

RESEARCH

Open Access

Segmentation of four-dimensional, X-ray computed tomography data

John W Gibbs* and Peter W Voorhees

*Correspondence:
jwgibbs@u.northwestern.edu
Department of Materials Science
and Engineering, Northwestern
University, Evanston, IL, USA

Abstract

The rapidly improving temporal resolution of X-ray computed tomography (CT) imaging methods makes it ever easier to do in-situ, time-resolved (4D) experiments. This work describes a method of segmenting 4D X-ray CT data that works well for extracting information on the interfacial properties, such as interfacial curvature and velocity. As an example of this method, a segmentation is performed on data from an isothermal coarsening experiment of an Al-Cu solid/liquid mixture.

Keywords: Segmentation; Registration; 4D data; Computed tomography

Background

The key characteristic of a 4D experiment is having an in-situ measurement method in which the duration of each scan is on the same timescale or faster than the evolution of the sample. X-ray computed tomography (CT) is a common way to do 4D studies because it can be used to make consecutive, non-destructive measurements of a sample in 3D. This technique has been used for the last decade [1] with the continually improving spatial and temporal resolution of synchrotron facilities enabling an ever wider variety of materials and processes to be studied. The recent review article by Rowenhorst [2] covers upwards of a dozen of these recent 4D studies and the review articles by Stock [3] and Landis [4] cover X-ray CT in great detail. These articles contain many examples in which either 3D or 4D characterization of microstructures has provided valuable information and also how these can be used to validate microstructural evolution simulations that are being used in materials-by-design methods. Segmentation methods like the one that is presented here are necessary to produce the accurate data that is required for these comparisons.

We have used X-ray CT to study the isothermal coarsening dynamics of solid/liquid mixtures of Al-Cu alloys. The goal of these experiments is to determine the interfacial locations between the liquid and solid phases and how the interfaces evolve over time. This makes it possible to study the interfacial curvature (which sets the chemical potential of the interface and therefore drives coarsening) and how it affects the microstructural evolution of the sample. 4D data is necessary to do this because calculating curvatures requires the interfacial locations in 3D and calculating the interfacial normal velocities requires interfacial locations over time.

The goal of this paper is to share some of our experiences working with 4D X-ray CT and the Al-Cu system and to show some of the analysis tools that we have used and developed to work with these datasets. The methods that will be discussed here have been developed for use with a very specific type of datasets (4D, containing three phases and were created with absorption contrast, X-ray CT) but the methods are in fact very general and could easily be applied to datasets containing greater or fewer phases, 2D or 3D data or even completely different characterization techniques like optical or scanning electron microscopy.

The segmentation technique that is presented here has been tailored to determine interfacial locations that are both smooth and accurate. This is in contrast to many other segmentation techniques in which the goal is to produce a segmentation that is as accurate as possible with little regard for the smoothness of the boundaries. To achieve this, an iterative technique is employed in which the goal is to determine smooth interface locations that are still consistent with the noisy data.

Methods

Experimental methods

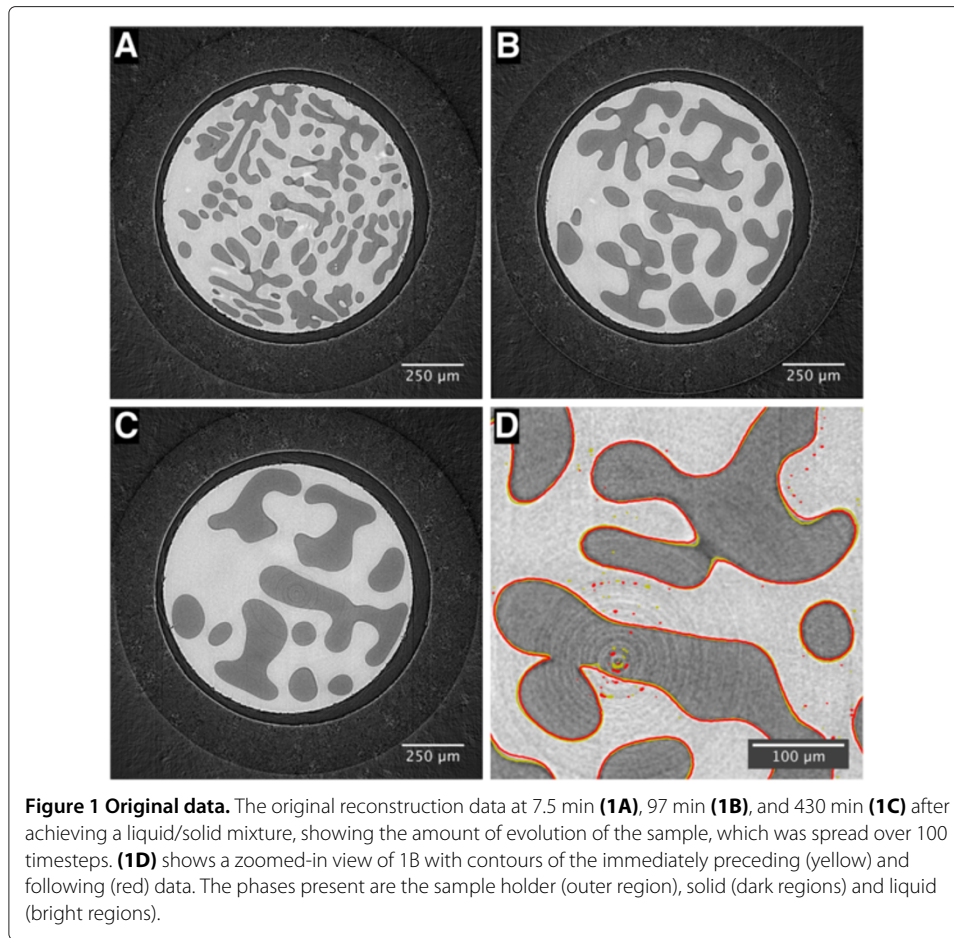
The samples used in these studies are all hypo-eutectic Al-Cu alloys with compositions between 14 wt% Cu and 28 wt% Cu. The samples are prepared by directionally solidifying a 12mm diameter rod, then cutting out 1mm diameter cylinders. These samples are held at 553°C, which is 5°C above the eutectic temperature, for the duration of the experiment to create a solid/liquid mixture with constant phase fractions. The furnace used in these experiments is able to hold the temperature to well within a degree of the setpoint, which negates the possibility of noteworthy microstructural changes due to temperature variations. Experiments are run for up to 12 hours to develop a significant amount of change in the microstructure.

These experiments were carried out at the TOMCAT beamline at the Swiss Light Source, Paul Scherrer Institut in 2008 and use an exposure time of 250 ms and 721 projections per reconstruction. This results in low noise reconstructions and a total scan time of about 4 minutes, which is well matched to the rate of change of the sample. The reconstructed data is $1024 \times 1024 \times 1024$ voxels and each voxel has an edge length of $1.4 \mu\text{m}$.

Reconstructions of the CT projection data are done with filtered back projection using only a high-pass filter. A low-pass filter is typically also applied to the projection data before back projecting to reduce the noise in the tomograms; however, we find that it is a better tradeoff to skip the low-pass filter to make the interfaces as sharp as possible and remove the noise in the segmentation method. Some examples of these reconstructions, which will henceforth be referred to as ρ , can be seen in Figure 1.

Registration methods

One advantage of using X-ray CT to perform a 3D characterization is that it results in a rigid 3D dataset that does not require any registration of various 2D sections as is required with many serial section methods of collecting 3D data. That said, registration is still an important step in the analysis of 4D CT data because the sample rotates through 180° during the CT scan and when it is done, it must return to the initial position. If it doesn't



return to the exact starting position of the previous scan, the resulting two datasets will be slightly rotated relative to one another.

This misalignment, which is typically on the order of fractions of a degree, can be removed by registering the multiple datasets. The registration process is an optimization problem in which a metric that quantifies the alignment of the two datasets is defined and optimized with respect to the degrees of freedom that are allowed for the spatial transformations. Since the 3D datasets can be considered rigid, only affine transformations need be considered.

Because the segmentations are done in 4D, registration must be done on ρ before segmenting the data. The registration starts with the first 3D dataset being the reference dataset and the second dataset being registered to it. When this is complete, the transformed second dataset is used as the reference and the third dataset is registered relative to the second. In this way, the differences between datasets that is due to sample evolution is negligible compared to the misalignment.

The matching metric that is used here is mutual information as in [5,6]. Mutual information measures the similarity between datasets using the Shannon entropy of the datasets. The entropy of a dataset, A , is given by:

$$H(A) = - \sum_i p(i) \log p(i) \quad (1)$$

where $p(i)$ is the probability of finding intensity i in dataset A . The joint entropy between any two datasets, A and B , is calculated as:

$$H(A, B) = - \sum_{i,j} p(i, j) \log p(i, j) \quad (2)$$

where $p(i, j)$ is the probability of voxels in at the same physical location in datasets A and B having intensity values of i and j , respectively. The mutual information between two datasets is calculated using:

$$I(A, B) = H(A) + H(B) - H(A, B) \quad (3)$$

Because mutual information is a measure of the similarity between two datasets, it will be maximized when the two datasets are properly aligned.

Segmentation methods

The main goal of these segmentations is to determine the interface locations between the liquid and solid phases with sufficient accuracy to be able to calculate interfacial curvature and velocity. Both of these quantities can be significantly affected by even slight amounts of noise in the data. Thus, the goal of the segmentation are to determine interfacial locations that are smooth in both time and space while keeping the interfacial location consistent with the reconstructed data.

The segmentation method succeeds by separating the image intensity information from the interfacial location information, then iteratively applying diffusive smoothing and accuracy maintaining steps to the interface location data. The accuracy maintaining step utilizes smoothing to operate on the larger features more than on the smaller features that are more likely to be noise; thus, maintaining the accuracy of the larger features while the smoothing reduces the noise.

To define the interfacial location information, a level-set based method is used to implicitly define the interface locations using a signed distance function (SDF). A SDF is a dense array of data in which the value at each voxel is the distance to the nearest interface with negative values being considered on one side of the interface and positive values on the other side of the interface; thus, the interface is defined as where the SDF crosses zero. The implicit interface locations allows for sub-voxel level interface positioning and also lends itself to simple, accurate curvature calculation.

All arrays used in this work are defined in 4D; however, there are some instances where operations are performed in 3D to every timestep individually. The reinitialization of a SDF is one of these instances because the definition of a SDF is only a spatial relationship and not explicitly a temporal one.

The segmentation technique is largely based on the piecewise-smooth method for segmenting two phases presented in [7]. In the current work, modifications have been made to account for multiple phases. Similar multi-phase modifications have been made in [8,9]; however, the current implementation is somewhat simpler than these at the expense of some generality.

The piecewise-smooth method that is used is based on having a few distinct regions, in which the intensity values are smooth and slowly varying with abrupt changes in

intensity at the boundaries of the phases. A variation of the piecewise-smooth method is used in this work in which the smooth intensities for each phase are replaced with constant values. The piecewise-constant approach is used here because the data used contains three distinct regions, each of which has a roughly constant intensity value. The piecewise-smooth method from [7] could easily be incorporated into these methods.

While this method can be used for 2D, 3D or 4D data, there are some distinct advantages to working with the highest dimensionality of data that is available due to the increasing data density with increasing dimensionality. For example, a single voxel in a 3D dataset has 6 neighbors within a one voxel radius and 32 neighbors within a two voxel radius, compared to 4 and 12 neighbors within one and two voxel radii for a 2D dataset. In 4D, this effect is compounded by the temporal neighbor data^a.

The overall scheme of this segmentation technique is comprised of an initialization stage and an iterative stage. In the initialization, the following are calculated:

- The initial interfacial locations
- The intensity values of the individual phases
- The piecewise constant approximation of the intensity values

Each iteration contains the following steps:

- Compute the difference between the original intensity data and the piecewise constant approximation
- Use this difference to update the interfacial locations
- Smooth the interfacial location arrays
- Recompute intensity values of the individual phases
- Recompute the piecewise constant approximation of the intensity data

The details of this procedure are given below and an example of its use, along with values used for the parameters, is given in the Results section. The software used to do these segmentations is written in Fortran and parallelized using MPI to take advantage of the large amounts of memory available on a distributed memory machine.

Initialization

The three, previously described regions which will be segmented are identified with Ω_1 , Ω_2 and Ω_3 which will represent the sample holder, the solid phase and the liquid phase, respectively. These Ω_i variables are sets in \mathbb{R}^4 . Voxel locations will be described by the 4D vector \vec{x} ; therefore, if a voxel is in the solid phase, it will be described as $\vec{x} \in \Omega_2$. The mathematical definition of these regions will be given later.

Two threshold values must be manually selected to determine the initial interface locations from the data, $\rho(\vec{x})$: $T_{1/2}$ will be used to segment between the darkest (Ω_1) and intermediate brightness (Ω_2) regions and $T_{2/3}$ will be used to segment the intermediate (Ω_2) from the brightest (Ω_3) regions. The interface locations, which will be stored in arrays, ϕ_1 and ϕ_2 , are initialized as follows:

$$\phi_1(\vec{x}) = \rho(\vec{x}) - T_{1/2} \quad (4)$$

$$\phi_2(\vec{x}) = \rho(\vec{x}) - T_{2/3} \quad (5)$$

A reinitialization procedure, as described in [10], is then used to turn these arrays into signed distance functions. These arrays can be used to mathematically define the different regions as follows:

$$\Omega_1 = \text{where } ((\phi_1(\vec{x}) < 0) \text{ and } (\phi_2(\vec{x}) < 0)) \quad (6)$$

$$\Omega_2 = \text{where } ((\phi_1(\vec{x}) > 0) \text{ and } (\phi_2(\vec{x}) < 0)) \quad (7)$$

$$\Omega_3 = \text{where } ((\phi_1(\vec{x}) > 0) \text{ and } (\phi_2(\vec{x}) > 0)) \quad (8)$$

The three intensity values are called $\hat{\rho}_1$, $\hat{\rho}_2$ and $\hat{\rho}_3$, corresponding to regions Ω_1 , Ω_2 and Ω_3 . These values are single-valued constants and are simply the average value of ρ within the bounds of the region that it represents:

$$\hat{\rho}_1 = \langle \rho (\vec{x} \in \Omega_1) \rangle \quad (9)$$

$$\hat{\rho}_2 = \langle \rho (\vec{x} \in \Omega_2) \rangle \quad (10)$$

$$\hat{\rho}_3 = \langle \rho (\vec{x} \in \Omega_3) \rangle \quad (11)$$

where $\langle \dots \rangle$ indicates taking the mean of those datapoints. The overall piecewise constant approximation of the microstructure is calculated as:

$$\hat{\rho}(\vec{x}) = \begin{cases} \hat{\rho}_1 & \text{for } \vec{x} \in \Omega_1 \\ \hat{\rho}_2 & \text{for } \vec{x} \in \Omega_2 \\ \hat{\rho}_3 & \text{for } \vec{x} \in \Omega_3 \end{cases} \quad (12)$$

Since this piecewise constant reconstruction will be compared to the original intensity values, the two datasets should be as alike as possible in terms of interfacial width or diffuseness. The CT data collection and reconstruction process results in blurring of the interfaces; therefore, $\hat{\rho}$ must be smoothed to match it. The smoothing that is used for this is diffusion smoothing or Gaussian blurring and is performed by evolving the following diffusion equation:

$$\frac{d\hat{\rho}}{d\tau} = \frac{\partial^2 \hat{\rho}}{\partial x^2} + \frac{\partial^2 \hat{\rho}}{\partial y^2} + \frac{\partial^2 \hat{\rho}}{\partial z^2} \quad (13)$$

where τ is an artificial smoothing time. This equation is evolved for a predetermined amount of time that is related to the interfacial width in ρ . This step is only necessary if the edges in the data are blurred; if they are sharp, this step can be omitted. This smoothing is applied in 3D and not 4D because the CT data collection process is a series of sequential, 3D measurements so any blurring due to data collection would be strictly spatial and not temporal.

Iterations

Each iteration is comprised of a smoothing step and a accuracy-preserving step. The smoothing step is done using diffusion-like smoothing^b on the interface location arrays, ϕ_1 and ϕ_2 , by evolving the following equation:

$$\frac{d\phi_i}{d\tau} = D_x \frac{\partial^2 \phi_i}{\partial x^2} + D_y \frac{\partial^2 \phi_i}{\partial y^2} + D_z \frac{\partial^2 \phi_i}{\partial z^2} + D_t \frac{\partial^2 \phi_i}{\partial t^2} \quad (14)$$

where ϕ_i is either ϕ_1 or ϕ_2 , τ is again an artificial smoothing time, and the pre-factors D_x , D_y , D_z and D_t are the diffusivity-like terms that are adjusted to account for differing resolutions in space and time in the experiment. After smoothing, a SDF reinitialization procedure is then applied to the ϕ_1 and ϕ_2 arrays to ensure they maintain their SDF properties. This is the same reinitialization procedure as used in the initialization of the segmentation method.

The accuracy update starts by calculating the difference between the intensity values in the collected data and those of the piecewise constant approximation:

$$\Delta\rho(\vec{x}) = \rho(\vec{x}) - \hat{\rho}(\vec{x}) \quad (15)$$

This $\Delta\rho$ term is then smoothed to reduce the effects of noise. The smoothing equation that is defined in Equation 13 is used here as well because it conserves the total intensity of $\Delta\rho$ thus, if there is a large area of deviation between ρ and $\hat{\rho}$, it will be largely unaltered; however, if the area is much smaller, or if it is rapidly alternating between positive and negative values (as would be the case for an interface that is smooth in ϕ but has some roughness due to noise in ρ), the difference term will be negligible. Once $\Delta\rho$ is calculated and smoothed, it is used to update the interface location arrays as follows:

$$\phi_1(\vec{x}) = \phi_1(\vec{x}) + \beta \Delta\rho(\vec{x}) \quad (16)$$

$$\phi_2(\vec{x}) = \phi_2(\vec{x}) + \beta \Delta\rho(\vec{x}) \quad (17)$$

where β is an adjustable parameter. These equations are applied once per overall iteration. Since these smoothing and accuracy-preserving steps work together to ensure the interfaces do not become too rough or too inaccurate, their coefficients must be selected to work together. In this work, a value was chosen for β , and the amount of smoothing was selected to work well with that value.

The intensity values of the individual phases are recomputed the same as before using Equations 9, 10 and 11 and the piecewise constant approximation is recalculated using Equation 12.

Analysis tools

One of the advantages of using a segmentation method that results in a signed distance function, as opposed to a binary dataset, is the ease with which interfacial curvature and velocity can be calculated from the SDF. The interfacial normal velocity can be calculated using:

$$V = \frac{-1}{|\nabla\phi|} \frac{d\phi}{dt} \quad (18)$$

where ϕ is used to represent either ϕ_1 or ϕ_2 . The mean curvature (H) and Gaussian curvature (K) can be calculated directly from a signed distance function using the equations [11]:

$$H = \frac{\begin{pmatrix} \phi_x^2(\phi_{yy} + \phi_{zz}) + \\ \phi_y^2(\phi_{xx} + \phi_{zz}) + \\ \phi_z^2(\phi_{xx} + \phi_{yy}) - \\ 2\phi_{yz}\phi_y\phi_z - \\ 2\phi_{xz}\phi_x\phi_z - \\ 2\phi_{xy}\phi_x\phi_y \end{pmatrix}}{2|\nabla\phi|^3} \quad (19)$$

$$K = \frac{\begin{pmatrix} \phi_x^2(\phi_{yy}\phi_{zz} - \phi_{yz}^2) + \\ \phi_y^2(\phi_{xx}\phi_{zz} - \phi_{xz}^2) + \\ \phi_z^2(\phi_{xx}\phi_{yy} - \phi_{xy}^2) - \\ 2\phi_x\phi_y(\phi_{xy}\phi_{zz} - \phi_{xz}\phi_{yz}) - \\ 2\phi_x\phi_z(\phi_{xz}\phi_{yy} - \phi_{xy}\phi_{yz}) - \\ 2\phi_y\phi_z(\phi_{yz}\phi_{xx} - \phi_{xy}\phi_{xz}) \end{pmatrix}}{|\nabla\phi|^4} \quad (20)$$

where the subscripts indicate partial derivatives.

The signed distance functions provide values of V , H , and K at points both on the interfaces where $\phi = 0$ and for ϕ both positive and negative. Since the objective is to determine interfacial properties, it is necessary to determine how much each voxel contributes to the overall interfacial properties. This is done by using a weighting factor that behaves like the interfacial area of a voxel and is determined by using an interfacial delta function [10]:

$$\delta(\phi) = \begin{cases} 0 & \text{for } \phi < -\epsilon \\ \frac{1}{2\epsilon} \left(1 + \cos\left(\frac{\pi\phi}{\epsilon}\right) \right) & \text{for } -\epsilon \leq \phi \leq \epsilon \\ 0 & \text{for } \epsilon < \phi \end{cases} \quad (21)$$

where ϵ is a parameter to determine the width of the interface; a value of $\epsilon = 1.5$ is used on the recommendation of [10]. The delta function has units of inverse length and can be used as an area weighting term when used inside a volume integral or summation; for example, the total area-weighted curvature is $\sum_{\vec{x}} \delta(\phi(\vec{x}))H(\vec{x})\Delta V$, where $H(\vec{x})$ is the curvature of a voxel at location \vec{x} and ΔV is the volume per voxel.

Results

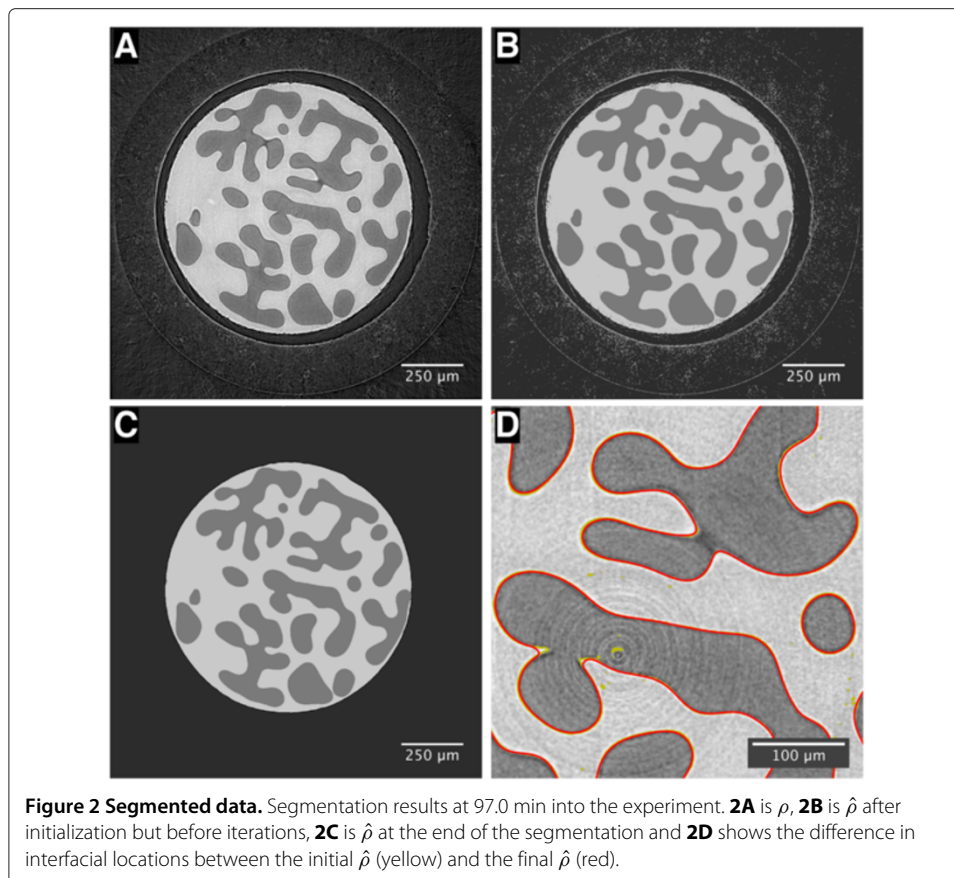
The optimal parameters in this segmentation technique will depend strongly on a variety of features, including the amount of noise in the data, the size scale of the features being segmented and the intensity range of the starting data. For the data used here, the average feature size is approximately 20 voxels at the first timestep and grows to approximately 40 voxels by the end of the experiment. The noise level in the data is relatively low and can be seen in Figure 1. The data in this figure has been scaled such that a value of 0 corresponds

to black and a value of 1.5 corresponds to white. Threshold values of $T_{1/2} = 0.4$ and $T_{2/3} = 1.0$ are used, resulting in initial intensity values of $\hat{\rho}_1 = 0.242$, $\hat{\rho}_2 = 0.669$ and $\hat{\rho}_3 = 1.184$. By the end of the segmentation, these values changed slightly to $\hat{\rho}_1 = 0.256$, $\hat{\rho}_2 = 0.724$ and $\hat{\rho}_3 = 1.182$.

For this data, smoothing was applied to $\hat{\rho}$ for 1.0 artificial time units to make the intensity profile through the interface similar to the data. This was then used to compute $\Delta\rho$, which was smoothed for 5.0 artificial time units to reduce the influence of noise. This value was chosen because it removes the majority of the noise in $\Delta\rho$ without significantly affecting the larger features.

The interface location arrays are updated using a value of $\beta = 0.5$; this value is the largest that could be used without numerical instabilities occurring. The value of β was chosen early in determining the optimal parameters and fixed, then the amount of smoothing was varied until the desired results were obtained. The arrays ϕ_1 and ϕ_2 were then smoothed with the 4D smoothing method for 0.5 time units, using pre-factors of $D_x = 1.0$, $D_y = 1.0$, $D_z = 1.0$ and $D_t = 0.5$. The lower temporal diffusivity was used because the structure is more finely discretized in time relative to its rate of evolution than it is in space relative to the average feature size. The individual intensity values, $\hat{\rho}_1$, $\hat{\rho}_2$ and $\hat{\rho}_3$, were updated at every iteration. A total of 25 iterations were used for this segmentation.

A 2D section of the original intensity values is shown in Figure 2A, along with the corresponding initial (2B) and final (2C) piecewise constant segmentations. The difference



in between the initial and final contours is shown in Figure 2D. The majority of the interfacial locations are very well maintained, although the highest curvature regions show some deviation from the initial contours. While this deviation of the interface is undesirable, this segmentation technique provides a better compromise between smoothness and accuracy than we have seen in 2D and 3D applications of this method. Furthermore, this iterative methodology has proven more effective at producing smooth and accurate interfacial locations than single-step methods such as 3D diffusion smoothing or motion by mean curvature. Figure 3 shows a comparison between a single-step method and the proposed method are shown. These two datasets show similar levels of interfacial displacement relative to the original data, but the proposed method results in much smoother interfaces and mean curvatures.

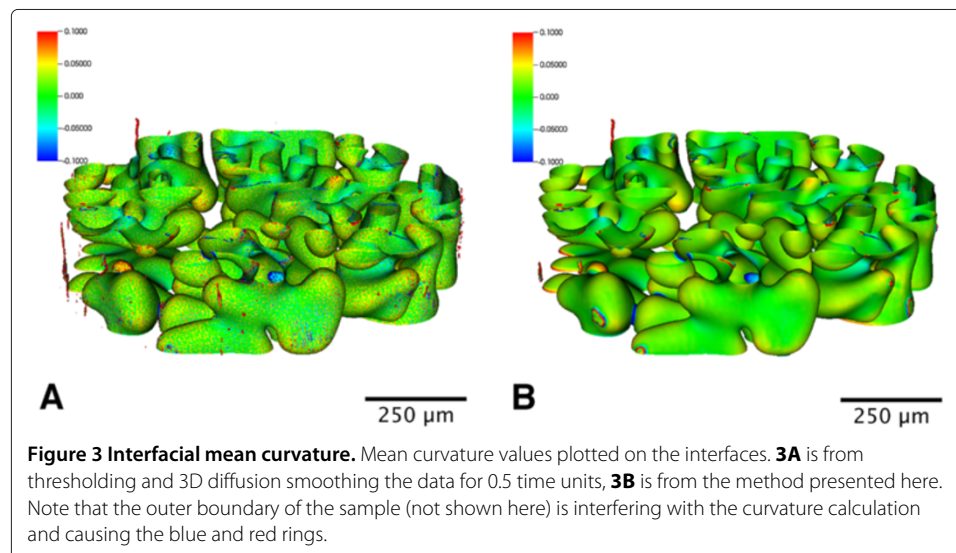
Discussion

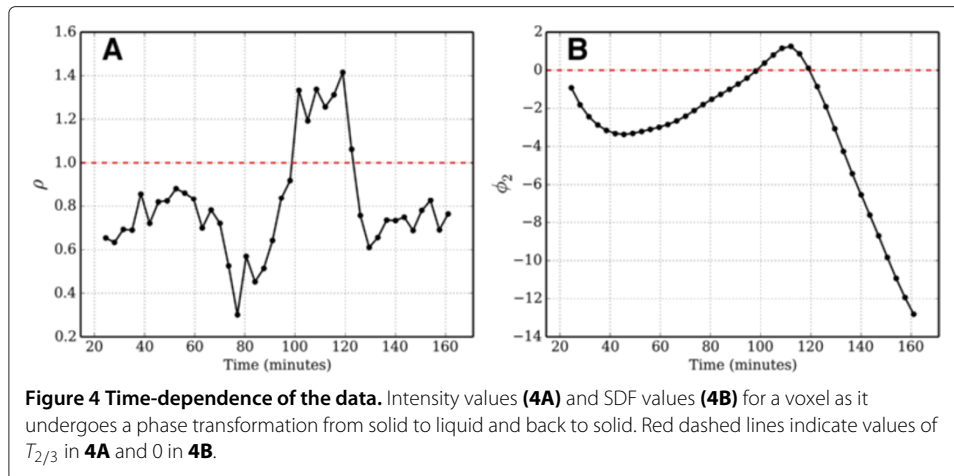
There are advantages to working with 4D data; however, there are also times when it makes more sense to treat time resolved data as individual, 3D datasets. One of these instances is when the temporal resolution is relatively low. If the sample changes significantly between timesteps, smoothing in time can lead to inaccuracies just like smoothing spatially under-resolved data can lead to problems. The other reason for possibly not using the full 4D dataset is the computational burden. This is primarily due to storing the full 4D dataset in memory since a dataset can be up to 10^{11} or 10^{12} voxels.

An advantage of smoothing the SDF that represents the interface, rather than the original intensity values, is that the SDF varies in a gradual and consistent way near the interface, instead of an abrupt change. This is highlighted in Figure 4, which shows both the intensity value and the SDF of a voxel as the phase changes from solid to liquid and back to solid again.

Conclusions

Registration and segmentation methods are presented that work quite well for processing 4D isothermal coarsening data of Al-Cu alloys. These methods work very well for determining smooth interface locations from noisy, 4D data.





This method has been presented for a particular case; however, it is quite easy to generalize to many other uses. For example, it could be used in 2D or 3D instead of 4D or for two or four or more phases instead of three. Also, some of the smoothing processes could be replaced with things that are better suited to different problems, such as motion by mean curvature or non-linear diffusion smoothing instead of the linear diffusion smoothing presented here.

Endnotes

^aThe number of neighbors for the 4D data is intentionally not reported because the differing spatial and temporal resolutions make this analysis less straightforward than the uniform 2D and 3D examples.

^bThis is not an actual diffusion equation due to the presence of the temporal component.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

JWG and PWV developed the methods described here, JWG performed the segmentations and JWG and PWV wrote this manuscript. Both authors read and approved the final manuscript.

Acknowledgements

This work was funded by the DOE under grant DE-FG02-99ER45782/A012 and by a DOE NNSA Stewardship Science Graduate Fellowship (grant DE-FC52-08NA28752). The authors would like to thank Dr. Julie L. Fife for the use of her data in the development of these methods and Dr. E. Begum Gulsoy for guidance with registration methods.

Received: 10 December 2013 Accepted: 24 March 2014

Published: 1 April 2014

References

- Schmidt S, Nielsen S, Gundlach C, Margulies L, Huang X, Jensen DJ (2004) Watching the growth of bulk grains during recrystallization of deformed metals. *Science* 305(5681): 229–232
- Rowenhorst DJ, Voorhees PW (2012) Measurement of interfacial evolution in three dimensions. *Annu Rev Mater Res* 42: 105–124
- Stock SR (2008) Recent advances in X-ray microtomography applied to materials. *Int Mater Rev* 53(3): 129–181
- Landis EN, Keane DT (2010) X-ray microtomography. *Mater Charact* 61(12): 1305–1316
- Pluim JPW, Maintz JBA, Viergever MA (2003) Mutual-information-based registration of medical images: a survey. *IEEE Trans Med Imaging* 22(8): 986–1004
- Gulsoy E, Simmons J, DeGraef M (2009) Application of joint histogram and mutual information to registration and data fusion problems in serial sectioning microstructure studies. *Scr Mater* 60(6): 381–384

7. Tsai A, Yezzi A, Willsky AS (2000) A curve evolution approach to smoothing and segmentation using the Mumford-Shah functional In: Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662). IEEE, New York, NY, pp 119–124
8. Yezzi A, Tsai A, Willsky A (2002) A fully global approach to image segmentation via coupled curve evolution equations. *J Vis Comm Image Represent* 13(1–2): 195–216
9. Chan TF, Vese LA (2001) A level set algorithm for minimizing the mumford-shah functional in image processing In: Proceedings. IEEE Workshop on Variational and Level Set Methods in Computer Vision, 2001. IEEE, New York, NY, pp 161–168
10. Osher S, Fedkiw R (2003) Level set methods and dynamic implicit surfaces. *Applied Mathematical Sciences*, vol. 153. Springer, New York
11. Goldman R (2005) Curvature formulas for implicit curves and surfaces. *Comput Aided Geomet Des* 22(7): 632–658

doi:10.1186/2193-9772-3-6

Cite this article as: Gibbs and Voorhees: Segmentation of four-dimensional, X-ray computed tomography data. *Integrating Materials and Manufacturing Innovation* 2014 **3**:6.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- ▶ Convenient online submission
- ▶ Rigorous peer review
- ▶ Immediate publication on acceptance
- ▶ Open access: articles freely available online
- ▶ High visibility within the field
- ▶ Retaining the copyright to your article

Submit your next manuscript at ▶ springeropen.com
