PNNL-26862

Pacific Northwest
NATIONAL LABORATORY
Proudly Operated by Battelle Since 1965

Detecting the Extent of Cellular Decomposition after Sub-Eutectoid Annealing in Rolled UMo Foils

July 2017

EJ Kautz
S Jana
A Devaraj
CA Lavender
LE Sweet
VV Joshi

U.S. DEPARTMENT OF ENERGY

Prepared for the U.S. Department of Energy
under Contract DE-AC05-76RL01830

PACIFIC NORTHWEST NATIONAL LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC05-76RL01830

Printed in the United States of America

Available to DOE and DOE contractors from the
Office of Scientific and Technical Information,
P.O. Box 62, Oak Ridge, TN 37831-0062;
ph: (865) 576-8401
fax: (865) 576-5728
email: reports@adonis.osti.gov

Available to the public from the National Technical Information Service
5301 Shawnee Rd., Alexandria, VA 22312
ph: (800) 553-NTIS (6847)
email: orders@ntis.gov <http://www.ntis.gov/about/form.aspx>
Online ordering: http://www.ntis.gov


This document was printed on recycled paper.
(8/2010)

PNNL-26862

Detecting the Extent of Cellular Decomposition after Sub-Eutectoid Annealing in Rolled UMo Foils

EJ Kautz
S Jana
A Devaraj
CA Lavender
LE Sweet
VV Joshi

July 2017

Prepared for
the U.S. Department of Energy
under Contract DE-AC05-76RL01830

Pacific Northwest National Laboratory
Richland, Washington 99352

Summary

Uranium alloyed with 10 weight percent (wt%) Molybdenum (U10Mo) is under development as a metallic nuclear fuel system in order to address the U.S. Department of Energy National Nuclear Security Administration's objective of converting all research reactors and radioisotope production facilities in the U.S. from using highly enriched uranium (HEU) fuels to low-enriched uranium (LEU). Replacement of HEU fuels is required in order to minimize risk of nuclear proliferation. During fuel fabrication, U10Mo is subjected to different time-temperature histories and mechanical deformation, and consequently the microstructure evolves from the as-cast condition to the final fuel foil.

Fuel microstructure has a direct influence on irradiation response in-pile. Quantitative description of U10Mo microstructures is therefore needed in order to relate microstructure to processing parameters, and ultimately material performance. A major goal in performing U10Mo microstructure characterization studies is to be able to consistently and accurately predict microstructure from processing parameters. Improved predictive capabilities is needed for selection of appropriate processing parameters that yield desired microstructure and thus material properties and performance.

This report presents an automated image processing approach to quantifying microstructure image data, specifically the extent of eutectoid (cellular) decomposition in rolled U-10Mo foils. An image processing approach is used here to be able to quantitatively describe microstructure image data in order to relate microstructure to processing parameters (time, temperature, deformation).

The goal of developing a more automated approach is to minimize bias introduced via the requirement for an expert to manually threshold image data, and more efficiently analyze a large amount of image data (i.e. hundreds of micrographs), thereby analyzing a larger sample area which would have been prohibitively time consuming if analyzed with manual methods alone.

Acknowledgments

This work was funded by the U.S. Department of Energy and the National Nuclear Security Administration under the Material Management and Minimization Reactor Conversion Program performed at Pacific Northwest National Laboratory (PNNL) under contract DE-AC05-76RL01830.

All materials used to generate images for work reported here were cast at the Y-12 Security Complex at Oak Ridge National Laboratory, and shipped to PNNL for characterization.

The authors would like to recognize the following individuals at PNNL: Shelly Carlson for her technical support in material handling and sample preparation, and Alan Schemer-Kohrn for his expertise in electron microscopy.

The authors also wish to acknowledge Professor Richard Radke (Rensselaer Polytechnic Institute, Electrical and Computer Systems Engineering Department) for his expertise in image processing and for developing course materials that were directly applicable to work reported here.

Acronyms and Abbreviations

BCC	body-centered cubic
BSE	backscattered electron
CDF	cumulative distribution function
CI	confidence interval
CLAHE	contrast-limited adaptive histogram equalization
GUI	graphical user interface
HEU	highly enriched uranium
LEU	low-enriched uranium
SEM	scanning electron microscopy
TTT	time-temperature-transformation
wt%	weight percent
U10Mo	uranium alloyed with 10 wt% molybdenum
UC	uranium carbide
UMo	uranium-molybdenum
XRD	x-ray diffraction

Contents

Figures

Tables

1.0 Introduction

1.1 Overview

Uranium alloyed with 10 wt% molybdenum (U10Mo) is a promising metallic nuclear fuel candidate that uses low-enriched uranium (LEU) for replacement of highly enriched uranium (HEU) fuels currently used in U.S. research reactors and radioisotope production facilities.

U10Mo experiences a eutectoid reaction at approximately 565°C, where the γ-U matrix phase forms α-U and U_2Mo (γ′) as decomposition products, as follows:

$$\gamma\text{-U} \rightarrow \alpha\text{-U} + U_2Mo\ (\gamma')$$

Here, this reaction is referred to as a eutectoid or cellular decomposition. The corresponding U-Mo phase diagram is provided below, for reference:

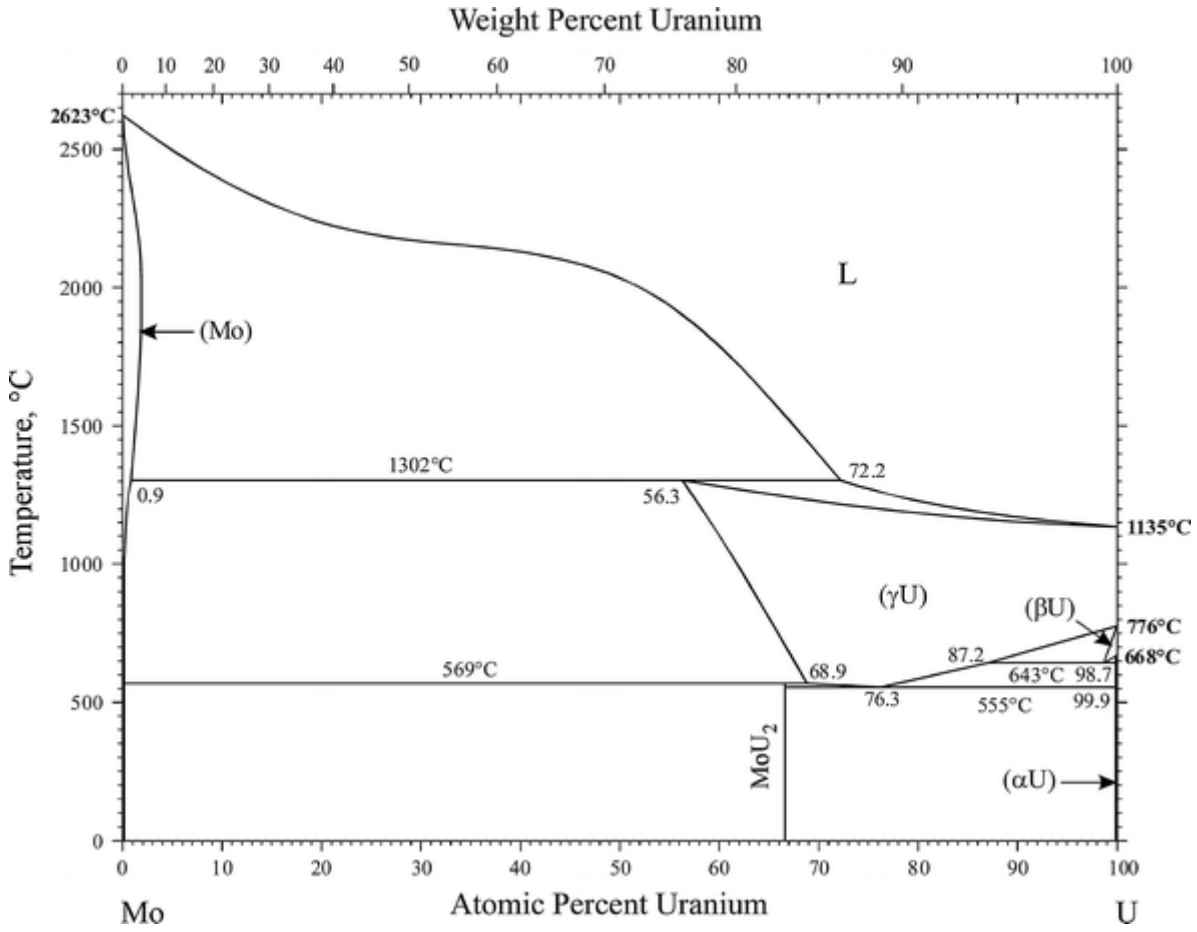

Figure 1. UMo equilibrium phase diagram reproduced from Okamoto (2012).

The γ-U matrix phase is the stable allotrope of U with body-centered cubic (BCC) structure at elevated temperature (above 565°C). It is possible to stabilize the high temperature BCC γ-U phase at room temperature as a metastable single phase by alloying with Mo. A minimum of 6–7 wt% Mo combined with slow furnace cooling (approximate cooling rate of 2°C/minute) is required for γ-U stabilization.

This slow cooling rate is needed to allow the eutectoid reaction (decomposition) to proceed. The reaction kinetics of this eutectoid decomposition slow further with increasing Mo content above 6–7 wt%. γ-U, or γ-UMo (when U is alloyed with Mo), is the desirable matrix phase for nuclear fuels due to its processing characteristics and irradiation behavior, whereas the α-U phase is undesirable because it shows anisotropic swelling during irradiation (ASM 1965, Sinha 2010, Hofman 1994). The γ-UMo phase is also more desirable than α-U due to its favorable processing characteristics, and corrosion resistance in water, and water and steam environments (IAEA 2003).

In order to determine appropriate processing parameters for development of a desirable microstructure that yields acceptable properties for a LEU metallic fuel, a reliable characterization of processed U10Mo microstructures is needed. This report presents work on characterizing U10Mo microstructures in order to quantify the extent of eutectoid decomposition so that phase transformation kinetics can be related to processing parameters.

This report is organized as follows:

Background information related to phase transformations in U10Mo is presented, followed by a discussion of previous characterization work done using image analysis and x-ray diffraction (XRD). Next, an image analysis approach (for characterization of image data obtained using scanning electron microscopy [SEM] image data) using the Python programming language is presented, including an overview of methods used, and hardware and software specifications.

In the Methods section of this report, each method used in image processing is described, and any key equations and descriptive schematics are presented in order to clearly convey what exactly is being done to process each image. Results of image processing (output images and area fraction calculated) are presented, followed by limitations of the existing method using Python. The report ends with a conclusion section, and recommendations for future work.

1.2 Background

During fabrication of uranium-molybdenum (UMo) fuel foils, the UMo alloy is frequently subjected to various processing steps (including hot rolling and hot isostatic pressing) that take place within the temperature range of 400–600°C for various times. Thus, there is always a possibility that the metastable γ-UMo phase will decompose and form the undesirable α-U phase during exposure to elevated temperatures.

Time-temperature-transformation (TTT) diagrams developed for UMo alloys with varying Mo content can provide a general guideline about what temperature and time combinations should be used during fuel fabrication in order to avoid α-U formation. However, our recent work published in Jana (2017) on phase transformation behavior in a cast and homogenized U10Mo alloy has shown that the available literature varies widely based on source.

The eutectoid transformation can initiate within one hour of heating U10Mo at 500°C and the major mode of transformation can change (e.g., discontinuous or cellular reaction vs. continuous precipitation) depending on the processing temperature (Jana, 2017). In addition, Mo homogeneity and the presence of

impurity elements can play a major role in determining phase transformation behavior. Thus, a recommended best practice is to identify transformation products from a fabricated UMo foil, rather than relying solely on earlier TTT data to determine what phases exist in the alloy (Jana, 2017).

Various characterization techniques can be employed to detect phase transformation in U10Mo alloys, including optical and scanning electron microscopy, XRD, dilatometry, resistivity measurement, and hardness testing. Our recent work published in Jana (2017) has shown that microscopy of samples prepared using traditional metallographic processes is the most sensitive technique that can help in detecting the extent of eutectoid decomposition in U10Mo, particularly during early stages when the estimated volume fraction of the transformation (based on area fraction) is less than 10%. Furthermore, metallography methods and imaging of prepared samples are well established techniques and reproducible results can be obtained fairly quickly, which is highly desirable for analysis of microstructures in an industrial/manufacturing environment.

For analysis of transformation products via microscopy, optical or SEM images need to be obtained at a magnification of 500-1000x from different regions of the sample to use the image processing approach described here. If a sample has undergone the eutectoid transformation, there is a distinct difference in grayscale intensity between the parent γ-UMo phase and transformation products, which appear as α and γ lamellae. By using this recommended magnification range (500-1000x), and by imaging various regions of the sample, the collected data can be assumed to accurately represent the entire volume of material from which the metallographic sample was sectioned.

The extent of phase transformation can be quantified from image data by computing the area fraction based on this observed grayscale difference between phases, where area fraction serves as a proxy for volume fraction. Per the fuel specification, the process shall yield U Mo alloy with at least 90 wt% gamma phase (i.e., having the body centered cubic crystal structure).

Previous work on determining area fraction from micrographs has been performed manually using ImageJ to threshold input images. This manual method using ImageJ has the following key disadvantages:

1. Operator bias is introduced into the thresholding process, since each image is thresholded separately.
2. Manual thresholding is a time-consuming process for multiple images.

In order to automate image processing and calculating area fraction, the OpenCV and scikit-image open-sources libraries were used with Python (version 2.7.13).

The goal of automating this process is to ultimately deliver a tool for engineers on a manufacturing floor to be able to quickly and reliably determine whether the microstructure of U10Mo is acceptable. Here, an “acceptable” microstructure is defined as having an area fraction of eutectoid decomposition less than 0.10 (10%).

1.3 Prior Work on Characterization of UMo Microstructures via Image Analysis and X-Ray Diffraction

As an example, a set of backscattered electron (BSE)-SEM images at a magnification of 250× is shown in Figure 1. The images were obtained from rolled/annealed U10Mo foils that were subjected to subeutectoid heat treatments for different durations. In Figure 1, darker regions represent transformation products (regions of α and γ lamellae). Area fraction of transformed regions was measured manually in ImageJ, and is indicated below each individual image. Change in the distribution of the darker phase is apparent in Figure 1, as total transformation area changes from 6 to 26%.

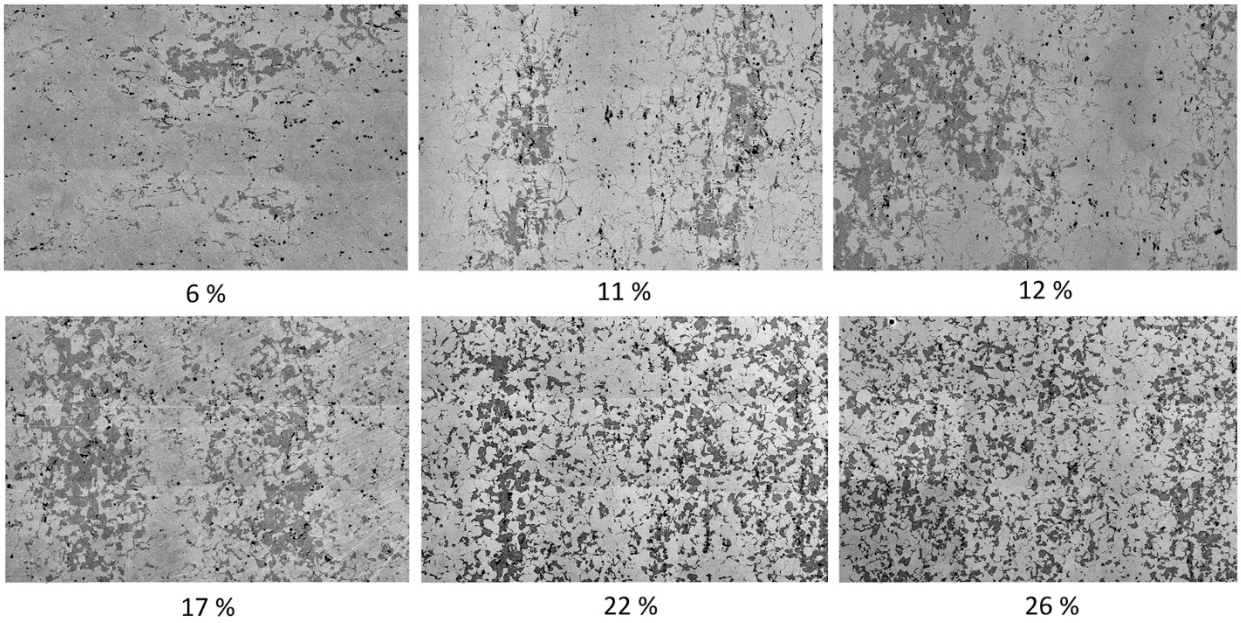

Figure 1. A set of BSE-SEM Images at a Magnification of 250× from rolled and annealed U-10Mo Foils after Sub-eutectoid heat treatment for different durations ¶Percentages under each image indicate the percentage of area transformed, which was measured by manually thresholding images using ImageJ software. Darker regions represent the transformed region.

A comparison was made to evaluate the effectiveness of metallography vs. XRD techniques in determining the extent of phase transformation in U10Mo alloy. Figure 2 shows an image of a heat treated U10Mo sample at a magnification of 250×, where the degree of transformation was found via image analysis to be ~7%. An XRD profile from the same sample is shown in Figure 3. Rietveld refinement of the XRD profile indicates presence of γ-UMo phase and a small fraction of UC phase. Presence of phase transformation products in heat treated U10Mo alloy, therefore, remains undetected with the XRD method until the area fraction reaches a critical value. The rationale for this is subsequently discussed.

As has been mentioned earlier, cellular decomposition is the major mode of transformation in U10Mo alloy, especially at 500°C or higher. Cellular reaction results in formation of a lamellar microstructure that primarily nucleates along the prior γ-UMo grain boundaries and grows inward into the γ-UMo grain interior with time. The lamellar microstructure is made of thin lamellae of α-U, and the interlamellar region is γ-UMo phase but with a higher Mo content than the parent γ-U10Mo alloy. At 250× magnification, the lamellar nature of the transformed regions is not discernable, and they appear as a darker contrast phase. Such contrast difference helps distinguish the transformed regions through image analysis techniques. On the other hand, the volume fraction of α-U present within the lamellar microstructure is very low. Use of the lever rule can help in finding the volume fraction of α-U within the lamellar microstructure in a conservative manner. However, the actual volume fraction of α-U is less than that predicted by the lever rule because of cellular decomposition. Therefore, XRD is unable to detect such a small volume fraction of α-U, especially at the early stage of transformation. Detecting the extent of phase transformation products at an early stage is of paramount importance during UMo fuel fabrication.

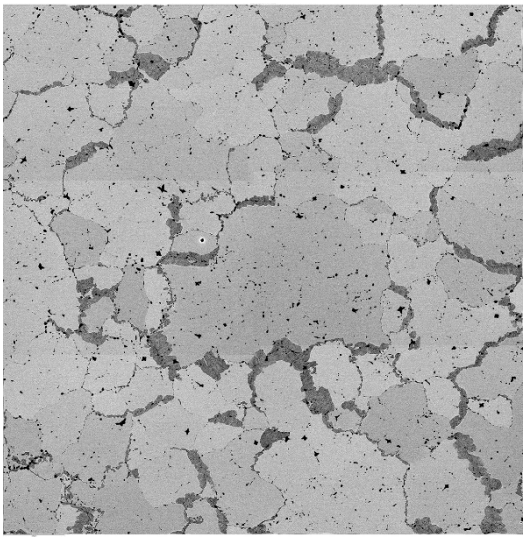

Figure 2. SEM-BSE Image at 250× of a heat treated U10Mo sample¶The percentage of transformed area was calculated to be ~7% using ImageJ software.

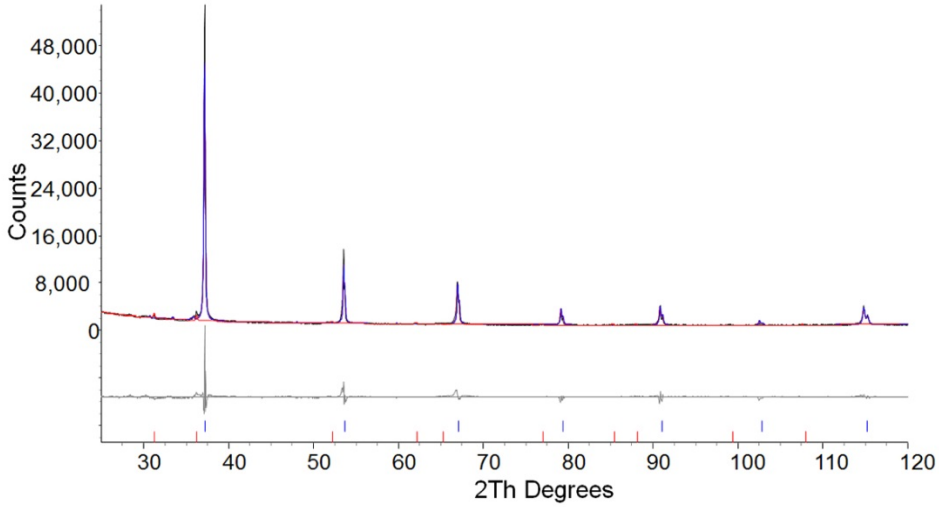

Figure 3. XRD profile of the heat treated U10Mo sample shown in Figure 2¶Rietveld refinement indicates presence of γ-UMo phase and a small fraction of UC. Phase transformation through cellular reaction remains undetected through XRD.

In the above example, the image analysis approach was able to determine the phase transformation where there was little phase transformation. However, it is to be noted that conventional metallography methods will not be able to determine phase transformation in U10Mo alloy when continuous precipitation is the major mode of transformation, which is the case at 400°C or lower. We have found that hardness measurement of the γ-UMo grain interior can help in determining the onset of continuous precipitation in these cases. One such example is shown in Figure 4. The hardness of the γ-UMo matrix increases more rapidly at 400°C than at 500°C; this is because continuous precipitation dominates at 400°C and results in nanoscale precipitates within the γ-UMo grain interior, which cause the rapid hardness increase.

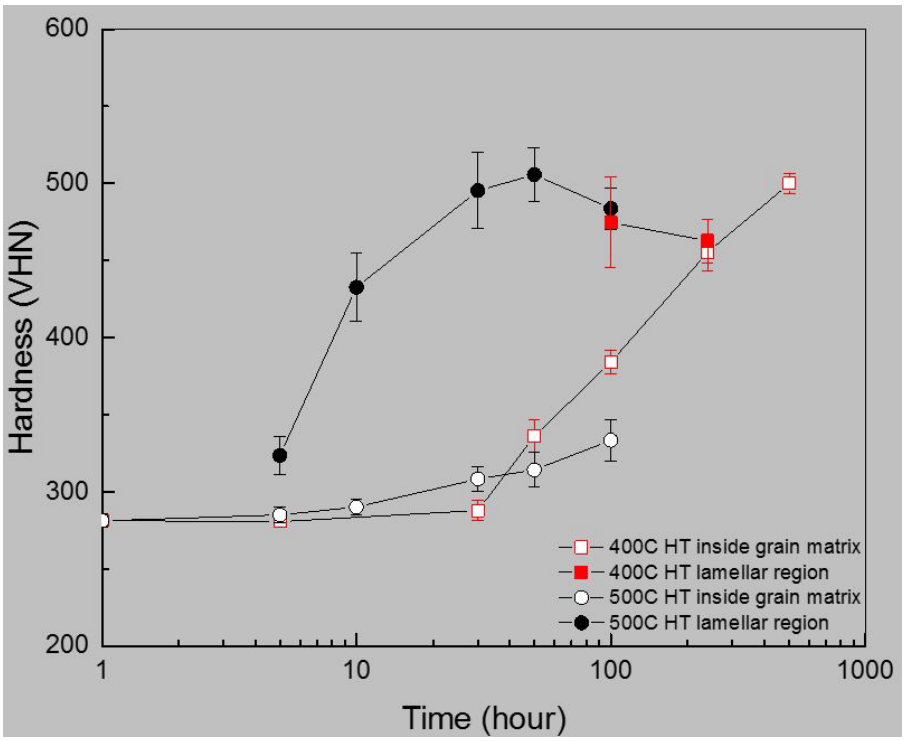

Figure 4. Hardness of a Heat Treated U10Mo Alloy Sample

In summary, previous work indicates that metallography and microscopy combined with hardness measurements are time efficient and effective methods for determining the extent of cellular decomposition in U10Mo.

2.0 Methods

The input requirements for images to be processed using the image analysis Python script discussed here (and provided in the appendix) include the following:

1. Image type: grayscale (1 color channel)
2. Image acquired using SEM
3. Magnifications: 500–1000×
4. Orientation: planar or longitudinal cross-sectional view
5. File Type: .tif
6. Image size: 1024 pixels (height) × 1280 pixels (width)

The output of the script is two values of area fraction, saved in a comma-separated values (.csv) file.

The general approach applied to image processing was first to crop all BSE images to remove the scale bar, then improve image contrast, perform thresholding, and fill in regions of the image that corresponded to the cellular precipitation/decomposition transformation. In this work, carbides were also included in the final calculated percent transformed in order to provide a conservative estimate. The methods used in this approach (shown schematically in Figure 5) are described in more detail in this section.

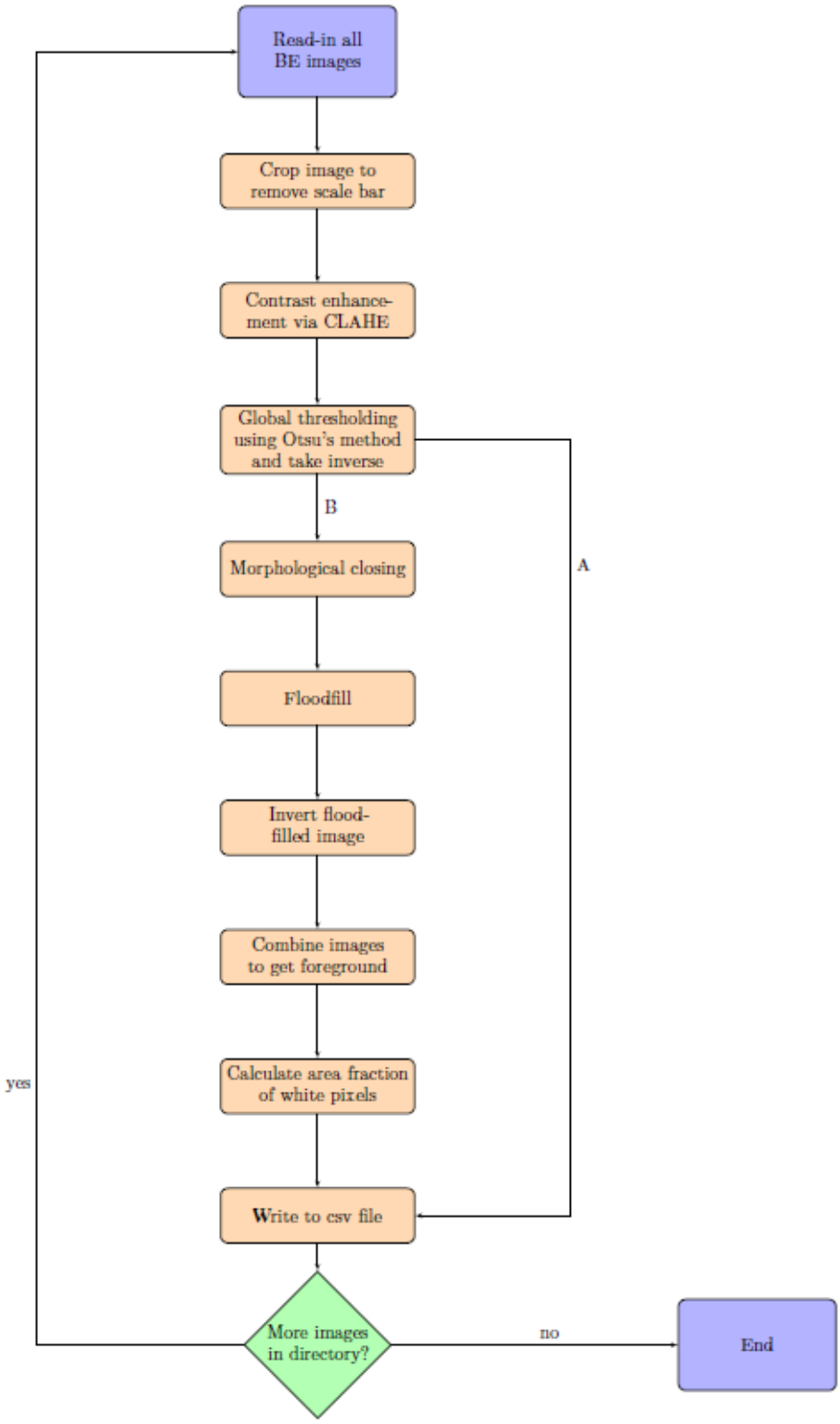

Figure 5. Flow Chart that summarizes the image processing script written in Python¶Here, CLAHE refers to contrast-limited adaptive histogram equalization. Routines A and B are labeled in the flowchart above as 'A' and 'B', respectively.

2.1 Contrast-Limited Adaptive Histogram Equalization

Contrast-limited adaptive histogram equalization (CLAHE) is a specific type of histogram equalization, which is a method that improves image contrast by stretching the intensity range of grayscale values in an image.

Histogram equalization requires mapping the starting image histogram (count versus grayscale intensity value) to another distribution that is wider and more uniform across the grayscale range of 0–255, hence the term "equalization." This concept of stretching an image histogram is shown schematically in Figure 6.

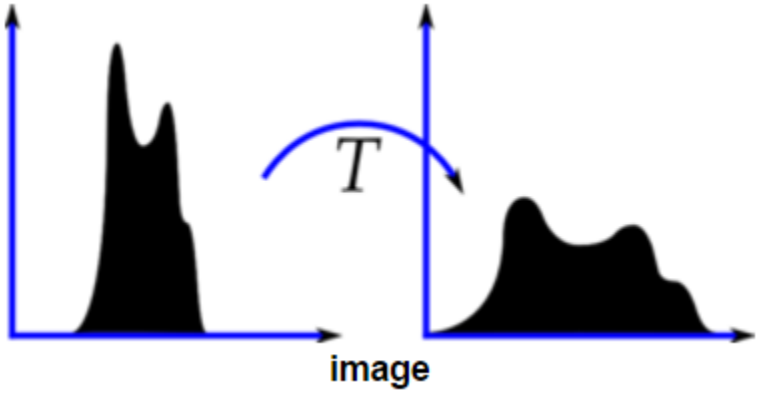

Figure 6. Schematic of Histogram Equalization¶from OpenCV documentation (OpenCV 2015). Here, "T" in the above schematic represents the function required for stretching the original image histogram to perform histogram equalization

Given an input image, $\{x\}$, with n_i as the number of occurrences of grayscale intensity value, i, the probability of a pixel having intensity $x = i$ is as follows:

$$p_x(i) = p(x = i) = \frac{n_i}{n}, 0 \leq i > L \quad (1)$$

where L is the total number of grayscale intensity values in an image (256 if all values 0 to 255 are represented), n is the total number of pixels in an image, n_i is the total number of pixels with intensity i, and $p_x(i)$ is the image histogram normalized to [0 1].

The cumulative distribution function (CDF) of the image can be expressed as

$$cdf_x(i) = \sum_{j=0}^{i} p_x(j) \quad (2)$$

In histogram equalization, a transformation function must be determined in order to produce a new image that has a flat histogram. This concept can be mathematically expressed as below, where y is the new image, x is the original image, and T is the transformation function:

$$y = T(x) \quad (3)$$

In order to flatten the image histogram, the new image must have a flattened CDF across the range of grayscale intensity values (0–255):

$$cdf_x(i) = iK \tag{4}$$

where K is a constant.

The transformation that needs to be performed is as follows, where k is in the range of [0 L]:

$$cdf_y(y') = cdf_y(T(k)) = cdf_x(k) \tag{5}$$

To map grayscale intensity values back to the original range, the following transformation is applied:

$$y' = y(\max\{x\} - \min\{x\}) + \min\{x\} \tag{6}$$

where y' is the new image and y is the original image.

The result is that the number of pixels is more evenly distributed through the intensity range.

In this work, CLAHE was performed. Adaptive histogram equalization means that histogram equalization was performed locally as opposed to globally (the entire image), in order to account for illumination variations across the image. The image was split into blocks of pixels, and histogram equalization was done for each block. Block size is a set parameter. For the purposes of this work, the image was split into nine blocks, and histogram equalization was performed on each block separately to yield the processed image that was then used for thresholding.

Contrast limiting is also applied in order to minimize any noise amplification that could arise due to the histogram equalization process in each block. When adaptive histogram equalization is performed in each block, the histogram is confined to that given block, and if there is noise in that block, histogram equalization could amplify this noise. CLAHE limits amplification of noise by clipping the histogram at a value before computing the CDF for each block. This clipping limits the slope of the CDF and consequently the transformation function applied to the block histogram. Regions of the histogram that exceed the clip limit are redistributed equally among histogram bins. The value at which the histogram is clipped is a parameter called the clip limit, and is set in the Python script.

2.2 Thresholding via Otsu's Method

Global thresholding was used in this work in order to reduce the grayscale image to a binary (black and white) image. This thresholding algorithm assumes the image has a bimodal histogram, and then calculates the optimal threshold value separating the two classes of pixel distributions that minimizes the intra-class variance (maximizes inter-class variance). This concept is illustrated in Figure 7.

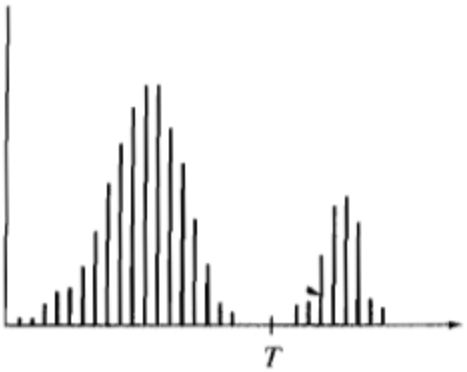

Figure 7. Schematic of an image histogram that is bimodal, where the y-axis is count and the x-axis is grayscale intensity value. In Otsu's Method of Thresholding¶the image is assumed to have a bimodal distribution of grayscale values, where the threshold value is T. Schematic is reproduced from Gonzalez and Woods (2018).

Otsu's method (Otsu 1979) is based on the simple concept of finding the grayscale value that minimizes the weighted within-class (intra-class) variance, $\sigma_w^2(t)$, and selecting this as the threshold value. Otsu's method operates on the image histogram (grayscale, single color channel). This method can be applied globally (to the entire image), or locally (to blocks of pixels, similar to what was done in CLAHE, previously). In this work, global thresholding after CLAHE yields acceptable results, so local thresholding was not applied in this case. However, local thresholding using Otsu's method could improve thresholding results by accounting for varying illumination across an image that causes the contrast in one block to vary from the contrast in a different block.

Mathematically, Otsu's method can be described by Equations (7)–(12) below. The goal is to calculate minimum weighted intra-class variance $\sigma_w^2(t)$ (Gonzalez and Woods, 2008).

$$q_1(t) = \sum_{i=1}^{t} P(i) \tag{7}$$

$$q_2(t) = \sum_{i=t+1}^{I} P(i) \tag{8}$$

$$\mu_1(t) = \sum_{i=1}^{t} \frac{iP(i)}{q_1(t)} \tag{9}$$

$$\sigma_1^2(t) = \sum_{i=1}^{t} [i - \mu_1(t)]^2 \frac{P(i)}{q_1(t)} \tag{10}$$

$$\sigma_2^2(t) = \sum_{i=t+1}^{I} [i - \mu_2(t)]^2 \frac{P(i)}{q_2(t)} \tag{11}$$

$$\sigma_w^2(t) = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t) \tag{12}$$

where

$q_1(t)$ is the class 1 probability
$q_2(t)$ is the class 2 probability
$\mu_1(t)$ is the class 1 mean
$\mu_2(t)$ is the class 2 mean
$\sigma_1^2(t)$ is the class 1 variance
$\sigma_2^2(t)$ is the class 2 variance
$\sigma_w^2(t)$ is the weighted intra-class variance

$P(i)$ is the probability of grayscale value i
t is the maximum grayscale value of class 1
I is the maximum grayscale value of class 2

2.3 Morphological Transformations

Morphological image processing is a term used to describe nonlinear operations related to the shape or morphology of features in image. Morphological operations are best suited for operating on binary images. Morphological techniques, such as erosion and dilation, probe an image using a small pixel template (a small matrix of ones and zeros) called a structuring element or kernel. The size and shape of the structuring element can be varied based on desired results. Possible shapes include rectangles, crosses, and disks.

A morphological operation on a binary image can be used to either “fatten” an image (connect components, referred to as dilation), or break thin connecting lines in an image (called erosion).

Erosion removes the boundaries of a foreground object, where the foreground object is white and the background is black (in a binary image).

The structuring element slides across the image, and a pixel in the original binary image (which is either a 1 or 0) will be 1 if all pixels under the structuring element are 1. The result is that pixels near the boundary of the structuring element will be discarded (depending on size and shape of structuring element).

Dilation has the opposite effect of erosion. Dilation is performed in the same manner as erosion, in that a structuring element slides over a binary image, except here, if there is any overlap between the structuring element and a pixel that is 1 in the original image, all pixels in the original image that coincide with the structuring element become a 1. The result is that objects in the image are “fattened” (the area of objects in an image increases).

A simple example of erosion and dilation are provided in OpenCV (2015) and is reproduced in Figure 8 in order to illustrate these concepts applied in this work.

(a) (b) (c)

Figure 8. Simple Illustration of Morphological Transformations (Erosion and Dilation) Used in This Work¶reproduced from OpenCV (2015), where (a) is the original image, (b) is the original image after erosion, and (c) is the original image after dilation.

2.4 Floodfill

The floodfill operation operates on a single-channel (grayscale) image, and fills in gaps in an image within connected components.

In this work, floodfilling was done in order to fill in space between lamellae that were thresholded in order to estimate total area of transformation, as opposed to just the area associated with the thresholded lamellar phase (and other thresholded objects, such as carbides).

2.5 Hardware and Software Specifications

For this work, a Dell laptop computer was used. The Dell laptop has an Intel Core i5-3340M CPU at 2.70 GHz, and is running a Windows 7 operating system with 8 GB of RAM.

With regard to software, Python version 2.7.13 was used in this work. The OpenCV library (version 3.2) interface for Python was installed and used for image processing. Other important libraries used include matplotlib, skimage, numpy, and csv.

2.6 Parameters

Key parameters in the image processing Python script for this work are listed in Table 1:

Table 1. Parameters used in Segmentation of SEM Images at 500× Magnification using Python

	Parameter	Value – SEM
Contrast-limited adaptive histogram equalization	tileGridSize	100 × 100
cv2.createCLAHE	clipLimit	1.0
Thresholding (global)	minValue	125
cv2.threshold	maxValue	255
Morphological Transformations	Structural element shape and size	Ellipse 3 × 3
cv2.dilate	Number of iterations (for both dilation	
cv2.erode	and erosion)	8
Floodfill		2 pixels greater
cv2.floodFill	mask size	than image size

3.0 Results and Discussion

An example of output images corresponding to image processing steps described in the Methods section are provided in Figure 9. The same steps were applied to a directory of images that were all planar cross-sectional views and to a separate directory of only longitudinal cross-sectional views. Due to the fact that 16 images were processed as part of this work, it is not feasible to include all intermediate output images here (80 total, including original images). Results, shown in Table 2, appear similar for all 16 images analyzed.

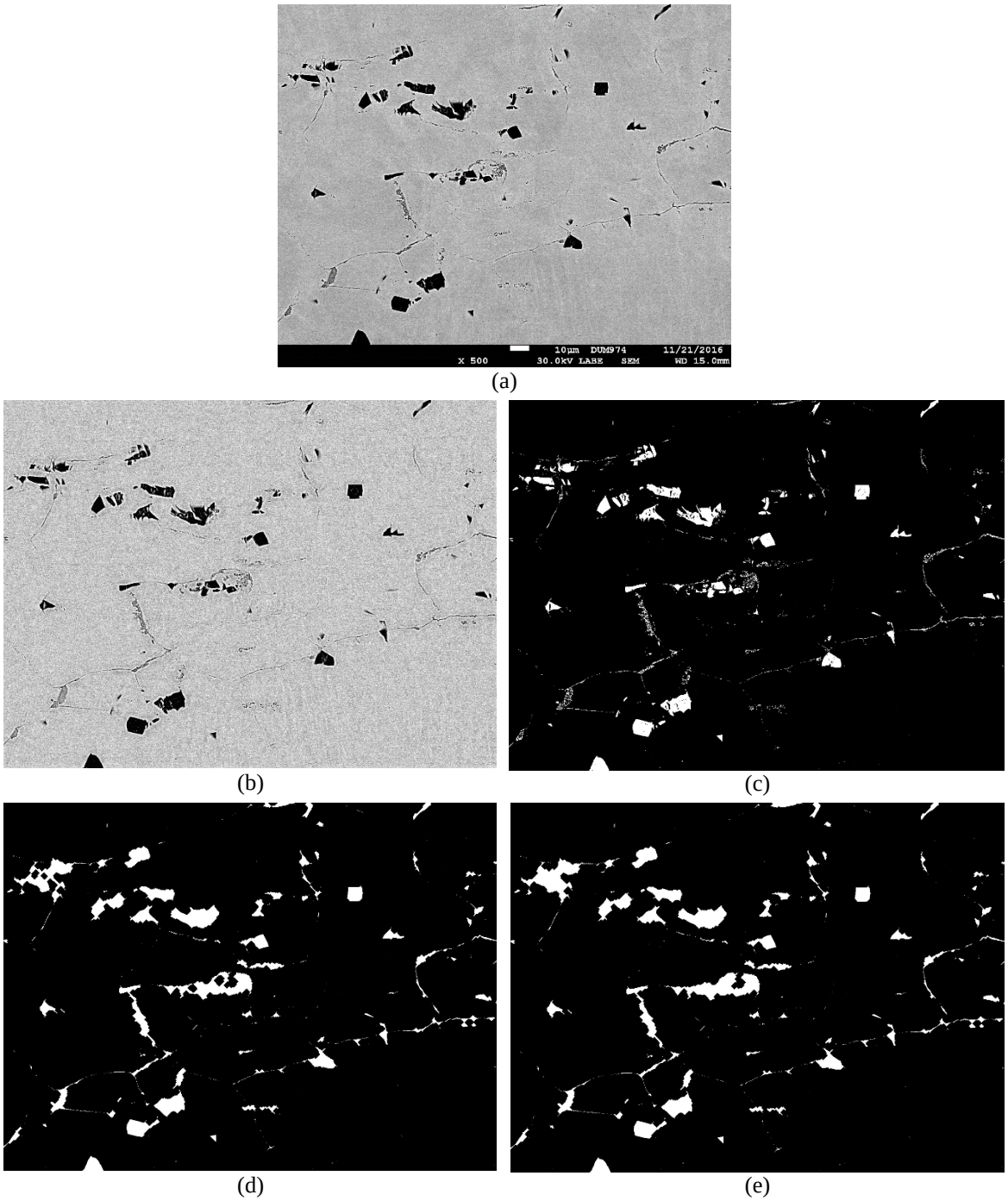

Figure 9. Series of Images that Illustrate the Results of Each Major Step in the Image Processing Script Used to Estimate Area Fraction of the Transformed Region¶The image processed to generate the above output is DUM974 image number 001, taken at 500× magnification using a 30 keV accelerating voltage and the BSE detector. This image is a planar cross-sectional view. Here, (a) is the original image, (b) after CLAHE, (c) after global thresholding using Otsu's method, (d) after erosion and dilation, and (e) after floodfill. The same steps were applied to all images processed; therefore DUM974 image 001 (above) serves as an example.

Table 2. Results from Image Processing using Python¶All values below are reported as percent area transformed (out of a total image area of 100%).

Sample ID	Cross-Sectional View	Magnification	Area Transformed (%)		
			Routine A	Routine B	Difference
DUM975	Longitudinal	500	1.75	2.96	1.21
			1	1.83	0.05
			0.36	0.79	0.36
			0.96	1.72	0.73
			0.96	0.88	0.52
			1.75	2.76	0.07
			0.67	1.98	0.56
			0.54	4.82	1.28
			1.14	4.82	1.28
			0.91	2.69	0.94
			0.6	2.1	1.16
DUM974	Planar	500	2.91	4.8	1.89
		1000	3.7	6.28	2.58
		500	1.25	2.29	1.04
		250	2.22	4.3	2.08
		500	2.56	5.31	2.75

In addition to processing images and estimating area fraction of the cellular decomposition transformation, statistical analysis to determine the number of images needed was performed. The question to answer was, “how many images are needed per specimen in order to best estimate the area fraction of the cellular decomposition reaction?” This question could be answered by plotting sample size, *N*, (number of images) versus confidence interval width for a 95% confidence interval. The goal in plotting this data was to find where on this plot confidence interval width is minimized for a reasonable number of samples (where increasing the number of samples beyond this number, *N*, only marginally decreases interval width).

Table 3 reports how confidence interval width varies with sample size (*N*), the number of images. Results reported in Table 3 are plotted in Figure 10.

Table 3. Results from Statistical Analysis of Data Presented in Table 2¶Confidence interval (CI) bounds and width are for a 95% CI. The data listed in this table is plotted in Figure 10.

Sample Size, *N*	CI Lower Bound	CI Upper Bound	CI Width
10	1.5236	3.4491	1.9255
15	1.7410	3.2318	1.4908
20	1.8565	3.1163	1.2598
25	1.9308	3.0419	1.1112
30	1.9839	2.9889	1.0050
41	2.0616	2.9112	0.8496
51	2.1077	2.8650	0.7572

A sample size of 20 is an optimal number of images for analysis, yielding a confidence interval width of 1.26. However, since obtaining 20 images per metallographic mount is very time consuming, a sample size of 10 or greater is acceptable: for 10 images the 95% confidence interval width is 1.93. The significance of performing this preliminary statistical analysis was to be able to state that the mean area fraction of the eutectoid decomposition for a 95% confidence interval would be within a certain range (1.26% for 20 images or 1.93% for 10 images).

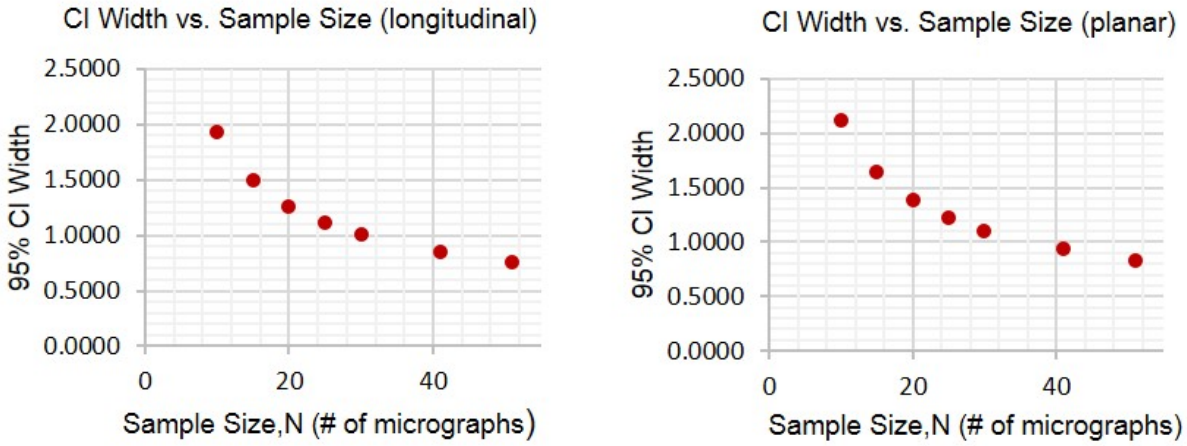

Figure 10. 95% Confidence Interval (CI) Width versus Sample Size for Percent Transformed for Longitudinal and Planar Cross-Sectional Views¶The percentage-transformed data from Routine B was used to calculate CI lower and upper bounds.

4.0 Conclusions

The following conclusions can be made based on results from automating image processing of U10Mo micrographs:

1. Image processing work previously performed manually using ImageJ can be successfully automated and improved using open-source libraries in Python (OpenCV and scikit image). The current image processing method in Python is sufficient for providing a conservative estimate of volume fraction of the eutectoid decomposition (transformation), in order to relate to processing parameters.

2. The use of CLAHE, thresholding, and flood-filling methods provide a conservative estimate of area fraction (which serves as a proxy for volume fraction) of the eutectoid decomposition. The result is viewed as conservative because carbides and silicides are included in the area fraction estimate

3. Area fractions calculated from planar cross sections differ slightly from area fractions calculated from longitudinal cross sections, which may indicate the eutectoid decomposition transformation is proceeding with some directionality.

5.0 Limitations

Limitations of the current image processing script are listed below:

1. A specific range of magnifications is required (500-1000x).
2. No graphical user interface (GUI) has been created, so the script is run using the Anaconda Prompt terminal. This is identified as a limitation because it is not as straightforward as a GUI would be for users who are not familiar with Python.

6.0 Recommended Future Work

Currently, the image processing script (written in Python) provides a conservative estimate for the area fraction (which serves as a proxy for volume fraction) of the transformed region of the U10Mo matrix.

Below, tasks to explore in future work (and the corresponding rationale) are listed:

1. Remove carbides and silicides from the segmentation process, in order to be less conservative and improve accuracy of calculated area fraction of the eutectoid decomposition
2. Segmentation of a higher magnification image (20,000×, for example) in order to estimate area fraction of α-U in the transformed region. Determining the fraction of the transformed region that is just α-U (which is the phase known to be detrimental to mechanical stability of U10Mo) could be used to improves accuracy of calculated area fraction of the eutectoid decomposition (because the current calculated area fraction is conservative).
3. Analysis of more images of planar versus longitudinal cross-sectional views in order to investigate whether transformation is proceeding with some directionality. Additional images need to be analyzed in order to draw conclusions based on statistics.
4. Experiment with additional image processing methods (such as low pass filtering) to determine if image quality prior to thresholding could be improved. Low pass filtering could be applied to remove background noise in raw image data, and ultimately improve thresholding results and accuracy of calculated area fraction.
5. Apply the current Python script to automated image analysis of more images for construction of TTT diagrams.

7.0 References

ASM International, Properties and Selection: Non Ferrous Alloys and Special Purpose Materials, vol. 2, 1990, pp. 1965.

Development status of metallic, dispersion and non-oxide advanced and alternative fuels for power and research reactors, IAEA-TECDOC-1374, Vienna, 2003, pp. 49.

G.L. Hofman, L.C. Walters, Materials science and technology, in: B.R.T. Frost (Ed.), Nuclear Materials, vol. 10A, VCH Publishers, N.Y., 1994.

Gonzalez, Rafael C., and Richard E. 1954- Woods. *Digital Image Processing*. 3rd ed. Upper Saddle River, N.J.: Prentice Hall, 2008.

Okamoto, H. J. Phase Equilib. Diffus. (2012) 33: 497. doi:10.1007/s11669-012-0095-z

OpenCV – Open Source Computer Vision. 2015. *OpenCV-Python Tutorials.* Accessed July 18, 2017, at http://docs.opencv.org/3.1.0/d6/d00/tutorial_py_root.html.

Otsu N. 1979. "A Threshold Selection Method from Gray-Level Histograms." *IEEE Transactions on Systems, Man, and Cybernetics* 9(1)62–66. DOI: 10.1109/TSMC.1979.4310076. Accessed July 18, 2017, at http://ieeexplore.ieee.org/document/4310076/.

S. Jana, A. Devaraj, L. Kovarik, B. Arey, L. Sweet, T. Varga, C. Lavender, V. Joshi, Kinetics of cellular transformation and competing precipitation mechanisms during sub-eutectoid annealing of U10Mo alloys, Journal of Alloys and Compounds, Volume 723, 2017, Pages 757-771, ISSN 0925-8388, http://dx.doi.org/10.1016/j.jallcom.2017.06.292.

V.P. Sinha, P.V. Hegde, G.J. Prasad, G.K. Dey, H.S. Kamath, J. Alloys Compd. 491 (2010) 753.

Appendix A

Python Script

Appendix A

Python Script

```
TITLE: Image Processing to Determine Area Fraction of Eutectoid
#Decomposition in U-10wt%Mo Micrographs
#AUTHOR: EJ Kautz
#DATE: 04/11/2017
#SUMMARY:
#reads-in images in a directory
#pre-processes (CLAHE)
#thresholds
#performs morphological operations (dilation and erosion)
#calculates area fraction (and converts to percentage)
#prints data to csv file
#-------------------------------------------------------------
from fnmatch import fnmatch
import scipy
import numpy as np
import matplotlib.pyplot as plt
import os
import cv2
import csv
import math
import operator
from cv2 import adaptiveThreshold
from decimal import *
```

```
import skimage

from skimage import data, io, exposure

from skimage.measure import regionprops

from skimage.measure import label

from skimage.filters import threshold_otsu, rank

from skimage.io import imread

from skimage.morphology import disk, rectangle

import matplotlib.cm as cm

from skimage.color import rgb2gray

import skimage.io as io

from skimage.util import img_as_ubyte

from skimage import exposure

#from skimage.morphology import disk

#------------------------------------------------------------

#source file directory

dir_path='C:/Users/kaut934/Documents/UMo_ImProc/SEM/input/SM160349 DU-Mo X1-X21C 10c_m
500C 90m FC Long DUM975'

#create empty array

n=[]

fn=[]

#out_dir='C:/Users/kaut934/Documents/UMo_ImProc/SEM/output'

#image file extension

#ext='.tif'

files=os.listdir(dir_path) #files is an array of file names

file_pattern='* LBE *.tif' #only process backscatter images

print('Pattern:'+str(file_pattern))

#print('files=' + str(files))
```

```
#amount=len([f for f in files if f.endswith('.tif')]) #number of images to process

#print('# of images to process=' + str(amount))

amount=len([f for f in files if fnmatch(f,file_pattern)])

print('# of images to process=' + str(amount))

#set output file directory here

output_dir='C:/Users/kaut934/Documents/UMo_ImProc/SEM/output/output_SM160349 DU-Mo X1-X21C 10c_m 500C 90m FC Long DUM975/'

f1_name='percent_transformed_long.csv'

f1=open(output_dir+f1_name,'w')

writer=csv.writer(f1)

writer.writerow(['File Name','Thresholded Only','With Dilation/Erosion','Difference','Result'])

#------------------------------------------------------------

for f in files:

    if fnmatch(f,file_pattern):

        fn.append(f)

        im=cv2.imread(f,0)

        im_cropped=im[1:956,1:1280]

#------------------------------------------------------------

        #plot cropped image next to image histogram

        fig=plt.figure("Original Image and Histogram")

        ax0=fig.add_subplot(2,2,1)

        ax0.imshow(im_cropped,cmap=cm.gray)

        ax0.axis('off')

        ax1=fig.add_subplot(2,2,2)

        im=im_cropped

        im = im[np.nonzero(im)]  # Ignore the background

        hist,bins = np.histogram(im.flatten(),256,[0,256])
```

```
        cdf = hist.cumsum()
        cdf_normalized = cdf * hist.max()/ cdf.max()
        ax1.plot(cdf_normalized, color = 'b')
        ax1.hist(im.flatten(),256,[0,256], color = 'r')
        #ax1.xlim([0,256])
        ax1.legend(('cdf','histogram'), loc = 'upper left')
        #plt.show()
        #ax1.xaxis.set_major_locator(MultipleLocator(10))
        ax1.minorticks_on()
        ax1.set_yticks([])
        ax1.set_xlabel('Intensity')
        ax1.set_ylabel('Count')
        fig_name=f+".png"
        plt.savefig(output_dir + fig_name)
        plt.close()
        #plt.show()
#------------------------------------------------------------------------
        #contrast-limited adaptive histogram equalization (CLAHE)
        clahe=cv2.createCLAHE(clipLimit=1.0,tileGridSize=(100,100))
        cl1=clahe.apply(im_cropped)
        cv2.imwrite(os.path.join(output_dir,"CLAHE-"+str(f)+".png"),cl1)
        min=np.amin(cl1)
        max=np.amax(cl1)
#------------------------------------------------------------------------
        #THRESHOLD(global) and take inverse/image complement
        r,t=cv2.threshold(cl1,125,255,cv2.THRESH_BINARY_INV)
```

```
        #r,t=cv2.threshold(cl1,125,255,cv2.THRESH_BINARY)
        cv2.imwrite(os.path.join(output_dir,"THRESH-"+str(f)+".png"),t)
#-----------------------------------------------------------------
        #Morphological operations - dilate/erode image
        kernel = cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (3, 3));
        im_morph=cv2.morphologyEx(t,cv2.MORPH_CLOSE,kernel,iterations=8)
        cv2.imwrite(os.path.join(output_dir,"MORPH-"+str(f)+".png"),im_morph)
        #Floodfilling
        # Copy the thresholded image.
        im_floodfill = im_morph.copy()
        # Mask used to flood filling.Notice the size needs to be 2 pixels than the image.
        h, w = im_morph.shape[:2]
        mask = np.zeros((h+2, w+2), np.uint8)
        # Floodfill from point (0, 0)
        cv2.floodFill(im_floodfill, mask, (0,0), 255)
        # Invert floodfilled image
        im_floodfill_inv= cv2.bitwise_not(im_floodfill)
        # Combine the two images to get the foreground.
        im=cv2.add(im_morph,im_floodfill_inv)
        #im = im_morph | im_floodfill_inv
        cv2.imwrite(os.path.join(output_dir,"FLOODFILL-"+str(f)+".png"),im)
#-----------------------------------------------------------------
        #Calculate area fraction for thresholded AND dilated/eroded image
        #and print to file
        x,y=t.shape
        im_area=x*y
```

```
        num_white_pix1=cv2.countNonZero(t) #thresholded image
        num_white_pix2=cv2.countNonZero(im) #dilated/eroded image
        af1=Decimal(num_white_pix1)/Decimal(x*y) #area fraction thresh. image
        af2=Decimal(num_white_pix2)/Decimal(x*y) #area fraction dilated/eroded image
        pt1=af1*100 #percent transformed (thresholded image)
        pt2=af2*100 #percent transformed (dilated/eroded image)
        ptr1=round(pt1,2) #percent transformed, rounded (thresholded image)
        ptr2=round(pt2,2) #percent transformed, rounded (dilated/eroded image)
        d=ptr2-ptr1 #difference
        if ptr2>10:
            r=['UNACCEPTABLE']
            print('UNACCEPTABLE, Percent Transformed='+str(ptr2))
            print('Percent Transformed from thresholded (only) image ='+str(ptr1))
        if ptr2<10:
            r=['ACCEPTABLE']
            print('ACCEPTABLE, Percent Transformed='+str(ptr2))
        n.append([f, ptr1,ptr2,d,r])
for values in n:
    writer.writerow(values)
f1.close()
print 'DONE'
```

Pacific Northwest
NATIONAL LABORATORY

Proudly Operated by **Battelle** *Since 1965*

U.S. DEPARTMENT OF
ENERGY

902 Battelle Boulevard
P.O. Box 999
Richland, WA 99352
1-888-375-PNNL (7665)

www.pnnl.gov