Clustering Algorithms for Nanomechanical Property Mapping and Resultant Microstructural Constituent and Phase Quantification

Bryer C. Sousa, Chris Viera, Rodica Neamtu, and Danielle L. Cote

Abstract Tacit assumptions have been made about the suitability of two primary data-driven deconvolution algorithms concerning large (10,000+) data sets captured using nanoindentation grid array measurements, including (1) probability density function determination and (2) k-means clustering and deconvolution. Recent works have found k-means clustering and probability density function fitting and deconvolution to be applicable; however, little forethought was afforded to algorithmic compatibility for nanoindentation mapping data. The present work highlights how said approaches can be applied, their limitations, the need for data pre-processing before clustering and statistical analysis, and alternatively appropriate clustering algorithms. Equally spaced apart indents (and therefore measured properties) at each recorded nanoindentation location are collectively processed via high-resolution mechanical property mapping algorithms. Clustering and mapping algorithms also explored include k-medoids, agglomerative clustering, spectral clustering, BIRCH clustering, OPTICS clustering, and DBSCAN clustering. Methods for ranking the performance of said clustering approaches against one another are also considered herein.

Keywords Nanoindentation · Clustering algorithms · Microstructures

Introduction

Advances in nanoindentation testing systems' application, understanding, and functionality have continued with regularity since the formalization of the original Oliver-Pharr (OP) in the late 1980s and early 1990s [1]. Such advancements include the in-situ integration of nanoindentation systems with scanning electron as well as transmission electron microscopes [2], the development of high-strain rate impact testing methods via nanoindentation [3], the ability to quantify stress–strain relations [4], and the ability to perform nanoindentation testing of materials at notably elevated and

B. C. Sousa (✉) · C. Viera · R. Neamtu · D. L. Cote
Worcester Polytechnic Institute, Worcester, MA, USA
e-mail: bcsousa@wpi.edu

TMS 2022 151st Annual Meeting & Exhibition Supplemental Proceedings,
The Minerals, Metals & Materials Series,
https://doi.org/10.1007/978-3-030-92381-5_68

cryogenic temperatures [5]. In addition to the advancements, considerable research and development have also been dedicated to the formalization of statistically significant and high-throughput mechanical property mapping at a rate of an indent per second [6, 7]. With the latter in mind, the present work aims to build upon the current state of nanoindentation-based mechanical property mapping and the analysis of the data obtained through such experimental protocols. That said, consideration of prior work related to the data-driven analysis of nanomechanically mapped datasets is considered first.

Background

When consideration was initially being given to the potential value of nanoindentation grid arrays for mechanical property mapping, Randall et al. noted that a grid array of 2500 preprogrammed and automated nanoindentation measurements could be successfully obtained over three to four days [8]. However, by 2012, nanoindentation "tomography" remained relatively limited in high-throughput functionality (relative to modern systems), which can be shown by way of considering the work of Tromas et al. via [9]. That is not to say that the work of Randall et al. and Tromas et al. was any less valuable; rather, detailing the history of nanomechanical mapping or grid array protocol implementation with a nanoindenter enables one to contextualize better the degree of advancement achieved since that period. Specifically, as nanoindentation technologies advanced in the 2010s, the rate at which individual indents could be measured continued to the point of an indent per second in the case of Nanomechanics, Inc. (now KLA), via a method named NanoBlitz3D [10]. Such a revolution in the high-throughput nature of nanoindentation mapping implementation can be exemplified by comparing the three-to-four-day timeframe encountered by Randall et al. for 2500 indents to the amount of time required to measure the same number of indents via NanoBlitz3D, which would only be 0.0289 days or just shy of 42 min. With such remarkable testing speeds, nanoindentation grid arrays and nanomechanical property mapping quickly enabled relatively massive datasets to be obtained in realistic timeframes and therefore enabled big data or data-driven techniques to be suitably applied for analyzing the results. For context, Fig. 1 presents a nanoindentation array measured using NanoBlitz3D and the iMicro Pro from KLA that houses 160,000 indentation measurements within one array measured on a 4xxx series steel.

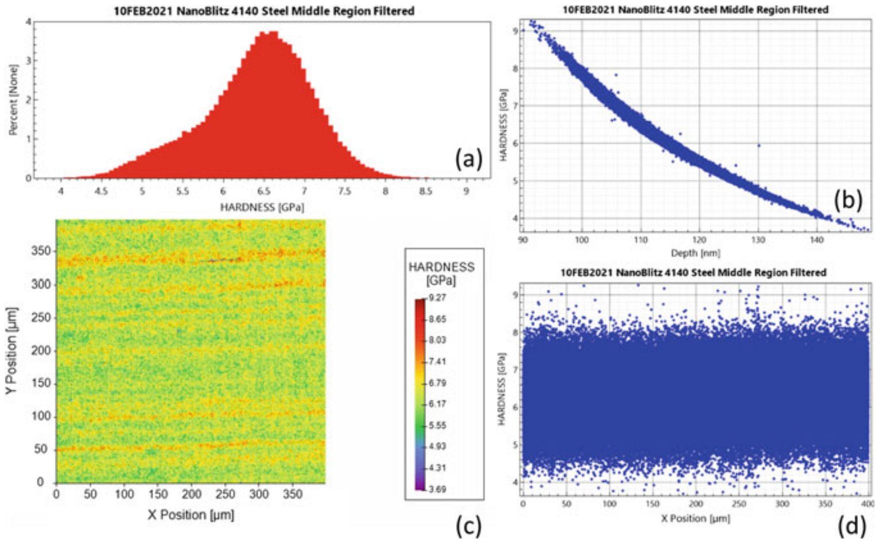

Fig. 1 **a** A NanoBlitz 3D 4140 steel hardness histogram. **b** The hardness versus depth data for the same material system displayed in the hardness histogram (**a**). **c** The hardness contour plot for the 160,000 indentation measurements within one array measured on a 4xxx series steel. **d** The hardness versus array's x-axis position within the nanomechanical property map shown in (**c**)

Methods and Materials

Data-Driven Details

The data analysis techniques applied herein include: (1) probability density function (PDF) and deconvolution, (2) k-means clustering and deconvolution, (3) k-medoids clustering and deconvolution, (4) agglomerative clustering and deconvolution, (5) spectral clustering and deconvolution, (6) balanced iterative reducing and clustering using hierarchies (BIRCH) and deconvolution, (7) ordering points to identify the clustering structure (OPTICS) and deconvolution, and (8) density-based spatial clustering of applications with noise (DBSCAN) and deconvolution.

(1) The probability density function determination and deconvolution method rely on the idea of fitting a variable number of normal curves to the PDF of a dataset. A normal curve represents each cluster. If a data point is in each normal curve, then that point and all other points in that normal curve will make up a cluster. The deconvolution method will iteratively fit new normal curves to the data set, keeping track of the best result so far as it does this. The new curves are generated by assigning them to random sections in the probability density function. The best so far is the combination of normal curves that best account for the probability density function of the original data. Combining these normal curves should result in a similar probability density function as

the original, assuming the number of curves is appropriately chosen for the data set. A limit can also be applied to determine an acceptable combination of normal curves to use, as shown by the decon 3.0 program [11], which the deconvolution process in this paper is based on.

(2) K-means clustering focuses on generating *k* number of centroids to represent *k* different clusters in a data set. A centroid is an arbitrary point in the range of data values representing a cluster's center. The distance between the point and each centroid must be calculated to determine what cluster a point is affiliated with. The data point will belong to the cluster paired with the centroid it is closest to. The locations of these centroids are determined through a random and iterative process. To start, *k* centroids are distributed randomly throughout the data set. The algorithm then moves these centroids to minimize the within-cluster sum-of-squares. These centroids are moved iteratively to locations where the within-cluster sum-of-squares is decreased. Once the within-cluster sum-of-squares has reached its minimum value, the centroids are in their final locations and returned as the clustering result.

(3) K-medoids clustering focuses on generating *k* number of medoids representing k different clusters in a data set. K-medoids is incredibly like that of k-means, differing in the use of medoids instead of centroids. A medoid differs from a centroid as it must be one of the data points themselves. This makes the algorithm more robust when dealing with outliers and noise as it is less likely to have its clusters' centers closer to undesirable points. Despite this difference, the method performs similarly to k-means, iteratively moving around its medoids until it finds the smallest within-cluster sum-of-squares.

(4) Agglomerative clustering is a form of hierarchical clustering. Agglomerative clustering involves creating a dendrogram with all the data points by pairing them together interactively. To start, each data point is represented as its own cluster or group. With each iteration, a metric is computed to determine the absolute difference between each data point and every other data point, making every possible pair of clusters. It will then take the two points with the best metric value for that iteration and combine them into their own cluster. The cluster of the two points now has a new value that represents it to compare with other data points. This first iteration forms the first step of the dendrogram. This process is then done iteratively until only the number of clusters left is specified at the start of the algorithm.

(5) Spectral clustering is a form of clustering which performs a low-dimensional embedding of the affinity matrix between samples and then clusters the result using k-means. Spectral clustering takes in all data points and then computes a similarity graph using either a radius (epsilon-neighborhood) or k-nearest neighbors. Once this is completed, it will create a Laplacian matrix. After this, it will compute the first *k* eigenvectors of its Laplacian matrix to define a feature vector for each object. Finally, after the original data points have been represented in this way, the algorithm runs k-means on these features to separate objects into *k* classes.

(6) Birch clustering is a form of clustering that builds a Clustering Feature Tree to create Cluster Feature Nodes (CF Nodes) to reduce data. These nodes represent several subclusters called Clustering Feature subclusters. Each of these subclusters stores information involving the data points, allowing them to represent them accurately. This algorithm reduces the amount of data by creating a tree of the data and then clustering the resulting CF nodes in the tree. The clustering algorithm used after this point to further cluster the data is arbitrary. The library used by this paper runs agglomerative clustering. In terms of how the respective tree is formulated, a new sample is inserted into the root of the CF Tree, which is a CF Node. It is then merged with the subcluster of the root that has the smallest radius after merging, constrained by the threshold and branching factor conditions. If the subcluster has any child node, then this is done repeatedly till it reaches a leaf. After finding the nearest subcluster in the leaf, the properties of this subcluster and the parent subclusters are recursively updated. If the radius of the subcluster is obtained by merging the new sample and the nearest subcluster is greater than the square of the threshold, and if the number of subclusters is greater than the branching factor, then a space is temporarily allocated to this new sample. The two farthest subclusters are taken, and the subclusters are divided into two groups based on the distance between these subclusters.

(7) OPTICS is a form of clustering incredibly like DBSCAN with a few additions. Along with the fundamental properties of DBSCAN, OPTICS also has two additional metrics. The first metric is a minimum distance to make a given point a core point. The second metric OPTICS uses are known as the reachability distance or the distance between density-reachable points. This reachability metric can then be used to separate clusters, separating clusters every time there are peaks in the reachability metric.

(8) DBSCAN is a form of clustering which focuses on the idea of a dataset being separated into areas of high-density data points and low-density data points. The goal of this algorithm is to identify the sections of high-density points into separate clusters. This algorithm requires that points have a minimum number of points in that cluster to be classified as a cluster. The process completes its clustering by creating a circle around each data point and classifying each data point based on the number of points within a radius. Once this is done, clusters can be formed from these points by forming cores. Iteratively, it goes through the process of joining points as follows. First, X is density-reachable from Y when X is in the radius of Y and Y is a core point. Next, X is density-connected to Y when there is a point O where both X and Y are density-reachable from O. All density-connected points become a separate cluster. Once this process is complete, there will be several clusters from the density-connected point sets and several outliers that did not fit into the requirements of being density-connected with a minimum number of points. Moreover, HDBSCAN is a form of clustering that combines DBSCAN and hierarchical clustering. It is like DBSCAN but does not use a fixed cutoff as the radius around a point to group points with. It instead handles any offshoots in the dendrogram by discarding

them using the minimum cluster size parameter. This creates a denser dendrogram and reduces the number of small extra clusters often present in DBSCAN. This method also relies on generating an estimated probability density function of the data through sampling, varying the number of samples to find a balance between a noisy PDF and one that may be too smooth. HBSCAN also has another parameter that specifies the minimum cluster size, which must be balanced to prevent too many extra clusters from forming from being too low or merging too many clusters together from being too high.

Data Pre-processing

Once several clustering methods had been explored, it became apparent that the data needed to be transformed for the clustering methods to create optimal models. This was done through interpolation, correcting outliers, and separating outliers from data sets to be added on later. While plotting the original data set, it became apparent that nulls were a constant issue. A single null value can prevent the software from displaying a map. To counter this, any nulls in the original data are interpolated by basing them off their neighbors. To do this, the project uses SciPy's interpolate library. It can find all points which are null and then interpolate them based on their neighbors. To interpolate the data, it runs through a two-step processing using two different interpolation algorithms. The first algorithm generates the most accurate guess of a data point possible based on the surrounding neighbors. However, this is not guaranteed to fill all null values in with a numerical value. If the null data point has many null neighbors, it will be unable to generate a value. To compensate for this, the data is run through another interpolation method which is guaranteed to fill in every null value regardless of the number of null neighbors at the cost of accuracy. Instead of using a calculation like the first algorithm, it picks the neighbor closest to it and uses that value. Once this process has been completed, all null values have been corrected.

Due to the focus on clustering, some required the data to be cleaned before clustering. K-means and agglomerative clustering, for example, is very prone to be skewed by outliers. If there is a small group of outliers far away from the data points, then it is very likely that a cluster will only be composed of outliers and take away from the analysis of the substance, especially when these outliers are defects. To counter this, the data were cleaned to remove all outliers. Due to the lack of a standardized metric, existing statistical methods were explored. The first method involved taking a sample's mean and standard deviation and defining any point 3 standard deviations or more away from the mean for an outlier. The second method involved calculating the first quartile, the first quartile, and the interquartile range. After this, any value outside the range of the first quartile -1.5 * IQR and the first quartile $+1.5$ * IQR were defined as outliers. The method involving the mean and standard deviation was much more successful in accurately removing outliers. Once outliers had been removed, the resulting null values would be interpolated just as they were in the previous section. This allowed an entire grid of data for clustering and mapping.

After exploring removing outliers, a sample was chosen to be mapped where the outliers were essential to identify. These involved a small section of a material whose hardness value was more considerable to an unexpected material. Another process was developed to prevent outliers from causing the clustering methods to fail but still consider outliers. This process involved defining the outliers the same as above. Instead of removing and interpolating the values, the data values would be separated into two sections, one containing outliers with higher-than-expected values and another containing outliers with lower-than-expected values. The expected data values, which were not classified as outliers, would then be clustered. After this, they would be recombined with the outliers. When recombining the outliers, they would be identified as being in their own clusters. All outliers with lower-than-expected values would exist in a cluster, and those with higher-than-expected values would exist. If outliers existed on both sides of a dataset and k-means clustering was used without outliers into three clusters, the resulting contour plot would have five clusters. This allowed the clustering algorithms to perform as intended while still marking off anomalies.

After developing a framework to generate clustering models on more optimal data and exploring the evaluation of these models as possible, it became necessary to use a standardized metric to compare them after implementing the clustering methods above. The following metrics were explored to solve this issue. After generating the clustering configurations used in this paper, it became practical to compare how similarly two clustering methods performed. Unlike the other evaluation techniques, this would not determine how well a clustering model performs, only how similar it is to another clustering model. The rand index score is used to compare the results of two clustering outputs and determine their similarity. If two sets were the same, it would produce 1.0; if they were completely different, it would produce 0.0. SciPy's adjusted rand index score was used for this project, which allowed for values to be lower after taking chance into account. This allowed the quantitative measurement of how similar clustering configurations were to one another after running them on the same data set. Most importantly, if the original data set had each data point with a labeled phase fraction, the metric could then be run between each clustering configuration and the labeled data. This would produce a metric for how well the clustering configuration scored to correctly identify the material phase at each (x, y) location.

Engineering-Driven Details and Initial Performance

The primary material considered during the present work was a Pb–Sn soldering alloy. The Pb–Sn soldering alloy utilized was formulated with 60% Sn and 40% Pb with a 2% (by weight) leaded rosin-activated flux core. The 60/40 soldering alloy system was selected due to the solders' near eutectic nature. Given the near eutectic nature of the selected alloy system, along with consultation of prior literature of relevance on 60/40 solder solidification microstructures, Pb40/Sn60 microstructures house two microstructural phases. Specifically, the two microstructural phases housed within Pb40/Sn60 solders include an alpha-Pb dendritic constituent in addition to an alpha-Pb/beta-Sn near eutectic constituent, which has previously been shown to have separable micromechanical properties (such as mean contact pressure, that is, indentation hardness); thus offering an economically viable and easily processable material for as-solidified microstructural property mapping and subsequent data analysis for dual-phase fraction quantification. Due to the thermodynamically unstable nature of the soldered microstructure obtained via a soldering iron, experimental methods were applied shortly after solidification and metallurgical preparation.

In terms of metallurgical preparation, the as-solidified 60/40 soldering alloy was compression mounted in black phenolic resin using mounting materials and a compression mounting system from Buehler (Lake Bluff, IL USA). Upon compression mounting in phenolic resin, the sample was mechanically polished to a mirror finish wherein a 0.05 um colloidal silica suspension-based final polishing step was employed using an automatic mechanical polishing suite sourced from Buehler. Buehler's automatic polisher and compression mounting system, which was utilized in the present study, are housed and maintained at Worcester Polytechnic Institute (Worcester, MA, USA) as part of the Buehler Center of Excellence affiliated with the Metals Processing Institute.

Following the mounting, grinding, and polishing procedures implemented, scanning electron microscopy (SEM), digital microscopy, and nanoindentation-based mechanical mapping was performed for a dual-phase fraction or phase area percentage determination benchmarking. Regarding SEM analysis, a tabletop Zeiss (Oberkochen, Baden-Württemberg, Germany) Evo MA-10 series scanning electron microscope was employed. An accelerating voltage between 5 and 10 kV was used during SEM analysis alongside a working distance of 10.5 mm and a secondary electron detector. An example SEM-captured micrograph, which was measured after nanomechanical mapping was performed, is presented in Fig. 2.

As noted above, nanomechanical mapping was performed before secondary electron-based SEM analysis. Consistent with the dual-phase microstructure discussed for the Pb40/Sn60 soldering alloy, the light gray constituents captured in Fig. 2 represent the near eutectic alpha-Pb/beta-Sn phase while the dark gray constituents represent the dendritic alpha-Pb phase. The relatively spherical and dark alpha-Pb features can also be observed in the near eutectic primary phase of the light gray beta-Sn. As for the 50-by-75 array of indents shown in the SEM micrograph of the soldering alloy presented in Fig. 2 followed from nanoindentation mapping using an InForce 1000 mN actuator, diamond Berkovich nanoindenter tip from Micro Star Technologies, Inc. (Huntsville, TX, USA), which has since been obtained by Bruker (Billerica, MA, USA), and KLA's (Milpitas, CA, USA) iMicro Pro, which was manufactured by Nanomechanics, Inc. (Oak Ridge, TN, USA), before KLA acquired Nanomechanics, Inc. Rapid nanomechanical mapping with the iMicro Pro system was achieved through the use of the NanoBlitz 3D test method.

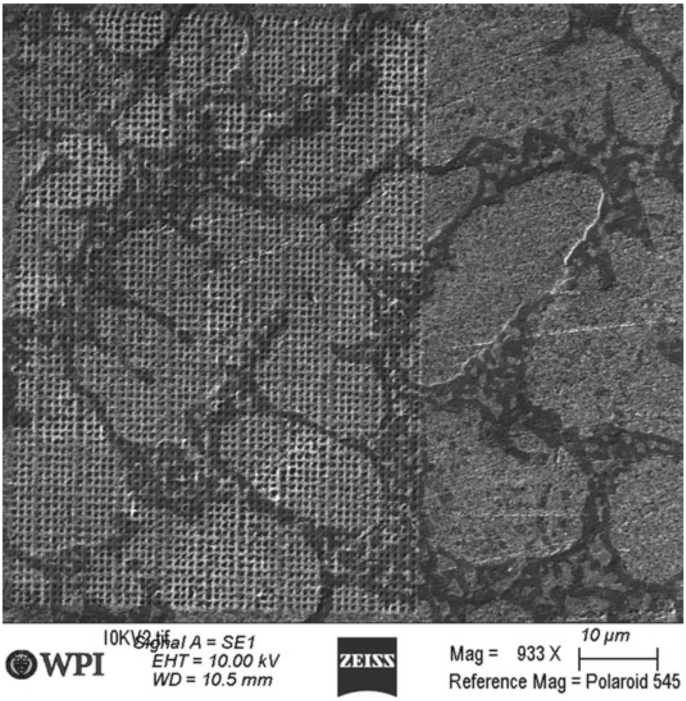

Fig. 2 SEM-captured micrograph, measured after nanomechanical mapping was performed on the as-solidified 60/40 soldering alloy

That said, dynamic or continuous stiffness measurement (CSM) nanoindentation testing was performed before applying the NanoBltiz 3D test method to identify an average target load associated with the desired target nanoindentation mapping depth of approximately 100 nm. After that, the nanomechanical map was obtained by defining the target applied load at each location within the grid array per the dynamically/CSM-determined load associated with a 100 nm nanoindentation depth. As a result, the relatively recently refined nanoindentation spacing of ten times the depth rule-of-thumb demonstrated in [10] was able to be applied herein as well for enhanced nanomechanical property contour plotting/mapping. Furthermore, implementing the k-means clustering protocol built into the commercially available nanoindentation data analysis software known as InView, which is associated with commercial nanoindentation systems from KLA, basic benchmarking initialization of clustered and deconvoluted phase fractions was achieved. Further details surrounding benchmarking nanoindentation insights obtained are subsequently presented too.

Additional benchmarking was procured by way of image analysis using ImageJ and computational thermodynamic analysis via Thermo-Calc. In terms of computational thermodynamic analysis, the commercial software used was Thermo-Calc

2021b, which enabled equilibrium-based volume fractions of the dendritic (alpha-Pb) phase as well as the primary constituent of the near eutectic (beta-Sn) phase such that the results could be compared as a benchmark against the various clustering techniques described in the Data-driven Details subsection of the Methods and Materials section of the present work. Furthermore, equilibrium-based computational analysis via Thermo-Calc is achieved through the "CALculation of PHAse Diagrams," or the CALPHAD technique. Moreover, the computationally assessed system was defined within Thermo-Calc 2021b via the soldering alloy-based database, denoted within Thermo-Calc as TCSLD3: Solder Alloys v3.3. Accordingly, the mass percent of Pb was set to 40%, while the mass percent of Sn within the system was set to 60%, given the Pb40/Sn60 composition of the soldering alloy experimentally considered herein. Furthermore, beyond the use of the database, the conditional temperature and pressure defined for the single-point equilibrium calculation were set to 294.15 °K and 100,000 Pa, respectively. At the same time, the system size was set to 1.0 mol.

Finally, one may consider the details surrounding the use of image analysis via ImageJ herein. First, a JPEG formatted digital micrograph was obtained using the digital microscope accompanying the iMicro Pro nanoindenter for image analysis via ImageJ. Then, said JPEG-based digital micrograph was initially opened within ImageJ 1.53e and cropped to remove regions containing shadowing and edge effects. After that, the JPEG-based image file was converted to a TIFF-based file format. Upon TIFF reformatting of the cropped JPEG-based image, thresholding was applied to the 8-bit TIFF-based image such that binarization of the alpha-Pb (light constituents shown in Fig. 3a) and the beta-Sn (darker constituents shown in Fig. 3a) was achieved. After that, the area of the binarized micrograph shown in Fig. 3b associated with alpha-Pb (the black features in Fig. 3b) and the beta-Sn (the white features in Fig. 3b) could be quantified as an area percentage relative to the total surface microstructural area shown in Fig. 3b post-thresholding.

To establish a ground truth for comparison of clustering and deconvolution algorithm results with one another and with independent methods of phase fraction determination, nanoindentation mapping via KLA's respective software and test method gave 29.1% of the Beta-Sn dominant phase, ImageJ suggested 31.79% of the same phase, and the Thermo-Calc volume phase fraction computed at ambient equilibrium for the 60–40 alloy gave 30.36%. Follow-on work based upon the framework detailed herein and using the methods detailed in the Data-driven Details subsection of the Materials and Methods section of the present work will enable identification of the optimal data clustering techniques for nanomechanical property and phase mapping.

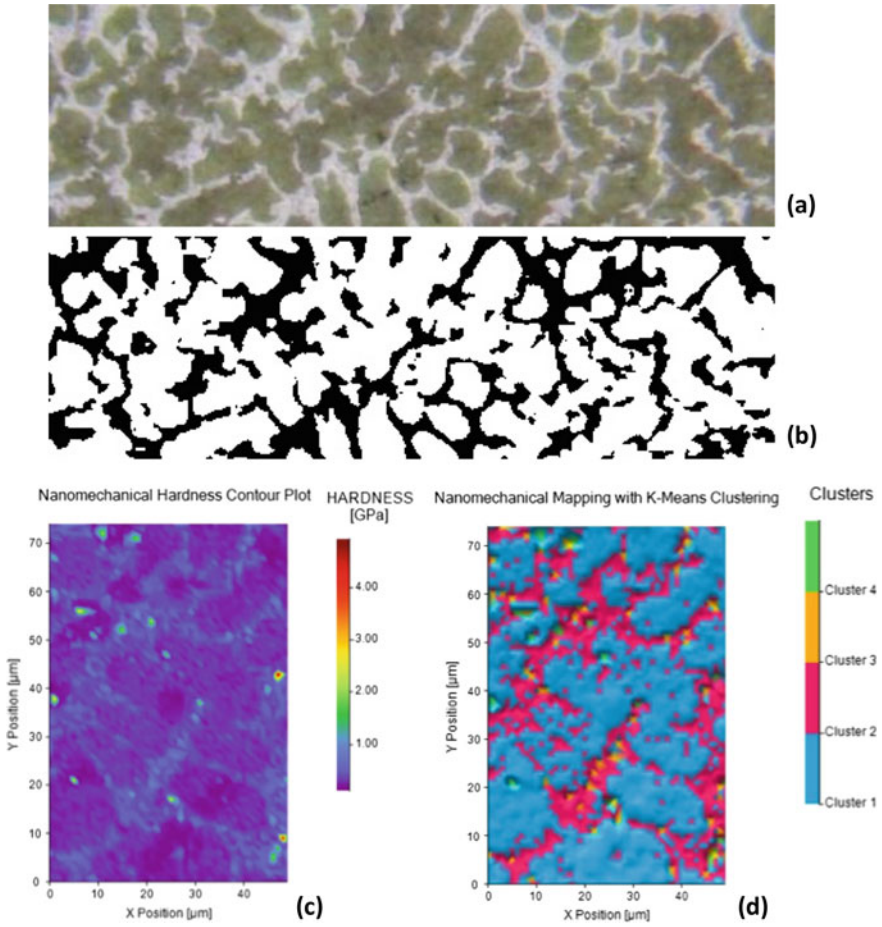

Fig. 3 **a** Digital micrograph of soldering alloy experimentally considered; **b** The binarized digital micrograph from ImageJ pre-processing; **c** Nanoindentation hardness contour plot obtained using NanoBlitz 3D and iMicro Pro; and **d** The k-means clustered phase map

References

1. Oliver WC, Pharr GM (1992) An improved technique for determining hardness and elastic modulus using load and displacement sensing indentation experiments. J Mater Res 7(6):1564–1583
2. Minor AM, Morris JW Jr, Stach EA (2001) Quantitative in situ nanoindentation in an electron microscope. Appl Phys Lett 79(11):1625–1627
3. Sudharshan Phani P, Oliver WC (2017) Ultra-high strain rate nanoindentation testing. Materials 10(6):663
4. Hay J (2019) U.S. Patent No. 10,288,540. U.S. Patent and Trademark Office, Washington, DC
5. Wheeler JM, Armstrong DEJ, Heinz W, Schwaiger R (2015) High-temperature nanoindentation: the state of the art and future challenges. Curr Opin Solid State Mater Sci 19(6):354–366

6. Vignesh B, Oliver WC, Kumar GS, Phani PS (2019) Critical assessment of high-speed nanoindentation mapping technique and data deconvolution on thermal barrier coatings. Mater Des 181:108084
7. Yang M, Sousa B, Smith R, Sabarou H, Cote D, Zhong Y, Sisson RD (2021) Bainite percentage determination and effect of Bainite percentage on mechanical properties in austempered AISI 5160 steel. Mater Perform Character 10(1):110–125
8. Randall NX, Vandamme M, Ulm F-J (2009) Nanoindentation analysis as a two-dimensional tool for mapping the mechanical properties of complex surfaces. J Mater Res 24(3):679–690
9. Tromas C et al (2012) Hardness and elastic modulus gradients in plasma-nitrided 316L polycrystalline stainless steel investigated by nanoindentation tomography. Acta Materialia 60(5):1965–1973
10. Phani PS, Oliver WC (2019) A critical assessment of the effect of indentation spacing on the measurement of hardness and modulus using instrumented indentation testing. Mater Des 164:107563
11. Němeček (2009) Nanoindentation of heterogeneous structural materials. PhD diss, Czech Technical University in Prague