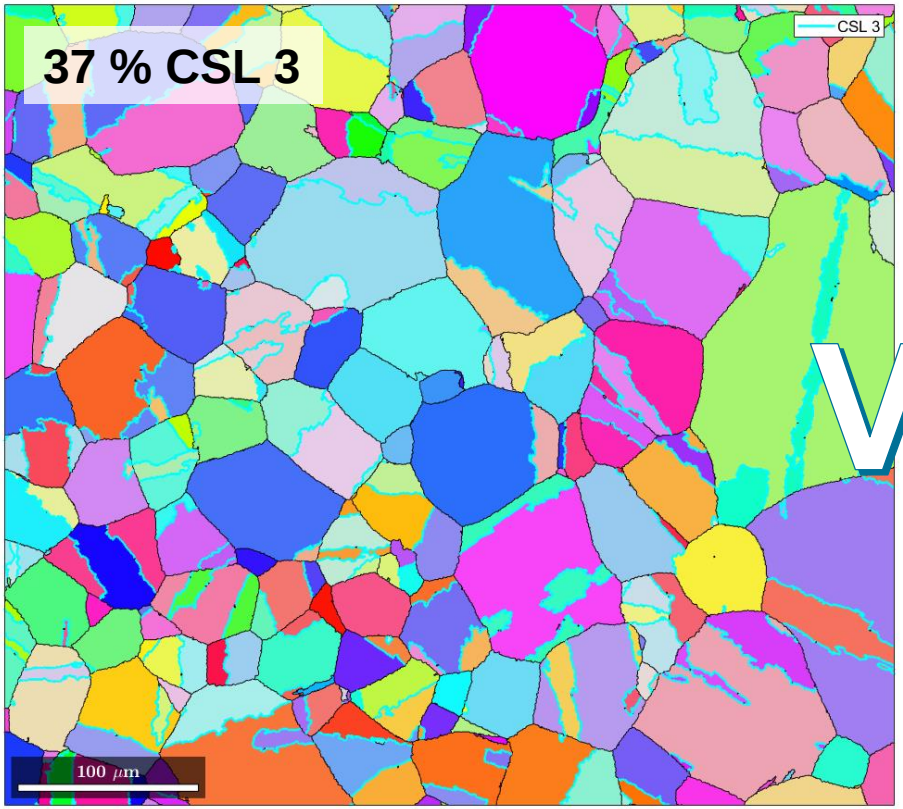
Kernel + growth method used to reconstruct a large area map



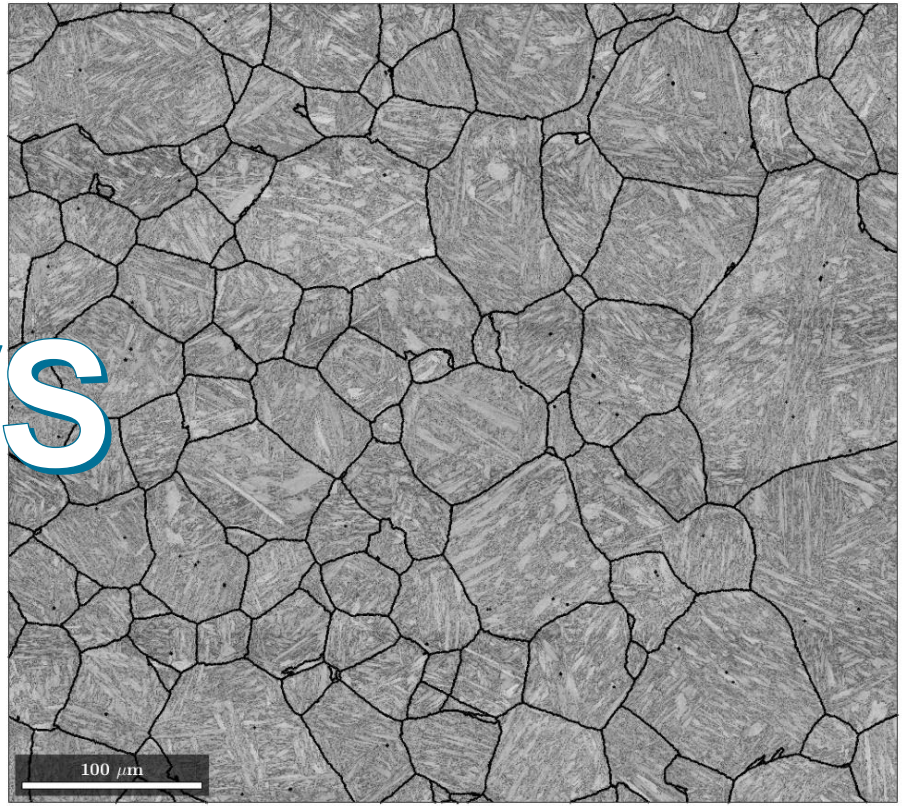
Kernel + growth method used to reconstruct a large area map



Twin boundaries are still challenging

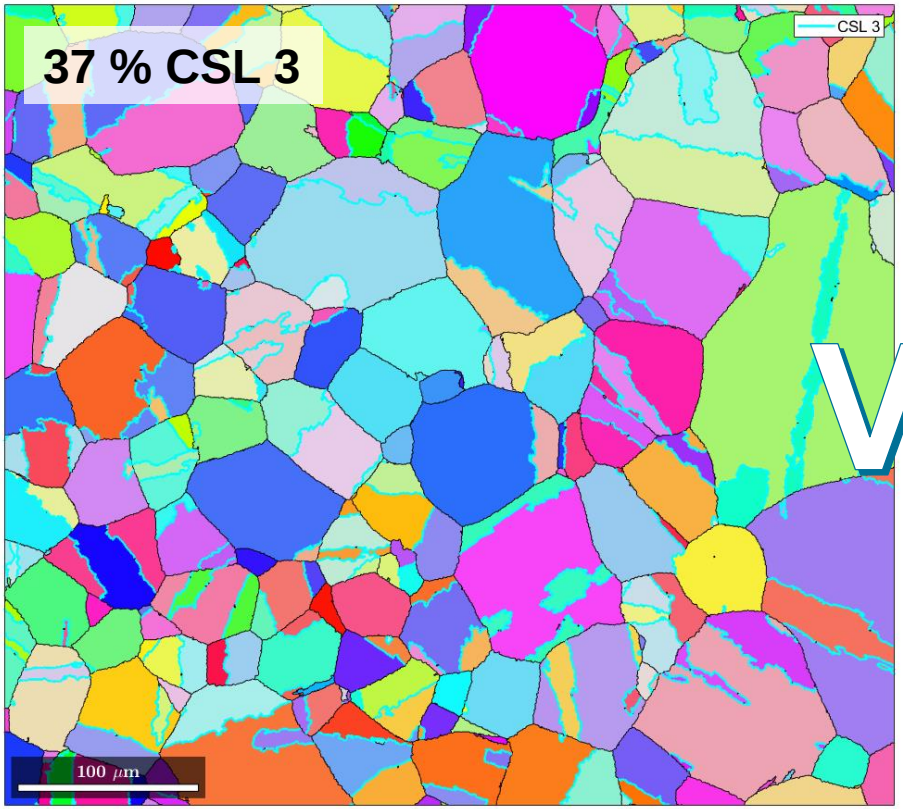


VS

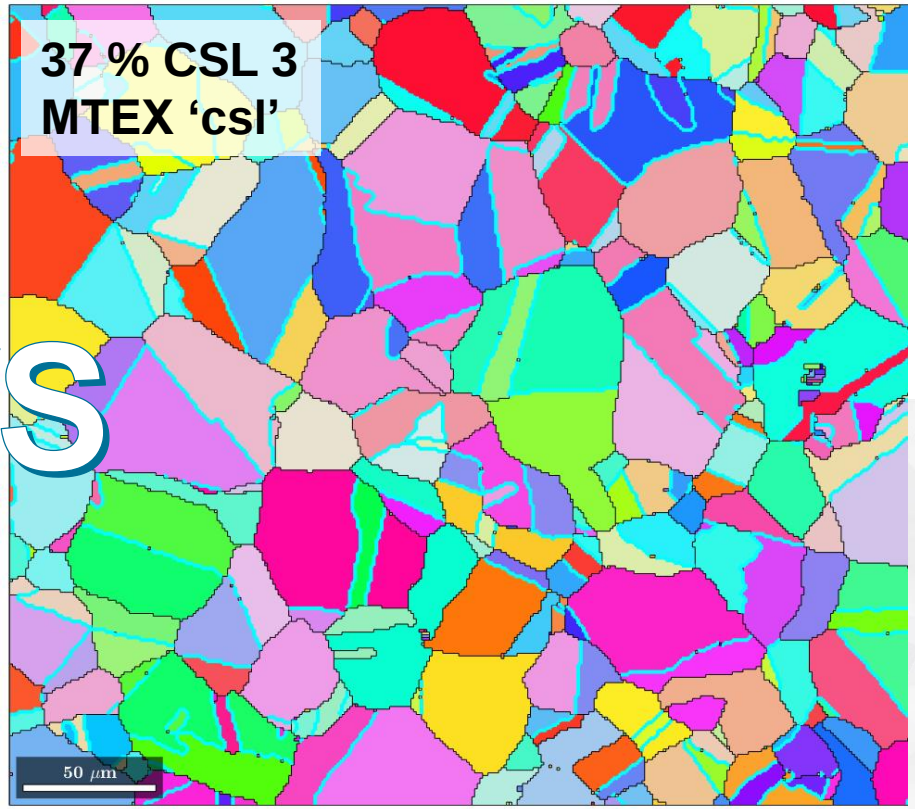


Twin boundaries are still challenging

SWERIM



VS



Conclusions

- Twin boundaries still difficult to index
- Kernel + growth approach helps a bit

- How to approach problem
  - Twin boundary does not cross packet boundary
    - Find problematic packets, test for best fit?

Tuomo Nyyssönen

tuomo.nyyssonen@swerim.se