Grain segmentation in atomistic simulations using orientation-based iterative self-organizing data analysis

M. Vimal [a], S. Sandfeld [a,b,c], A. Prakash [a,*]

[a] *Micromechanical Materials Modelling (MiMM), Institute of Mechanics and Fluid Dynamics, TU Bergakademie Freiberg, Lampadiusstraße 4, Freiberg 09599, Germany*
[b] *Institute for Advanced Simulation – IAS-9: Materials Data Science and Informatics, Forschungszentrum Juelich GmbH, Juelich 52425, Germany*
[c] *Chair of Materials Data Science and Materials Informatics, Faculty 5 – Georesources and Materials Engineering, RWTH Aachen University, Aachen 52056, Germany*

A B S T R A C T

Atomistic simulations have now established themselves as an indispensable tool in understanding deformation mechanisms of materials at the atomic scale. Large scale simulations are regularly used to study the behavior of polycrystalline materials at the nanoscale. In this work, we propose a method for grain segmentation of an atomistic configuration using an unsupervised machine learning algorithm that clusters atoms into individual grains based on their orientation. The proposed method, called the OrISODATA algorithm, is based on the iterative self-organizing data analysis technique and is modified to work in the orientation space. The working of the algorithm is demonstrated on a 122 grain nanocrystalline thin film sample in both undeformed and deformed states. The OrISODATA algorithm is also compared with two other grain segmentation algorithms available in the open-source visualization tool OVITO. The results show that the OrISODATA algorithm is able to correctly identify deformation twins as well as regions separated by low angle grain boundaries. The model parameters have intuitive physical meaning and relate to similar thresholds used in experiments, which not only helps obtain optimal values but also facilitates easy interpretation and validation of results.

1. Introduction

Polycrystalline materials, made of aggregates of single crystals or grains, constitute a large fraction of materials used today. The properties of such materials depend on the constituent microstructure, comprising of topological entities such as grain interior, grain boundaries (GBs), triple junctions and quadruple points. A well-known example is the Hall-Petch equation which relates the yield stress in the material to its mean grain size [1].

Of particular interest have been nanocrystalline (NC) materials, with mean grain sizes in the range of $10 \sim 100$ nm. The increased interest in NC materials in recent years has been spurred both by advances in processing and by insights obtained via computations [2–5]. Atomistic simulations of the molecular dynamics/statics kind have played a key role in elucidating deformation mechanisms in NC materials [6,7]. Such simulations have now clearly detailed the role of GBs in NC materials. For instance, the negative Hall-Petch effect observed in almost all NC materials at very small grain sizes is due to a transition of deformation mechanism from dislocation mediated plasticity to that dominated by grain boundary sliding [8,9]. Furthermore, GBs not only act as nucleation sites for dislocations due to the absence of intra-granular sources like Frank-Read or spiral sources, defect structures at GBs such as ledges

act as effective pinning points for nucleated and/or propagating dislocations [10–13].

It hence follows that understanding the influence of GBs and related topological entities is the key to obtaining structure property relationships for polycrystalline materials. An important task to this end in atomistic simulations is the partitioning of atoms into grain interior and grain boundary regions, and furthermore, to identify and track individual grains. Conventional characterization methods, like common neighbor analysis (CNA) [14,15], centrosymmetry parameter (CSP) [16], coordination analysis (CA) based on the local structural environment around an atom, or filtering methods based, e.g., on the potential energy of the atom, can be used to identify local crystal structure and assign atoms a corresponding label as "*bulk*" or "*defect*" atoms. Such an approach is taken by Tucker and Foiles [17] in their algorithm where fcc atoms (determined by CNA) are designated as *grain-center* or *grain-edge* atoms. But they are not universally applicable: CNA, CSP are not suitable for cases with low angle GBs; CA does not identify coherent twin boundaries.

Identification of grains hence requires calculation of and accounting for an inherent property: orientation. The grain tracking algorithm (GTA) developed by Panzarino *et al.* [18,19], involves the calculation of a local per atom crystallographic orientation using the geometry of

the unit cell (obtained by CNA and CSP). This is followed by an iterative process to identify individual grains as contiguous regions where the misorientation between nearest neighbors of atoms is less than a pre-defined threshold. A similar approach is taken by Hoffrogge and Barrales-Mora [20] who suggest an additional "*global*" criterion to track the misorientation to the mean orientation of the grain. Such methods are, however, sensitive to local perturbations in the structural environment resulting from, e.g., strain or temperature.

Partitioning of a dataset into different clusters, e.g. grains, is a well researched unsupervised machine learning problem. The idea is to form a cluster with data points that are as similar as possible to each other within the cluster, whilst being as different as possible to data points in a different cluster. The open-source visualization tool OVITO [21] provides implementations of two hierarchical clustering methods in its grain segmentation modifier. The first approach is similar to the GTA of Panzarino *et al.* [18,19], but uses the minimum spanning tree representation of the input structure, and additionally, computes the local atomic orientation using the polyhedral template matching [22] algorithm. The second approach also uses graph clustering, but with different weights for the graph edges in comparison to the former. Grains are subsequently built by contracting graph edges using the node sampling method [23]. With its automatic mode, the algorithm chooses a good threshold value using a sequence of graph merging steps. The drawback, however, is that the threshold value (in both automatic and manual) has no intuitive physical meaning.

In this work, we propose an alternative approach that uses a centroid based partitioning technique and shows advantages to the graph clustering algorithms particularly for large datasets [24,25]. Our approach, called OrISODATA algorithm, is based on the iterative self organizing data analysis (ISODATA) method [26], an unsupervised learning algorithm that is widely employed in remote sensing applications [27–30]. An extension of the K-means algorithm, ISODATA method has the advantage of automatically selecting the final number of clusters based on certain heuristics. The self organizing capabilities and the ability to *split* clusters with larger spread and *merge* similar clusters based on the thresholds are its key advantages.

The OrISODATA algorithm retains the basic structure of the ISODATA algorithm suggested by Ball *et al.* [26], but modifies the *split* and *merge* procedures to account for the non-Euclidean nature of the orientation space. The method works for both undeformed and deformed states, and is able to identify both low angle GBs and twinned regions well. In any clustering problem, the threshold parameters can have a significant influence on the final clustering results and must be carefully chosen [31,32]. Herein lies the advantage of the OrISODATA algorithm: the intuitive nature of the threshold parameters, which essentially split and merge clusters using orientation spread and misorientation between clusters, help in obtaining optimal parameter values easily and ensure robust and reliable results.

2. Details of the algorithm

The generic ISODATA algorithm involves an iterative approach of splitting clusters based on the standard deviation of the data points along each dimension, and merging two clusters based on their Euclidean inter-cluster distance. A detailed description of the ISODATA algorithm can be found in Ball *et al.* [26]. The overall idea is to reduce the variation in each cluster and to combine similar clusters over the iterations.

The basic steps involved in a generic ISODATA clustering algorithm are,

1. Randomly sample the initial centroids for the given initial number of clusters
2. Sort data points based on their proximity to the cluster centroids
3. Recompute the cluster centroids
4. Split clusters if the *standard deviation* along any dimension is greater than the user defined split threshold

5. Merge pairs of clusters if their *inter-cluster Euclidean distance* is lower than the user-defined merge threshold
6. Go to step 2 until convergence is achieved

2.1. Orientation-based ISODATA clustering

The use of standard deviation and the Euclidean distance metric in the generic ISODATA algorithm, as measures for variation within a cluster and similarity between clusters, respectively, is insufficient for identifying grains based on orientations, since orientations do not reside in Euclidean space. We hence propose a modified ISODATA algorithm —the OrISODATA algorithm— by incorporating orientation-based metrics into the cluster split and merge procedures. The general structure of the main program that executes the OrISODATA algorithm is shown in Algorithm 1 .

Algorithm 1: Main program.

Input:
1. Atomistic configuration (AtomId, Position, Orientation)
2. Lattice constant: a_0
3. Lattice structure type: *e.g. fcc/bcc/hcp*
4. Per-atom volume: V_i
5. Cluster split threshold angle (Maximum disorientation spread within agrain): Ψ_{split}
6. Cluster merge threshold angle (Minimum disorientation angle between twograins): Φ_{merge}
7. Initial number of clusters: n_{clus}^{init}
8. Minimum number of atoms per cluster *or* Minimum grain size: D_g^{min}
9. Maximum number of iterations: n_{iter}
10. Convergence tolerance: TOL

Output: Clusters (Grains)
1 Data import;
2 Extract bulk atoms (using Lattice structure type) from therest of the sample;
3 OrISODATA clustering → Algorithm 2; `// Only the bulk atoms data are used as input for clustering`
4 **for** $i \leftarrow 1$ **to** *Number of orphan atoms* **do**
5 | Assign orphan atom → Nearest large cluster
6 **end**
7 Merge clusters → Algorithm 4; `// Iterative merge until no clusters get merged`
8 **for** $j \leftarrow 1$ **to** *No of clusters* **do**
9 | Calculate geometric centroid of each cluster
10 **end**
11 Data export

The input to the OrISODATA clustering algorithm is a snapshot of an atomistic simulation, which contains the positions, local crystal structure and orientation of individual atoms. We use the polyhedral template matching algorithm (PTM) [22] as implemented in OVITO [21] to determine the local orientation and atomic structure of each atom. Atomic volume is computed using a Voronoi tessellation. The structure of the OrISODATA algorithm is shown in Algorithm 2 .

Briefly, the program works as follows: atoms in the snapshot are first classified into bulk and non-bulk atoms using the local atomic structure (here, fcc). Subsequently, the OrISODATA algorithm (see *Algorithm* 2) is applied on the bulk atoms, which organizes them into individual clusters. Herein, all clusters that have fewer than a predefined number of atoms are designated as small/orphan clusters. In an iterative procedure, large (non-orphan) clusters are then split and merged using user-defined thresholds for the same. The details of the split and merge routines are presented below. Subsequently, all atoms in orphan clusters and non-bulk atoms that were previously neglected, are assigned to a large clus-

---

Algorithm 2: OrISODATA clustering.

Input: 1. Position and orientation of atoms, 2. Values of clustering parameters

Output: 1. Clusters, 2. Mean orientations of large clusters

1   Assign atoms to clusters randomly ;   `// Number of clusters,` $n_{clus} = n_{clus}^{init}$

2   Calculate mean orientation of each cluster ; `// L2-chordal mean and atomic volume as weights at every calculation`

3   **for** $i \leftarrow 1$ **to** n_{iter} **do**

4   Calculate orientation distance of each atom to mean orientation of each individual cluster;

5   Assign atoms to the nearest cluster based on orientation distance;

6   Update mean orientation of each cluster;

7   Ignore clusters with number of atoms $< D_g^{\min}$;   `// Data points in the ignored clusters are not deletedbut only ignored for the remaining iterations`

8   Update array of mean orientations and n_{clus} ;   `// If number of ignored clusters` ≥ 1

9   **Split clusters** → Algorithm 3**if** *Clusters split* **then**

10   Update array of mean orientations and n_{clus} ;

11   Calculate orientation distance of each atom to the mean orientation of all clusters ;

12   Assign atoms to the nearest cluster based on orientation distance;

13   Update array of mean orientations and n_{clus}

14   **end**

15   **Merge clusters** → Algorithm 4Update array of mean orientations and n_{clus} ;   `// If any cluster gets merged`

16   Δ mean orientation = Current mean orientation - Last mean orientation;

17   Last mean orientation ← Current mean orientation ;

18   **if** $\left(i == n_{iter} \right)$ **OR** $\Big($ *absolute max (Δ mean orientation)* $<=$ TOL $\Big)$ **then**

19   **for** $j \leftarrow 1$ **to** n_{clus} **do**

20   Position based cluster split ;   `// mean of` 1st `and` 2nd `neighbor distance` → `Cut-off distance`

21   **end**

22   **Split clusters** → Algorithm 3;   `// Iterative split until all the newly split (position based) clustersmaintain their grain orientation spread (GOS) lower than` Ψ_{split}

23   Append all the ignored clusters (line 7) to the array of currentclustersDesignate clusters as large and orphan clusters based on $D_g^{\min}$;

24   Calculate mean orientation of each large cluster**return** Clusters, mean orientations of clusters

25   **end**

26   **end**

---

ter to which they are in close positional proximity. Finally, the average orientation and center of mass of each cluster are calculated.

The orientation distance metric used in the current work is the misorientation angle defined via the geodesic distance metric [33,34] and is given by the dot product of two orientations (here, quaternions) q_A and q_B as follows:

$$\theta = 2\cos^{-1}\left(q_A \cdot q_B\right). \tag{1}$$

Further details on the calculation of the orientation distance are provided in the supplementary material.

For calculating the mean orientation of each cluster, the chordal L2 mean (projected/induced arithmetic mean) [35,36] based on the cost minimization method is used. The cost function is given by

$$C(R) = \sum_{i=1}^{n} (R_i - R)^2 , \tag{2}$$

where R is the average orientation of a set of orientations (R_i) that minimizes the cost function $C(R)$.

After convergence, the final clusters are split based on their positional connectivity by analyzing the connected components in a sparse graph, constructed using the atomic positions of each cluster [37] with the neighbors of each atom within a given cutoff distance. Fig. 1a shows the position based cluster split of 3 regions having identical orientations, as a result of which they would be classified to a single cluster. These regions are, however, not directly connected to each other. The average of the 1st and 2nd neighbor distance for the given lattice structure type is chosen as the neighbor cutoff distance (d_{cutoff}) for evaluating the connectivity.

2.2. Cluster split

The algorithm for splitting clusters using an orientation-based metric is shown in Algorithm 3 . For computing the spread of orientations in

---

Algorithm 3: Split clusters.

Input:

  1. Clusters

  2. Ψ_{split}

Output: Clusters after split

1   **Function** *get GOS(Cluster)* **is**

2   Calculate orientation distance of all atoms with respect to the mean orientation(q_m) of the given *cluster*;

3   Find the corresponding orientation (q_{max}) that has the maximumorientation distance (θ_{max});

4   Calculate the GOS (Ψ_{spread}) fromthe orientation distances**return** q_{max}, Ψ_{spread};

5   **end**

6   **for** $i \leftarrow 1$ **to** *Number of clusters* **do**

7   Call *get GOS(i^{th} cluster)***if** $\Psi_{spread} > \Psi_{split}$ **then**

8   Split cluster into two clusters;

9   Assign q_{max} and q_m as mean orientations of thenew clusters ;   `// Only for initial data sorting`

10   Assign atoms from the unsplit cluster to the new clusters by calculating theangle-based nearest-neighbor;

11   **end**

12   **end**

---

each cluster, the grain orientation spread (GOS) is used. This metric is also used to characterize deformed microstructures using electron backscatter diffraction (EBSD) techniques [38–40]. The GOS (Ψ_{spread}) of a cluster is given by Allain-Bonasso et al. [41]:

$$\Psi_{spread} = \frac{1}{N} \sum_i \theta_{q_i \leftrightarrow q_m}, \tag{3}$$

where N is the number of orientations (atoms) in the cluster under consideration and ($\theta_{q_i \leftrightarrow q_m}$) is the orientation distance between the orientation (q_i) of the atom and mean orientation (q_m) of the cluster. Fig. 1b shows a schematic of a cluster with dissimilar orientations split into two different clusters. All clusters satisfying the split condition, $\Psi_{spread} > \Psi_{split}$, are split into two different clusters.

2.3. Cluster merge

Clusters are merged (see Algorithm 4) based on the inter-cluster disorientation (ϕ) calculated using the mean orientations of the clus-
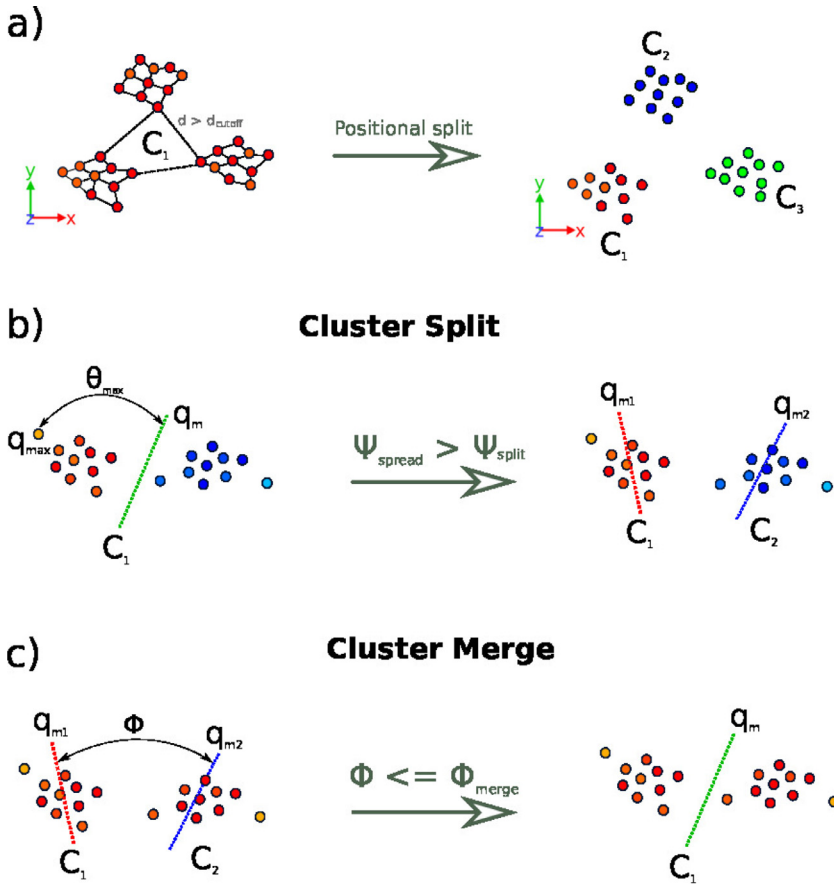
Fig. 1. Schematic representation of the a) positional split, b) cluster split and c) cluster merge processes.

ters. Fig. 1c shows a schematic of two clusters with similar orientations, merged into a single cluster. Any two clusters satisfying the condition, $\phi <= \phi_{merge}$, with ϕ_{merge} being the user-defined merge threshold, qualify as a cluster pair for the merge operation. All such cluster pairs are sorted in ascending order of their disorientation angle (ϕ) to allow the cluster pairs having the lowest disorientation angle to merge first. Finally, for each of such cluster pair, if neither of the clusters is already merged to other clusters and additionally, if the maximum spread angle (Ψ_{spread}) of the final cluster after merging is less than the user defined split threshold angle (Ψ_{split}), they are merged together as a single cluster. This additional condition using the split threshold is to avoid repeated split and merge of the same clusters over the iterations due to the incompatible input threshold values, i.e. split threshold << merge threshold.

3. Methods for comparison

To verify the working of our OrISODATA algorithm, we compare the results with two other algorithms – automatic graph clustering and minimum spanning tree – implemented in the open-source visualization tool, OVITO [21]. A common feature of both algorithms is that they use a nearest neighbor graph to perform agglomerative hierarchical clustering. These algorithms are available under the *Grain Segmentation Modifier* in OVITO and are documented as "experimental" versions in the OVITO manual [42]. As a result, we only use them for comparing the results of our algorithm for a polycrystalline sample and do not undertake an in-depth study of the algorithms itself. For this study, the options *Adopt orphan atoms* and *Handle coherent interfaces/stacking faults* are enabled. Note that these options are only available with the grain segmentation algorithms of OVITO. The former option assigns orphan atoms at, e.g. GBs, to the nearest grain and is similar to the approach used in our OrISODATA algorithm. The latter merges atoms having hcp

crystal structure with atoms having cubic crystal structures at stacking faults and/or other types of coherent interfaces [42]. This feature is, however, not a part of our OrISODATA algorithm.

3.1. Graph clustering

The graph clustering algorithm employs the node pair sampling method [23] for building up the grains by contracting edges of a graph. The edge weights of the graph are initialized as:

$$W = \exp\left(-\theta^2/3\right), \tag{4}$$

where θ is the misorientation angle in degrees between two neighboring atoms. The algorithm contains two clustering parameters - *merge threshold* and *minimum number of atoms per grain*. The *merge threshold*, however, has *no intuitive physical meaning* [42]. The modifier in OVITO allows for an automatic mode, where the merge threshold is chosen automatically using a statistical analysis of a sequence of graph merging steps; merging of clusters is stopped as soon as a deviation from the regular exponential behavior is observed [42].

3.2. Minimum spanning tree

The second algorithm employs misorientation angles between neighboring atoms as the edge weights of a minimum spanning tree. Grains are built up by contracting edges in a sorted order based on misorientation, with the latter also acting as a measure of the merge distance. The approach is similar to that proposed by Panzarino and Rupert [18] but uses a hierarchical graph which reduces computation times. The algorithm is fast and has low memory usage. It is pointed out in the OVITO manual [42] that this method can lead to poor results in the presence of local perturbations due to, e.g., thermal noise.

Algorithm 4: Merge clusters.

Input:
1. Clusters
2. Φ_{merge}
3. Ψ_{split}
4. Positional connectivity flag ; `// flag = True, only during merge clusters call from Algorithm 1 line 7`

Output: Clusters after merge

1 **for** $i \leftarrow 1$ **to** *Number of clusters* **do**
2   Calculate disorientation angle (ϕ) between clusters with their meanorientations;
3 **end**
4 Find cluster pairs having $\phi <= \Phi_{\text{merge}}$ and sort these pairs based on their disorientation angle in ascendingorder;
5 **for** $i \leftarrow 1$ **to** *Unique cluster pairs* **do**
6   **if** *GOS (Ψ_{spread}) of the combined cluster aftermerge $<= \Psi_{split}$ **AND** Both clusters remainunmerged in the current merge clusters routine call* **then**
7   **if** *(Positional connectivity flag == True)* **then**
8   Check the positional connectivity of the cluster pair; `// To avoid re-mergingof clusters that were already positionally split`
9   **if** *Connected* **then**
10   Merge the cluster pair
11   **end**
12   **end**
13   **else**
14   Merge the cluster pair
15   **end**
16   **end**
17 **end**

---

4. Testing and validation

The OrISODATA algorithm is first validated on a bicrystal dataset containing a coherent twin boundary (CTB). For all validations, the maximum number of iterations and the convergence tolerance were set to 200 and 1e-5, respectively. All clustering results presented below are the final results obtained only after achieving convergence to the given tolerance value. Visualization of the data is performed using OVITO [21].

Fig. 2 a shows a Au bicrystal with a $\Sigma 3$ (111) CTB constructed using Atomsk [43]. The sample has approximate dimensions of 25 x 56 x 7 nm^3 and contains 600,000 atoms. Periodic boundary conditions are applied in all directions. The bicrystal sample is then relaxed using the conjugate gradient and FIRE [44] minimization algorithms as implemented in the atomistic simulation software LAMMPS [45,46]. The interatomic forces are modeled with an embedded atom method (EAM) potential for Au [47]. Fig. 2b,c show the orientations in the unrelaxed and relaxed samples, respectively.

The clustering parameters used for the validation of the OrISODATA algorithm, unless specified otherwise, are:

- Split threshold angle: $\Psi_{\text{split}} = 0.001°$
- Merge threshold angle: $\Phi_{\text{merge}} = 0.001°$
- Initial number of clusters: $n_{clus}^{init} = 1$
- Minimum number of atoms per cluster: $D_g^{\min} = 10$

The split and merge thresholds are set to very low numerical values since there exists almost no orientation spread in both the unrelaxed as well as the relaxed samples. The initial number of clusters $n_{clus}^{init} = 1$ is chosen so as to avoid any bias in the final result, since the expected number of clusters is known.

Two further synthetic samples, with an artificial orientation spread in each individual grain, are created to further validate the clustering parameters. The first of the synthetic samples (Fig. 2d) has an orientation spread of $\approx 0.5°$ introduced linearly from the GB to the boundary of the simulation box. The second synthetic sample has a central cylindrical region in each individual grain (Fig. 2e), with a disorientation angle of approximately 9.5° to the surrounding matrix.

The results from our OrISODATA clustering are shown in Fig. 2f–j. For the unrelaxed and relaxed cases, the algorithm correctly identifies two clusters delimited by the CTB. For the case with a linear orientation spread, the algorithm identifies four clusters. Increasing the split threshold (Ψ_{split}) beyond $\approx 0.125°$ results in the two clusters in the top and bottom grain merging together to form a single individual grain. For the sample with a cylindrical inclusion, we obtain as expected, four clusters with the inclusions identified as separate clusters.

5. Application on a polycrystalline sample

We demonstrate the working of our OrISODATA algorithm on a nanocrystalline thin film sample. The initial structure is generated by means of a constrained Voronoi tessellation [48,49] so as to reduce non-equilibrium junctions [50]. The initial dimensions of the thin film are 180 x 120 x 15 nm^3 and contains 122 grains with a mean grain diameter of 15 nm. Each grain is assigned an orientation with < 111 > along the film thickness and a random rotation in the plane of the thin film, resulting in purely tilt grain boundaries in the structure. The atomistic structure generated using the open-source toolbox *nano*SCULPT [51] contains approximately 19 Mio atoms (see Fig. 3a). Periodic boundary conditions (PBC) are applied in the plane of the thin film; free boundaries exist along the thickness of the film.

The atomistic sample is then relaxed using the FIRE algorithm in standard molecular statics simulations, and subsequently equilibrated and thermalized at 300 K for 40 ps. Fig. 3b,c shows the defect structure – computed using common neighbor analysis – and the orientations of individual grains in the thin film. The equilibrated pressure-free structures are then subjected to uniaxial tension at a constant strain-rate of $10^8 \ s^{-1}$. Simulations are performed with the ITAP molecular dynamics (IMD) code [52]. The interatomic forces are modeled with an EAM potential for Au [53].

5.1. Au undeformed polycrystalline thin film sample at 300 K

The results of clustering obtained via the OrISODATA algorithm on the undeformed (relaxed + thermalized) sample are shown in Fig. 3d–h. The influence of the split and merge threshold values on the final clustering is clearly visible from the results. As the threshold values decrease, the number of clusters increases. With parameter set #1, the OrISODATA algorithm is able to identify clusters which conform approximately to the orientation map in Fig. 3c. For $\Phi_{\text{merge}} \leq 1°$ (see Fig. 3e–g), small clusters emerge at GBs and triple junctions (TJ), where the local orientation of individual atoms differs from the mean orientation of the grains.

Increasing the threshold for the minimum number of atoms per cluster to an equivalent grain size of 7 nm ensures that such small clusters are now treated initially as orphan clusters and assigned to the closest cluster after the final iteration. Nonetheless, grains separated by very low angle GBs (< 1°) can still be identified, as shown exemplarily by the highlighted grains in Fig. 3h. Using threshold values beyond those defined by the orientations of such grains results in the coalescence of such grain pairs to a single cluster, see highlighted boxes in Fig. 3h.

Since we do not expect the orientation of individual grains to change as a result of relaxation and equilibration of the sample, atoms forming the original grain must be assigned to a new distinct cluster, as long as the threshold criteria are satisfied. This is evident in Fig. 4b which shows the fraction of atoms of each grain assigned to different clusters, thus providing a measure of effectiveness of the algorithm. Such a plot
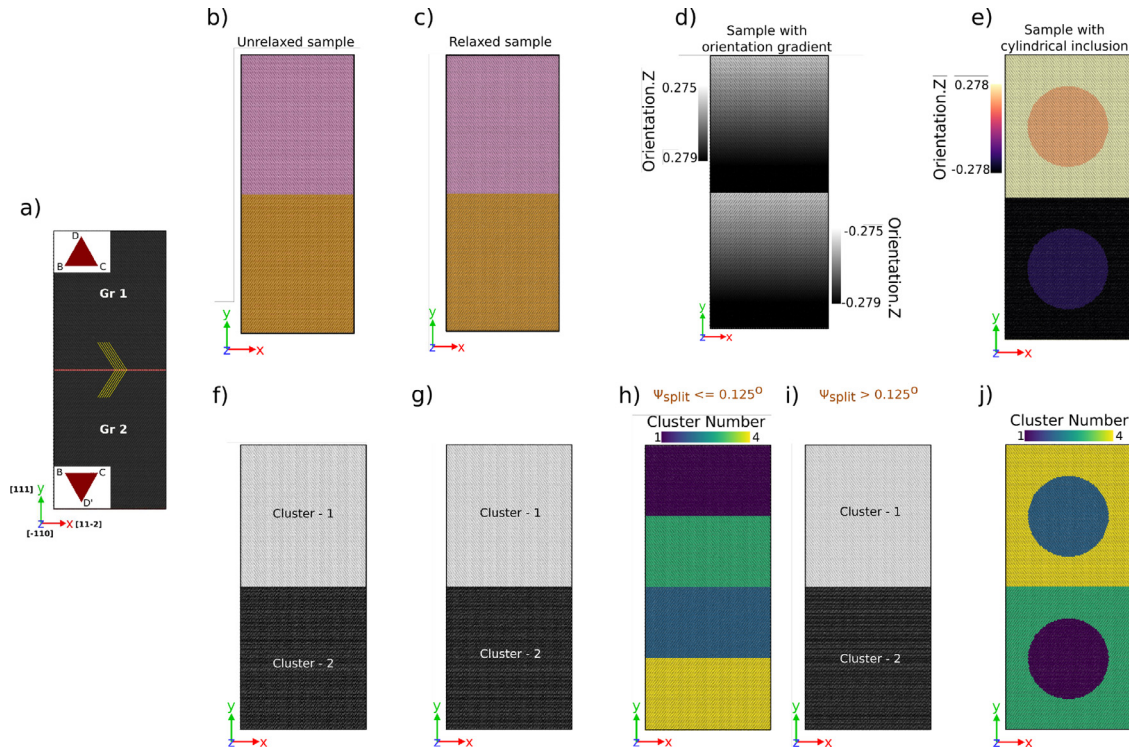
Fig. 2. Results of validation and testing of our ORISODATA algorithm. (a) Au bicrystal sample with a coherent twin boundary and colored based on common neighbor analysis (black: fcc, red: hcp). The orientation of the two grains is shown using the Thompson tetrahedron. The yellow solid lines indicate coherent atomic planes in the two grains. (b,c) Unrelaxed and relaxed samples colored based on orientation of individual atoms. (d) Synthetic bicrystal sample with an artificial orientation spread in each grain. (e) Synthetic bicrystal sample with a cylindrical inclusion with a misorientation of $\approx 9.5°$ from the matrix. (f–j) Clustering results with atoms colored according to their corresponding cluster numbers.

is particularly helpful to track, e.g., whether the initial grains are split or merged with the neighboring grains whilst being assigned to a cluster. The diagonal trend indicates that by-and-large each grain is assigned uniquely to a new cluster. Multiple points along a single *horizontal* line indicate the coalescence of multiple grains into a single cluster. This is, for instance, the case with grains 53, 98, 121 which are now assigned to the cluster 53. Such merging of two or more grains is a result of very low misorientation between the grains, which is less than the threshold of 2°; the GBs between such grains contain no defect atoms as seen in Fig. 3c. On the other hand multiple points along a single *vertical* line indicate that atoms in the original grain are split into multiple clusters as a consequence of local rearrangement close to GBs. Only three grains evidence a split where more than 10% of the original grain is assigned to a neighboring cluster.

The results of clustering with the grain segmentation algorithms in OVITO are shown in Fig. 4c–f. The automatic graph clustering (AGC) algorithm results in fewer clusters (64) than with our ORISODATA algorithm (cf. Fig. 4c). In particular, grains separated by low angle GBs are organized to a single cluster with as many as five grains sometimes grouped into a single cluster. Hence a significant deviation from a purely diagonal trend is seen in the correlation plot shown in Fig. 4d. The threshold value determined by the AGC algorithm is 22 units, which results in a misorientation angle of approximately $\approx 3°$ using Eq. (4). Increasing the threshold value in ORISODATA to 3° leads, expectedly, to fewer number of 98 clusters, but is still much higher than that obtained with the AGC algorithm (see *supplementary* Fig. S5).

By contrast, with the minimum spanning tree (MST) algorithm Fig. 4e we obtain an almost equal number of clusters as with our ORISODATA algorithm – 106 wth MST vs. 105 with ORISODATA. The clustering pattern obtained from the MST algorithm is very similar to that observed from our ORISODATA algorithm. The low merge threshold of 0.05° is seemingly the reason behind the ability to recognize low angle

GBs that are apparently neglected with the AGC method. This threshold value is, however, not directly comparable with the merge threshold in the ORISODATA algorithm since the latter is a measure of distance between the mean orientations of two grains, whereas the former defines a threshold for the misorientation between neighboring atoms.

Despite these noticeable differences in the clustering results from the three algorithms, no large differences can be seen in the mean orientations of the identified clusters. The pole figures shown in *supplementary* Fig. S1 are very similar and show the presence of a {111} texture in the thin film.

5.2. Au deformed polycrystalline thin film sample at 300 K

The defect structure in the thin film sample, after 10% tensile strain in the global y-direction, obtained via CNA is shown in Fig. 5a. The deformed configuration is dominated primarily by stacking faults and deformation twins, which form via the motion of partial dislocations. Very few full dislocations are visible. The orientation map (see Fig. 5b) shows a larger spread of orientations in many grains, than that seen in the undeformed configuration. Twinned regions are visible as regions with a different orientation in comparison to the orientation of the parent grain. It must be expected that a good grain segmentation algorithm is able to identify such regions and help track the evolution of, e.g. twin volume during deformation.

Comparison of clustering results from ORISODATA with those from the grain segmentation algorithms in OVITO throws up some interesting observations, see Fig. 6. The number of clusters identified by ORISODATA is significantly higher than the other two algorithms. A purely diagonal trend in the correlation between the undeformed cluster/grain and the identified cluster (Fig. 6b,d,f) is no longer visible in any of the algorithms. This observation can be ascribed to the larger spread of orientation gradients in many grains, as a result of which, atoms belonging
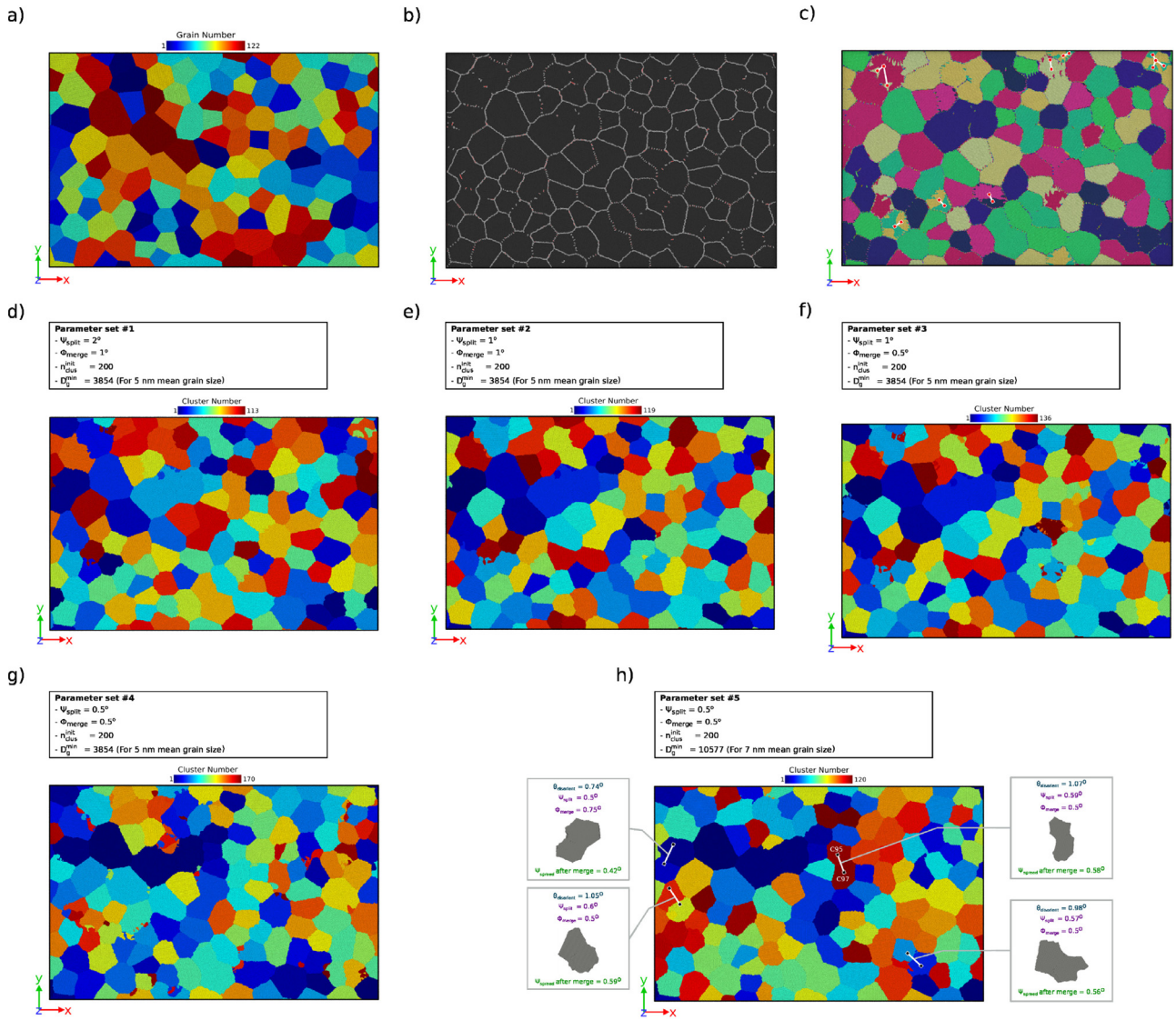
Fig. 3. Au polycrystalline thin film undeformed sample at 300 K (sliced in the thickness for better visualization). (a) Thin film sample with 122 grains. Atoms are colored according to their initial grain numbers. (b) CNA results from OVITO (Color code for atoms: black - fcc, red - hcp, grey - others). (c) Orientations of individual grains in the thin film sample (in terms of quaternions) of individual atoms are converted and represented in RGB values. The white lines with red marker connect regions with symmetrically equivalent orientations. (d–h) Clustering results with five different parameter sets. Atoms are colored according to their cluster numbers. Higher threshold values result in coalescence of certain grain pairs as shown in (h).

to individual undeformed grains/clusters are split into multiple clusters in the deformed state. Furthermore, many clusters have now formed via the conglomeration of multiple initial grains, indicating significant changes in GB characteristics, topology and network. Of the two algorithms in OVITO, the AGC method performs better than the minimum spanning tree approach in identifying local twinned regions. We note that the option *Handle coherent interfaces/stacking faults* is switched on during clustering with the grain clustering algorithms in OVITO. Turning this option off results in improved identification of the twinned regions with the AGC method (see *supplementary* Fig. S2). No change in the identification of twins is seen with the MST algorithm, but switching off the option results in substantially more clusters (see *supplementary* Fig. S2).

The mean orientations of individual clusters confirm the aforementioned observations. With the MST algorithm, almost no change is observed in the overall texture (cf. *supplementary* Fig. S3). By comparison, a strong deviation from the strong initial {111} texture is observed in a few clusters (mostly corresponding to deformation twins) with both OrISODATA and the AGC algorithms. The few points which deviate from the initial {111} texture with the MST algorithm correspond to a single twin identified by the algorithm.

6. Discussion

The primary difference between the grain segmentation algorithms in OVITO and our OrISODATA algorithm is in the approach taken. With the graph based methods implemented in OVITO, a more-or-less bottom-up approach is forged, wherein clustering begins by the creation of a graph representation of the input structure using atoms as the graph nodes and the bonds with neighbors as the graph edges. If the threshold criterion is satisfied, graph edges are collapsed resulting in a hierarchical clustering of individual grains. By contrast, in the OrISODATA algorithm we follow a top-down approach. Clustering begins directly in the data space without reference to individual atoms. Clusters are then split and merged if certain criteria are met. The reference to individual atoms appears only now: If a cluster has multiple distinct regions which are not connected to each other in terms of immediate neighbors, we split such clusters further.
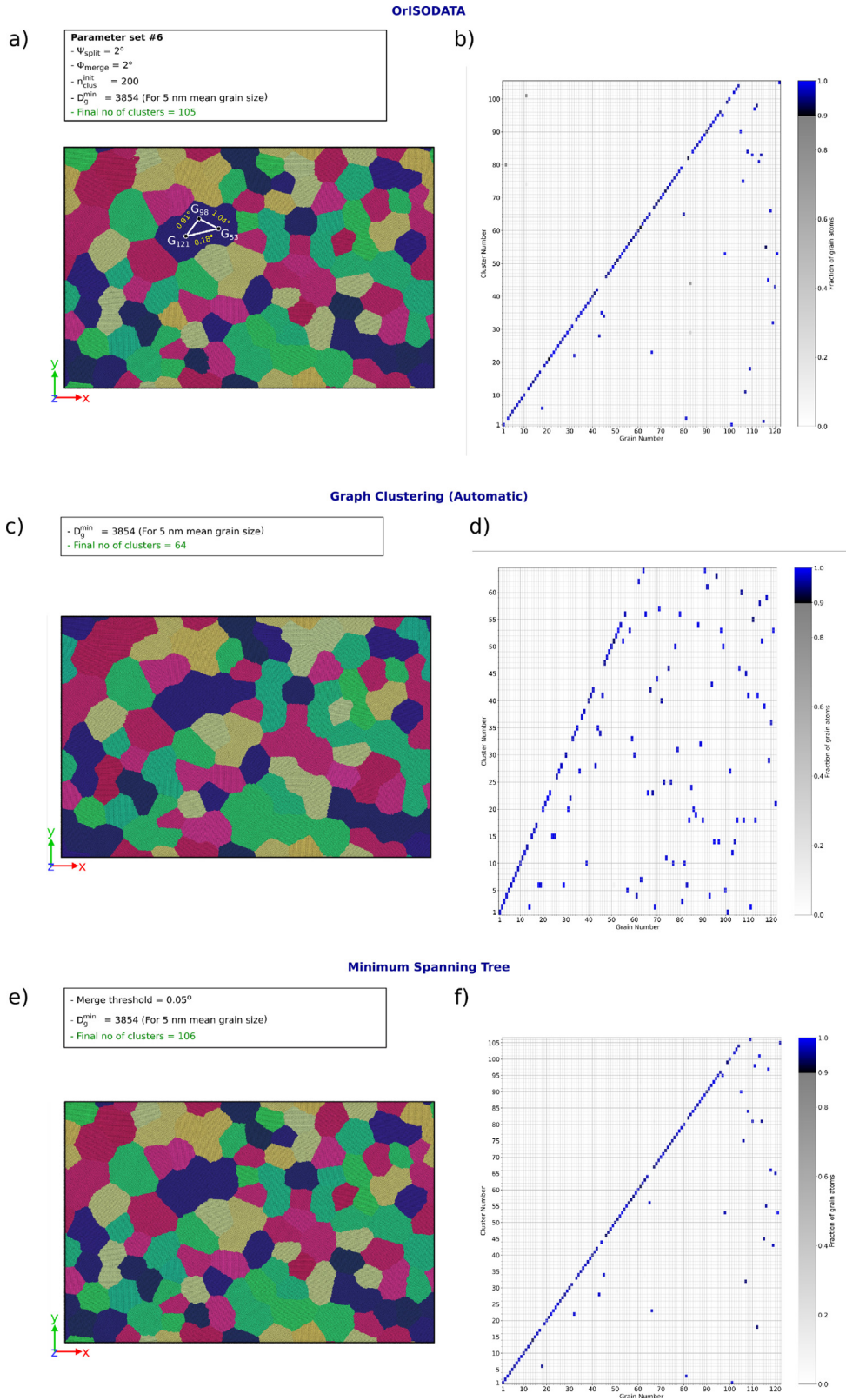
OrISODATA

a)

Parameter set #6
- $\Psi_{split} = 2°$
- $\Phi_{merge} = 2°$
- $n_{clus}^{init} = 200$
- $D_g^{min} = 3854$ (For 5 nm mean grain size)
- Final no of clusters = 105

b)

Graph Clustering (Automatic)

c)

- $D_g^{min} = 3854$ (For 5 nm mean grain size)
- Final no of clusters = 64

d)

Minimum Spanning Tree

e)

- Merge threshold = 0.05°
- $D_g^{min} = 3854$ (For 5 nm mean grain size)
- Final no of clusters = 106

f)

Fig. 4. Comparison of clustering results from OrISODATA algorithm with those from grain segmentation algorithms in OVITO. *left*: Individual clusters colored by their mean orientations by mapping the corresponding quaternion (q_x, q_y, q_z, q_w) to red, green and blue values, respectively. *right*: Correlation plot between initial grain number and the final cluster identified by the corresponding algorithm.

Despite the different approaches taken, the working of the algorithms can be understood by comparing the characteristics of the output clusters and evaluating the influence of the corresponding parameters on the clustering. One aspect of the clustering process is the treatment of orphan atoms. Figs. 7 and 8 show the clustered configurations of the undeformed and deformed samples, respectively, from the three algorithms, before the adoption of orphan atoms. It is evident that for the parameter sets chosen, the fewest number of orphan atoms is with the AGC method and the highest with the MST algorithm. Approximately 35% of the undeformed sample and 65% of the deformed sample is iden-
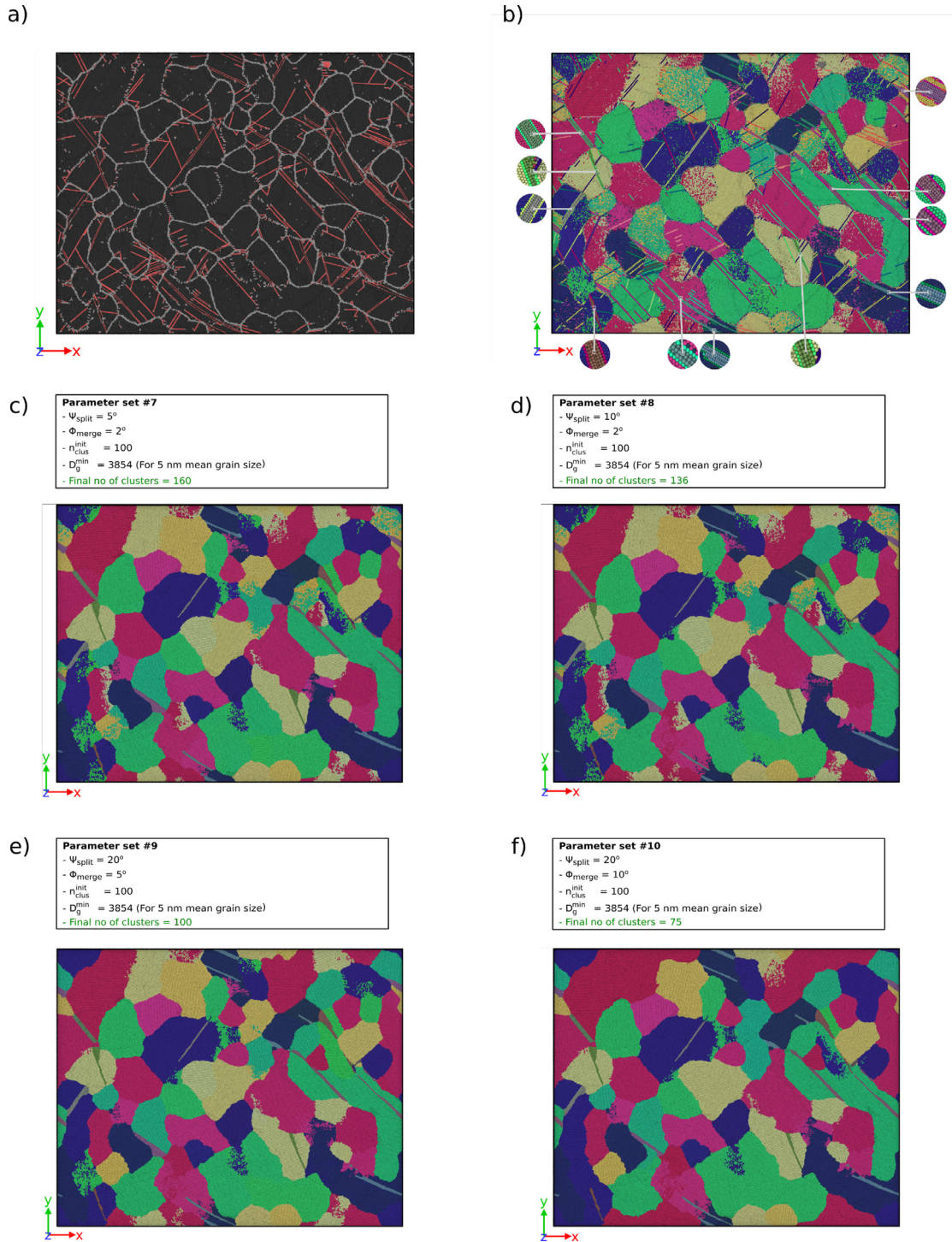
a)

b)

c)

Parameter set #7
- $\Psi_{split} = 5°$
- $\Phi_{merge} = 2°$
- $n_{clus}^{init} = 100$
- $D_g^{min} = 3854$ (For 5 nm mean grain size)
- Final no of clusters = 160

d)

Parameter set #8
- $\Psi_{split} = 10°$
- $\Phi_{merge} = 2°$
- $n_{clus}^{init} = 100$
- $D_g^{min} = 3854$ (For 5 nm mean grain size)
- Final no of clusters = 136

e)

Parameter set #9
- $\Psi_{split} = 20°$
- $\Phi_{merge} = 5°$
- $n_{clus}^{init} = 100$
- $D_g^{min} = 3854$ (For 5 nm mean grain size)
- Final no of clusters = 100

f)

Parameter set #10
- $\Psi_{split} = 20°$
- $\Phi_{merge} = 10°$
- $n_{clus}^{init} = 100$
- $D_g^{min} = 3854$ (For 5 nm mean grain size)
- Final no of clusters = 75

Fig. 5. Au deformed polycrystalline thin film sample (Sample has been sliced in the Z direction for better visualization). (a) CNA results from OVITO (Color code for atoms: black - fcc, red - hcp, grey - others). (b) Orientations (in terms of quaternions) of individual atoms are converted and represented in RGB values. Some of the stacking fault regions present within the grains are highlighted. (c–f) Clustering results with four different parameter sets. Atoms are colored according to their mean orientations.

tified as orphan atoms with the MST algorithm (see Figs. 7f and 8f). The number of orphan atoms with the ORISODATA algorithm is close to that of the AGC method; the slight differences seen in the distributions are ascribed to the chosen parameter set which results in many more clusters being identified with our ORISODATA algorithm in comparison to the AGC method.

This increased fraction of orphan atoms is directly responsible for the poor identification of deformation twins with the MST algorithm. With all the three algorithms, stacking faults are identified as orphan atoms due to their hcp crystal structure. However, with MST, regions around hcp atoms are also identified as orphan atoms. These orphan atoms are later assigned to a parent cluster, resulting in twinned regions appearing much larger or being completely assimilated into the parent grain.

The threshold parameter used in the MST algorithm evidences a high-sensitivity to small perturbations. Choosing the right threshold parameter in the MST algorithm can hence be quite tricky, as seen in *sup-*
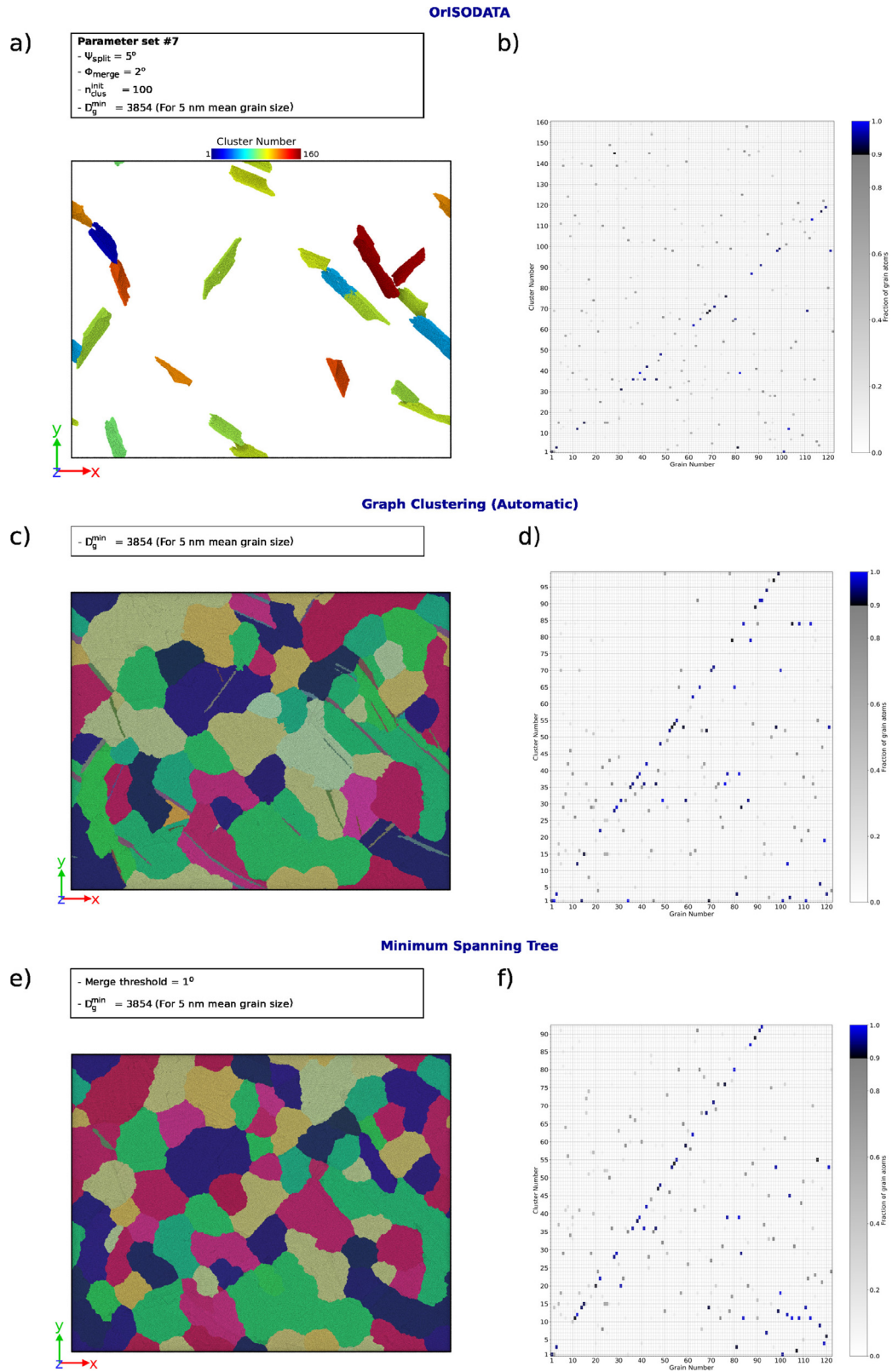
OrISODATA

a)

Parameter set #7
- $\Psi_{split} = 5°$
- $\Phi_{merge} = 2°$
- $n_{clus}^{init} = 100$
- $D_g^{min} = 3854$ (For 5 nm mean grain size)

b)

Cluster Number

Graph Clustering (Automatic)

c)

- $D_g^{min} = 3854$ (For 5 nm mean grain size)

d)

Minimum Spanning Tree

e)

- Merge threshold = 1°
- $D_g^{min} = 3854$ (For 5 nm mean grain size)

f)

Fig. 6. Comparison of clustering results from OrISODATA algorithm with those from grain segmentation algorithms in Ovito on the Au polycrystalline sample deformed to 10% tensile strain. *left*: Individual clusters colored by their corresponding mean orientation. *right*: Correlation plot between initial grain number and the final cluster identified by the corresponding algorithm.
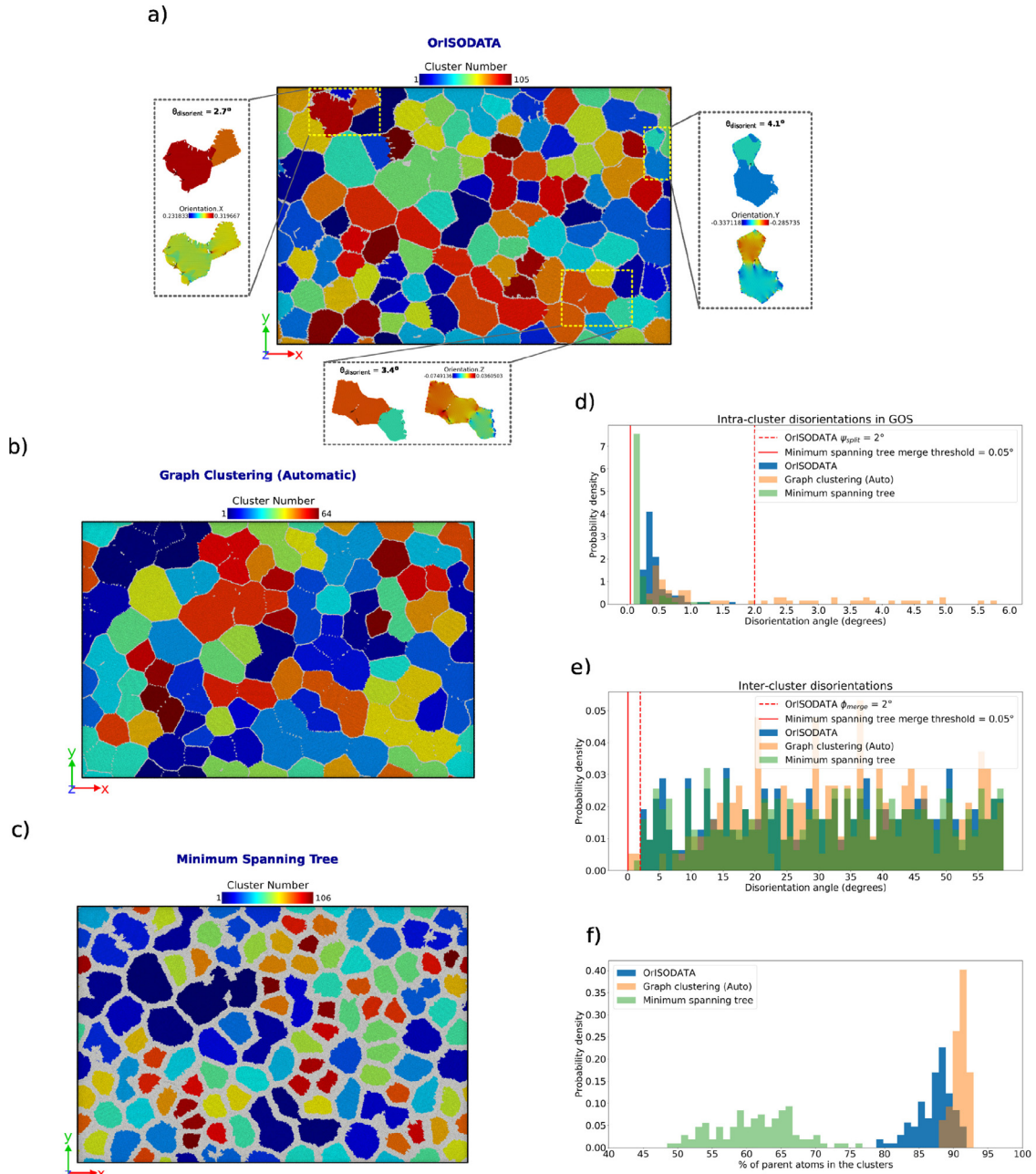
Fig. 7. Comparison of clustering results on the undeformed sample of OrISODATA algorithm with the grain clustering algorithms in OVITO. The samples are sliced in the thickness for better visualization. (a) Clustering results from OrISODATA for the parameter set #6. Disorientation angles and the color map showing the quaternion component having maximum variation between the sub-clusters formed within grains, are highlighted separately. (b) Clustering results from graph clustering (automatic). (c) Clustering results from minimum spanning tree algorithm. (d) Intra cluster disorientation. (e) Disorientation between neighboring clusters. (f) Distribution of percentage of parent atoms (non-orphan atoms) in individual clusters. The threshold values used for the corresponding algorithms are marked in vertical lines. Color code in (a–c) - Atoms are colored according to their cluster numbers except the orphan atoms which are colored in light gray.

plementary Fig. S4. For the deformed configuration, the chosen value of 1° results in the best clustering pattern that is comparable with the other algorithms, as shown in Fig. 6. Decreasing the threshold to 0.9° results in almost equal number of clusters, but significantly different clustering pattern. Increasing the threshold to 1.1° results in a higher number of clusters, see *supplementary* Fig. S4. This behavior is non-intuitive, since we expect the number of clusters to either increase or remain constant with a decrease in the threshold value, and can again be attributed to the presence of large number of orphan atoms with the MST algorithm. Decreasing the minimum number of atoms per cluster to 1000 reduces the number of orphan atoms and results in the expected behavior of in-

creased number of clusters with a lower threshold. The predicted clusters, however, are quite different to those observed with the AGC and OrISODATA algorithms.

A quantitative comparison of the three algorithms can be made by looking into the intra-cluster disorientation, and the inter-cluster disorientation between immediate neighbors of a cluster. The results of such a comparison is shown in Figs. 7 and 8 for the undeformed and deformed samples, respectively. With MST and OrISODATA algorithms where criteria can be specified, such threshold criteria are well satisfied. The intra cluster disorientation in terms of GOS is higher than the threshold of 0.05∘ between atoms in the case of the MST algorithm in
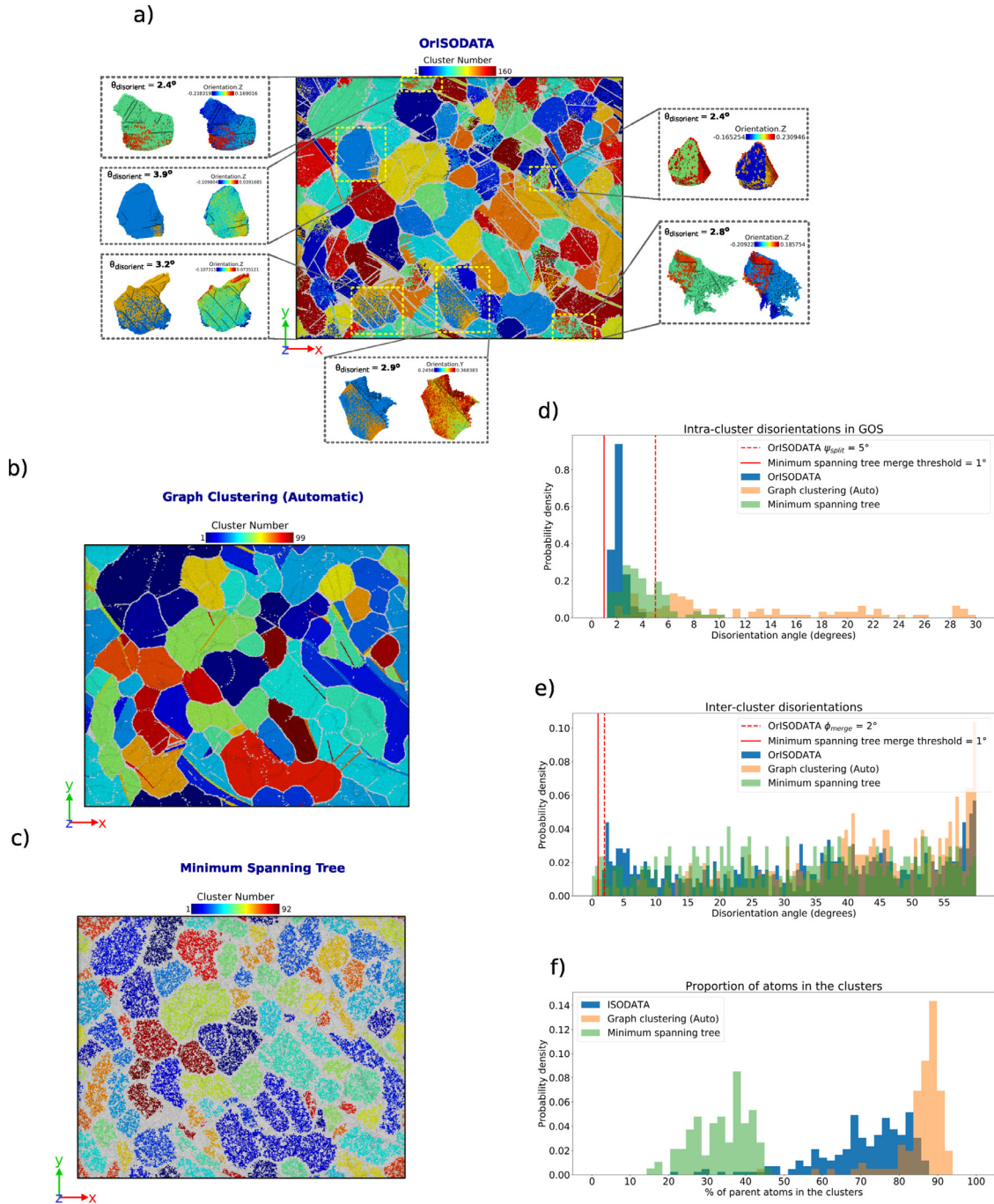
Fig. 8. Comparison of clustering characteristics from OrISODATA, graph clustering and minimum spanning tree algorithms in the strained sample. The samples are sliced in the thickness for better visualization. (a–c) Clusters determined by the three different algorithms before the adoption of orphan atoms. Cluster numbers for the parameter set #7 is shown in (a). (d) Intra cluster disorientation. (e) Disorientation between neighboring clusters. (f) Distribution of percentage of parent atoms (non-orphan atoms) in individual clusters. The threshold values used for the corresponding algorithms are marked in vertical lines. Color code in (a–c) - Atoms are colored according to their cluster numbers except the unadopted atoms which are colored in light gray.

the undeformed configuration. This seemingly translates to a maximum GOS of $1.5°$ and an average value of $0.1°$ in the formed clusters. With the AGC algorithm, the maximum GOS observed was close to $6°$.

In the deformed configuration, a threshold of $1°$ with the MST algorithm results in a maximum intra-cluster disorientation of roughly $10°$. In comparison, the maximum GOS is $\approx 30°$ with the AGC model, which is, however, an artifact of hcp atoms identified as part of the parent cluster. The inter-cluster disorientations evidence a distribution over a range of angles, similar to that seen in the undeformed config-

uration; both OrISODATA and AGC algorithms result in higher number of clusters with large inter-cluster disorientations due to their ability to identify twinned regions. The OrISODATA algorithm is, furthermore, clearly able to identify clusters with low angle inter-cluster disorientations, whereas very few of such clusters were identified with the AGC model (see Fig. 8e).

Using the manual version of the grain clustering algorithm can provide the user with increased flexibility in identifying clusters. Decreasing the threshold value results, as expected, in increased number of clusters

(see *supplementary* Fig. S6). However, the non-physical nature of the threshold used to collapse graph nodes essentially hinders robust interpretation and verification of results. With the undeformed configuration, a threshold of 22 units was used to generate the clustering results shown in Fig. 4c, which translates to a disorientation angle of approximately 3°, suggesting a minimum inter-cluster disorientation of the same measure. By fine tuning the parameters in the OrISODATA algorithm, we were able to drastically reduce the number of clusters to match that of the graph clustering algorithm (see *supplementary* Fig. S7). A similar clustering pattern was only obtained for split and merge thresholds of 10° with our OrISODATA algorithm. Furthermore, with the graph clustering algorithm, whilst clusters with intra-cluster disorientations of as high as ≈ 4° are identified as a single cluster, clusters with lower intra-cluster disorientation of ≈ 3.2 are split into two clusters whose inter-cluster disorientation is a mere 1.2°. This essentially leads us to conclude that the threshold parameter in the graph clustering algorithm, and consequently, the clustering results, lack simple interpretation.

The clustering results with our OrISODATA algorithm depend strongly on the two threshold parameters – higher numerical values of these thresholds result in fewer number of final clusters, and vice versa. The choice of numerical values for the thresholds depends, however, on the sample and application at hand. For example, to identify low angle GBs, the *merge threshold* must be set to a low value that corresponds to the disorientation between two neighboring grains. On the other hand, the *split threshold* influences the position of a GB in a contiguous domain with significant orientation gradients. In other words, the intra-cluster disorientation must be less than the split threshold and the inter-cluster disorientation between nearest neighbors must be greater than the merge threshold. In both the undeformed and deformed configurations, the results clearly follow this rule, see Fig. 7d,e and 8d,e.

Two further parameters form a part of the OrISODATA algorithm: (a) Minimum number of atoms or minimum grain size ($D_g^{\min}$) for any cluster to be accepted as a possible solution, (b) Initial number of clusters (n_{clus}^{init}). The former has an effect on the fraction of orphan atoms, and eliminates the formation of very small clusters (see Fig. 3g,h). Nevertheless, for the same threshold parameters, the influence of the minimum grain size on the clustering results is negligible. The influence of the latter parameter, n_{clus}^{init}, depends on the orientation gradients in the sample – larger the gradients in orientation, larger is the influence on the final number of clusters, as is the case with the deformed thin film sample (see *supplementary* Fig. S9). Nevertheless, this variation is less than ±5% indicating the robustness of the entire procedure.

The advantage of the OrISODATA algorithm over the other two algorithms is in the intuitive nature of the threshold parameters which is expressed as follows:

- *Split threshold*: How much of a deviation from the mean orientation is to be allowed within the domain of the grain
- *Merge threshold*: How close in terms of orientation should two neighboring grains/clusters be in order for them to be treated as a single grain

This intuitiveness of the parameters ensures easy verification and interpretation of results. Furthermore, these threshold parameters are similar in interpretation to those used in experiments (e.g. EBSD) making the methodology appealing to a larger user base and facilitating increased synergy between simulations and experiments.

7. Conclusions

In this work, we propose an orientation-based iterative self organizing data analysis (OrISODATA) algorithm to identify grains in atomistic simulations of polycrystalline materials. The algorithm takes a data based approach to cluster points in the orientation space. We first validate the algorithm on a bicrystal sample containing a coherent twin boundary. Subsequently, the working of the OrISODATA is demonstrated on a polycrystalline thin film sample in the undeformed state as well as

the deformed state after 10% tensile strain at 300 K. Furthermore, the results from our OrISODATA algorithm are compared with those from the automatic graph clustering and minimum spanning tree algorithms implemented in OVITO. With the latter, we are able to identify very low angle GBs which was not possible with the AGC algorithm. On the other hand, deformation twins are not identified by the MST algorithm, but were clearly identified by the AGC algorithm.

The findings of the current study can be summarized as follows:

- The OrISODATA algorithm uses parameters which are intuitive and physically relevant for the application. This makes it substantially easy to choose optimal parameters, and to validate and interpret the clustering results.
- Very low angle grain boundaries, i.e. inter-cluster disorientation less than 2° can be identified with both the OrISODATA and the MST algorithms, but not with the AGC algorithm.
- Twinned regions in the deformed polycrystalline thin film are identified with the OrISODATA and AGC algorithms, but not with the MST algorithm.
- The proportion of atoms identified as *orphan atoms* is a significant influencing factor on the clustering results of the three algorithms. Perturbations due to the applied strain and local elastic fields can result in a significant fraction of atoms – 60~70% fraction – in the deformed thin film being classified as orphan atoms with the MST algorithm.
- A smaller fraction of orphan atoms is the key to proper identification of twinned regions in the deformed state of the polycrystal. Thin twinned regions are otherwise classified as orphan atoms and are later assigned to the nearest cluster.
- Choosing the right parameter set for the usage of the AGC algorithm is particularly difficult since the threshold parameter has no intuitive meaning. Furthermore, intra-cluster grain orientation spread and inter-cluster disorientation distributions are also not along expected lines.

The two threshold parameters in the OrISODATA algorithm are intuitive and relate to similar thresholds used in experiments: the *split threshold* is a measure of the maximum orientation spread within a grain, whereas the *merge threshold* is a measure of proximity between the mean orientation of two grains. We conclude that although the choice of the parameter values strongly influences the clustering results, as is the case with any algorithm, having model parameters that are intuitive and physically relevant to the application, is substantially helpful for choosing optimal parameter values and also for analyzing and rightly interpreting the final results.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

Supplementary material

Supplementary material associated with this article can be found, in the online version, at 10.1016/j.mtla.2022.101314.

References

[1] E.O. Hall, The deformation and ageing of mild steel: III discussion of results, Proc. Phys. Soc. Lond. Sect. B 64 (9) (1951) 747.

[2] K.S. Kumar, H. Van Swygenhoven, S. Suresh, Mechanical behavior of nanocrystalline metals and alloys, Acta Mater. 51 (19) (2003) 5743–5774.

[3] D. Wolf, V. Yamakov, S.R. Phillpot, A. Mukherjee, H. Gleiter, Deformation of nanocrystalline materials by molecular-dynamics simulation: relationship to experiments? Acta Mater. 53 (1) (2005) 1–40.

[4] M.A. Meyers, A. Mishra, D.J. Benson, Mechanical properties of nanocrystalline materials, Prog. Mater. Sci. 51 (4) (2006) 427–556.

[5] H. Van Swygenhoven, J.R. Weertman, Deformation in nanocrystalline metals, Mater. Today 9 (5) (2006) 24–31.

[6] D. Farkas, Atomistic simulations of metallic microstructures, Curr. Opin. Solid State Mater. Sci. 17 (6) (2013) 284–297.

[7] E.N. Hahn, M.A. Meyers, Grain-size dependent mechanical behavior of nanocrystalline metals, Mater. Sci. Eng. A 646 (2015) 101–134.

[8] J. Schiøtz, F.D. Di Tolla, K.W. Jacobsen, Softening of nanocrystalline metals at very small grain sizes, Nature 391 (6667) (1998) 561–563.

[9] H. Van Swygenhoven, P.M. Derlet, Grain-boundary sliding in nanocrystalline fcc metals, Phys. Rev. B 64 (22) (2001) 224105.

[10] J. Schiøtz, Atomic-scale modeling of plastic deformation of nanocrystalline copper, Scr. Mater. 51 (8) (2004) 837–841.

[11] H. Van Swygenhoven, P.M. Derlet, A.G. Frøseth, Nucleation and propagation of dislocations in nanocrystalline fcc metals, Acta Mater. 54 (7) (2006) 1975–1983.

[12] J.F. Panzarino, Z. Pan, T.J. Rupert, Plasticity-induced restructuring of a nanocrystalline grain boundary network, Acta Mater. 120 (2016) 1–13.

[13] A. Prakash, D. Weygand, E. Bitzek, Influence of grain boundary structure and topology on the plastic deformation of nanocrystalline aluminum as studied by atomistic simulations, Int. J. Plast. 97 (2017) 107–125.

[14] J.D. Honeycutt, H.C. Andersen, Molecular dynamics study of melting and freezing of small Lennard-Jones clusters, J. Phys. Chem. 91 (19) (1987) 4950–4963.

[15] D. Faken, H. Jónsson, Systematic analysis of local atomic structure combined with 3D computer graphics, Comput. Mater. Sci 2 (2) (1994) 279–286.

[16] C.L. Kelchner, S.J. Plimpton, J.C. Hamilton, Dislocation nucleation and defect structure during surface indentation, Phys. Rev. B 58 (17) (1998) 11085.

[17] G.J. Tucker, S.M. Foiles, Molecular dynamics simulations of rate-dependent grain growth during the surface indentation of nanocrystalline nickel, Mater. Sci. Eng. A 571 (2013) 207–214.

[18] J.F. Panzarino, T.J. Rupert, Tracking microstructure of crystalline materials: a post-processing algorithm for atomistic simulations, JOM 66 (3) (2014) 417–428.

[19] J.F. Panzarino, J.J. Ramos, T.J. Rupert, Quantitative tracking of grain structure evolution in a nanocrystalline metal during cyclic loading, Modell. Simul. Mater. Sci. Eng. 23 (2) (2015) 025005.

[20] P.W. Hoffrogge, L.A. Barrales-Mora, Grain-resolved kinetics and rotation during grain growth of nanocrystalline aluminium by molecular dynamics, Comput. Mater. Sci 128 (2017) 207–222.

[21] A. Stukowski, Visualization and analysis of atomistic simulation data with OVITO–the open visualization tool, Modell. Simul. Mater. Sci. Eng. 18 (1) (2009) 015012.

[22] P.M. Larsen, S. Schmidt, J. Schiøtz, Robust structural identification via polyhedral template matching, Modell. Simul. Mater. Sci. Eng. 24 (5) (2016) 055007.

[23] T. Bonald, B. Charpentier, A. Galland, A. Hollocou, Hierarchical graph clustering using node pair sampling, arXiv preprint arXiv:1806.01664

[24] S.K. Popat, M. Emmanuel, Review and comparative study of clustering techniques, Int. J. Comput. Sci. Inf. Technol. 5 (1) (2014) 805–812.

[25] D. Xu, Y. Tian, A comprehensive survey of clustering algorithms, Ann. Data Sci. 2 (2) (2015) 165–193.

[26] G.H. Ball, D.J. Hall, ISODATA, a novel method of data analysis and pattern classification, Technical Report, Stanford research inst Menlo Park CA, 1965.

[27] S.A. El Rahman, Hyperspectral imaging classification using ISODATA algorithm: Big data challenge, in: Proceedings of the Fifth International Conference on e-Learning (econf), IEEE, 2015, pp. 247–250.

[28] A.K. Jain, R.C. Dubes, Algorithms for Clustering Data, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988. URL http://portal.acm.org/citation.cfm?id=46712

[29] I. Manakos, T. Schneider, U. Ammer, A comparison between the ISODATA and the eCognition classification methods on basis of field data, IAPRS 33 (Suppl. B7) (2000) 133–139.

[30] A.W. Abbas, N. Minallh, N. Ahmad, S.A.R. Abid, M.A.A. Khan, K-Means and ISODATA clustering algorithms for landcover classification using remote sensing, Sindh Univ. Res. J. SURJ (Sci. Ser.) 48 (2) (2016).

[31] P. Berkhin, A survey of clustering data mining techniques, in: Grouping Multidimensional Data, Springer, 2006, pp. 25–71.

[32] M.Z. Rodriguez, C.H. Comin, D. Casanova, O.M. Bruno, D.R. Amancio, L.d.F. Costa, F.A. Rodrigues, Clustering algorithms: a comparative approach, PLoS ONE 14 (1) (2019) e0210236.

[33] R. Hartley, J. Trumpf, Y. Dai, H. Li, Rotation averaging, Int. J. Comput. Vis. 103 (3) (2013) 267–305.

[34] D.Q. Huynh, Metrics for 3D rotations: comparison and analysis, J. Math. Imaging Vis. 35 (2) (2009) 155–164.

[35] R. Hartley, K. Aftab, J. Trumpf, L1 rotation averaging using the Weiszfeld algorithm, in: Proceedings of the CVPR 2011, IEEE, 2011, pp. 3041–3048.

[36] M. Moakher, Means and averaging in the group of rotations, SIAM J. Matrix Anal. Appl. 24 (1) (2002) 1–16.

[37] D.J. Pearce, An improved algorithm for finding the strongly connected components of a directed graph, Technical Report, Victoria University, Wellington, NZ, 2005.

[38] D. Jorge-Badiola, A. Iza-Mendia, I. Gutierrez, Study by EBSD of the development of the substructure in a hot deformed 304 stainless steel, Mater. Sci. Eng. A 394 (1–2) (2005) 445–454.

[39] S. Mitsche, P. Pölt, C. Sommitsch, Recrystallization behaviour of the nickel-based alloy 80 a during hot forming, J. Microsc. 227 (3) (2007) 267–274.

[40] A. Ayad, N. Allain-Bonasso, N. Rouag, F. Wagner, Grain orientation spread values in IF steels after plastic deformation and recrystallization, in: Materials Science Forum, volume 702, Trans Tech Publ, 2012, pp. 269–272.

[41] N. Allain-Bonasso, F. Wagner, S. Berbenni, D.P. Field, A study of the heterogeneity of plastic deformation in IF steel by EBSD, Mater. Sci. Eng. A 548 (2012) 56–63.

[42] Grain segmentation modifier, OVITO 3.3.5 manual, (https://www.ovito.org/docs/current/reference/pipelines/modifiers/grain_segmentation.html), Accessed: 01-08-2021.

[43] P. Hirel, Atomsk: a tool for manipulating and converting atomic data files, Comput. Phys. Commun. 197 (2015) 212–219.

[44] J. Guénolé, W.G. Nöhring, A. Vaid, F. Houllé, Z. Xie, A. Prakash, E. Bitzek, Assessment and optimization of the Fast Inertial Relaxation Engine (FIRE) for energy minimization in atomistic simulations and its implementation in LAMMPS, Comput. Mater. Sci 175 (2020) 109584.

[45] S. Plimpton, Fast parallel algorithms for short-range molecular dynamics, J. Comput. Phys. 117 (1) (1995) 1–19.

[46] S. Plimpton, R. Pollock, M. Stevens, Particle-mesh Ewald and rRESPA for parallel molecular dynamics simulations, in: Proceedings of the PPSC, Citeseer, 1997.

[47] P.A.T. Olsson, Transverse resonant properties of strained gold nanowires, J. Appl. Phys. 108 (3) (2010) 034318.

[48] A. Prakash, D. Weygand, E. Bitzek, Influence of grain boundary structure and topology on the plastic deformation of nanocrystalline aluminum as studied by atomistic simulations, Int. J. Plast. 97 (2017) 107–125.

[49] T. Xu, M. Li, Topological and statistical properties of a constrained voronoi tessellation, Philos. Mag. 89 (4) (2009) 349–374.

[50] P.H. Serrao, S. Sandfeld, A. Prakash, Optimic: a tool to generate optimized polycrystalline microstructures for materials simulations, SoftwareX 15 (2021).

[51] A. Prakash, M. Hummel, S. Schmauder, E. Bitzek, NanoSCULPT: a methodology for generating complex realistic configurations for atomistic simulations, MethodsX 3 (2016) 1–9.

[52] J. Stadler, R. Mikulla, H.R. Trebin, IMD: a software package for molecular dynamics studies on parallel computers, Int. J. Mod. Phys. C 8 (05) (1997) 1131–1140.

[53] H.S. Park, J.A. Zimmerman, Modeling inelasticity and failure in gold nanowires, Phys. Rev. B 72 (5) (2005) 054106.