THE RELATION BETWEEN IRRADIATION ASSISTED STRESS CORROSION CRACKING AND DISLOCATION CHANNELING: THE ROLE OF SLIP INTERACTION AT THE GRAIN BOUNDARY

by

Michael David McMurtrey

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Nuclear Engineering and Radiological Sciences)
in The University of Michigan
2014

Doctoral Committee:

Professor Gary S. Was, Chairman
Professor Michael Atzmon
Associate Professor Samantha H. Daly
Associate Research Scientist Kai Sun

To my family

ACKNOWLEDGEMENTS

Finally, I would like to thank my family. My parents, Gary and Sheri, as well as my siblings (Brian, Daniel, Erin, Kristi, Julie and Kelly), for their unconditional support and encouragement over the years. My wife, Kate McMurtrey, for her love and patience even in times of stress and for providing me with encouragement and optimism throughout this endeavor.

Table of Contents

LIST OF FIGURES

xiv

LIST OF TABLES

LIST OF APPENDICES

ABSTRACT

**THE RELATION BETWEEN IRRADIATION ASSISTED STRESS CORROSION
CRACKING AND DISLOCATION CHANNELING: THE ROLE OF SLIP
INTERACTION AT THE GRAIN BOUNDARY**

by

Michael David McMurtrey

Chair: Gary S. Was

The objective of this study was to determine the role of localized deformation in
irradiation assisted stress corrosion cracking. First, the character of grain boundaries
susceptible to cracking was examined. It was observed that boundaries intersected by
discontinuous dislocation channels were more susceptible to cracking. Channels were
then characterized based on their interactions with the grain boundary, and cracking
susceptibility was examined based on these channel-boundary classifications. Results
show that dislocation channels are connected to irradiation assisted stress corrosion
cracking, however, the mechanism connecting the two is still unknown

Grain boundary susceptibility to cracking was characterized based on boundary
misorientation, angle with respect to the tensile axis, Schmid Factor, Taylor Factor, and
the continuity of the channels intersecting the boundary. It was found that cracking

propensity was higher at random high angle boundaries, boundaries normal to the tensile axis, boundaries adjacent to low Schmid Factor or High Taylor factor grains and boundaries that were intersected by discontinuous channels.

Digital image correlation and confocal microscopy were used to characterize the channels, with the focus being at the channel-boundary intersection. Channels either were transmitted across the grain boundary and classified as continuous, were discontinuous and terminated at the grain boundary (discontinuous), or were discontinuous at the grain boundary but induced grain boundary slip (discontinuous with grain boundary slip). Continuous channels were found to undergo the largest amount of slip, while discontinuous channels underwent the least. Despite the low amount of slip within the discontinuous channels, these were found to be the most likely to induce cracking. This is believed to be caused by the high level of stress that results from the unaccommodated slip. Other areas of high stress, such as triple junctions, were also found to be susceptible to cracking.

This work was able determine the relative amounts of cracking occurring from discontinuous channels that induced grain boundary slip and those that resulted in dislocation pile-ups at the grain boundaries. The findings of this work indicate that the localized deformation in irradiated austenitic stainless steel causes areas of unaccommodated slip which results in high stress, and leads to IASCC.

Chapter 1

INTRODUCTION

In 2014, the US Nuclear Regulatory Committee (NRC) reported that there are 100 NRC-licensed nuclear reactors in the US, generating approximately 20% of the total electricity generation of the country. These consist of both boiling water reactors (BWRs) and pressurized water reactors (PWRs), which are classified together as light water reactors (LWRs). These plants were originally licensed by the NRC to operate for 40 years. Current regulations allow these licenses to be renewed and extend the lifetime of the reactors for 20 additional years. As reported by the US Nuclear Regulatory Committee [1], as of June 2013, 72 of the US reactors have received license renewals. Currently, 20 reactors have been operating for over 40 years, and another 42 reactors have been operating between 30-39 years. As the age of the reactors increase, and the possibility of additional life extensions to 80 years is considered, it is becoming increasingly more important to understand the state of the materials used in these nuclear reactors.

Light water reactors use austenitic stainless steels for a variety of reactor core components. Stainless steel is desirable due to its high corrosion resistance, however, under irradiation, it has been found to be susceptible to intergranular stress corrosion cracking (IGSCC) [2–4]. This has been termed irradiation assisted stress corrosion

cracking (IASCC). The exact mechanism causing IASCC in stainless steel is not well understood. Irradiation causes a number of changes to the microstructure of the steel, including radiation induced segregation (RIS) of the elements in the alloy, an increase in hardness, and a change in the deformation mechanism from relatively homogeneous slip to very heterogeneous slip, with deformation confined to coarsely spaced bands called dislocation channels.

The complexity of the irradiation effects on stainless steel make it difficult to separate out the individual irradiation effects on stress corrosion cracking (SCC). Initially, chromium depletion at the grain boundary was considered a likely cause for IASCC, however, detailed studies have found that the lower chromium content at the grain boundary is not likely the controlling mechanism causing the increased cracking susceptibility [5,6]. In recent years, studies have shown a connection between the localization of deformation in irradiated stainless steel and IASCC [2,3,7–10]. While a connection between localized deformation and IASCC has been observed, the exact mechanism causing cracking remains unknown.

Localized deformation is a result of defects created by the irradiation severely hindering the motion of dislocations through the grains of the stainless steel. When a critical resolved shear stress is reached, the dislocations begin to slip through the defects created by the irradiation. As the dislocations pass through the irradiation damaged zone, defects are annihilated, creating a pathway with less resistance to slip for subsequent dislocations to pass through. As additional dislocations pass through the channel defects are progressively removed from the planes where slip has occurred. This forms what is termed dislocation channels, which consist of parallel slip planes where irradiation

defects have been cleared, creating a passage through which dislocation may pass with relatively low resistance. Dislocation channels are typically ~100 nm wide and space one to three microns apart.

One difficulty in understanding the role of localized deformation is determining an accurate way to analyze the amount of strain in the dislocation channel, and its effect on the grain boundary (amount of plastic/elastic strain it induces at the boundary). In this work, topographical measurements have been used to determine the height of the channel where it intersects the surface, and use this height measurement to approximate channel strain [11]. Height measurements can only be used as an approximation of total strain, as any strain in the surface plane is not measured. In plane strain has been studied in both unirradiated and irradiated metals using digital image correlation (DIC) [10,12], which refers to the comparison of two images of the sample surface at the same location, one prior to strain, one after plastic strain has been induced. Localized in-plane strain measurements are determined by measuring shifts in a pattern applied to the sample surface prior to strain. A random gold nano-particle speckle pattern was used in this work to allow for high resolution DIC to determine the in-plane displacement of the dislocation channels. By combining topographic measurements with DIC results, it was possible to measure displacement along all of the axes of the sample.

This thesis focuses on the interactions of dislocation channels with grain boundaries and the role of dislocation channeling on IASCC. Chapter 2 of this thesis describes relevant background information, in particular describing prior research that lead to this study. Chapter 3 describes the experimental procedures and the systems/techniques used in this study. Chapter 4 contains the results of the deformation

and cracking experiments. Chapter 5 is the discussion of the results presented in chapter 4, focusing on the relationship between the deformation measurements and cracking. Chapter 6 ends this thesis with the conclusions of this study.

Chapter 2

BACKGROUND

In this chapter, a review of published literature provides the background information relevant to the objectives and goals of this thesis. The first section describes the properties of the material of interest, austenitic stainless steel. The second section describes the deformation behavior of unirradiated polycrystalline materials such as stainless steel. This lays the foundation for understanding localized deformation that occurs in irradiated materials. The third section describes microstructure characteristics of importance to this study, namely grain orientation and grain boundary geometry. The fourth section discusses the effects of irradiation on austenitic steel. The fifth section describes the localized deformation processes that occurs in irradiated steel. The sixth section reviews irradiation assisted stress corrosion cracking (IASCC), the increase in cracking susceptibility that occurs due to irradiation effects in the steel. The seventh section discusses the current understanding of the relationship between localized deformation and IASCC. The final section states the objective of this work and the experimental approach used to accomplish the stated objective.

2.1   Properties of austenitic stainless steel

Steel is an iron alloy containing small amounts of carbon, generally less than 2 wt%. Steel may be separated in to three general phases: austenite, ferrite, and martensite.

Austenitic stainless steel is a form of steel containing chromium and nickel. The austenite phase is stabilized by nickel, carbon, nitrogen and manganese. Chromium decreases the stability of the austenite phase, but is added to increase corrosion resistance [13,14]. It is heavily used due to its strength, malleability and good corrosion resistance.

2.1.1 Physical and mechanical properties

While physical properties may vary depending on composition, generally austenitic stainless steels have a density of ~7.9 g/cm^3 and are non-magnetic. It has a liquidus temperature of around 1400 °C and a low mean coefficient of expansion, ~1.9 $\times$ 10^{-5} °C^{-1} for temperatures between 20-1000 °C, which make it suitable to be used in high temperature conditions such as in nuclear power plants.

Typical yield strength and ultimate tensile strength for various commercial stainless steels is shown in Table 2.1. Mechanical properties of steel, such as yield strength, are dependent on the composition, grain size and level of cold work. The effect of grain size on yield strength can be characterized by the Hall-Petch relation [15],

$$\sigma_y = \sigma_0 + \frac{k_y}{d^{\frac{1}{2}}}$$

(2.1)

where σ_y is the measured yield stress, σ_0 is a friction stress, k_y is a coefficient used to characterize the transfer of slip through the grain boundaries and d is the grain diameter. Larger grain diameters result in softer steels. Austenitic stainless steels do not lose strength as rapidly at high temperatures as ferritic steels [16]. As depicted in the plot of yield strength vs. temperature of an austenitic steel in Figure 2.1, the yield strength of

austenitic stainless steel has a very shallow negative slope with increasing temperature between 100-600°C.

2.1.2 Phases

Steel is often composed of multiple phases, with combinations of austenite, ferrite and martensite. Unlike ferrite (BCC) and martensite (BCT), austenite has a face centered cubic (FCC) structure. The FCC structure is formed from three planes of close packed atoms, generally referred to as A, B and C, so as to form the repeating pattern ABCABC. The location of atoms relative to each other in these three planes is depicted in Figure 2.2.

As stated previously, different alloying elements stabilize different phases. This is depicted in a Schaeffler diagram, which relates percent ferrite to equivalent chromium and equivalent nickel values in the alloy. The equivalence equations relate other alloying elements in the steel to either a weighted chromium or nickel concentration. Various groups have created Schaeffler diagrams using different weighting factors and taking in to consideration different alloying elements [14,17,18], such as the one shown in Figure 2.3. Martensite may form in austenite during cold work. Equation (2.2) [19,20] is used to calculate the temperature at which 50 vol% martensite is formed after a true tensile strain of 30%:

$$\begin{aligned} M_d(°C) = 413 &- 13.7(\%\,Cr) - 9.5(\%\,Ni) - 8.1(\%\,\text{Mn}) - 18.5(\%\,\text{Mo}) \\ &- 9.2(\%\,\text{Si}) - 462(\%\,\text{C} + \%\,\text{N}). \end{aligned} \tag{2.2}$$

Other phases that may form in austenitic steel include carbides, nitrides, and a σ-phase, which is generally an iron-chromium intermetallic that forms during prolonged heating in the temperature range of 650-870 °C [14]. Carbon is much more soluble in

austenite at high temperatures. In some austenitic steels, carbon is only retained in the solution by rapid cooling, so that the carbon is trapped in solution, not having enough energy to form carbides, which would otherwise be more stable at the lower temperatures. If these steels are held at higher temperatures (450-850 ºC), carbides may precipitate and form, typically at the grain boundaries [13]. The σ-phase will always form after carbides have begun to precipitate. When the intermetallic phase forms, it can lower the Cr and the Mo content of the steel, which increases the carbon solubility and can lead to partial dissolution of carbides [20].

There are several types of carbides that may form. $M_{23}C_6$ commonly forms in austenitic steels, where M represents Fe, Mo, Cr and Ni. MC may also form, with M representing Ti, Nb, or V. If Mo is present, M_6C may form, from Fe, Mo and Cr. Carbon content in austenitic steels is typically not high enough to form M_7C_3 (M representing Cr and Fe). However for steels with high carbon contents, or in areas where C concentration is higher, M_7C_3 may form [20]. These carbides are typically undesired, as they are can cause cracks to form at their interfaces and further cracking can result due to cracking propagation through the material [21].

$M_{23}C_6$ in austentic steels is typically composed of Cr and C, however Fe, Mo, and Ni atoms may be substituted in place Cr atoms, and N and B atoms may substituted in for C atoms. Thermal history has a strong influence on the composition of the carbide. It forms an fcc crystal structure, and the lattice parameter is approximately three times that of austenite, though the lattice parameter is a function of the composition. $M_{23}C_6$ will most likely precipitate at the grain boundaries, specifically at grain boundaries with high Σ values. After grain boundaries, the most likely locations for the carbide to precipitate

dislocations within the grain. Cold deformation can accelerate the carbide precipitation, specifically within the grains. $M_{23}C_6$ can cause intergranular corrosion and decreases the ductility and toughness of the steel. It has been shown, however, to make grain boundary sliding more difficult when it is present at the boundary, thus improving creep ductility [20].

The MC carbide typically forms whenever Ti, Zr, Hf, V, Nb, or Ta are present in the austenitic steel. These elements are typically called stabilizing elements. While they lower the solubility of carbon in steel, they also lower the formation of $M_{23}C_6$ so that there is less of a tendency for intergranular cracking. The MC carbide forms typically within the grain, on dislocations and stacking faults, though precipitation at the grain boundary can occur[20]. It is an fcc crystal structure and nitrogen can substitute in for the carbon to form a MN nitride.

M_6C forms in steels containing Mo, and as most austenitic steels do not contain high levels of Mo, M_6C is typically not found, or only found in small amounts. When it does form, it has an fcc crystal structure. M_7C also does not typically occur in austenitic steels, as the carbon content generally isn't high enough. It requires a very high carbon:chromium ratio, higher than would be found in normal austenitic steel. It has a pseudo-hexagonal crystal structure. Table 2.2 summarizes the information for each of the four carbides previously discussed.

2.1.3 Corrosion

While stainless steel is generally considered to be corrosion resistant, it has been found to be susceptible to stress corrosion cracking (SCC) [13,22,23], as well as general

and localized (pitting/crevice) corrosion [24]. The good corrosion resistance of stainless steel largely comes from the passive oxide layer that the chromium forms over the metal surface. This compact, thin oxide layer significantly reduces any further corrosion.

When exposed to specific corrosive environments, and being under stress, usually from some combination of residual stress from the cold working of the material and an applied stress occurring during service, SCC has occurred in steel. Higher nickel compositions (>30 wt%) make the alloy more resistant to SCC, but do not completely mitigate the cracking. The heat treatments the steel has undergone will also affect the SCC susceptibility. Internal residual stresses may provide the stress necessary for SCC in certain environments. Heat treatments may relieve these stresses and therefore reduce SCC susceptibility, provided stress is not applied to the steel after the heat treatment. If carbides precipitate during the heat treatment (generally occurs between ~430 -870°C), however, cracking susceptibility will increase. A corrosive environment must be present for SCC to occur. In the case of stainless steel, SCC generally occurs in solutions that either contain chloride ions or are caustic. Dissolved oxygen, and high temperatures also will increase SCC susceptibility.

General corrosion is the term for the roughly uniform loss of material over the entire exposed surface. Due to the passive layer that readily forms over stainless steel, general corrosion is not typically a problem. However, if the steel does not have high enough levels of alloying elements which stabilize the passive film, in particular chromium, or is located in extreme environments such as acids and hot caustic solutions, general corrosion may occur.

In some cases, localized areas on the steel will corrode, rather than the entire surface. This typically results in pitting or crevice corrosion and is a result of imperfections in the oxide film or effects of the localized environment. Localized corrosion is generally enhanced by the presence of halogenides, such as chlorides, which can hamper the reformation of the protective oxide layer [25]. Chloride and sulfate ions in particular promote corrosion and cracking in stainless steel. Chlorides promote pitting and crevice corrosion, as well as increasing the rate of stress corrosion cracking. Chlorides effect sensitized steel in particular, though have been known to increase the corrosion of nonsensitized steels. Sulfates are more aggressive than chlorides in promoting IGSCC [26]. As little as 100 ppb sulfate ions can decrease the time to failure by a factor of three, due to increased IGSCC [27].

Generally, stainless steel suffers from intergranular stress corrosion cracking (IGSCC) in boiling water reactor (BWR) normal water chemistry (NWC) environments, though pitting does occur. Cases of cracking in stainless steel were observed as early as the 1950s in stainless steel fuel rods [28]. Table 2.3 outlines observations of different cases of cracking occurring of stainless steel in BWR environments. Studies of the corrosion of the steel components has been of interest due to economical, as well as safety concerns. The rate of cracking is a factor of the impurities in the water, as well as the oxygen content [26]. Figure 2.4 depicts the crack growth rate of two stainless steels with respect to water conductivity: the higher the conductivity (the higher the ion concentration), the more severe the cracking. Corrosion of steel in nuclear reactors will be discussed in more detail in a later section where irradiation assisted stress corrosion cracking is discussed.

2.2   Microstructure characteristics of polycrystalline materials

Grain boundaries, a result of the difference in orientation between the two adjacent grains, affect the mechanical properties of the material. In this section the grain orientations, or crystallographic texture, as well as the grain boundaries, where different crystal lattices meet between grains, will be discussed in context of effects on the properties of the polycrystalline material.

2.2.1 Grain orientation and crystallographic texture

Crystallographic texture refers to the distribution of grain orientations within a polycrystalline material. Grain orientation is related to mechanical properties, such as the resolved shear stress acting on slip systems as described by the Schmid factor. Texture may be examined locally by examining individual grain orientations, or by examining the general orientations within the material as a whole.

Orientation maps, such as the one shown in Figure 2.5, are one method for depicting individual grain orientations within a polycrystalline material. These maps show grain orientation, as well as spatial information about the location of the grain. The spatial information is particularly important when examining local phenomena, such as crack characterization at particular locations, as well as providing information not only about the grain orientation, but also the orientation of neighboring grains.

When examining the general orientation of all the grains in the material, it is common to simplify the analysis by examining only the orientation information of the grains, and not any of the spatial information, such as the location of the grain within the material. Pole figures are stereographic projections of the lattice orientations and are a

common method of presenting orientation data. Figure 2.6 demonstrates how a pole figure is created. A sphere is made to surround a unit cell of the lattice, and a stereographic projection is made of the points where lines normal to the lattice planes intersect the sphere.

2.2.2 Schmid and Taylor factors

A number of models have been created to describe macroscopic stress/strain by crystallographic properties, namely based on grain orientation [29–31]. Schmid and Taylor factors will be described in this section. The Schmid factor does not directly describe slip within a grain, however, it does examine the resolved shear stress acting on a single slip system, which is important when considering the critical resolved shear stress needed to initiate slip on that slip system. Taylor factor describes slip within a grain during straining of a polycrystalline material.

As stated previously, slip will not occur until a critical resolved shear stress, τ_c, has been reached on the slip system. For uniaxial applied tensile stress, this may be calculated as

$$\tau_c = \sigma \cos \lambda \cos \varphi \,, \tag{2.3}$$

where σ is the applied tensile stress, λ is the angle between the slip direction and the tensile axis, and φ is the angle between the slip plane normal and the tensile axis. This is depicted graphically in Figure 2.7. Equation (2.3) may be shortened to

$$\tau_c = m\sigma \,, \tag{2.4}$$

where m is the Schmid factor and is defined as

$$m = \cos\lambda\cos\varphi \qquad\qquad (2.5)$$

For a slip system aligned to maximize shear stress from the applied tensile stress, the Schmid factor is at a maximum (m = 0.5). For a slip system oriented such that the slip direction is either parallel or perpendicular to the tensile axis, there is no shear stress acting on the system and the Schmid factor is minimized (m = 0). Grains may also be described by the Schmid factor of the slip system with the highest Schmid factor. Figure 2.8 shows the relationship between the orientation of the tensile axis and Schmid factor for a FCC crystal.

In Taylor's plasticity analysis [30], it was assumed that all grains undergo the same change in shape as the entire polycrystalline material. This assumption is depicted graphically in Figure 2.9. Taylor also assumed that all deformation occurred through crystallographic slip and the shear stress required to cause slip was the same for all slip systems. He concluded that the least work possible to impose the shape change in the grains would require that at least five (of the twelve in FCC) slip systems were active. For any given crystal orientation, Taylor selected the combination of five slip systems that resulted in the lowest sum of shear strains required to meet the overall shape change of the grain, assuming the other seven slip systems underwent zero shear strain.

When minimizing work performed in the slip systems, the incremental work/volume (d*w*) cause by slip in a given grain was defined as

$$dw = \sum \tau_i d\gamma_i \;, \qquad\qquad (2.6)$$

where τ_i is the shear stress required for slip to occur in slip system i, and γ_i is the shear strain in that slip system. Given the assumption that shear stress required to cause slip is the same for all slip systems, τ_i is a constant and may be moved to the outside of the summation. Using

$$\mathrm{d}w = \sigma_x \mathrm{d}\varepsilon_x \tag{2.7}$$

for the work/volume expressed in terms of external (applied to the macroscopic polycrystalline material) stress, σ_x, and strain, ε_x, along the tensile axis x. Equating the work in the grains and the external work,

$$\sigma_x \mathrm{d}\varepsilon_x = \tau \mathrm{d}\gamma \ , \tag{2.8}$$

Where $\mathrm{d}\gamma$ is the sum of $\mathrm{d}\gamma_i$ from Equation (2.6). The Taylor factor (M) is defined as

$$\mathrm{M} = \mathrm{d}\gamma \,/\, \mathrm{d}\varepsilon_x = \sigma_x \,/\, \tau \ . \tag{2.9}$$

A high Taylor factor means that the grain had to undergo a lot of slip in order to match the overall shape change of the polycrystalline material, which means that a high Taylor factor grain will require more applied stress in order to undergo the necessary strain. For a FCC material, Taylor factor can range from 2.449 to 3.674, depending on crystal orientation, as described by Figure 2.10. Detailed descriptions of Taylors work can be found in more recent books describing mechanical behavior of materials by authors such as Hosford [32].

2.2.3 Grain boundary geometry

15

A grain boundary is defined as the surface between two crystal lattices, which are at different orientations from each other. Due to mismatch of the lattices and an imperfect union between the two adjacent lattices, more free volume exists in the grain boundary than a perfect lattice. Read and Shockley have shown that this is similar to a network of dislocations and that grain boundaries may be modeled as an array of dislocations [33]. The amount of free volume is dependent on the mismatch of the adjacent lattices.

In 1949, a special subclass of grain boundaries were classified by Kronberg and Wilson [34]. These boundaries, called coincident site lattice (CSL) boundaries, are special due to the similarity in adjacent lattices which causes the two grains to match well at the boundary. To better understand the structure of CSL boundaries, the lattices of the two adjacent grains are visualized as overlapping, as shown in Figure 2.11, which depicts a $\Sigma 5$ boundary. CSL boundaries are denoted by a Σ value, which is the reciprocal of the ratio of the number of coincident sites to all lattice sites (e.g. so in the $\Sigma 5$ boundary type, one in five lattice sites are in coincidence). CSL boundaries may also be described by rotation and an axis of rotation. These are listed for CSL boundaries $\Sigma < 30$ in Table 2.4. A number of studies have shown that the energy associated with grain boundaries decreases sharply when the boundary is a CSL when compared to RHABs of similar orientations [35–37]. This is especially true for $\Sigma 3$ (twin) boundaries. An example of this is shown in Figure 2.12, where grain boundary energy for a number of boundaries is compared. Reported in these results is the relative grain boundary energy, with the boundary at 129° serving as the reference energy.

In general, a grain boundary is just an interface. As could be expected of any type of interface, the cohesion strength may be affected by defects at the interface. In the case of grain boundaries, the cohesion strength may be weakened by mechanical defects (non-bonded regions, such as those caused by missing atoms from dislocation or voids) and chemical segregation of impurities to the interface (could affect the spacing of atoms in the lattice or create additional, weaker interfaces, that lead to stress concentrations at the boundary interface) [38]. Temperature and emission of dislocations are also stated to affect cohesion strength, as well as the angle of the grain boundary tilt [39].

While a stress may be applied uniformly to a polycrystalline sample, the stress will not be uniformly distributed throughout the sample, due largely to the anisotropy of the deformation properties of grains, and the random crystal orientations in most polycrystalline material. In particular, stress tends to be higher at grain boundaries, likely due to the effect of the crystallographic orientations of neighboring grains. An example from a finite element model done by Kamaya et al. [40] is shown in Figure 2.13. In particular, stress may be particularly high at triple junctions, the point where three grains meet [40–42].

2.3   Deformation behavior in unirradiated face centered cubic (FCC) polycrystalline material

Under high levels of stress, crystalline materials will deform through the movement of dislocations along slip systems. On the macroscopic level, deformation begins at the yield stress of the material, which will elongate along the direction of the applied stress and reduce in the dimensions normal to the applied stress. On the microscopic level, strain is more complex in polycrystalline materials. Depending on the

orientation of the grains, slip may occur in certain grains before the applied stress has reached the macroscopic yield stress.

2.3.1 Dislocations and slip systems

In general, metals deform through the slip of dislocations through the crystal lattice. Dislocations are generally defined as edge dislocations and screw dislocations. Edge dislocations act as an extra half plane of atoms within the lattice and produce slip in the direction the dislocation moves. The magnitude and direction of the distortion in the lattice caused by the dislocation is equal to the Burgers vector (**b**) of the dislocation. For edge dislocations, the Burgers vector is perpendicular to the dislocation. In the case of screw dislocations, the Burgers vector is parallel to the dislocation line.

In an FCC crystal structure, edge dislocations tend to move along the {111} planes and in the <110> directions, for a total of 12 slip systems. Dislocations will not move along the slip plane until a critical resolved shear stress has been reached. This was discussed more in the section on Schmid factor. Obstacles within the grain, and the grain boundaries themselves, act to impede the movement of dislocations. When encountering an obstacle that the dislocation is unable to move through, edge dislocations may move perpendicular to the slip plane through a process called climb. Climb is a diffusion process, where interstitials or vacancies diffuse to the dislocation, causing the movement perpendicular to the slip plane. In the case of interstitials, the dislocation half plane is lengthened, whereas vacancies cause it to move in the opposite direction, shortening the half plane. As it is a diffusion process, climb is more likely to occur at high temperatures. Interactions between dislocations and grain boundaries will be discussed in more detail later in this section.

2.3.2 Stacking fault energy and partial dislocations

      A stacking fault results when a plane of atoms is added or removed from the repeating ABCABC pattern of the FCC close packed planes. The stacking fault is considered intrinsic when a plane is removed, resulting in ABCA|C, with the stacking fault represented by the |. An extrinsic stacking fault results from the addition of a plane within the pattern, such as ABCA_C_BC, where an additional C plane is added, represented by the _C_. There is an energy associated with the change in order of the close packed lanes, called the stacking fault energy (SFE), which is an intrinsic property of an alloy.

      SFE is related to the composition of an alloy. There are several empirical correlations relating SFE and composition, though they are not consistent with each other. In general, they follow the form

$$SFE(\mathrm{mJ/m^2}) = X_{Const} + \sum_i X_i C_i(\mathrm{wt.\%}),$$

(2.10)

where the constants (X) were determined empirically, and the concentrations (C) of the alloying elements are known. Table 2.5 depicts three sets of constants from different empirically derived fits of composition to SFE. As a result of the high degree of variation in SFE models, when accurate values are required, they must be derived experimentally. Equation (2.10) and the constants found in Table 2.5 are better suited for relative approximations [7].

      Stacking faults form when a perfect dislocation dissociates into two partials (the stacking fault exists between the two partials). In order to minimize the energy required to move the dislocation, it is often divided into two partial dislocations, which each act as

half of the full dislocation. A perfect dislocation with a Burgers vector of $\mathbf{b} = \frac{1}{2}[\bar{1}01]$ may divide into $\mathbf{b} = \frac{1}{6}[\bar{1}\,\bar{1}2]$ and $\mathbf{b} = \frac{1}{6}[11\bar{2}]$ partial dislocations, as shown in Figure 2.14. These two partial dislocaitons tend to be divided by a number of atomic planes, rather than occurring over a single plane as shown in Figure 2.14. The two partials repel each other, however, the existence of the stacking fault between them also increases the overall energy of the lattice, and so the SFE pushes them together in order to minimize the increase in energy caused by the stacking fault. In materials with a low SFE, the partials will be separated by a larger distance than in a material with a high SFE. Due to the large spacing between partials, it is difficult for the dislocation to cross slip between slip planes, as the partials must recombine prior to the cross slip. As a result, low SFE materials generally undergo planar slip. Austenitic stainless steels generally have low SFEs.

2.3.3 Interactions between dislocations and grain boundaries

As stated previously, and depicted in Equation (2.1), grain size affects the strength of a material. Specifically, smaller grains result in higher yield stresses. This is a result of grain boundaries acting as obstacles to dislocation slip. When a dislocation encounters a grain boundary, there are several different forms of interactions that may occur, which may be generally divided into discontinuous slip and slip accommodation. Discontinuous slip refers to the case where a dislocation encounters a grain boundary and is pinned with no further slip occurring. Slip accommodation may refer to direct slip transmission across the grain boundary, cross slip into a different slip system within the same grain, nucleation of new dislocations, or absorption into the grain boundary. These cases of dislocation interactions with the grain boundary are depicted in Figure 2.15.

When a dislocation cannot be accommodated at the grain boundary, it will remain in the lattice near the grain boundary. As the crystal is strained further, additional dislocations will move through the lattice towards the grain boundary and form a dislocation pile-up. Dislocations of like signs will repel each other, however, the applied stress on the crystal will push the dislocation towards one another, as they are unable to pass through or into the grain boundary. This creates an area of high stress around the pile-up. Britton et al. [43] used high resolution electron backscatter diffraction (EBSD) to measure the stress caused by a dislocation pile-up of screw dislocations at the grain boundary and found a stress field ahead of the pile-up which decayed at a rate proportional to one over the square root of the distance from the pileup, similar to the model proposed by Eshelby et al. [44].

The mechanism governing slip transmission has been studied using in-situ straining in electron microscopes [45–48]. Using scanning electron microscopy (SEM) imaging of indentations near grain boundaries, Wo et al. [48] found that slip transmission was related to a misorientation factor based on the orientations of the closest {111} slip planes and <110> directions of the two neighboring grains. Robertson et al., using TEM, determined more detailed factors that govern the emerging slip system that will activate from the incoming dislocations in slip tranmsission [47]. Those factors are as follows:

1) The angle between the lines of intersection of the incoming and outgoing slip planes with the grain boundary should be a minimum.

2) The resolved shear stress acting on the outgoing slip system should be a maximum.

3) The Burgers vector of the residual dislocation left in the grain boundary after the emission of a dislocation into the adjoining grain should be a minimum.

Of these three factors, the third has been found to be the most important in the slip transmission process. While the first and the third deal with the passage of dislocations through the grain boundary, the second is connected to the propagation of those dislocations through the grain adjacent to the incoming dislocations.

Dislocations observed in the TEM to absorb into the grain boundary may either be glissile or sessile and retain the lattice Burgers vector or decompose into grain boundary dislocations [49]. In cases of slip nucleation, which appears similar to the case of direct transmission that occurs only for screw dislocations with a line direction parallel to the slip plane intersection in the grain boundary, the dislocations are absorbed into the grain boundary, and a new dislocation is nucleated in the grain boundary and ejected into the adjacent grain.

2.3.4 Deformation by twinning

In general, mechanical twinning occurs in in BCC or HCP materials, though it has been observed in FCC metals [50–52]. Slip is the dominant deformation mechanism in FCC metals, however, changes in the material can cause the critical resolved stress for slip to be higher than the stress needed to cause twinning. Twins tend to form on {111} planes oriented such that the resolved shear stress is high on the lattice plane [53,54]. Low temperatures and high strain rates are known to increase the occurrence of mechanical twinning, in particular in low stacking fault energy metals.

Attempts to measure the twinning stresses of materials has resulted in a large scatter of data, and so, though twinning stresses are reported, generally these have a large error associated with them [53]. Trends have been reported, however, that twinning stress is related to the SFE of the material [53,55,56]. Materials with lower SFEs tend to have lower twinning stresses [55,56], as shown in Figure 2.16, thus forming deformation twins more readily during straining. Deformation twinning has also been found to be strongly affected by temperature. The first case of observed deformation twinning in an FCC metal, which was previously believed to not undergo deformation twinning, was copper strained at 4 K [51,57]. Since the first twinning observation of FCC metals, low SFE FCC metals have been observed to twin at temperatures above room temperature [53]. Hamada et al. [58] found that some stainless steels underwent deformation twinning at temperatures between 0 and 100 degrees Celsius. Below 0 °C, deformation martensite formation became the dominant deformation mechanism. Twinning stress itself does not vary much with temperature, rather, the increase in yield stress at low temperatures suggests that the critical resolved shear stress required for slip has increased to the point of being greater than the twinning stress[53]. Strain rate may also be adjusted to change the deformation mechanism to that of twinning. High strain rates are much more likely to induce deformation twinning, such as shock loading or severe impact experiments [53,59].

In general, twinning produces very little macroscopic strain. The maximum extension that can be obtained by twinning a crystal of zinc is only 7.39% [60]. While twinning may not account for much of the deformation within metals, it can bring slip

planes into a more favorable orientation for slip to occur more readily in the twinned region.

2.3.5 Inhomogeneous strain and strain tensors

As discussed previously, while a uniform stress may be applied to a polycrystalline material, due to the anisotropy of grain properties and the geometry of crystal grains, inhomogeneous strain occurs within the material. As Kamaya points out [61], it is difficult to predict the local strain from general information, such as Schmid factors and Taylor factors. It must be measured in detail or simulated taking in to account the effects of the grain orientations in the polycrystalline material.

The plastic strain tensor describes the three dimensional strain of a given area. Plastic strain may be related to the displacement within a sample from the unstrained state to the strained state. This displacement is measured in terms of the displacement vector $u_i(x_i)$, which describes the motion of each point in the material from its original position, described by the vector x_i (represented by the three Cartesian coordinates x_1, x_2, x_3, where 1, 2, and 3 represent the three Cartesian axes), to a new position, described by the vector y_i, such that

$$y_i = x_i + u_i(x_i) \tag{2.11}$$

A graphical example of x_i, y_i, and u_i is demonstrated in Figure 2.17. The gradient of u_i, is the displacement gradient tensor:

$$u_i \otimes \nabla = \frac{\delta u_i}{\delta x_j} = \begin{vmatrix} \dfrac{\delta u_1}{\delta x_1} & \dfrac{\delta u_1}{\delta x_2} & \dfrac{\delta u_1}{\delta x_3} \\[2ex] \dfrac{\delta u_2}{\delta x_1} & \dfrac{\delta u_2}{\delta x_2} & \dfrac{\delta u_2}{\delta x_3} \\[2ex] \dfrac{\delta u_3}{\delta x_1} & \dfrac{\delta u_3}{\delta x_2} & \dfrac{\delta u_3}{\delta x_3} \end{vmatrix}. \tag{2.12}$$

The gradient of y_i is defined as the deformation gradient tensor (F_{ij}), and is equal to the displacement gradient tensor summed with the identity matrix (δ_{ij}), as shown here:

$$F_{ij} = y_i \otimes \nabla = \frac{\delta}{\delta x_j}(x_i + u_i) = \delta_{ij} + \frac{\delta u_i}{\delta x_j}. \tag{2.13}$$

These tensors describe how a point changes in position after a change in shape. There are several different methods to convert this to strain. Traditionally, strain (ε) in a direction may be calculated as a change in length (Δl) compared to the original length (l),

$$\varepsilon = \frac{\Delta l}{l}, \tag{2.14}$$

or, in three dimensions,

$$\varepsilon_{ij} = \frac{\Delta l_i}{l_j}. \tag{2.15}$$

Here, Δl_i is the same as the displacement vector described earlier. For infinitesimal lengths (l_j), and using the displacement vector, strain may be defined as the displacement gradient tensor:

$$\varepsilon_{ij} = \lim_{l_j \to 0} \frac{\Delta l_i}{l_j} = \frac{\delta u_i}{\delta x_j} \tag{2.16}$$

The first subscript refers to the axis being strained, the second to the direction of the strain. Normal strain is represented by the cases where i=j, whereas when i≠j, the strain is shear strain. Generally, shear strain is reported as engineering shear strain, γ_{ij}, where i and j may be interchanged, as $\gamma_{ij} = \gamma_{ji}$. It is calculated as

$$\gamma_{ij} = \frac{1}{2}\left(\varepsilon_{ij} + \varepsilon_{ji}\right).$$ (2.17)

There are other methods of defining the strain tensor, such as the Lagrange strain tensor and the Eulerian strain tensor. These definitions begin with the Green definition of strain,

$$\varepsilon_G = \frac{l^2 - l_0^{\,2}}{2l_0^{\,2}}\;.$$ (2.18)

As these definitions of strain were not used in this work, they will not be discussed further here. More details, however, may be found in mechanics books [62].

2.4   Irradiation effects on stainless steel

Steel undergoes several changes during an irradiation. These changes include microstructural changes (radiation induced hardening, the formation of dislocation channels and loops) and micro-compositional changes (radiation induced segregation). These changes are largely a result of the formation, and subsequent movement, of point defects during irradiation. The formation of these defects, and their subsequent diffusion are the root cause for the many irradiation effects observed in stainless steel. Many of these effects will be discussed briefly in this section. Dislocation channeling and irradiation assisted stress corrosion cracking are directly related to

the research performed in this thesis and will be discussed in more detail than the other irradiation effects.

2.4.1 Defect formation

As stated previously, many of the defects formed during irradiation are annihilated through the recombination of a vacancy and an interstitial. Some, however, aggregate to form Frank loops. These Frank loops can be composed of either interstitials or vacancies, however at higher temperature, temperatures greater than 573 K, the vacancy loops lose their stability and form cavities or voids [63]. Defect loops accumulate much more quickly in fcc metals, such as austenite, than they do in bcc metals. A Frank loop may lower its energy by splitting into Shockley partial dislocations, as long as it is parallel to the [110] direction. The final shape is a tetrahedral with sides made up of stacking faults. This defect is called a stacking fault tetrahedral.

As stated previously, voids form as vacancies conglomerate and have several effects on the material. Voids change the strength of the material. High void density, roughly around 10% by volume, can cause a material to become brittle [64]. Void formation can also lead to a phenomenon called void swelling, which is a volume increase in the material due to these stable vacancy clusters. Void swelling can cause serious problems in nuclear reactors, as it can cause changes in geometry and make pieces not fit the same way as they originally did.

2.4.2 Radiation hardening

Radiation induced hardening is largely a result of these defects that form during irradiation and can be divided into two forms: source hardening and friction hardening.

Source hardening is the increase in the amount of stress required to cause dislocations to form at the point of origin, called the dislocation source (e.g. grain boundaries, precipitates, defects). Friction hardening has to do with the resistance to the dislocation's motion caused by obstacles lying close to or in the slip place. Voids and loops act as weak obstacles that temporarily pin dislocations as they glide through the crystal, causing irradiated material to be harder, though more brittle. An example of dislocation pinning can be seen in Figure 2.18.

A comparison of the irradiation hardening of several austenitic steel alloys can be seen in Table 2.6. Hardening is the difference between the normal alloy and the irradiated alloy. Irradiation hardening is also apparent in the stress-strain curves of the metals. Irradiated metals have higher yield stresses and tend to be more brittle than unirradiated metals. Stress-strain curves of a commercial 316 stainless steel at various doses of irradiation are depicted in Figure 2.19.

2.4.3 Radiation induced segregation (RIS)

The point defects formed during irradiation, whether interstitials or vacancies, are mobile. Two segregation mechanisms are often considered when RIS is discussed. The first mechanism is called inverse Kirkendall behavior, where atoms move preferentially with specific defects. When a vacancy moves, atoms move in the opposite direction. An interstitial is an atom that is out of place in the lattice, so its movement means the atom is moving in the same direction. As a result of some atoms moving more easily due to the vacancy mechanism than they do as interstitials, specific elements have a tendency to move toward defect sinks, or away from them. The interstitial movements and the vacancy movements explain why the undersized and oversized atoms tend to respectively

segregate towards and away from grain boundaries and other defect sinks [65]. The second mechanism that may play a role is due to a binding between alloying elements and vacancies. This would cause the vacancy to drag atoms in the same direction that the vacancy moved.

The movement of atoms and irradiation induced defects is what causes steel to undergo RIS. These segregation profiles are typically between 5-10 nm wide [66]. Figure 2.20 shows typical elemental profiles across a grain boundary in austenitic steel. RIS in stainless steel, changes the grain boundary composition from that of the bulk grain. The segregation causes a decrease in chromium concentration at the grain boundaries, but increases the concentration of silicon and nickel. Further examples of RIS in austenitic steels may be seen in Table 2.7. RIS is affected by the composition of the bulk material, as well as the radiation dose and the temperature of the irradiation [66]. For irradiation dose, segregation in austenitic steels was found to initially increase quickly with increasing irradiation dose, however, as irradiation levels increased, the rate of RIS increase slowed considerably. Irradiation temperature also has a large effect on RIS. With each alloy, there is a temperature where a maximum amount of segregation occurs, though this temperature changes based on alloy composition.

2.5 Dislocation channeling

2.5.1 Conditions for channel formation

Two major modes of deformation occur during the straining of irradiated austenitic stainless steel. These are deformation twinning and dislocation channeling. The deformation twinning in irradiated stainless steel is considered microtwinning, as the twins generally are around 50 nm wide [67,68]. Both the microtwins and the dislocation

channels are cleared of irradiation induced defects [67] and can be difficult to distinguish. Strain rate, total strain level, deformation temperature, and irradiation dose all affect the deformation mode of the steel.

Similar to the unirradiated metal, as discussed in section 2.2, twinning in irradiated stainless steel is the dominant deformation mode at low deformation temperatures and high strain rates [67–72]. At room temperature, irradiated stainless steel deforms by both dislocation channeling and twinning [68,72]. Microtwins can be distinguished from channeling by streaks that will be observed in the diffraction pattern due to the narrow changes in the latticed caused by the twins [67,72].

Dislocation channeling is the major form of deformation in irradiated metals strained at high temperatures. Irradiation causes austenitic steels to harden by forming defect clusters, which act as obstacles to dislocation movement when stress is applied to the steel. As dislocations move through the defects (point defects, as well as small defect clusters) formed in the sample, they annihilate them and create paths through the grains with fewer defects. Dislocations have been observed annihilating stacking fault tetrahedra through in-situ TEM straining, which has been coupled with molecular dynamics models to explain methods through which dislocations clear defects to form channels [73,74]. The dislocation channels themselves form as more dislocations tend to move in these defect clear paths, since there are fewer obstacles. Some obstacles are easier for dislocations to remove than others. For example, helium bubbles are more difficult to annihilate than voids, so steels irradiated with helium ions tend to undergo less channeling and more twinning.

2.5.2 General characteristics of dislocation channels

Channels form on the slip systems, so for FCC, the {111} glide planes and the [110] directions. When they intersect the surface of the material, visible steps are formed, as seen in the SEM image in Figure 2.21. Channels may form in low numbers even before the macroscopic yield point has been reached for the irradiated material [75,76]. Dislocation channeling causes strain localization when irradiated steel is deformed, as all of the plastic strain has been found to occur within the channels [76].

Dislocation channels may be characterized by their width, spacing, and the amount of shear strain in the channel. Typical width for channels in irradiated stainless steel would be on the order of 0.1 μm, with a spacing around 1 to 3 μm between channels [77]. As irradiation dose increases, the width and spacing between channels both increase. It has also been observed, in a study of dislocation channels in 316 steel, that the channel width, and the resolved shear stress associated with each channel, are greatest when the angle between the direction of the tensile stress and the slip plane normal are around 45° with each other [72]. The relation between channel width and resolved shear stress and the change in channel width with different slip directions are shown in Figure 2.22. In this figure, the data was collected from a steel that was irradiated to less than one dpa, which is likely the reason behind the overall small size of the channels. Hashimoto et al. [72] use this data to indicate they believe that there may be a stress dependent mechanism that limits the width of the dislocation channel evolution during deformation.

Some research also suggest that stacking fault energy (SFE) is important in planar deformation. Low stacking fault alloys tend to deform in planes, whereas high SFE tends

to have wavy slip. The low SFE alloys tend to have higher average strain in the channels than high SFE alloys [11]. Recalling the studies on SFE discussed earlier in this chapter, low SFE alloys tend to have high loop densities [78], which could explain the increased susceptibility to deformation localization, as there are more loops to impede dislocation motion.

2.5.3 Strain and stress associated with channeling

The amount of shear strain in the dislocation channels is difficult to analyze experimentally. The step height caused by the channel has been used to determine strain [7], as well as the offset the channel caused when it intersected other microstructural features [79]. Using these methods of measurement, strain within dislocation channels was found to be around two orders of magnitude higher than bulk strain. Strain is especially high in channels of low stacking fault materials [80], as seen in the comparison of channel heights in Figure 2.23.

Dislocation channeling occurs in irradiated austenite when a high enough stress levels is reached to propagate dislocations through the irradiation damaged matrix. Once a critical resolved shear stress has been reached, dislocations begin to slip in coarse dislocation channels, due to radiation effects in the steel. The critical resolved shear stress for irradiated materials is higher than that of unirradiated [81], which causes the overall stress of the metal to reach higher levels during straining before stress is relieved by plastic strain through dislocation slip. Normal stress at grain boundaries perpendicular to the tensile axis is particularly high, with the possibility of being more than double the applied stress according to a finite element model by Kamaya, et. al. [40].

Evrard and Sauzay [82,83] used a finite element model to simulate the intersection between channels and grain boundaries and free surfaces. Dislocation channels were modeled as soft channels (low critical resolved shear stress) in the hard grain (high critical resolved shear stress). Stress was examined along the grain boundary, as seen in Figure 2.24, with r = 0 being the point of channel and grain boundary intersection. Areas of high stress were found at the channel and grain boundary intersection. This figure also shows that the channel thickness affects the stress imposed by the channel on the boundary [82,83]. When these results are considered alongside those presented in Figure 2.22, it is expected that channels at 45° to the tensile direction induce more stress, due to the increased width.

2.6 Irradiation assisted stress corrosion cracking

Intergranular cracking has been found to occur in irradiated materials at strains and environments that would normally not cause cracking in that material. This has been termed irradiation assisted stress corrosion cracking (IASCC), as it occurs in certain irradiated alloys (like austenitic steel) and in the presence of stress and a corrosive environment. IASCC was first observed in reactor core components over 40 years ago, and a number of reviews have been written over the years discussing the current state of knowledge in to the problem [3,4,84,85]. The exact mechanism causing the increased cracking susceptibility in irradiated materials has been difficult to determine, due to the many irradiation effects that occur simultaneously.

2.6.1 Susceptible grain boundaries

IASCC is often studied using constant extension rate tensile (CERT) tests. These tests are generally performed in a simulated reactor water environment with a slow strain

rate, on the order of 10^{-7} s^{-1}, applied to the tensile samples. IASCC has been found to occur at a fraction of the yield stress of the material (on the order of 0.4-0.5 of the yield stress) [75], though generally does not occur in materials irradiated to doses below 1 dpa [3]. Certain grain boundaries are more prone to cracking than others. Similar to normal IGSCC, $\sum 3$ boundaries are resistant to IASCC [47,86,87].

In her cracking analysis, West developed a model to explain the cracking due to normal stresses [88,89]. The model takes into account the normal stress on grains, based on the boundary surface angle vs. tensile direction, and the Schmid factor pair types around the boundary. The model gives this final expression for normal stress, σ_N, acting on the boundary:

$$\sigma_N = \sigma_{f_g} (\cos \alpha)^2 \, ,$$
(2.19)

where

$$\sigma_{f_g} = \sigma_f \frac{m_{avg}}{m_g} \, ,$$
(2.20)

and σ_{f_g} is the flow stress, or the stress required to maintain plastic deformation, m_{avg} is the average Schmid factor of the sample, which West reported as being 0.45, m_g is the Schmid factor of the grain of interest, and α is the angle between the normal to the grain boundary plane and the tensile axis. The cos α term in eq. 4 is a maximum when the grain boundary normal is parallel to the tensile direction, which one would expect in order to maximize normal stress. If the Schmid factor of the grain is less than that of the bulk sample, then normal stresses increase, as the grain isn't able to accommodate the strain as well as its surrounding grains.

2.6.2 Possible mechanisms

Early in the study of IASCC, RIS was often considered as being the driving force for the enhanced cracking susceptibility in irradiated materials. Depletion of chromium from the grain boundary, as well as enrichment of silicon, were believed to be weakening the protective oxide layer at the boundary. However, it has been found that it is not likely the dominant factor in IASCC by Busby et al. [5]. Though irradiation effects occur simultaneously, Busby found that they are removed at different rates during post-irradiation annealing. Through these experiments, it was found that IASCC susceptibility returns to that of the unirradiated state of the material much more quickly than RIS, suggesting that RIS is not the dominant factor and that some other irradiation effect was controlling the cracking.

As focus moved away from RIS, localized deformation became a point of interest. A number of studies suggested that localized deformation was linked to IASCC [8,75,90]. Jiao et. al. [7], whose results are shown in Figure 2.25, has recently reported a good correlation between the degree of localized deformation and cracking, suggesting that dislocation channeling has an important role in IASCC. The connection between localized deformation and IASCC will be discussed in more detail in the next section.

In some cases, such as with SCC, cracking serves a stress relief mechanism in areas of high stress that cannot be relieved by other mechanism. As described by Kamaya [40], there are several types of stress that may be the governing stress related to cracking: resolved shear stress on the surface, which occurs in fatigue, normal GB stress on the surface, as is the case with SCC, or normal GB stress inside the body, which can occur during creep. Though IASCC is not a purely mechanical process, as there are

environmental effects, localized strain at the grain boundary has been reported as having an important role [10,75].

2.7 Localized deformation and IASCC

As mentioned previously, correlations between the degree of localized deformation and cracking [7,67,90] has been observed. A study comparing two irradiated stainless steels found the one with a higher propensity for dislocation channeling is also the one with a higher susceptibility to cracking [67]. Cracks have also been found to form where dislocation channels intersect grain boundaries, as seen in Figure 2.26. Even in unirradiated metals, heterogeneous deformation is believed to cause for crack initiation [91]. However, the exact mechanism linking localized deformation to IASCC is still not well understood.

In 2004, Was et al. [90] discussed possible mechanisms relating localized deformation and IASCC. Three forms of dislocation channel-grain boundary (DC-GB) interactions were considered, as shown in Figure 2.27, slip transmission, dislocation absorption into the GB resulting in GB sliding, and discontinuous slip resulting in a dislocation pileup. Of these three, the latter two were considered the likely locations for crack initiation, due to high strain in the case of grain boundary sliding, and high stress in the case of the dislocation pileup.

Indirect evidence and correlations of cracking data [80,90,92] have suggested that the grain boundary slip is particularly important. The high strain in the area of the grain boundary intersected by the channel causes the grains to shift and cracks the oxide. This process is repeated as the oxide reforms, and the crack widens. In the cited work by Jiao

[80], cracking propensities of two alloys were compared with propensity for GB sliding, and the alloy with a higher propensity for GB sliding was found to be more susceptible to cracking. However, due to the nature of the measurement technique, only a select few boundaries were examined, and the GB sliding measurements and the cracking analysis were performed on different samples. While these results provide evidence of the mechanism connecting localized deformation and IASCC, they do not directly establish the cause and effect of the DC interacting with the GB.

The second DC-GB intersection likely to induce cracking, as described by Was et al. [90] is the discontinuous DC, where the dislocations pileup at the GB. In this case, the inability of the boundary to accommodate the stress in other forms results in the stress reaching a critical level where it is relieved by the formation of a crack. West and Was [89] found a connection with stress at the grain boundary and IASCC, with IASCC susceptibility being higher at grain boundaries that experience high normal stress. In unirradiated materials, non-homogeneous stress has also been linked to cracking [40,93]. Discontinuous dislocation channels at the grain boundary intensify this effect of high stress where deformation is constrained. Nishioka et al. [75] found that there were two stresses working on the boundary, from the localized strain, and normal stress from the applied strain. They were unable to determine the relative importance of the two types of stresses. Lack of three dimensional data for the channels and the boundary make it difficult for them to determine the actual stress on the boundary needed to initiate grain separation.

While the cases of GB slip and dislocation pileups are generally considered more likely to cause IASCC, in instances where slip is continuous across the grain boundary,

partial dislocations will be left behind due to the mismatch of Burgers vectors between the incoming and outgoing dislocations and may build up and cause crack initiation. This was observed by Simkin [94] in TiAl, which cracks where deformation twins intersect grain boundaries, causing localized deformation similar to dislocation channeling in irradiated metal, though not to the same degree of localized deformation. Cracking was found to be related to a high Schmid Factor of the twinning plane (in FCC, same as slip plane) and the ability of the grain boundary to accommodate the localized strain well by emitting dislocations in the adjacent grain. It was also found, that for cracking to occur, the shear direction of the twinning plane should be close to the tensile direction. The cracking was determined to be a result of the residual defects remaining in the grain boundary after the twin shear was accommodated.

A recent study by West has shown a correlation in cracking trends and slip discontinuity [88]. Only grain boundaries that were intersected by dislocation channels were characterized. If more than 50% of the channels were continuous, the boundary was characterized as continuous. This work supports the earlier work by Was et al. [90], suggestion that it is the discontinuous channels, or the channels that cause grain boundary sliding that are important. As GB slip was not measured in this cracking study, the two cannot be distinguished.

2.8   Objective and Approach

While IASCC is generally agreed to be related to localized deformation, more work is needed to understand the exact mechanism. As recent as 2009, Bieler [91] commented on how little work has been done to examine how slip interactions can lead to damage nucleation. This is particularly true in the nuclear field with localized

deformation. ***The objective of this work is to understand the role of localized deformation in IASCC and how the slip interactions at the grain boundary are related to the crack initiation.*** This will be accomplished in a two phase experiment. First, plastic strain will be fully characterized on a microscopic level, so that slip in channels and grain boundaries can be quantified and the DC-GB intersection may be classified as continuous slip, grain boundary slip or discontinuous slip. After quantifying the slip, cracking will be induced and characterized to determine where cracking is occurring.

Plastic strain may be characterized on a microscopic level by combining two techniques. Digital image correlation (DIC), when performed in a scanning electron microscope (SEM) allows for high resolution, in-plane strain to be measured. By combining this with a laser confocal microscope, which is able to characterize out-of-plane displacement, the displacement may be fully characterized. Not only will this allow slip within dislocation channels and grain boundaries to be quantified, but it will also allow the DC-GB intersections to be accurately characterized, as at times in-plane slip is not readily visible when imaged in a microscope. Characterized DC-GB intersections will be divided into cases of continuous slip, discontinuous slip that induced grain boundary slip, and discontinuous slip. Strain will be induced into the sample by a constant extension rate tensile test performed in high temperature (288 °C) argon environment.

After the strain has been characterized, the tensile samples will undergo further straining in a simulated boiling water reactor environment. This will induce cracking in the samples, which will then be linked to the previously characterized DC-GB intersections where the cracking occurs.

Additionally, high resolution EBSD scans were collected at DC-GB intersections in order to perform a cross-correlation and determine the elastic strain tensor. This allowed for stress maps to be created and areas of high stress to be determined. Included with this work as well is a study of cracking susceptibility based on grain boundary character, such as misorientation, Schmid and Taylor factors of adjacent grain, and grain boundary surface trace angle with respect to the tensile axis (the surface trace angle being the angle between the tensile axis and the line where the grain boundary plane intersects the specimen surface).

Table 2.1 Room temperature mechanical properties of some austenitic stainless steels [13].

AISI	Mechanical Properties	
	0.2% Yield strength (MPa)	Ultimate tensile strength (Mpa)
302	215	490-690
304	195	490-690
309	240	540
314	240	540
316	205	500-700
321	205	500-700
347	205	500-700

Table 2.2 Information on carbides that may form in austenitic steel. Of the four listed, $M_{23}C_6$ and MC are the most common [20].

Carbide	Crystal structure	Atoms per unit cell	lattice parameter (nm)	Principal metallic elements
$M_{23}C_6$	fcc	118	a = 1.057 - 1.068	Cr, Fe, Mo, Ni
MC	fcc	8	a = 0.4131 - 0.4698	Ti, Nb, V, Zr
M_6C	fcc	112	a = 1.085 - 1.128	Fe, Mo, Cr
M_7C_3	pseudo hex.	40	a = 0.6928 - 0.6963 c = 0.4541 - 0.4511	Cr, Fe

Table 2.3 Stress corrosion cracking occurrences of stainless steel in BWR environments [28].

Event	Year of detection
Fuel Cladding IGSCC	Late 1950s and early 1960s
IGSCC of 304 during construction	Late 1960s
IGSCC of furnace sensitized 304 during operation	Late 1960s
IGSCC of welded small diameter piping	Mid 1970s
IGSCC of large diameter 304 piping	Late 1970s
Crevice-induced cracking of 304L and 316L	Mid 1980s
Cracking of low carbon and stabilized stainless steels in vessel locations	Late 1980s to present

Table 2.4 Rotation angle and axes for coincidence site lattices of Σ < 30 [95].

Σ	θ (°)	Axis of rotation
1	0	Any
3	60	<111>
5	36.9	<100>
7	38.2	<111>
9	38.9	<110>
11	50.5	<110>
13a	22.6	<100>
13b	27.8	<111>
15	48.2	<210>
17a	28.1	<100>
17b	61.9	<221>
19a	26.5	<110>
19b	46.8	<111>
21a	21.8	<111>
21b	44.4	<211>
23	40.5	<311>
25a	16.3	<100>
25b	51.7	<331>
27a	31.6	<110>
27b	35.4	<210>
29a	43.6	<100>
29b	46.4	<221>

Table 2.5. Empirically derived constants for Pickering, Schramm and Rhodes SFE derivation based on alloy composition.

	X_{const}	X_{Cr}	X_{Ni}	X_{Mn}	X_{Mo}	X_{Si}	X_C	X_N
Pickering [96]	25.7	-0.9	2.0	-1.2	NA	-13.0	410.0	-77.0
Schramm [97]	-53.0	0.7	6.2	3.2	9.3	NA	NA	NA
Rhodes [98]	1.2	0.6	1.4	17.7	NA	-44.70	NA	NA

Table 2.6 Irradiation hardening of several lab purity austenitic steels and one commercial purity 304 steel. Lab purity alloys are designated by their chromium and nickel content. All alloys were irradiated to 5 dpa with protons at an irradiation temperature of 360 °C.

Alloy	Sample ID	Pre-irradiation hardness	Irradiated hardness	Overall hardening
CP 304	Bar 1	163	388	225
	Bar 2	174	410	236
	Bar 3	179	421	242
18Cr-12Ni	Bar 1	147	387	240
	Bar 2	138	394	256
13Cr-15Ni	Bar 1	106	329	223
	Bar 2	122	340	218
	Bar 3	133	308	175
	Bar 4	126	342	216
21Cr-32Ni	Bar 1	122	318	196
	Bar 2	118	331	213
	Bar 3	133	318	185
	Bar 4	150	300	150
	Bar 5	137	336	199
16Cr-12Ni	Bar 1	142	324	182

Table 2.7 RIS in irradiated austenitic stainless steel. Bulk composition is shown compared to composition at the grain boundary (gb).

	304 steel [66]	304L steel [6]	304 steel [99]	304 steel [99]	348 steel [99]	316 steel [6]	PCA steel [100]
Irradiation Temp.	400 °C	400 °C	290 °C	290 °C	290 °C	400 °C	420 °C
dose (dpa)	1 (3.2 MeV protons)	0.8 (E>1 MeV nuetrons)	3.5 (E>1 MeV nuetrons)	10 (E>1 MeV nuetrons)	3.5 (E>1 MeV nuetrons)	0.8 (E>1 MeV nuetrons)	9 (E>1 MeV nuetrons)
Cr bulk	19	18.1	18.9	18.9	18.2	16.7	16.6
Cr gb	13	11.3	17.5	17.2	17.4	14.9	11.8
Ni bulk	8.5	8.7	11.4	11.4	10.6	11.1	15.3
Ni gb	12	20.7	15.2	18.8	12.9	22.1	24
Si bulk	-	0.52	0.05	0.05	0.09	0.66	1.17
Si gb	-	1.9	0.5	0.5	0.5	0.88	4.4
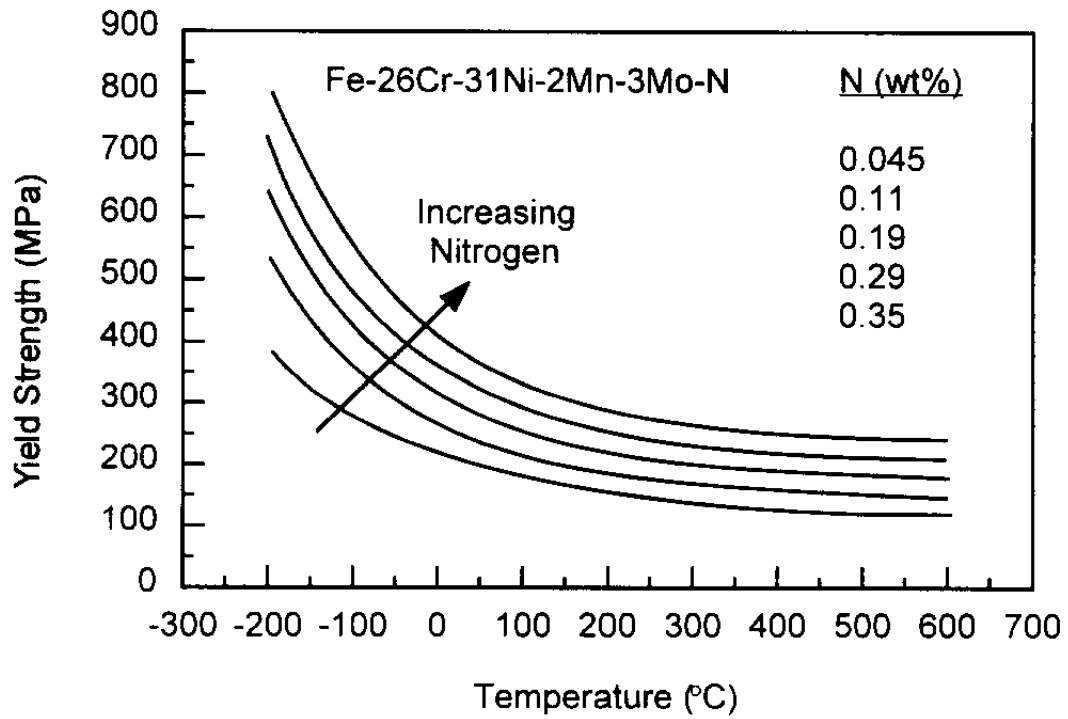
Figure 2.1 Effects of nitrogen content and temperature on yield strength of a 26 Cr, 31 Ni austenitic steel [101].
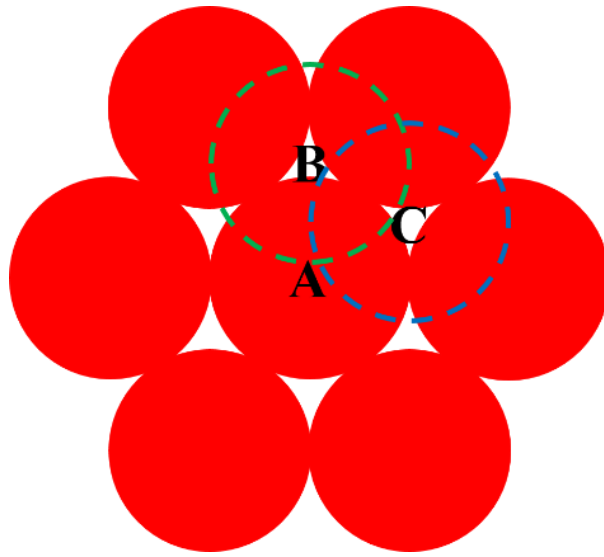
Figure 2.2 Close packed {111} planes in an FCC crystal.
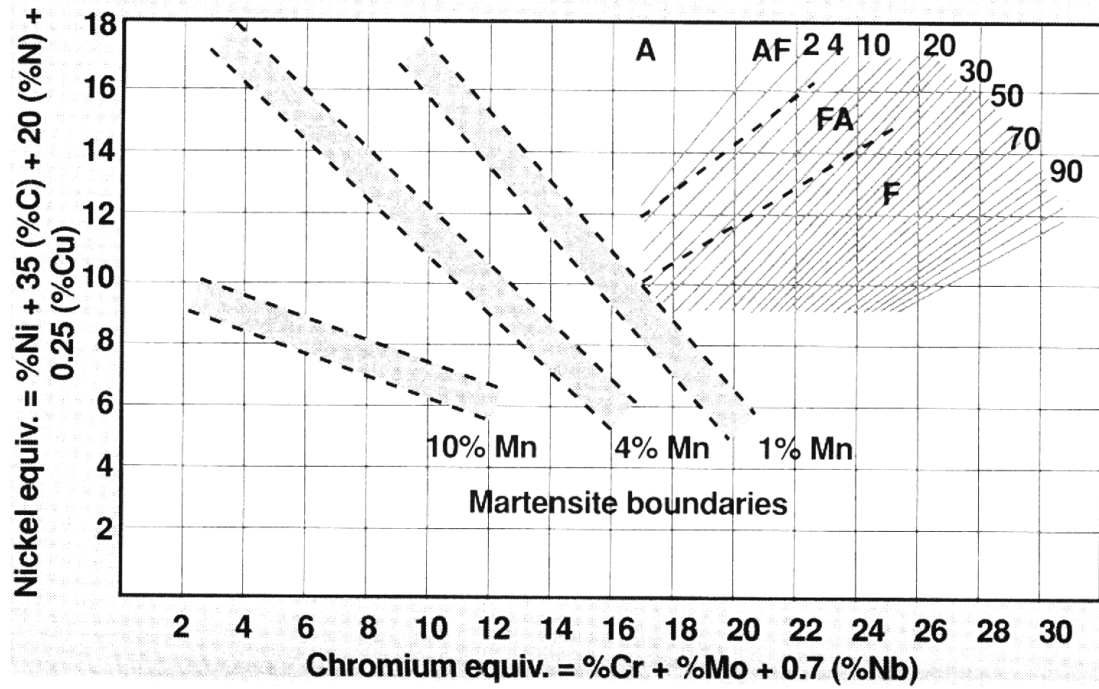
Figure 2.3 Phase diagram showing percent austenite, ferrite and martensite based on steel composition [18].
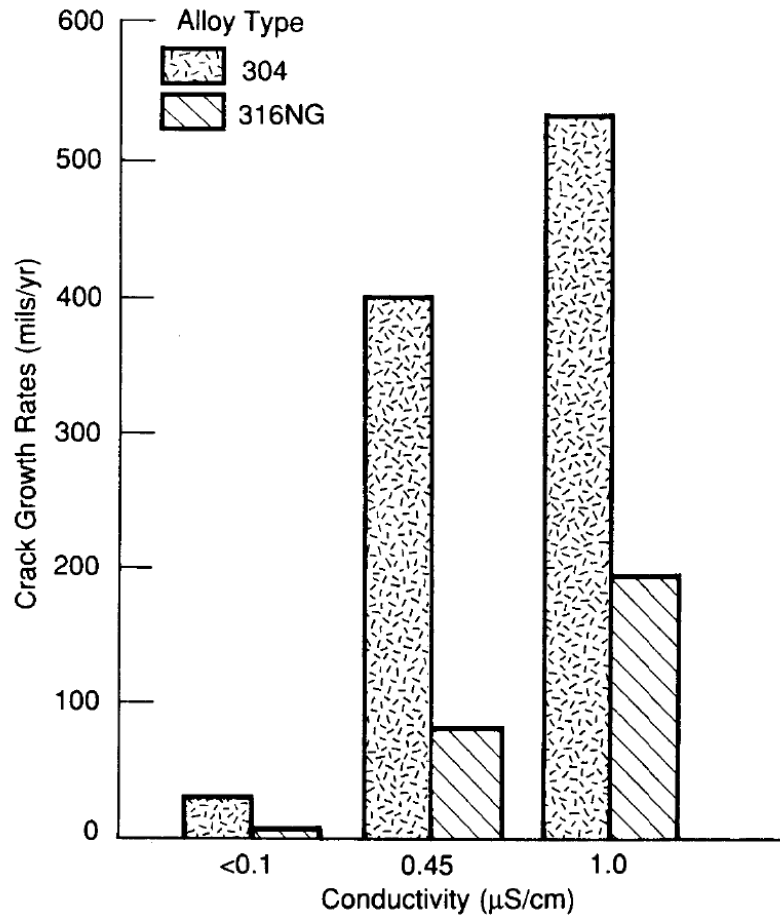
Figure 2.4 Crack growth rates as a function of water conductivity in 288 °C water with 200ppb oxygen [26].
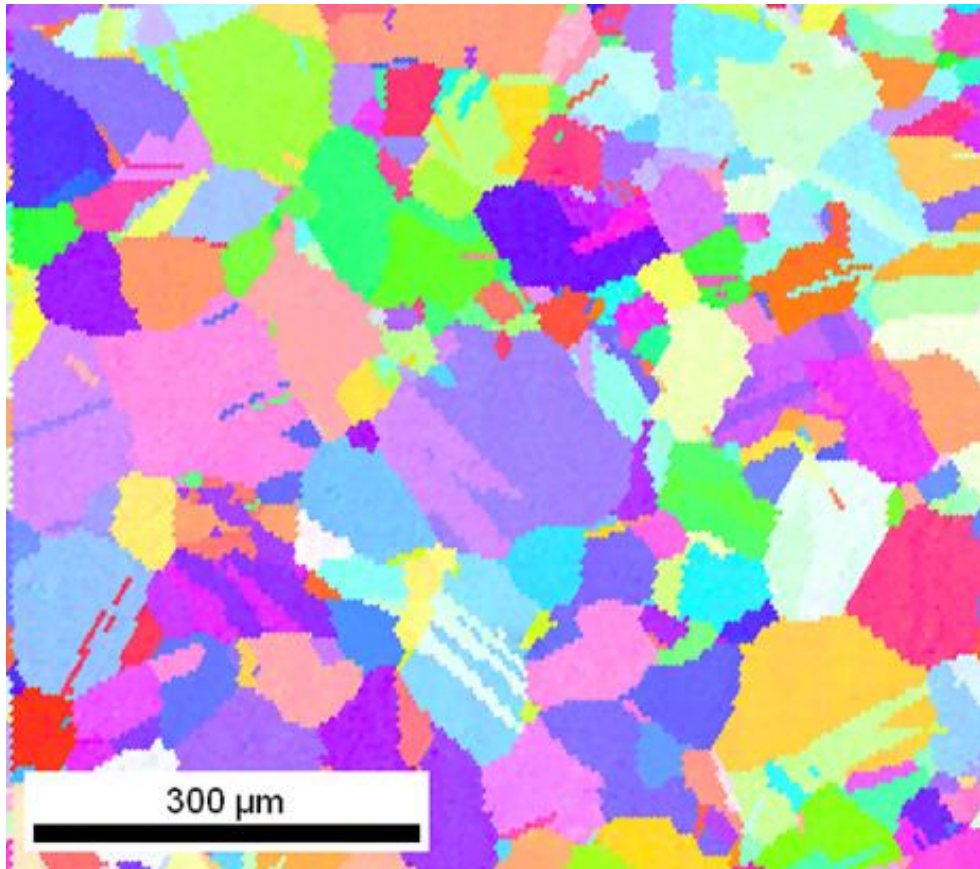
Color Coded Map Type: Inverse Pole Figure [001]

Steel Austenitic
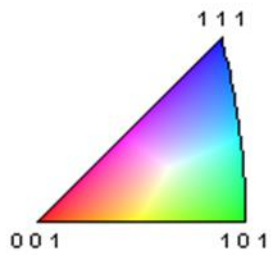
Figure 2.5 Orientation map depicting a section of austenitic stainless steel.
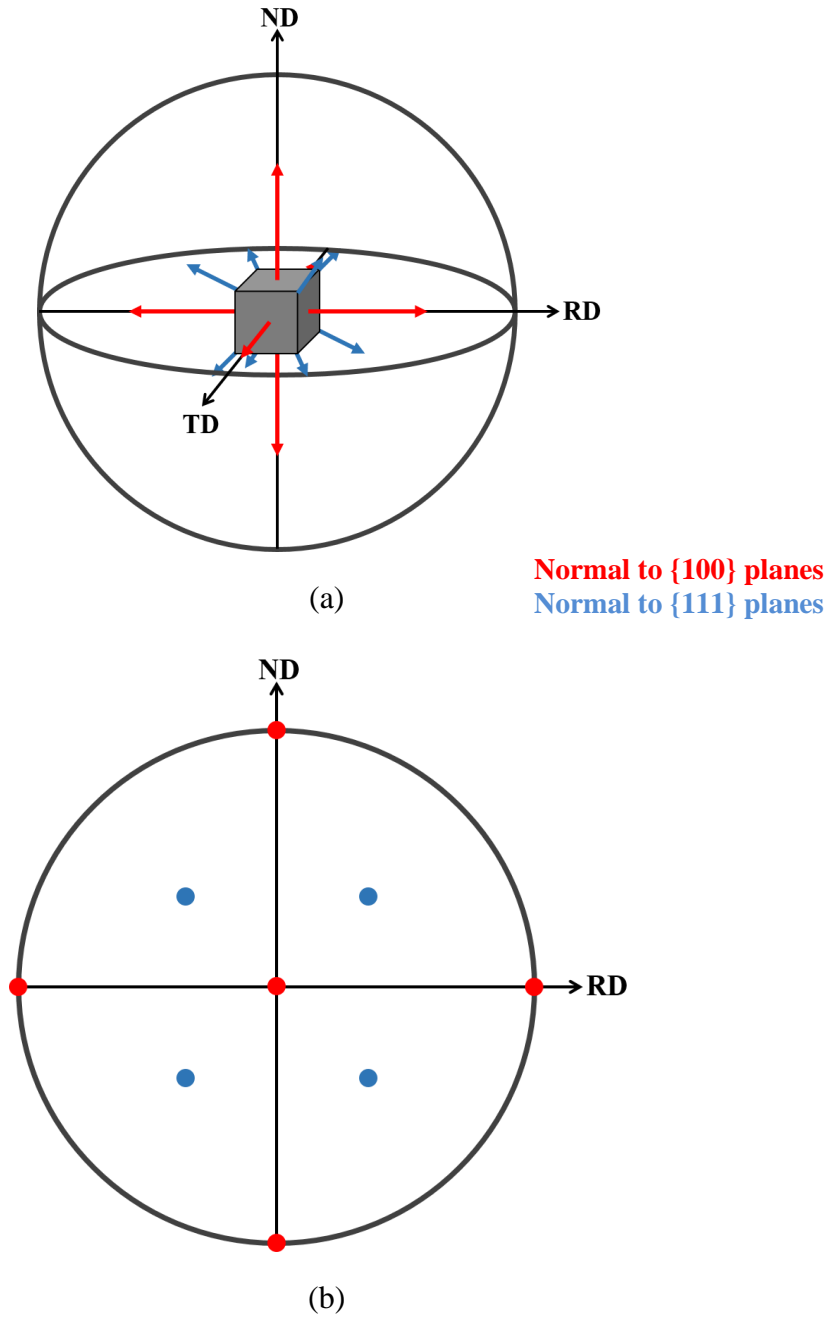
Figure 2.6 Pole figure demonstration. (a) depicts a unit cell of the crystal, with arrows showing normal to {100} and {111} planes. (b) depicts the stereographic projection of (a), which is the pole figure.
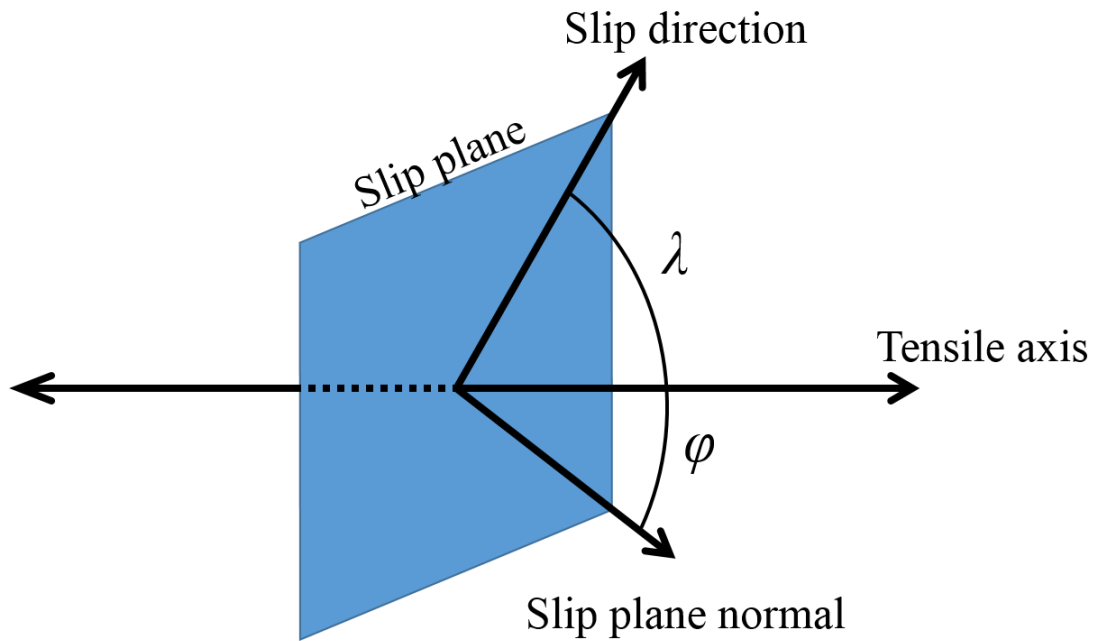
Figure 2.7 Schematic showing angles used in Eq. 2.1. λ is the angle between the slip direction and the tensile axis, and φ is the angle between the slip plane normal and the tensile axis.
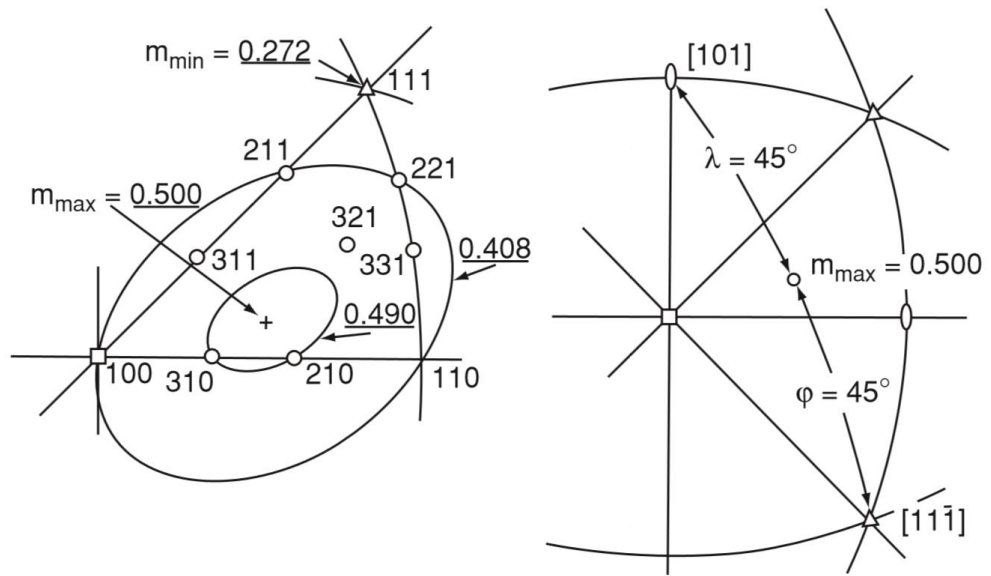
Figure 2.8 Schmid factor based on crystal orientation in an FCC structure [32].
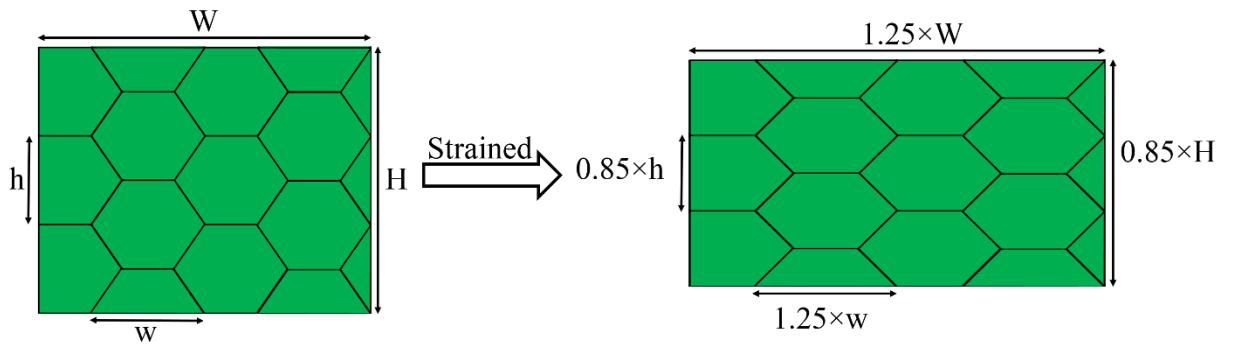
Figure 2.9 Representation of Taylor's [30] assumption that each grain in a polycrystalline material undergoes the an equivalent deformation as the full macroscopic material.
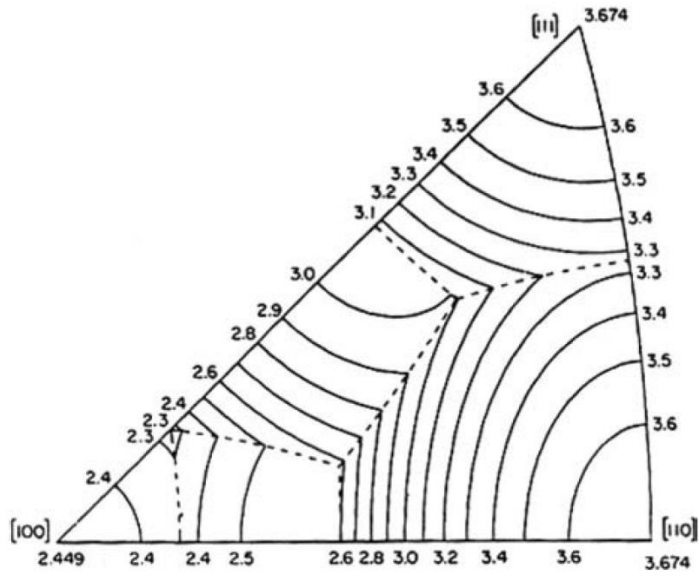
Figure 2.10 Grain orientation dependence of the Taylor factor for fcc materials [102].
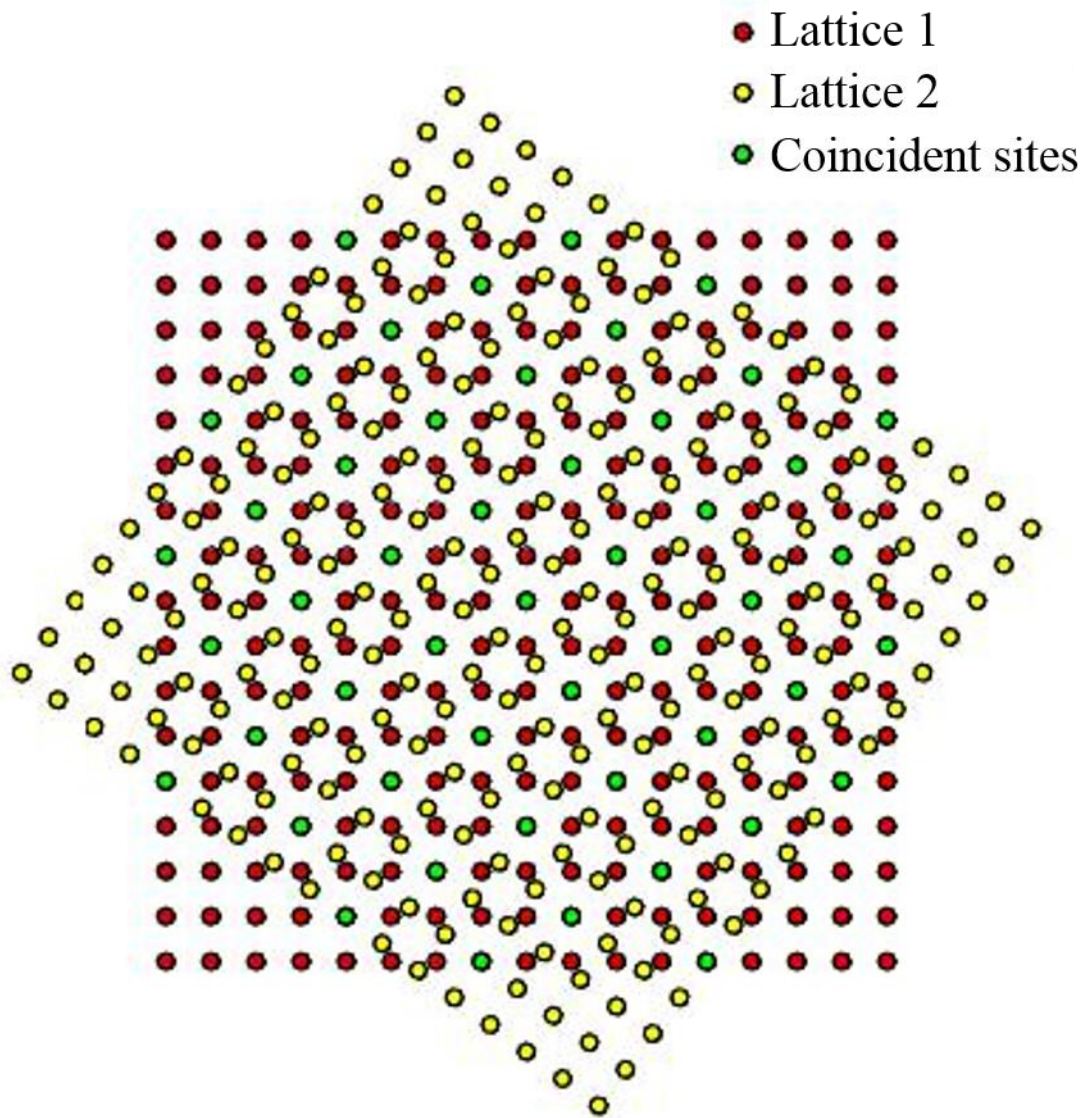
Figure 2.11 Depiction of the coincident sites if adjacent lattices from a Σ5 boundary are overlapped on top of each other.
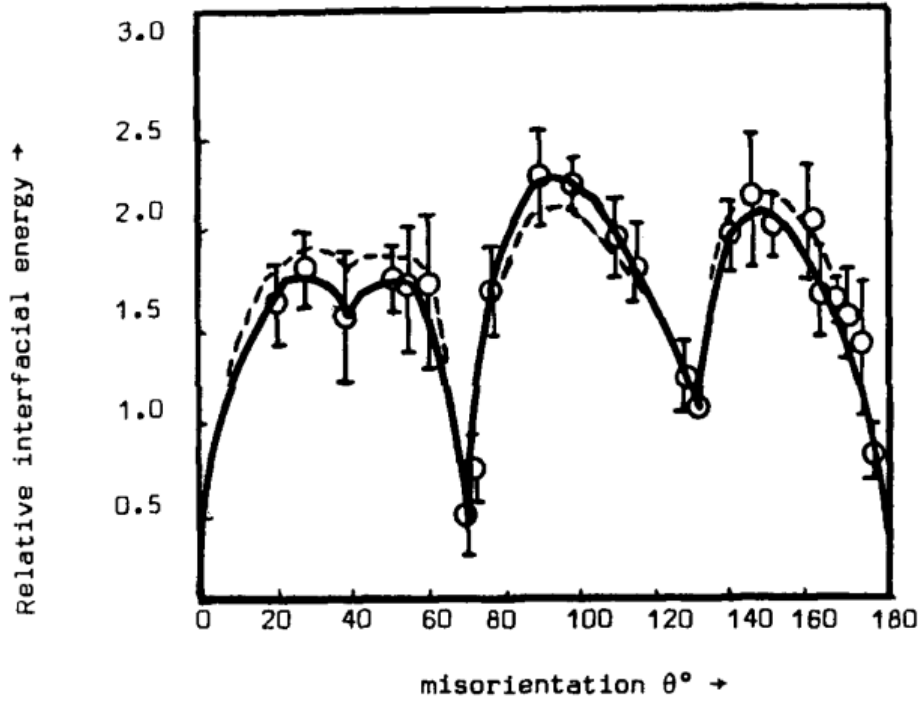
Figure 2.12 Measured relative energies of [110] tilt boundaries in aluminium. All boundary energies are reference to the 129° boundary. The 70° boundary is equivalent to a (111) twin plane, as would be seen in austenitic steel. [35]
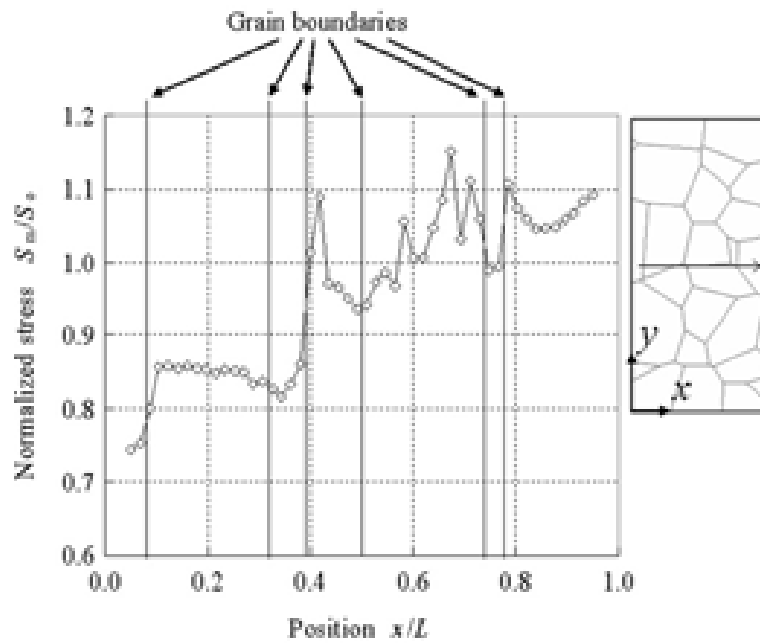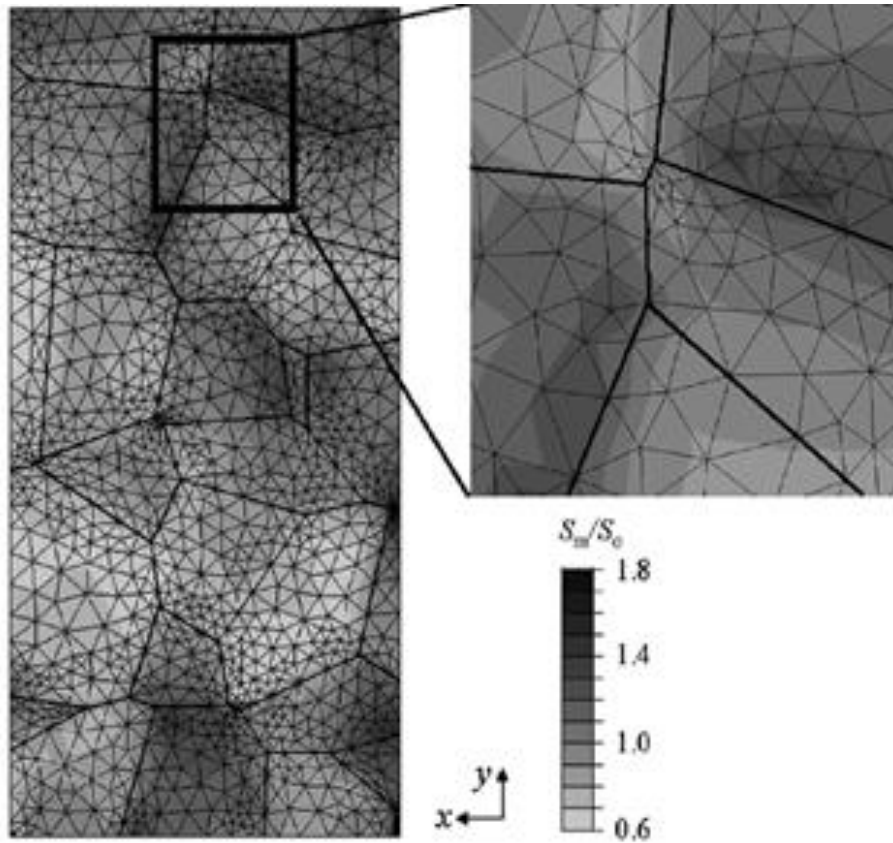
Figure 2.13 Mises equivalent stress distribution from a finite element analysis of a polycrystalline material, showing stress peaks at grain boundaries [40].

60
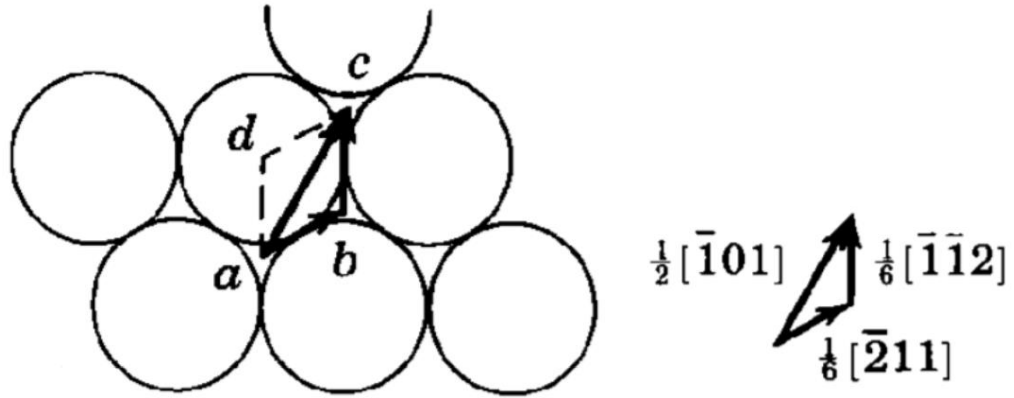
Figure 2.14 Schematic showing a [111] glide plane and the movement of a perfect dislocation above it from a to c being divided into partials which move from a to b and then to c [103].
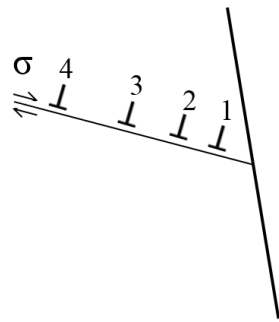
Discontinuous slip
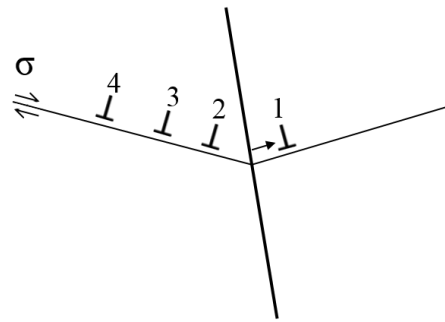
Direct slip transmission
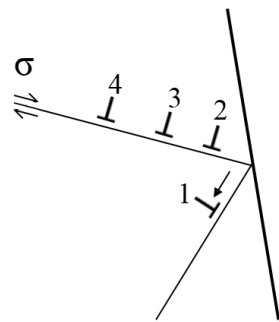
Cross slip
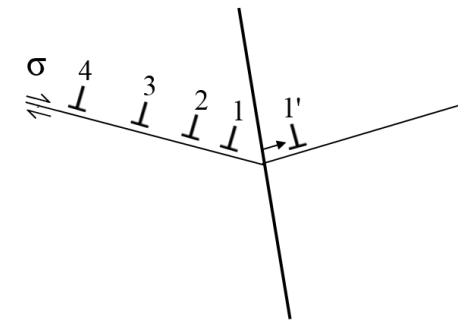
Nucleation at boundary
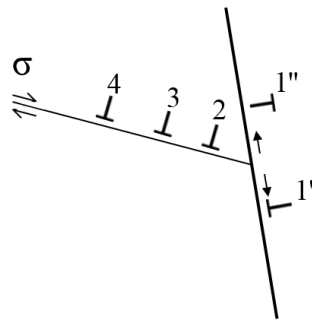
Boundary absorption
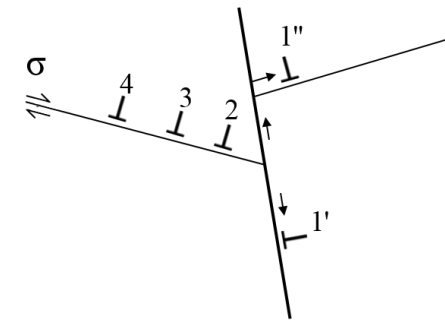
Absorption and emission

Figure 2.15 Methods of slip accommodation at a grain boundary [45,47].
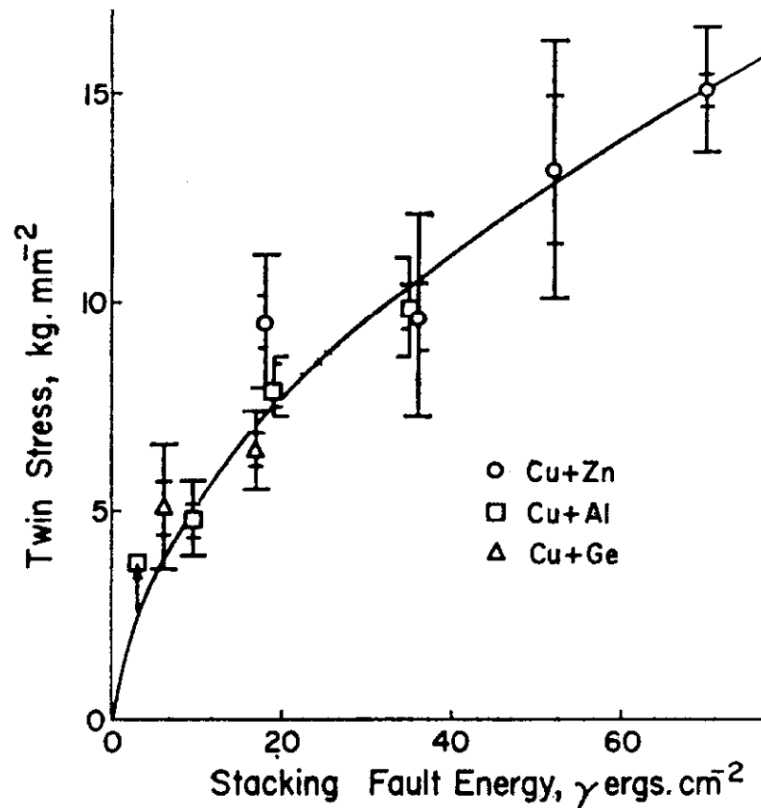
Figure 2.16 Deformation twinning of copper alloys were tested by Venables [55] and lower SFE materials were found to tend towards lower twinning stresses, resulting in easier deformation twin nucleation.
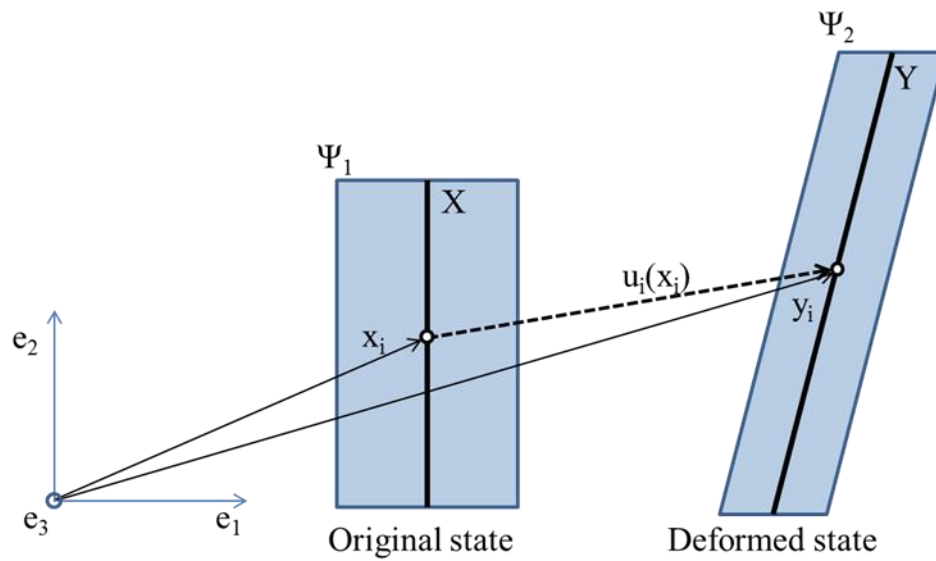
Figure 2.17 Example of deformation: Shape Ψ_1 deforms into Ψ_2. These shapes are three dimensional and extend into the page. Plane X is depicted within Ψ_1, and deforms into plane Y within Ψ_2. A point on plane X is described by vector x_i, where i has values of 1, 2, and 3, and represent the Cartesian components of x, with 1 being the horizontal component, 2 being the vertical component, and 3 being the component normal to the paper (represented by the e_i axis). x_i deforms to vector y_i by the displacement vector $u_i(x_i)$
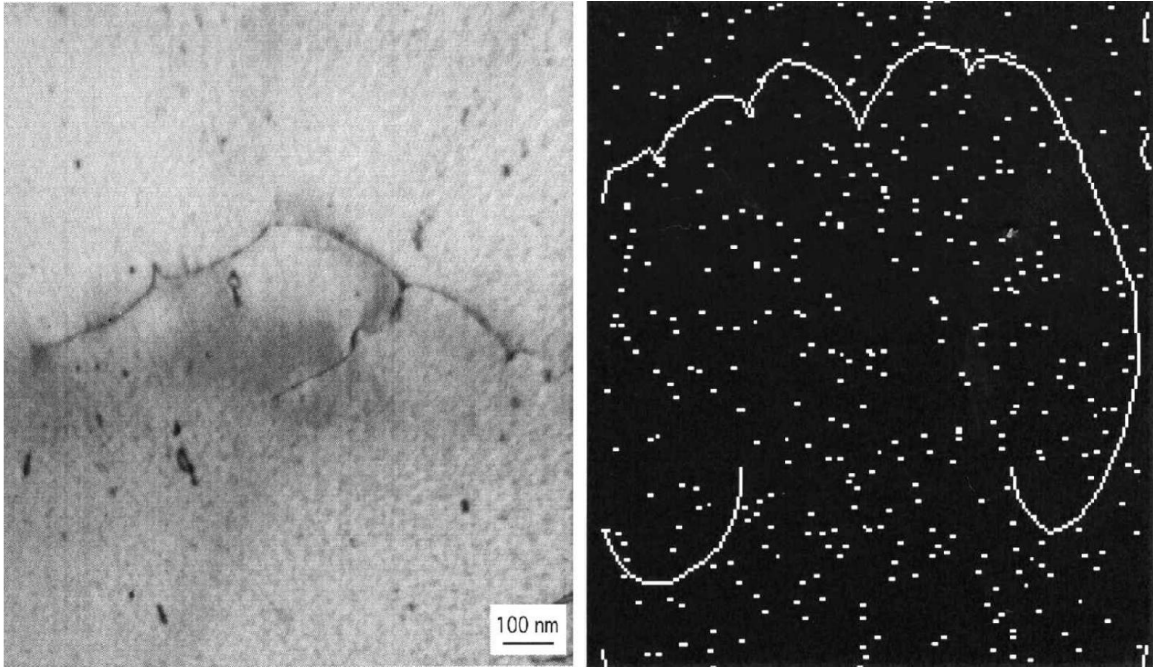
Figure 2.18 A dislocation line being pinned by defects, as seen by in a TEM image (left) and as seen using a computer simulation [104].
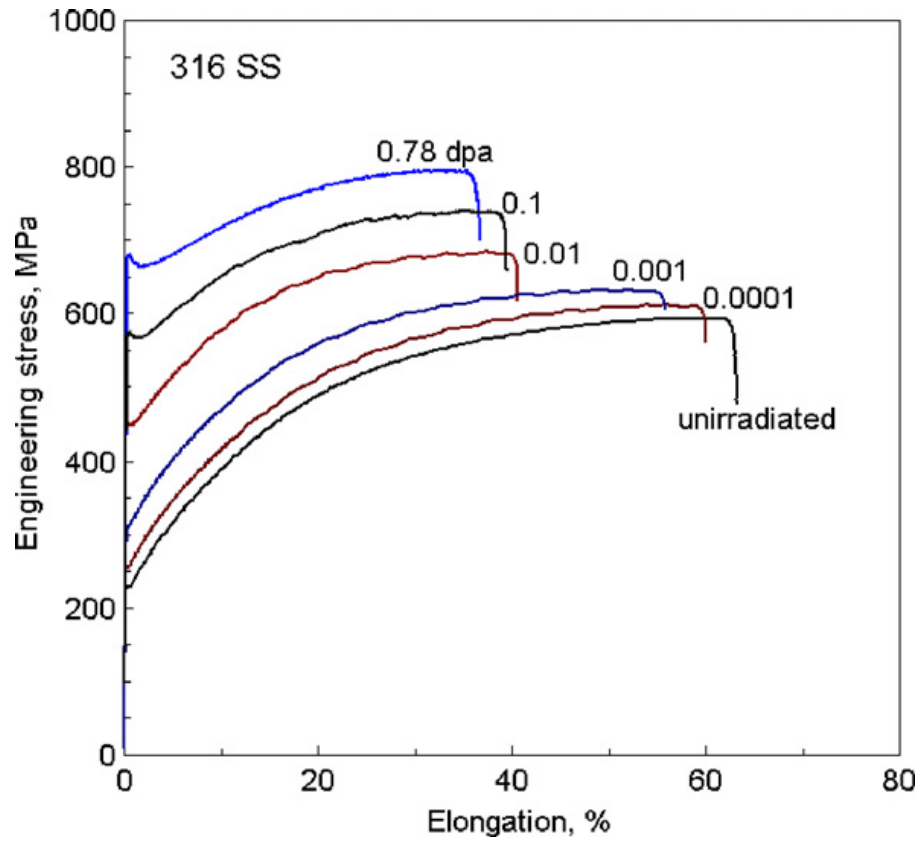
Figure 2.19 Room temperature stress-strain curves for 316 steel irradiated up to 0.78 dpa [72].
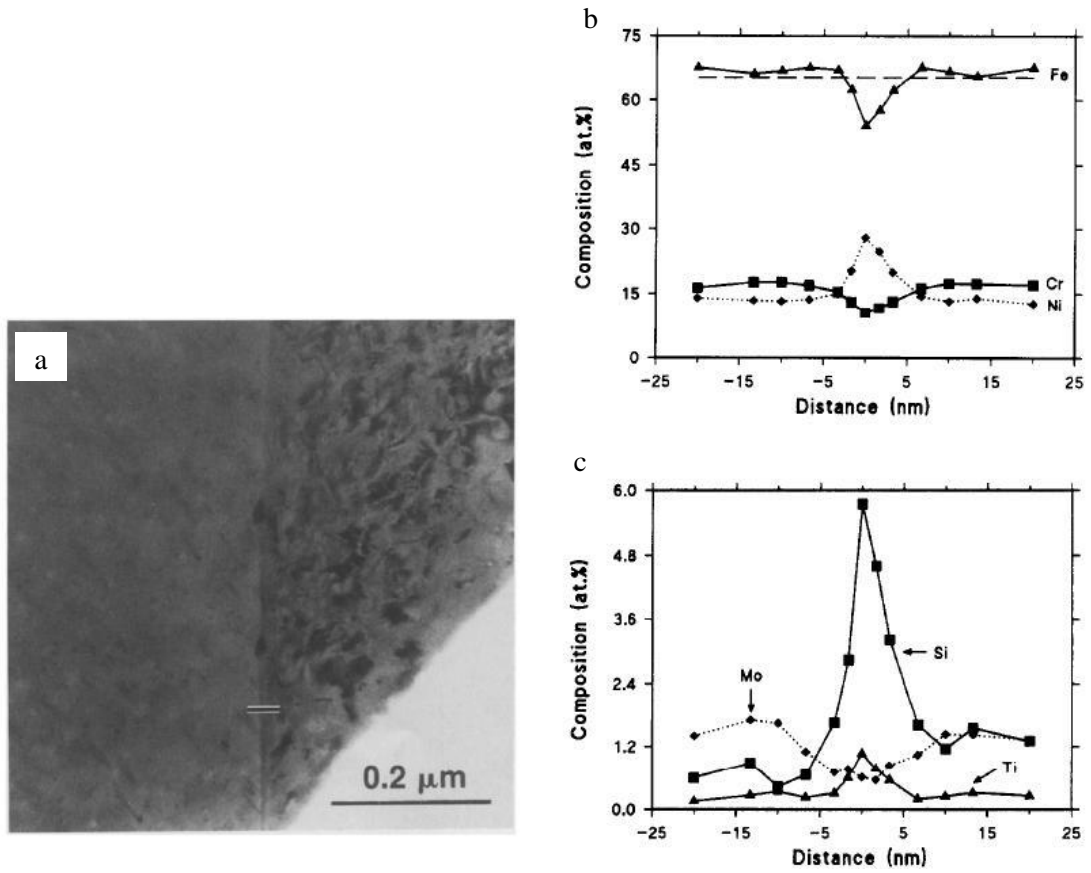
Figure 2.20 Composition across grain boundary (a), showing major (b) and minor (c) alloying elements. Dashed line in (b) indicates bulk iron concentration [100].
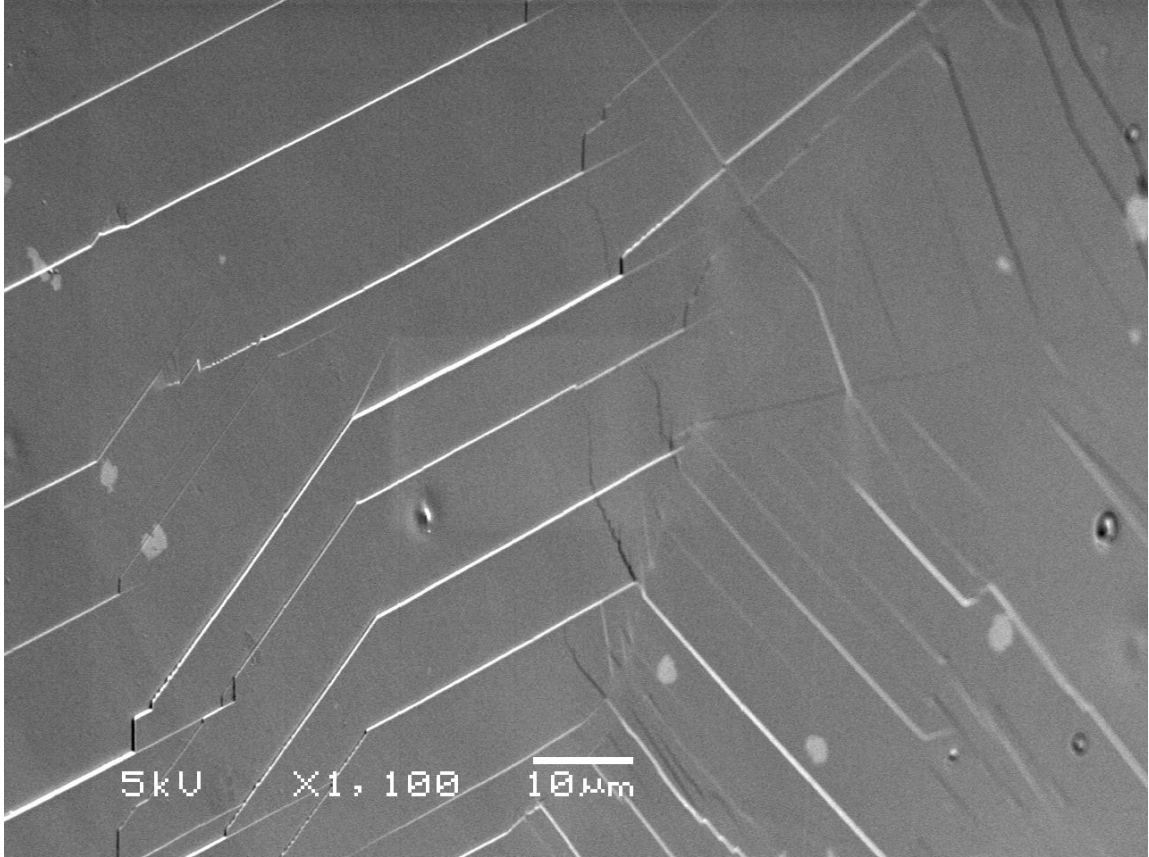
Figure 2.21 Slip steps from dislocation channels visible on the surface of a 5 dpa proton irradiated austenitic steel with 21 Cr, 32Ni strained to ~2% in a 288°C argon environment.
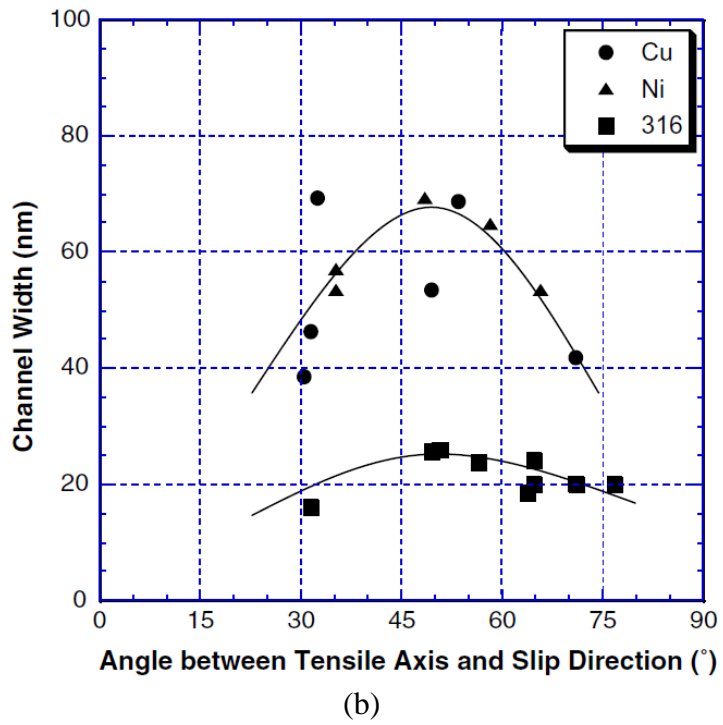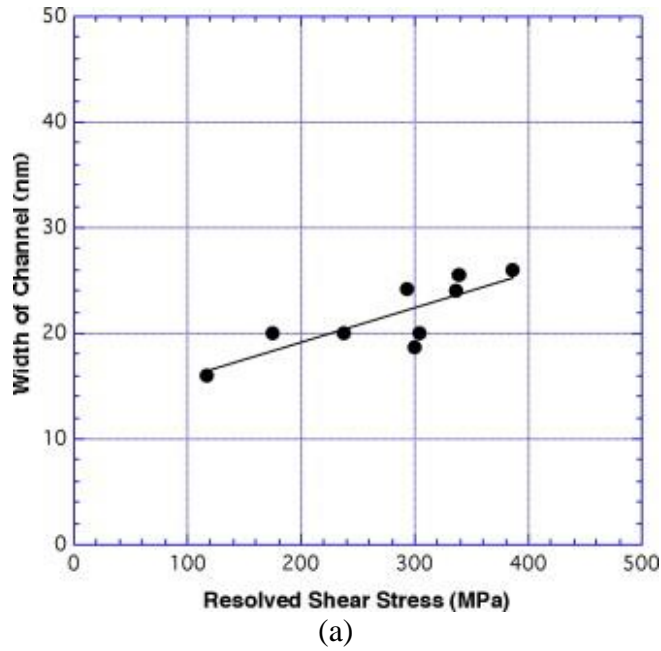
Figure 2.22   (a) Channel width versus resolved shear stress in a 316 SS [72] and (b) channel width versus angle between tensile axis and slip direction [105]. In both graphs, material has been irradiated to less than 1 dpa.
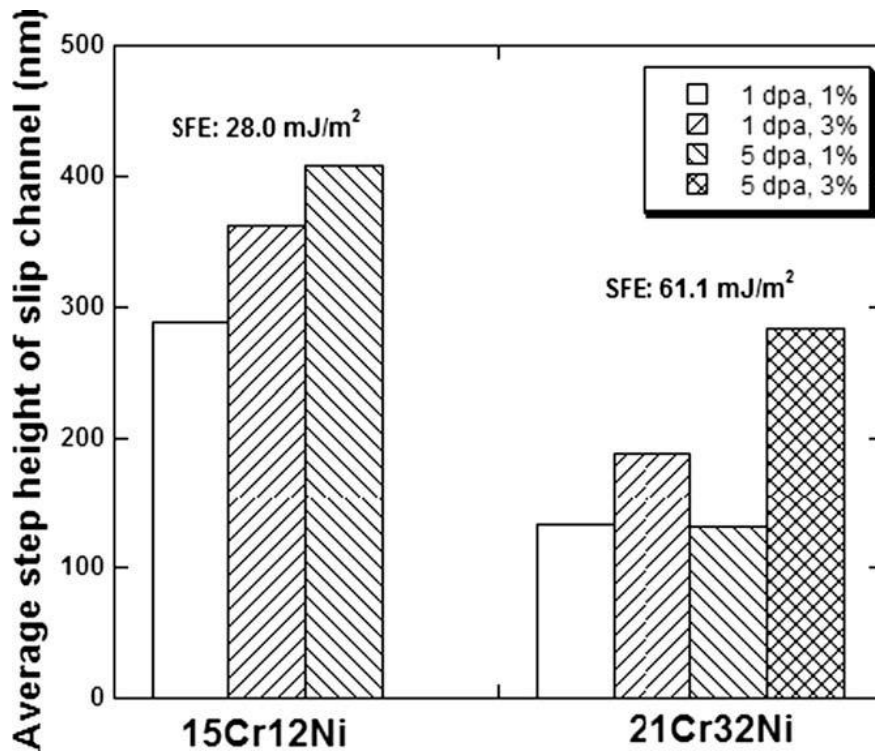
Figure 2.23 Average channel height of strained irradiated austenitic alloys of different stacking fault energies [80].
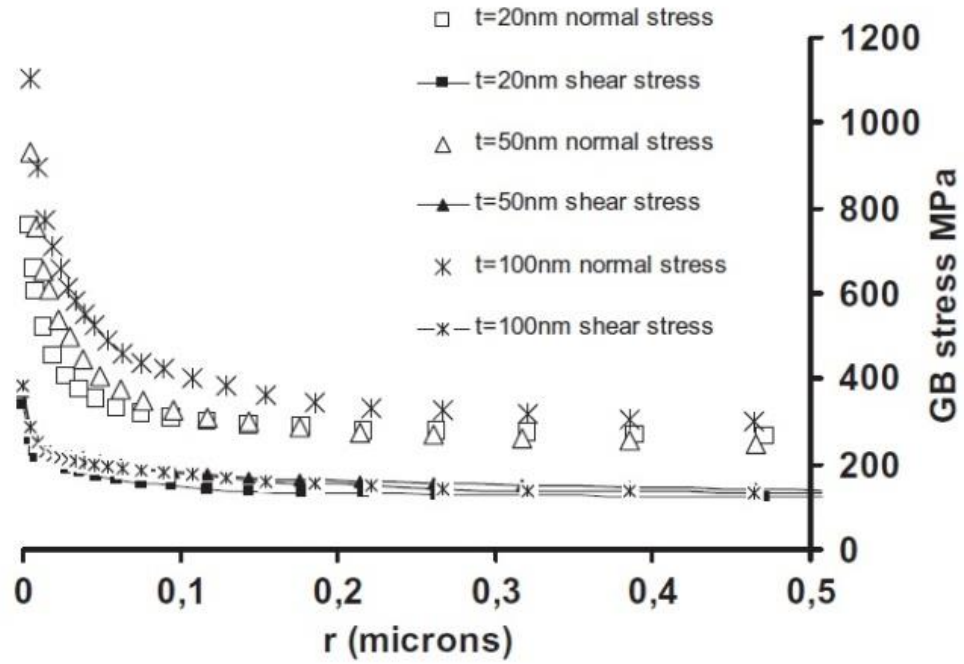
Figure 2.24 Effect of channel width, or thickness, (t) on stress along the grain boundary at r distance from the GB-channel intersection using a finite element model [83].
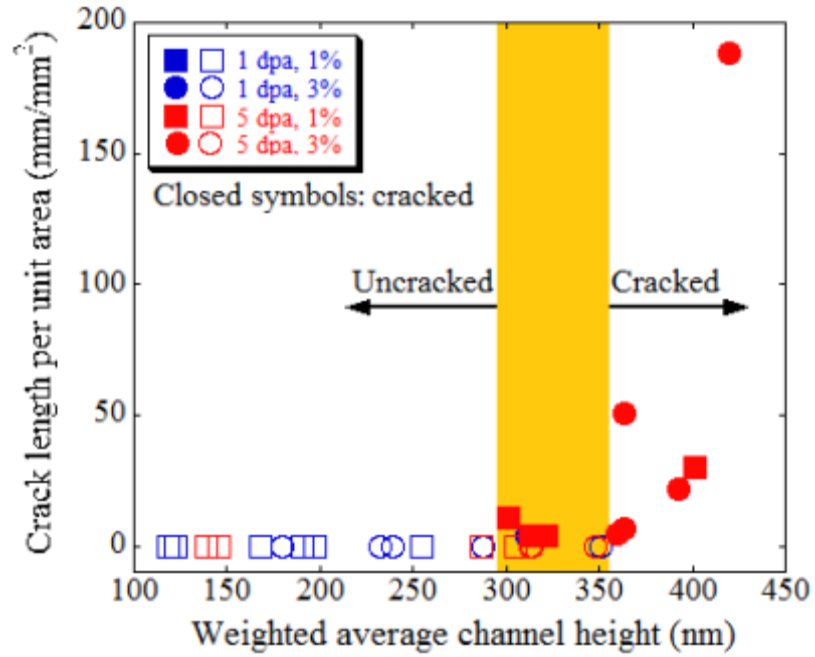
Figure 2.25 Correlation between IASCC and weighted averaged dislocation channel height, suggesting a strong connection between the degree of localized deformation and IASCC [7].
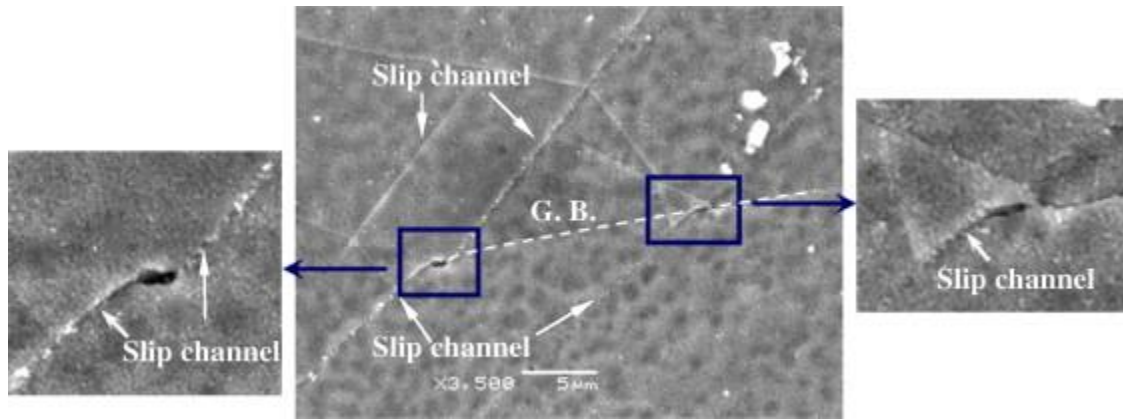
Figure 2.26 Micro-cracks initiated at DC-GB intersections in a proton irradiated 15Cr12Ni stainless steel strained to 1% in a simulated BWR environment [80].
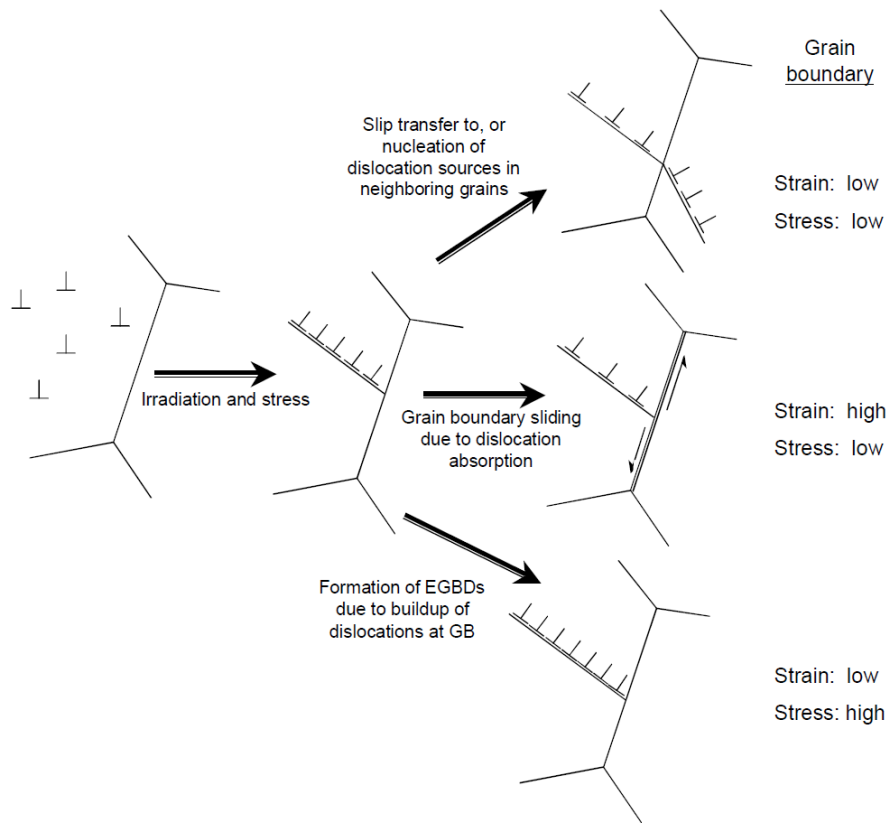
Figure 2.27 Options for dislocation channel interaction with a grain boundary, as proposed by Was, et al. [90]. Cracking propensity was believed to low for slip transfer and high for GB sliding and dislocation pileups, based on correlations in cracking data.

Chapter 3

EXPERIMENTAL PROCEDURES

This study aims to increase understanding of the connection between localized deformation and IASCC. This was accomplished by first performing a preliminary cracking study, to determine where cracking is occurring. In a separate set of experiments, full characterization of the plastic strain of the dislocation channels and the grain boundaries near the DC-GB intersections was performed. Elastic strain/stress was also analyzed at these intersections. The strain and stress measurements was then compared to DC-GB intersections that crack to determine the controlling cracking mechanism.

3.1 Alloy description

Five austenitic stainless steel alloys were be used in this study, with compositions as seen in Table 3.1. The four lab purity alloys, 13Cr15Ni, 21Cr32Ni, 16Cr12Ni, and 18Cr12Ni1Si were selected due to their range in stacking fault energies and expected cracking susceptibility. All four were pure austenite, with no ferrite detected (alloys were non-magnetic). One commercial alloy was also be tested, 304L, which has been used in nuclear BWR reactors for components such as the core shroud, which has been found to undergo IASCC [84]. The 16Cr12Ni was provided by Ames Laboratory and came in the

form of an annealed extruded rod. The other four alloys were annealed plates provided by General Electric Global Research (GEGR).

3.2 Sample preparation and characterization

3.2.1 Sample geometry

Tensile bars were cut using electrical discharge machining. The majority of the tensile bars were cut with a 23 mm gage section length and a cross section of 2.5 mm by 1.5 mm. The 16Cr12Ni and 18Cr12Ni1Si bars were cut to slightly smaller dimensions, with a 21 mm gage section length, and a 2.0 mm by 1.5 mm cross section. The full dimensions of the larger samples are shown in Figure 3.1. Electrical discharge machining was used as the cutting method in order to reduce mechanical damage depth in the bars. With the exception of the 16Cr12Ni alloy, the tensile bars were cut with the tensile direction parallel to the rolling direction. The 16Cr12Ni was cut from an extruded rod, with the tensile direction parallel to the length of the rod.

3.2.2 Polishing method

The samples were prepared first by a mechanical polish. Rough grit silicon carbide paper was used to grind the surface of the tensile bars so that all were even, and the EDM damage was removed. Successive polishes at finer grits removed the damage from the previous grit. Following the final silicon carbide paper (P4000, median SiC particle diameter of 2.5 µm), electropolishing was performed at 30V in a solution of 10% perchloric acid and 90% methanol at -40 °C for 90 seconds. A stainless steel cathode was used, which resulted in a current of 0.1-0.2 A. The electropolishing step was to remove the mechanical damage from mechanical polish, and give the sample a damage free, mirror like surface.

3.2.3 Orientation imaging microscopy (OIM)

After polishing, the tensile bars were divided into segments by placing indents at regular intervals, creating rectangles along the gage section (1.5mm by 0.7mm), marked with an indent in each corner. These were used as fiducial marks to create a reference system to locate grains and areas of interest. This was allow the same area to be located in multiple characterization methods.

The gage section of the tensile bars were characterized using a coarse EBSD scan in the SEM to determine grain orientation. This EBSD analysis was performed at the Michigan Electron Microscope Analysis Laboratory, using a Philips XL30 FEG SEM with a TSL OIM system for automated EBSD measurements. A schematic of the EBSD setup is shown in Figure 3.2. To perform this analysis, the tensile bars were tilted so that the surface normal was 70° to the electron beam. The electrons in the beam strike the surface and are scattered both elastically and inelastically in different directions. Some of the inelastically scattered electrons are scattered in directions such that Bragg's criterion is satisfied (depicted graphically in Figure 3.3), where the electron wavelength (λ) is related to the atomic plane spacing (d) and the angle of the incoming electrons (θ) are related by

$$\lambda = 2d \sin \theta .$$
(3.1)

These electrons that satisfy the Bragg criterion are scattered elastically and impinge on the EBSD detector, composed mainly of phosphor screen, which emits photons when electrons impinge on it, and a CCD camera to collect the photons and the resulting EBSD pattern.

The inelastically scattered electrons move in all directions, so the subsequent elastic scattering sends electrons out in a cone shape. As only a small portion of the cone intersects the screen, it appears as a straight line, or band, which is referred to as a diffraction band, or Kikuchi band. Each band is associated with a lattice plane (the plane where the Bragg's diffraction originally occurred).

The movement of the electron beam was controlled, such that it moved in steps of 5 μm in a raster pattern over a selected area. At each step, the beam was held for 0.03 seconds to collect EBSD patterns from that location. At each collection point, the TSL OIM^{TM} Data Collection Software automatically indexes the Kikuchi bands using a Hough transform to change the Kikuchi bands into points, which are read automatically by the computer. Based on the determined EBSD pattern, the software determines the Euler angles of the lattice.

The EBSD results were analyzed with the TSL OIM^{TM} Analysis Software to create orientation maps of the grains, based on the measured Euler angles. A legend is created by color coding the pole figure, such that each orientation refers to a particular point on the pole figure, and is colored according to that position. This is typically called an inverse pole figure map, as the pole figure is used to determine the crystal orientation, rather than the crystal orientation being used to create a pole figure, such as has been shown previously in Figure 2.5. These orientation maps were also used to determine grain boundary misorientations (based on the difference in orientation of the grains on either side of the boundary), as well as Schmid factors (SF) and Taylor factors (TF), Taylor and Schmid factors are calculated using the measured orientation data and Equations (2.5) and (2.9). Every grain has a single Taylor factor value, as it is calculated

based on the five slip systems that best allow macroscopic strain to be accommodated within the grain. Each slip system, however, possesses a unique Schmid factor, resulting 12 possible Schmid factors for each grain in austenitic steel. In this work, the Schmid factor of a grain refers to the Schmid factor of the slip system with the highest Schmid factor, so that every grain is referred to by its maximum Schmid factor.

Schmid and Taylor factors were divided into three bins, low (L), medium (M), and high (H). The limits of the bin were set so that approximately one third of the grains were allocated to each bin, with the max bin values for Schmid factor set to 0.44, 0.47 0.5 for L, M, and H, respectively. For Taylor factor, 2.89, 3.32, 3.67 were used for the maximum limits of L, M, H. Grain boundaries were characterized by the Schmid factors of the adjacent grains. A boundary is described by the combination of the bins of the adjacent grains so that a grain boundary adjacent to grains with "L" and "M" Schmid factors is denoted as "LM".

Occasionally, EBSD patterns were not high enough quality to be indexed by the program. In these cases, erroneous orientations are recorded by the software. These were removed from the results by following a cleaning procedure. The first step in the cleanup procedure involved removing points with a confidence index (CI) below 0.1. The CI is a value that is saved along with the orientation of each EBSD pattern analyzed, and measures the confidence of the software in the results. A CI of 0.1 refers to a pattern index that will be correct 90-95% of the time. Lower CIs result in lower confidence, so by removing these points, each point has more than a 90% chance of being correctly indexed. The second step in the cleanup was to restrict the grain size so that a grain must contain more than 2 adjacent data points. This removes additional errors by removing

points where a single location was incorrectly indexed, resulting in a grain of just one indexed point. Finally, for data where each grain was assigned a single value (such as SF or TF), each grain was given a single, average orientation. As defects and dislocations can cause lattice rotation, it is common to see slight changes in orientation throughout a single grain. This final cleanup step gives an average orientation value for each grain, and as such, an average SF and TF. For information where change in orientation within the grain was important, this final step was not performed.

3.2.4 Proton irradiation

After polishing and EBSD characterization, all tensile bars were irradiated using the 1.7 MV Tandetron accelerator at the Michigan Ion Beam Laboratory, located at the University of Michigan. The temperature was monitored by a two-dimensional thermal imager (IRCON® Stinger thermal imaging system) that tracked surface temperatures of the samples at high spatial resolution throughout the irradiation. Three areas of interest (AOIs) were created on each bar, where the temperature was tracked and controlled using electrical current into a heater behind the stage, and air flow, also behind the stage, to adjust sample temperature. AOI temperatures were controlled to maintain the sample temperature to within 9°C of the set temperature (360°C).

The samples were irradiated to 5 dpa at a rate of ~0.8 dpa/day with 2-3.2 MeV protons at 360°C, which creates an irradiated surface to a depth of 20-40 μm. The damage rate profile is shown in Figure 3.4, as determined using the full cascade model in SRIM^{TM}. It should be noted that recent results have found that the full cascade method over calculates the damage[106]. These damage rates should be compared to only others which have been calculated using the full cascade method. The damage rates, as well as

the total damage, were calculated based on the region of the profile that was relatively flat, away from the peaks at 20 μm for 2 protons or 40 μm for 3 MeV protons. For 2 MeV protons, this is the area around 12 μm depth from the sample surface, for 3 MeV, it is a depth of about 24 μm.

The damage was calculated based on the measured current of the proton beam impinging on the sample surface. From this current, the number of counts of protons impinging on the sample were recorded by a computer monitoring system. The number of counts per dpa was determined as

$$\frac{counts}{dpa} = \frac{N \times A \times q}{rate_{displacement}},$$

(3.2)

where N is the atomic density of the steel (8.8×10^{22} cm^{-3} was used as a general value for steel), q is the charge per ion, in this case protons (1.6×10^{-19} C), A is the irradiated surface area, and the displacement rate was determined from SRIM^{TM}.

3.2.5 Gold nano-particle deposition

Digital image correlation (DIC) was used to map plastic strain on a microscopic level across the tensile bar surface. This refers to the matching of similar features between two images, in this case, one prior to, and one after straining. This will be discussed in more detail in the following section. For this correlation to occur, a pattern must exist on the sample surface. As the tensile bars were polished to mirror-like finish, no features existed that could be used for the correlation. Gold nano-particles were deposited in order to create a pattern, as part of the sample preparation, which will be discussed here.

Gold nano-particle suspensions (particles ~40 nm in diameter) were created using the method developed by Frens [107]. 200 mL of a solution of 10^{-2}%, by weight, of $HAuCl_4$ in distilled water was heated until boiling (~100°C). Once this solution had reached boiling temperatures, 2 mL of a solution of 1% Na_3-citrate by weight in distilled water was added. The citrate reduced the gold chloride and created nano-spheres of gold, roughly 40 nm in diameter after about 5 minutes.

Gold nanoparticle deposition to the surface used a technique developed originally for surface enhanced Raman [108], but used by Kammers [109] for a DIC speckle pattern. The tensile bars were first soaked in a solution of 20% (3-aminopropyl) trimethoxysilane in methanol. The methoxy groups react with the hydroxyl groups in the metal oxide, to create firmly attached silane layer over the sample surface.

After soaking in the silane solution, the sample was rinsed with methanol to remove excess silane, and placed in the gold nano-particle suspension. The amines in the silanes attached to the gold, creating a random speckle pattern on the steel surface. A random speckle pattern is ideal for DIC, as it allows for higher resolution than repeating patterns, such as grids. This process is depicted in Figure 3.5. The small particle size, high contrast of gold with respect to steel in the SEM, and the stability of how the particles are adhered to the surface make this an ideal method for creating an SEM DIC speckle pattern.

In the initial depositions on the ASCC2 sample, SEM analysis determined that the gold particle density varied greatly across the sample surface, from areas with zero

particles/μm^2, to areas of 120 particles/μm^2, with an average of about 45 particles/μm^2. An example of the gold speckle pattern from this sample may be seen in Figure 3.6a.

In this case, after the gold deposition, following the procedure outlined in the experimental section of this thesis, the ASCC2 tensile bar was cleaned in an ultrasonic cleaner in acetone, methanol and distilled water, prior to imaging in the SEM. All other subsequent DIC samples were not cleaned in the ultrasonic cleaner directly after gold deposition, but rather were rinsed by flowing the solvents over the sample in a beaker. This improved the gold particle density significantly, given a uniform high density (~120 particles/μm^2) over the entire surface, as shown in Figure 3.6b. During the analysis of the ASCC2 sample, DC-GB intersections near areas of low particle density were avoided and not characterized.

3.3 Constant extension rate tensile (CERT) tests

All CERT tests were performed in the High Temperature Corrosion Laboratory at the University of Michigan. These tests were performed in autoclaves where the environment could be carefully controlled and up to four tensile bars could be strained at once, each with an individual load cell for stress measurements on each sample. The four pull rods attached to the samples were also connected to a moveable crosshead, where a linear variable differential transformer (LVDT) was affixed to measure displacement. A general depiction of these systems is shown in Figure 3.7. Two environments were used for this study: an inert argon environment, and a simulated BWR NWC environment.

<u>3.3.1 Argon environment</u>

In order to perform local strain measurements on the tensile bar surfaces, samples were strained in argon to avoid oxidation and cracking, as both would interfere with the strain measurements (oxide would change the surface pattern created by the gold nano-particles, cracking would cause huge displacements to be measured where the crack had opened). High temperature (288°C) argon was used so that the resulting deformation would be similar to that which occurs in BWR NWC environments (288°C water). The argon, initially 99.999% purity, was purified further to 99.99999% purity to remove any trace water and oxygen that could cause oxidation, using a 27601-U Supelco purifier.

For strain characterization, a 13Cr15Ni bar (ASCC2) was strained to 3.5%. Three additional bars were strained in argon, another 13Cr15Ni sample (ASCC3), a 16Cr12Ni sample (CSCC1) and a CP304 sample (304SCC2) were strained to roughly 2%. Strain was characterized in these three samples, prior to additional straining in water afterwards. The ASCC3 sample was removed at 1.5% strain and examined prior to being reloaded and strained the remaining amount. Summaries of all the CERT experiments may be found in Table 3.2.

3.3.2 Simulated boiling water reactor (BWR) normal water chemistry (NWC) environment

To cause cracking, the samples were strained in CERT tests using a simulated boiling water reactor (BWR) normal water chemistry (NWC) environment. The specifications of the BWR NWC environment consist of 288 °C water, with an oxygen concentration of approximately 2 ppm and an outlet conductivity of 0.2 μS/cm. The conductivity was controlled by additions of sulfuric acid. The water was original purified

to a conductivity below 0.1 µS/cm, at which point sulfuric acid was added to maintain the desired conductivity. The samples were strained at a rate of approximately 3×10^{-7} s^{-1}.

This study focused on crack initiations. As such, tensile bars were never strained to failure, where cracks had propagated through the entire sample. Rather, CERT tests were generally stopped at low strain levels, to attempt to catch cracks during initiation, prior to propagating beyond a single grain boundary. Often these tests were run in increments, examining the cracking behavior after each incremental straining. Along with the argon CERT tests, a summary of the BWR NWC CERT tests may be found in Table 3.2.

Two sets of BWR NWC CERT experiments were conducted. The first was conducted prior to any argon experiments and included the ASCC1, BSCC1, HSCC1 and 304SCC1 tensile bars from Table 3.2. Two straining increments were performed, with the goal of studying general cracking behavior. Cracking densities were examined, and cracked grain boundary character was studied, as well as the angle of the cracks with respect to the tensile axis, and the general behavior of dislocation channels around the cracks, as seen using SEM imaging.

The second set of BWR NWC CERT tests were performed after the argon CERT test. The ASCC3, CSCC1, and 304SCC2 tensile bars were used in this study and strained in BWR NWC conditions following a prior argon strain increment. This allowed straining to be characterized in detail prior to inducing cracking in the BWR NWC strain increment.

3.4 Plastic strain analysis

As discussed in the introduction, plastic strain may be related to the measurable displacement within a sample from the unstrained state to the strained state by equations (2.11) through (2.17). Attempts have been made previously to measure GB slip and strain in DCs using scratches, which results in crude measurements with low resolution (due to width of scratch and ability to determine displacement in scratch across grain boundaries and DC) [11]. In order to accomplish the objectives of this work, a method for higher resolution strain measurements was required, which would allow sub-micron imaging of the strain at DC-GB intersections.

To accomplish the micro-scale strain measurements, two techniques were combined: SEM DIC and confocal microscopy. These two techniques allowed the displacement vectors to be measured across the sample surface. Specifically, DIC measures the in-plane displacement, and confocal microscopy measures the out-of-plane displacement. These techniques will be discussed in more detail later in this section.

3.4.1 SEM image correction

In order to accurately make quantitative measurements in a SEM, distortions in the imaging process must be corrected. There are a variety of sources that cause these distortions, which are generally divided into two categories: Electron beam distortions and geometric distortions. The major electron beam distortions include astigmatism, spherical aberration, chromatic aberration, electromagnetic interference and E-beam rotation. The geometric distortions refer to vibration, perspective distortion, drift, and projection distortion.

Astigmatism and chromatic aberration are both related to the electronic lenses in the SEM. These lenses are coils surround the beam column, used to create electromagnetic fields that shape the electrons in the beam. Astigmatism results from a non-circular shaped electron beam. Spherical aberration refers to the phenomenon of refraction strength being greatest towards the edge of the lens, thus electrons that pass near the outer edge to have a different focal point than those passing through the center.

The final three forms of electron beam distortion are related directly to the electron beam itself, outside of the lens. Chromatic aberration is caused by electrons changing energies from the expected accelerating voltage, and thus changing focal points. Electromagnetic interference is a result of external electromagnetic fields acting on the beam in a similar way as the lenses, however, uncontrolled by the microscope itself. E-beam rotation refers to the fact that the electrons in the beam are not actually traveling in a straight line, but rather in a spiral path, due to the forces from the lenses. This can cause rotation in the image as the topography changes.

Vibration, perspective distortion and drift all refer to physical phenomena that may affect SEM imaging. Vibration distortion is a result of physical vibrations shaking the microscope resulting in poor image quality. Perspective distortion is a result of the changing topography of a sample surface. While depth of field of the SEM view may cause all of the sample to appear in focus, changes in elevation will cause distortion, as the SEM is a 2-D viewing instrument. Drift is a result of the sample moving with respect to the electron beam, in an unexpected and uncontrolled manner. This can be a result of the sample itself sliding, or the stage on which it is mounted. It can also occur due to thermal expansion gradients within the column, or on the sample and stage.

Projection distortion refers to the distortion due to the difference of the distance the electron beam travels when at the outer edge of the imaging area, with respect to the center of the image. This can cause magnification changes towards the edge of the viewing area, and can be a result of several processes, some already discussed in this section. Spherical aberrations can result in projection distortion, as well as E-beam rotation, as the electrons travel further at the outer edges of the image, resulting in additional spiraling of the electrons within the beam. It is also a result of the larger step sizes for the same change in beam angle (Gnomic projection) at the outer edges, as shown graphically in Figure 3.9. There is also the possibility of defects in the beam scan pattern causing nonlinearity, especially near the edges of the viewing area.

The design of the JEOL JSM-6480 SEM used in this study allows for many of these distortions to be corrected directly in the lens and SEM itself. Careful SEM design by the manufactures and selection of location of the SEM within the lab has minimized the amount of external sources of distortion, such as vibration and electro-magnetic fields. The projection distortion is more difficult to correct for, and can cause significant errors in measurement. To correct for the inherent distortion that occurs in SEM imaging, an Ultrasharp TGX01 AFM calibration grid (3 µm grid spacing) was loaded alongside the tensile bar on the SEM stage and imaged at the same working distance as the tensile bars surface. Matlab® was used to determine a transformation function to correct the SEM distortion in the grid image, and the same transformation function was then applied to the tensile bar images. Each time the tensile sample was loaded in to the SEM for imaging, the grid was also loaded and imaged so that a calibration image was available for each one.

Using the Ultrasharp grid, the distortion was quantified by imaging the grid, and measuring the difference in location between the imaged location of the grid points with the expected spacing of 3 µm that the grid was created to be, within 5 nm. The overlap of the measured vs. expected grid points is shown in Figure 3.10, and actual measurements are shown in Figure 3.11and Figure 3.12.

3.4.2 Digital image correlation

DIC is a general technique referring to the comparison of points between two different images. In this case, it will be used to measure localized displacement on the surface of a sample. The technique tracks changes in gold speckle pattern on the surface, due to displacement during straining. The images of the sample will be taken in the SEM before and after straining at a 15 keV accelerating voltage with a spot size set to 50, a 10 mm working distance and 5000X magnification, which results in 233 pixels per 5 micrometers, after using the previously described image correction process.

Post-strain images were taken in the SEM at the same locations imaged prior to loading. Images were analyzed using a DIC code developed in Matlab®. The first step in the analysis was to set up a grid of coordinates to be analyzed. The grid was set up around DC-GB intersections, with a spacing of ~100 nm between each point. An aperture window of ~600x600 nm was placed around each point within the grid. The section of the image contained within this aperture was taken from the pre-strain image, and located within the post-strain image using the CPCORR function, part of the Matlab® image analysis library. Minor modifications were made to the CPCORR function to increase the accuracy of the image comparisons.

Displacement of each point within the grid was determined, and a displacement map was created from this data. By taking the partial derivatives of the horizontal (X) and vertical (Y) displacements with respect to the X and Y axes, the partial plastic strain tensor was determined by assuming that the channels were 100 nm in width according to previous TEM observations [110] of dislocation channels in similar irradiated materials and similar straining conditions. An example of these results is shown in Figure 3.13. With the in-plane strain measured, only the components in the direction normal to the surface (Z) were not accounted for.

In order to determine the amount of experimental error in the DIC process, the same area of the tensile bar with a gold speckle pattern was imaged several times, being unloaded and loaded back into the SEM between images. As no strain was applied between images, a perfect experimental setup would have shown zero displacement in a DIC analysis of the images. Prior to using the calibration grid to correct the images, the DIC analysis was found to erroneously show an average of ±30.6 nm of displacement across images ~7x7μm in area. Using the correction grid, this error was reduced to an average of ±14.8 nm displacement. This error tended to occur gradually over the 7 μm length of the area, rather than a random error that creates a noisy signal with many peaks. While the theoretical resolution of the DIC measurement is 2 nm, realistically, the error limits this resolution.

3.4.3 Confocal microscopy

Topographical maps were taken with the LEXT confocal microscope in the Michigan Lurie Nanofabrication facility, which has a spatial resolution of ~125 nm and a height resolution on the order of 10 nm. The confocal microscope creates a

topographical map by illuminating the sample with a laser and varying the focus between a lower limit and a higher limit, set by the user. By using an aperture to block all light that is out of focus, the microscope can detect which portion of the imaged area is in focus at any given time, and map out the heights of different areas based on when they come into focus. This topographical map may be considered a displacement map in the direction normal to the surface, and can then be combined with the DIC displacement maps to create three dimensional displacement maps. The combination is performed using Matlab® by reading in the topographical data from the confocal microscope and selecting the same features on the topographical map as the DIC map, allowing the maps to be aligned.

The Z displacement measurements were used to determine the partial derivatives of Z with respect to the X and Y axes, giving the XZ and YZ strains, which were combined with DIC measurements to produce a nearly complete plastic strain tensor, an example of which is shown in Figure 3.13. As the collected measurements determine displacement on the measured surface only, it is not possible to determine the third column of the strain tensor, which would be calculated using the partial of the displacement measurements with respect to the Z axis.

3.5 Elastic strain/stress analysis

Unlike plastic strain, elastic strain is not a permanent deformation, and will relax when stress is removed. Elastic strain is difficult to measure based on displacement of the sample, as it causes distortions of the crystal lattice. It must, therefore, be measured by determining changes in the lattice parameters. Tensile stress will increase the lattice parameter, while compressive stress will decrease it. Shear stress will cause the cubic

structure to take on a triclinic structure, meaning the crystal lattices no longer are orthogonal. A technique has been developed using EBSD to measure changes in the crystal lattice and relate these to elastic strain.

Wilkinson, *et al* [111] showed that shifts in Kikuchi bands in EBSD are related to the elastic strain tensor. This can be shown by first looking at the relationship between the shift in the EBSD pattern (q_i), as measured perpendicular to the direction (r_i) along some selected feature of the EBSD pattern (such as a zone axis), and the displacement gradient tensor,

$$q_i = \frac{\partial u_i}{\partial x_j} r_i - \left(\frac{\partial u_i}{\partial x_j} r_i \cdot r_k \right) r_i,$$

(3.3)

where

$$\frac{\partial u_i}{\partial x_j} r_i = \begin{pmatrix} \dfrac{\partial u_1}{\partial x_1} & \dfrac{\partial u_1}{\partial x_2} & \dfrac{\partial u_1}{\partial x_3} \\[2mm] \dfrac{\partial u_2}{\partial x_1} & \dfrac{\partial u_2}{\partial x_2} & \dfrac{\partial u_2}{\partial x_3} \\[2mm] \dfrac{\partial u_3}{\partial x_1} & \dfrac{\partial u_3}{\partial x_2} & \dfrac{\partial u_3}{\partial x_3} \end{pmatrix} \begin{pmatrix} r_1 \\ r_2 \\ r_3 \end{pmatrix}$$

(3.4)

and u_i is again the displacement tensor at some position x_i. It should be noted that r_i is a unit vector. In (3.3), the term $\dfrac{\partial u_i}{\partial x_j} r_i \cdot r_k$ cannot be determined by measuring Kikuchi band pattern shifts. By combining equations for individual components of q_i, it is possible to eliminate $\dfrac{\partial u_i}{\partial x_j} r_i \cdot r_k$ with the following two simultaneous equations derived from (3.3) instead:

$$r_1 r_3 \left[\frac{\partial u_1}{\partial x_1} - \frac{\partial u_3}{\partial x_3} \right] + r_2 r_3 \frac{\partial u_1}{\partial x_2} + r_3^2 \frac{\partial u_1}{\partial x_3} - r_1^2 \frac{\partial u_3}{\partial x_1} - r_1 r_2 \frac{\partial u_3}{\partial x_2} = r_3 q_1 - r_1 q_3$$

(3.5)

and

$$r_2 r_3 \left[\frac{\partial u_2}{\partial x_2} - \frac{\partial u_3}{\partial x_3} \right] + r_1 r_3 \frac{\partial u_2}{\partial x_1} + r_3^2 \frac{\partial u_2}{\partial x_3} - r_1 r_2 \frac{\partial u_3}{\partial x_1} - r_2^2 \frac{\partial u_3}{\partial x_2} = r_3 q_2 - r_2 q_3 \qquad (3.6)$$

To solve these equations, q_i must be measured at four distinct regions of the EBSD pattern. These four regions must have different directions r_i, otherwise the system of equations becomes singular. More than four regions allows for a more rigorous statistical analysis, so at least 20 regions will be used in this project.

Once the displacement gradient tensor has been determined, the rotation (w_{ij}) must be separated in order to determine the elastic strain tensor (ε_{ij}). These may be calculated separately as follows:

$$\varepsilon_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \qquad (3.7)$$

and

$$w_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i} \right). \qquad (3.8)$$

The rotation tensor may be used to determine the number of geometrically necessary dislocations needed to cause the rotation or curvature in the crystal lattice, however, this is not an accurate method for determining plastic strain, as it does not determine the statistically necessary dislocations, which may make up a large fraction of the total number of dislocations caused by plastic strain. For this reason, DIC and confocal microscopy were used to measure slip and the rotation tensor was not used. It was

important only to separate the rotation from the displacement gradient tensor, so that true elastic strain could be determined.

Once residual elastic strain has been measured, elastic stress may be determined. Elastic stress is related to elastic strain through Hooke's law,

$$\sigma_{ij} = c_{ijkl}\varepsilon_{kl}, \tag{3.9}$$

where σ_{ij} is the elastic stress tensor, ε_{kl} is the elastic strain tensor, and c_{ijkl} is the fourth order tensor describing the relationship between the stress and strain tensors, referred to as the stiffness tensor. The stiffness tensor may be defined using the elastic modulus (E) and Poisson's ratio (v) of the material. This results in the following form of Hooke's law,

$$
\begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{23} \\ \sigma_{31} \\ \sigma_{12} \end{bmatrix} =
\frac{E}{(1+\upsilon)(1-2\upsilon)}
\begin{bmatrix}
1-\upsilon & \upsilon & \upsilon & 0 & 0 & 0 \\
\upsilon & 1-\upsilon & \upsilon & 0 & 0 & 0 \\
\upsilon & \upsilon & 1-\upsilon & 0 & 0 & 0 \\
0 & 0 & 0 & (1-2\upsilon)/2 & 0 & 0 \\
0 & 0 & 0 & 0 & (1-2\upsilon)/2 & 0 \\
0 & 0 & 0 & 0 & 0 & (1-2\upsilon)/2
\end{bmatrix}
\begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ 2\varepsilon_{23} \\ 2\varepsilon_{31} \\ 2\varepsilon_{12} \end{bmatrix}
\tag{3.10}
$$

From the measured elastic strain, and (3.10), the stress tensor may then be determined. This would then result in the plastic and elastic strain tensor at any given point of the material, as well as the stress tensor.

This form of EBSD analysis was performed on DC-GB intersection using BLG production's CrossCourt 3 EBSD analysis software. These intersections are where the presence of dislocation within the DC creates areas of high elastic strain, even after the applied tensile strain has been removed. CrossCourt 3 uses EBSD patterns to measure the displacement between a measured pattern and a reference, unstressed pattern to determine q in Equation (3.3).

EBSD analysis requires a nice surface finish. First the gold particles need to be removed. This will be accomplished by lightly wiping the surface with a clean polishing pad and distilled water. As high quality EBSD patterns need to be collected, the exposure time at each collection point will be 3 seconds. The points will be spaced 100 nm apart. No data cleanup will be performed on this EBSD data, as each individual pattern is needed for the analysis. Points where a pattern is not clear enough to analyze will be removed from the scan, rather than replaced by neighboring values as was done previously in when making the orientation maps. The collected patterns are then loaded into CrossCourt 3 for analysis.

3.6 Cracking analysis

As mentioned previously, the purpose of the BWR NWC CERT tests is to induce crack initiation. In general cracks were 1-5 μm. Cracks smaller than ~1 μm could not be confidently analyzed using the SEM, as it is difficult to determine what constitutes a crack when it is so small. Cracks larger than 5 μm generally extended beyond a single grain boundary, and therefore, the initiation site becomes difficult to determine. After cracks formed, they were located within the SEM and characterized, based largely on the OIM characterization.

3.6.1 Oxide stripping

After the BWR NWC CERT tests, large oxide particles coated the tensile bar surface, obscuring the view of crack initiation locations. After every BWR NWC CERT test, with the exception of the first increment for the ASCC1, BSCC1, HSCC1 and 304SCC1, the oxide layer was chemically stripped off. For the previously mentioned exceptions, the oxide particles hadn't grown to fully cover the sample surface, and the surface underneath was still visible enough for this analysis, so the oxide was not removed.

The chemical stripping process consisted of two solutions: sodium hydroxide and potassium permanganate, and a solution of ammonium oxalate. First, a water based solution of 100 g/L sodium hydroxide and 30 g/L potassium permanganate was brought to a boil at a temperature ~100 °C. Upon reaching this temperature, the tensile bar was submerged in the solution for 5 minutes. Following this, the sample was then submerged for 5 minutes in another water based solution with 100 g/L ammonium oxalate at the same temperature. The sample was then cleaned in distilled water using an ultrasonic cleaner, and the process was repeated 3 times. Following the third repetition, the metal surface appeared clean, with a luster similar to before the BWR NWC CERT test.

3.6.2 Crack characterization

The samples were examined for cracking in a JOEL JSM-6480 SEM. Cracked boundaries were located and characterized. This characterization included the grain boundary misorientation, the Schmid factors of the adjacent grains, and the angle between the surface trace of the grain boundary, the tensile direction (trace inclination), as well as the types of interactions exhibited by the DC when intersecting the GB.

The orientations of the grains adjacent to a boundary, as measured by EBSD, were used to determine Schmid and Taylor factors, and to characterize the cracking behavior in terms of grain boundary type. The location of the crack on the boundary (mid boundary or at a triple junction) and the angle of the trace of the crack with respect to the tensile axis were measured directly from the SEM images. Certain grain boundary characteristics are unevenly distributed and so the data was normalized so as avoid unfairly weighting the data towards characteristics that were more strongly represented in the sample. Examples of this include the boundary type and Schmid factor. The majority of the boundaries in the sample are either random high angle boundaries (RHABs) or coincident site lattice boundaries (CSL), of which most are $\Sigma 3$. There are few low angle boundaries (LAB). In these results, LABs and CLS boundaries will be grouped together as special boundaries.

In the cases where the samples were strained previously in argon, the locations of the crack will be compared to the plastic strain maps to understand the state of the DC and GB prior to cracking. For those samples where strain maps were not available (ASCC1 and HSCC1 in particular), slip continuity across grain boundaries was also characterized in the SEM, based on SEM images taken after the boundaries cracked. Dislocation channels that intersect the sample surface create ledges that are visible in the SEM. When these channels appear to be continuous across a boundary (allowing for a change in angle at the boundary) slip is classified as continuous. If channels intersect the boundary on one side, but are not observed to "emerge" from the boundary in the adjacent grain, slip is classified as discontinuous. An example of distinguishable continuous and discontinuous slip at a boundary is shown in Figure 3.15. If it is possible

97

to determine the continuity of slip across a boundary then the boundary is considered

distinguishable and used in the data analysis.

3.6.3 Statistical analysis

There is a statistical error associated with all of the collected data. This error

decreases with an increasing amount of data collected, however, there is a limit to the

amount of cracking data that it is feasible to collect, so it is important to calculate and

include the statistical error with all cracking data. This work uses the statistical analysis

outlined by Alexandreanu [112].

For this statistical analysis, the cracking was considered a binomial distribution,

as the boundaries could be considered to be in two states: cracked and uncracked.

Binomial distributions are modeled as

$$P(x) = \frac{n!}{(n-x)!x!} p^x (1-p)^{n-x}, \tag{3.11}$$

where P(x) is the predicted probability distribution function, x is the number of cracks at

the GB type being studied, n is the number of cracks studied and p is the probability of

the crack occurring at the GB type in question, defined as

$$p = \frac{x}{n}. \tag{3.12}$$

GB types were divided into categories of interest, such as misorientation type, specific

Taylor or Schmid factor pair types, specific GB surface trace angles, and based on

surrounding DC-GB intersection types.

For binomial distributions, the standard deviation (σ) may be used to determine the amount of error in the measurements. Standard deviations for binomial distributions are calculated as

$$\sigma_x = \sqrt{x(1-p)}\,, \tag{3.13}$$

or, in terms of the fractional uncertainty of the probability of cracking (p),

$$\frac{\sigma_p}{p} = \sqrt{\frac{1}{n}\frac{(1-p)}{p}}\,. \tag{3.14}$$

This statistical analysis was used to calculate error in a variety of measurements, beyond just cracking measurements. In particular, it was also used to determine the error in the DC-GB classification measurements. In these cases, x represents the number of DC-GB intersections of a particular classification (i.e. continuous, disc. w/ GB slip, or discontinuous), and n is the total number of DC-GB intersections examined. The error analysis then proceeds in the same fashion as the cracking analysis.

Table 3.1 Composition (wt%) of 4 alloys examined in this study.

Material designation	Fe	Cr	Ni	Mn	Si	P	C
CP304	Bal.	18.3	8.5	1.38	0.65	0.03	0.04
18Cr12Ni1Si	Bal.	18.2	12.4	1	1.05	<0.01	0.02
13Cr15Ni	Bal.	13.41	15.04	1.03	0.1	<0.01	0.016
21Cr32Ni	Bal.	20.73	31.16	0.94	0.1	0.014	0.014
16Cr12Ni	Bal.	16	12	1	0.1	<0.01	0.01

Table 3.2 List of CERT experiments, by alloy type. Includes level of strain, as well as environment.

Alloy	Sample designation	Strain %	Max applied tensile stress MPa	Environment
13Cr15Ni	ASCC1	2.3	149	BWR
		5.6	213	BWR
	ASCC2	3.5	185	Argon
	ASCC3	1.5	105	Argon
		2.5	141	Argon
		4.5	175	BWR
		7.2	192	BWR
21Cr32Ni	BSCC1	2.2	167	BWR
		5.2	210	BWR
16Cr12Ni	CSCC1	1.7	183	Argon
		3.5	205	BWR
18Cr12Ni1Si	HSCC1	2.5	234	BWR
		8.8	279	BWR
CP304	304SCC1	1.0	228	BWR
	304SCC2	2.1	186	Argon
		3.3	194	BWR
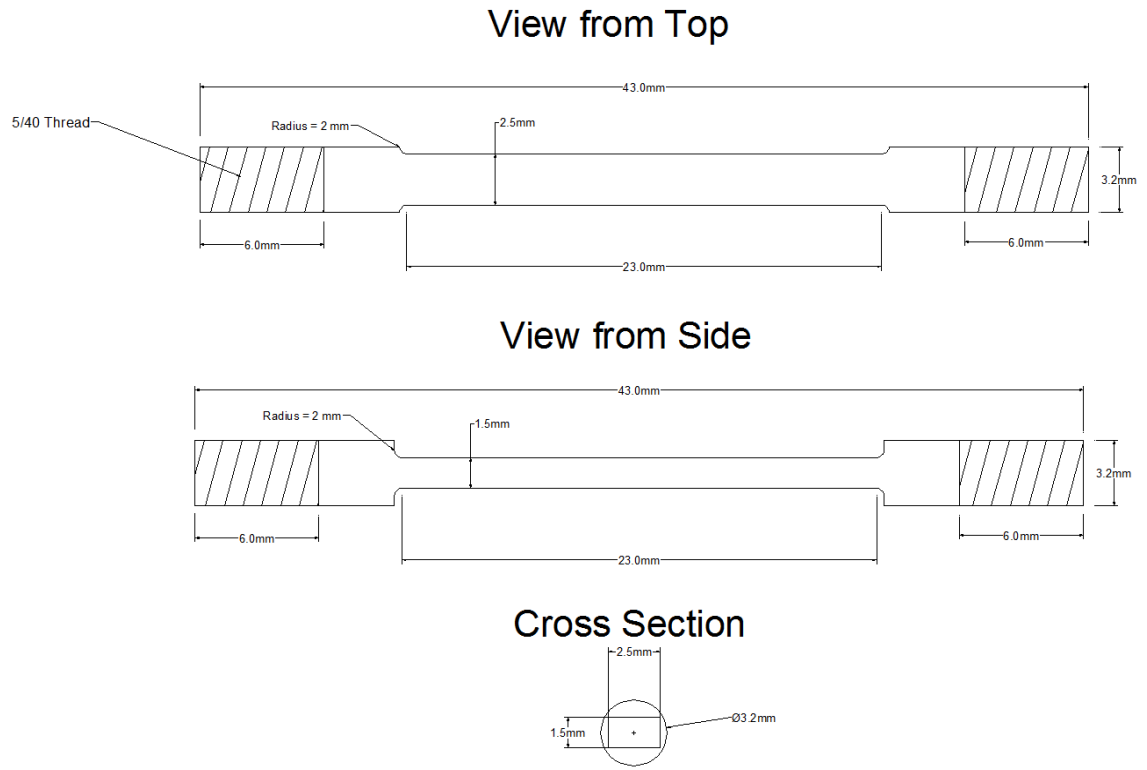
Figure 3.1 Tensile bar dimensions
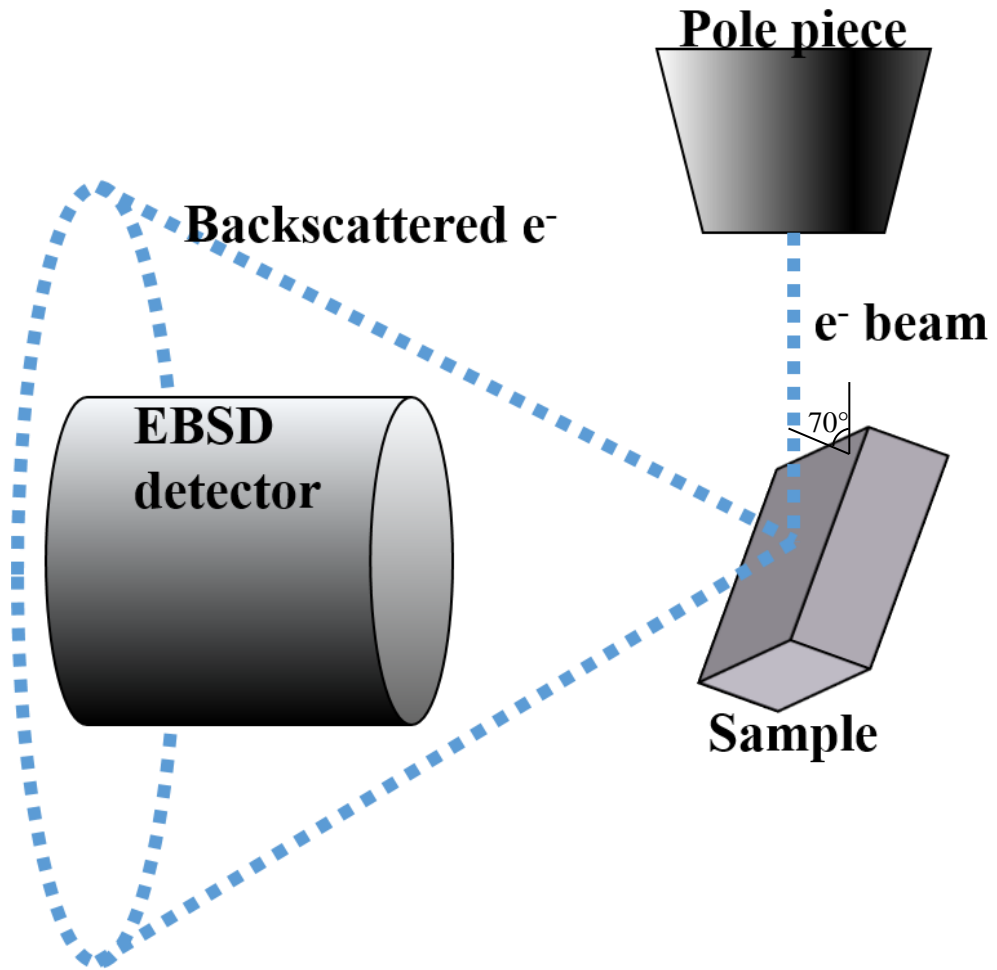
Figure 3.2 EBSD configuration inside an FEG SEM, with sample surface normal at 70° to the incoming electron beam.

Electron beam

θ

d
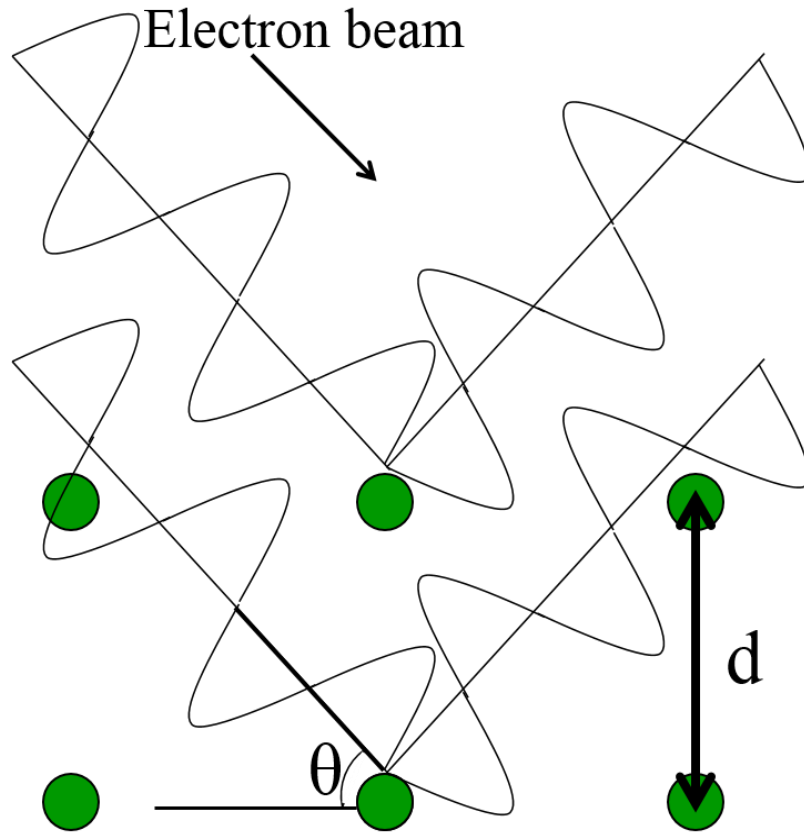
Figure 3.3 Depiction of Bragg's criterion, Equation (3.1). Two electrons impinging on a lattice at the Bragg's angle will scatter elastically, with the waves still coherent.
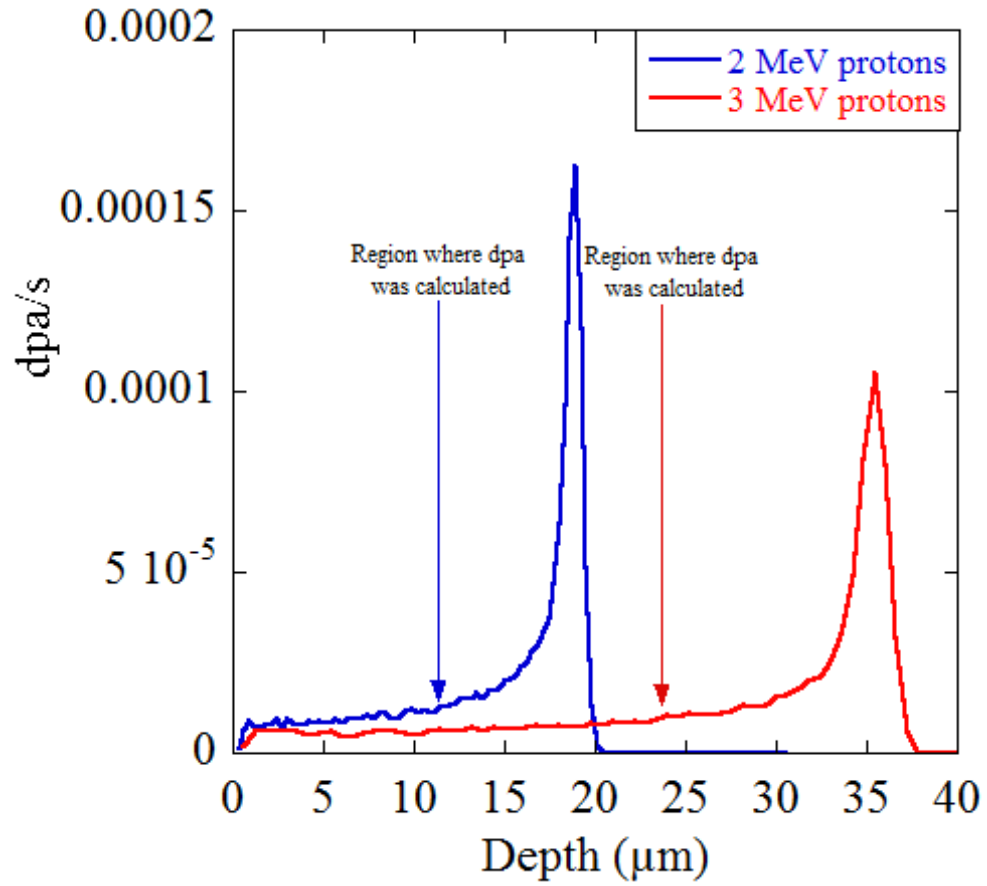
Figure 3.4 Damage rate depth profile for an austenitic stainless steel irradiated with 2 and 3 MeV protons.
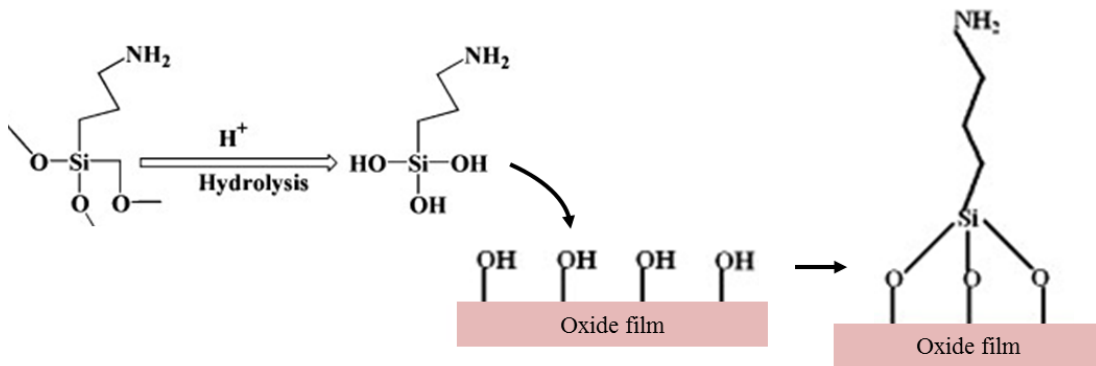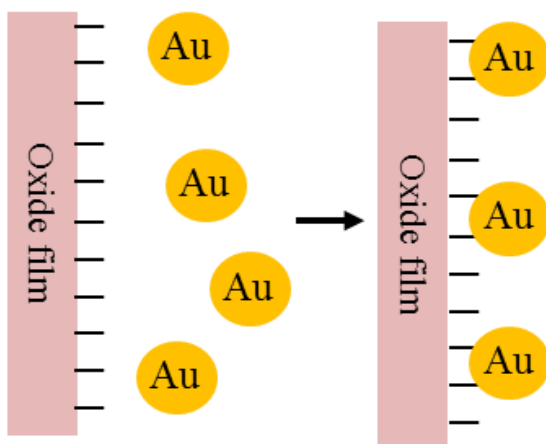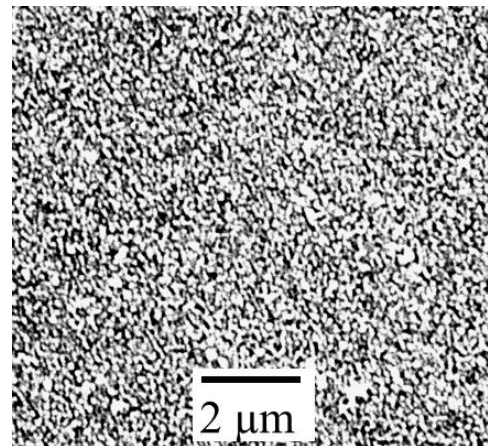
Figure 3.5 Gold pattern deposition process. First, silane is attached to the hydroxide groups in the metal oxide [113] (a). The amine groups in the silane then attach to gold nano-particles [108] (b), resulting in a speckled pattern on the steel surface (c), where the white in the SEM image is gold and the black steel.
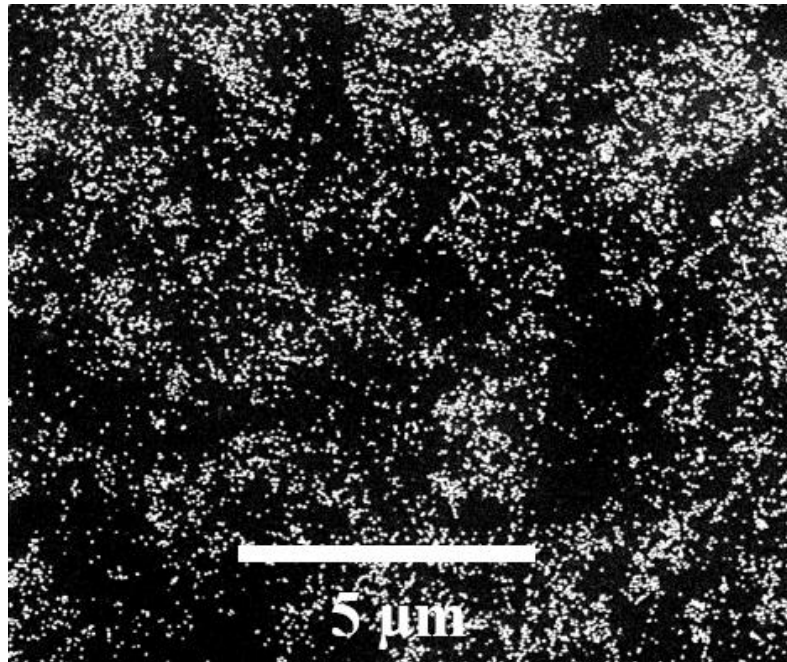
(a)
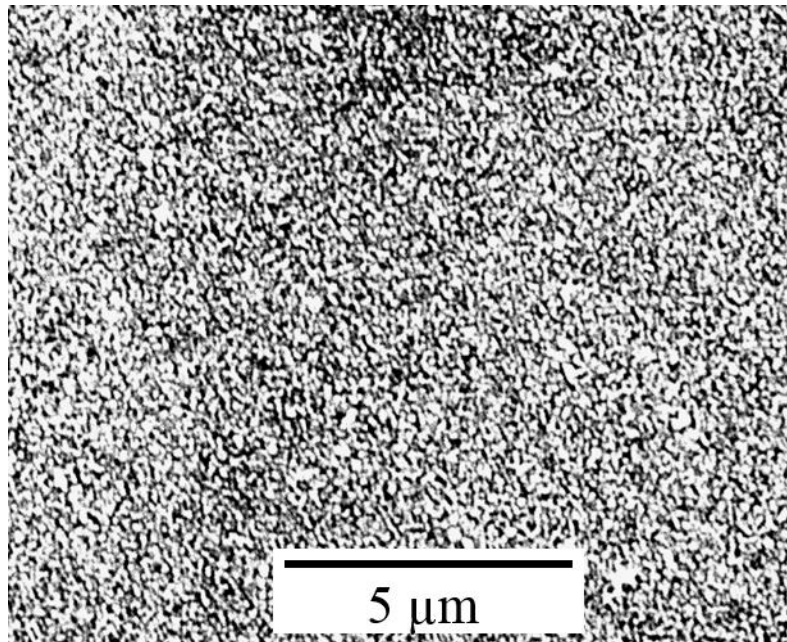
(b)

Figure 3.6 Gold deposition on (a) ASCC2, the first sample where gold deposition was used to create a DIC speckle pattern, and (b) ASCC3, which has a similar gold particle density as all other DIC samples, except ASCC2.

(a)

KEY

Pump
Emergency Rupture Valve
Emergency Pressure Release Valve
Back Pressure Regulator Valve
Pressure Gauge
Metering Valve
Butterfly Valve
Three Way Valve
Micron Filter
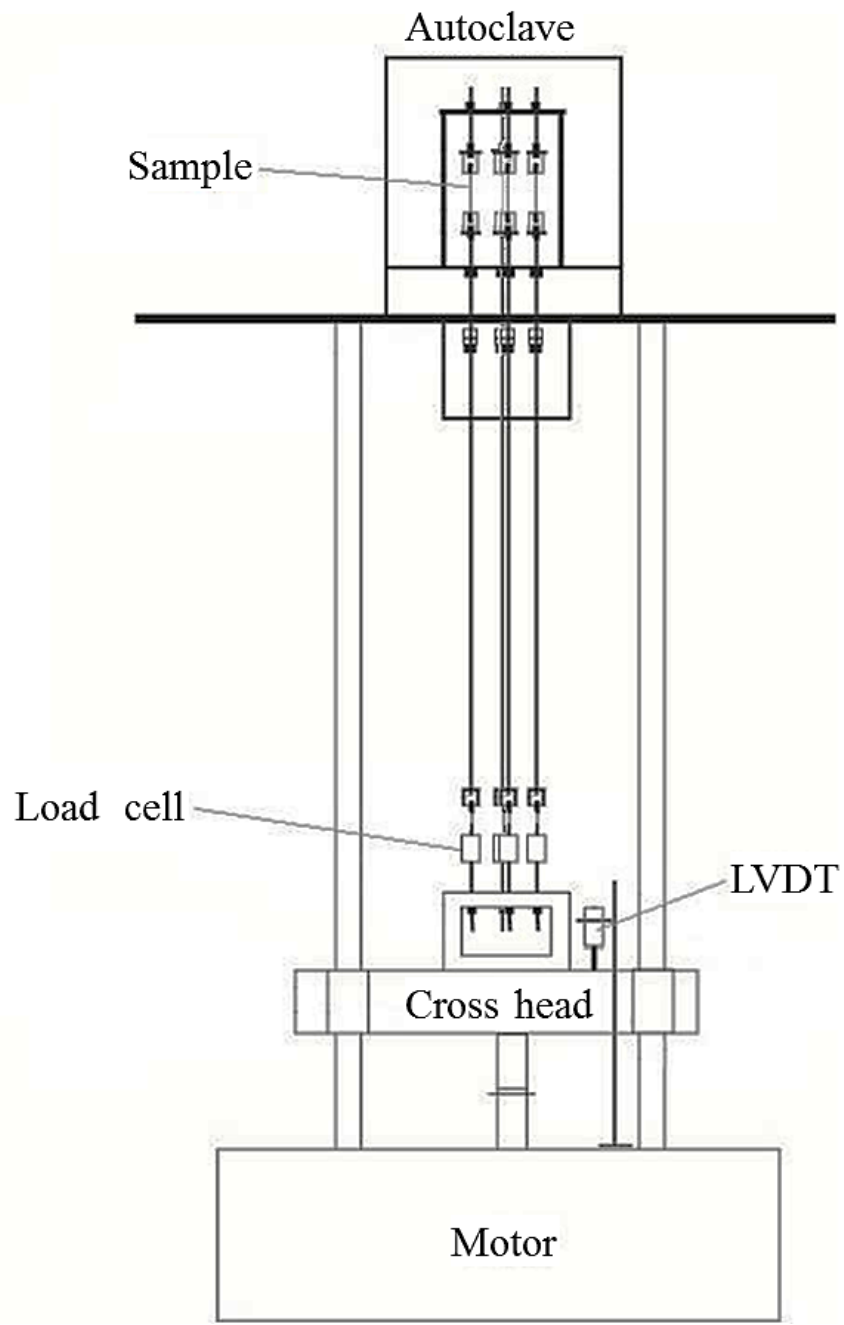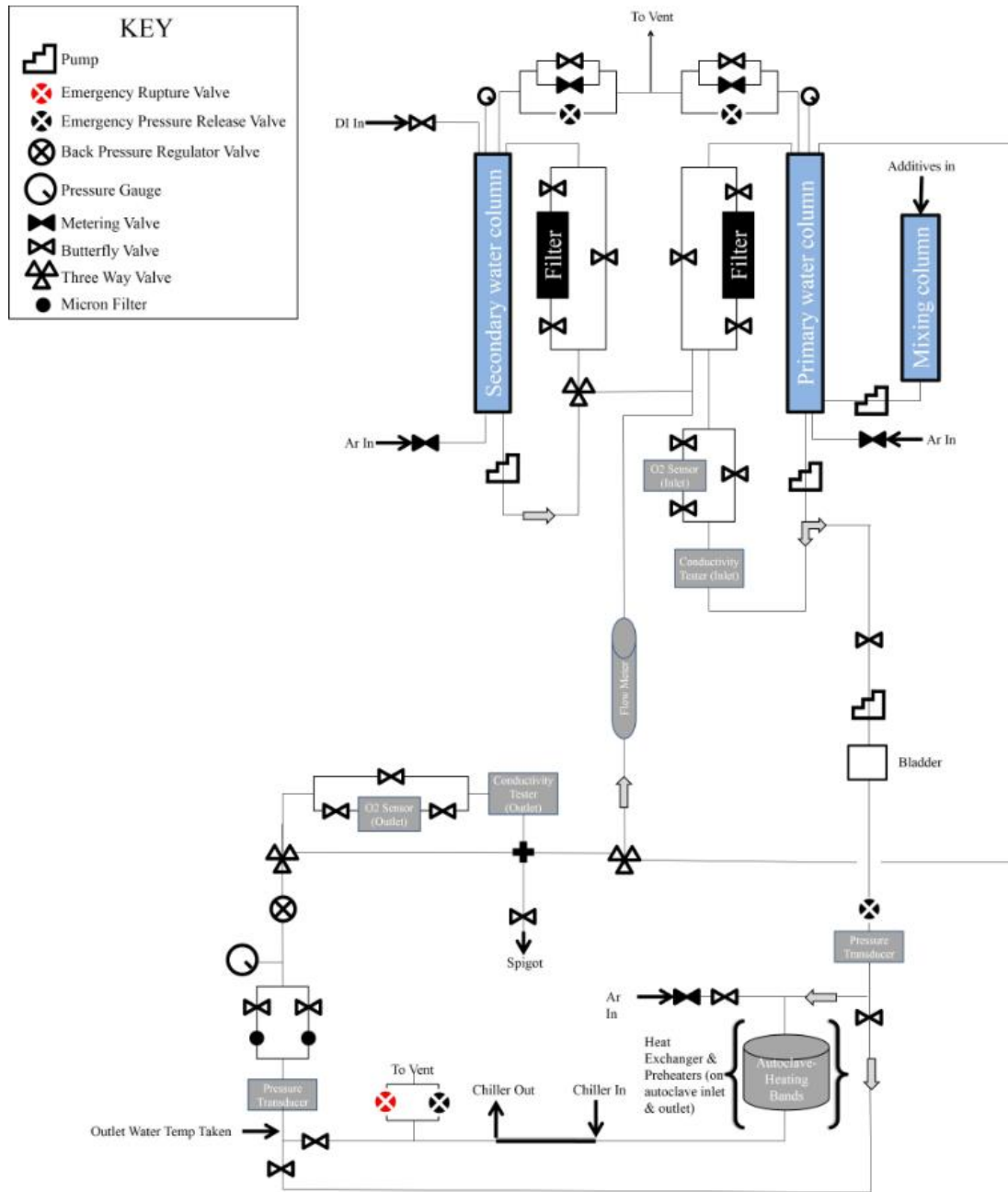
(b)

Figure 3.7 (a) Schematic for CERT testing apparatus at the High Temperature Corrosion Laboratory at the University of Michigan and (b) accompanying waterloop that feeds into the autoclave (modified from [114]).
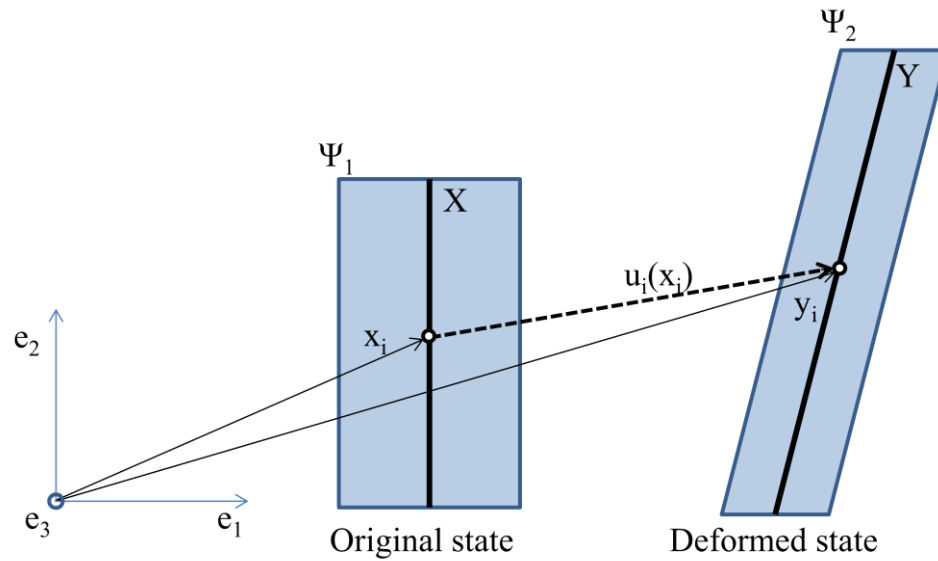
Figure 3.8 Example of deformation: Shape Ψ_1 deforms into Ψ_2. These shapes are three dimensional and extend into the page. Plane X is depicted within Ψ_1, and deforms into plane Y within Ψ_2. A point on plane X is described by vector x_i, where i has values of 1, 2, and 3, and in this case represent the Cartesian components of x, with 1 being the horizontal component, 2 being the vertical component, and 3 being the component normal to the paper (represented by the e_i axis). x_i deforms to vector y_i by the displacement vector $u_i(x_i)$
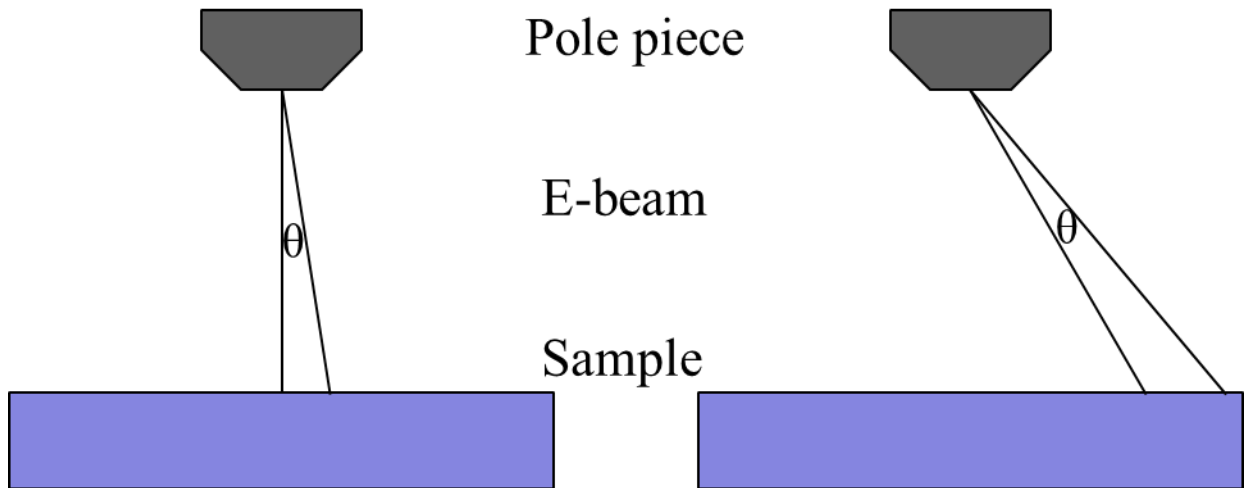
Figure 3.9 Graphic depicting one source of projection distortion: Larger step sizes from the same change in beam angle at the outer edges of the viewing area that the center of the viewing area.
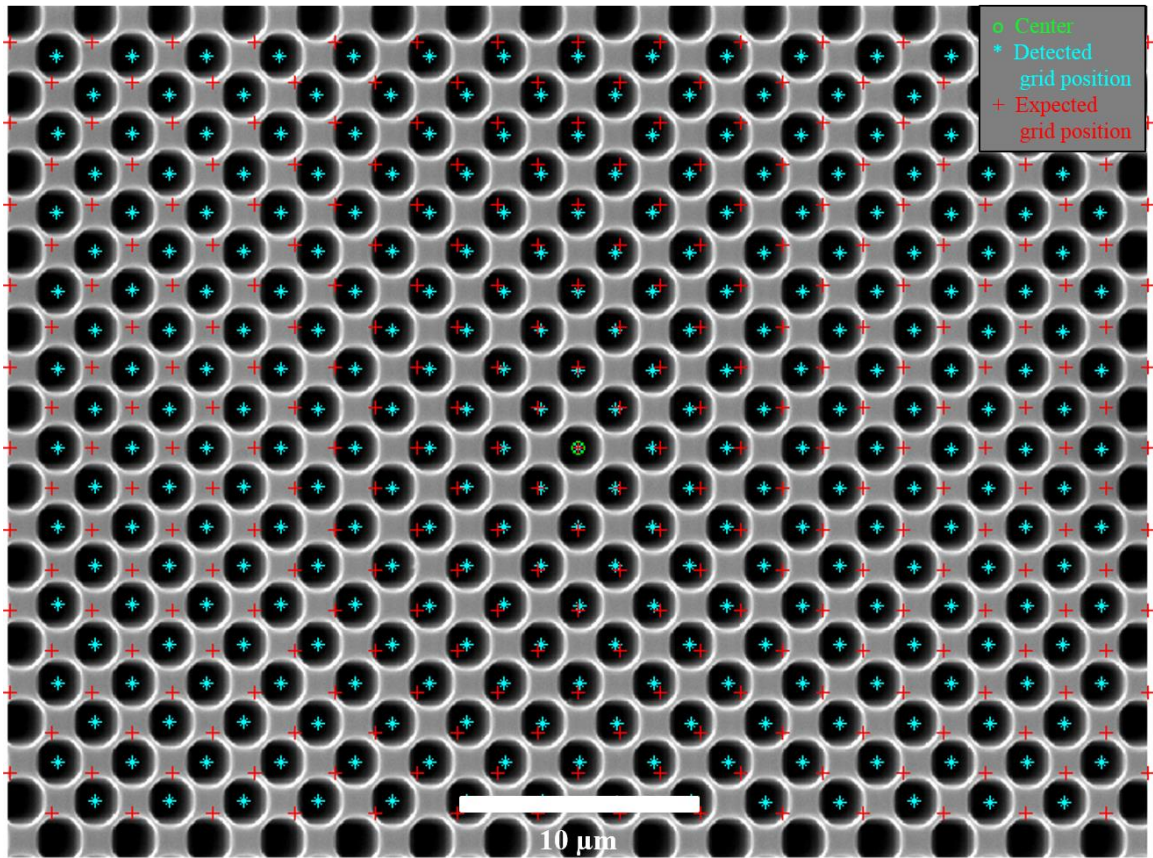
Figure 3.10 Corrected grid, taken at 3000x in the JEOL SEM. The red crosses mark the location where the center of the circles should appear and the blue asterisk mark the actual measured location of the circle centers.
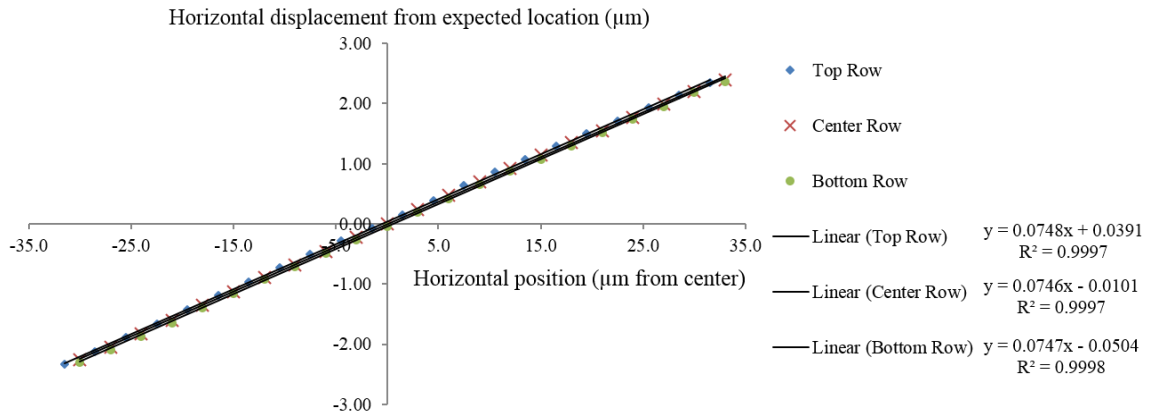
(a)
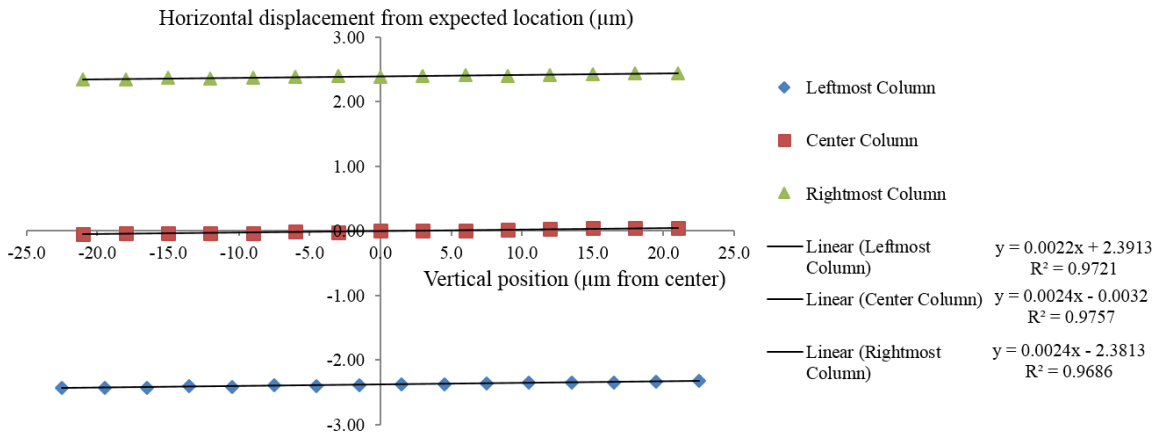
(b)

Figure 3.11 Measure of distortion based on the difference of horizontal displacement between the measured grid locations and the expected grid locations. (a) shows the measurements taken horizontally for the top, center and bottom rows of the grid, (b) shows the measurements taken moving vertically for the left, center and right columns of the grid.

Vertical displacement from expected location (μm)

(a)

Vertical displacement from expected location (μm)

(b)
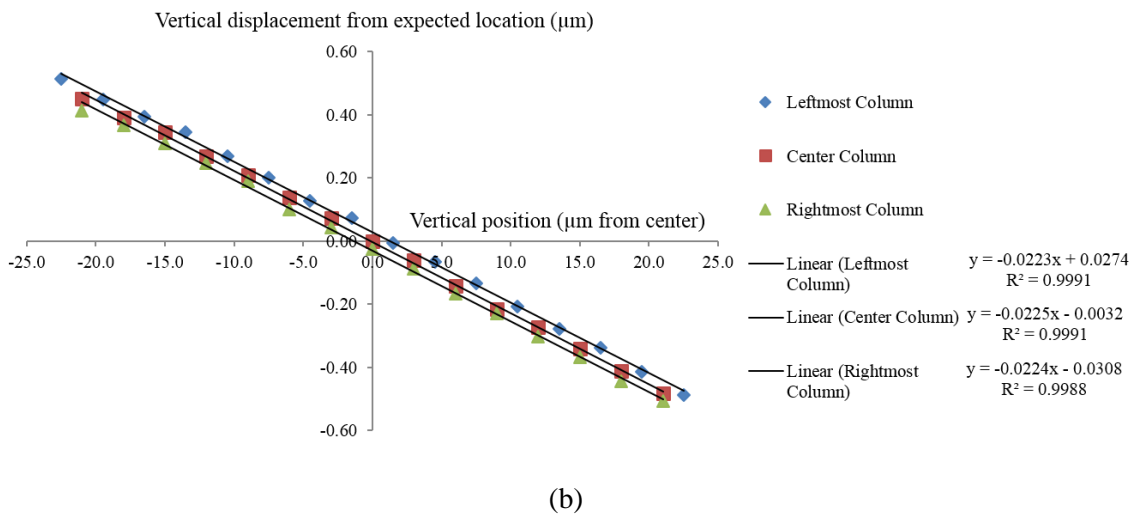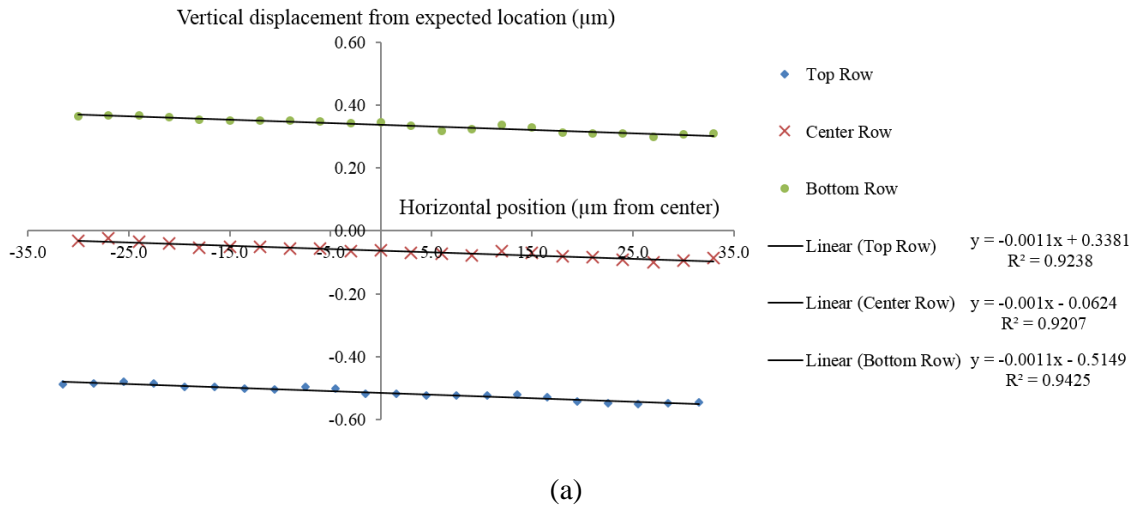
Figure 3.12 Measure of distortion based on the difference of vertical displacement between the measured grid locations and the expected grid locations. (a) shows the measurements taken horizontally for the top, center and bottom rows of the grid, (b) shows the measurements taken moving vertically for the left, center and right columns of the grid.
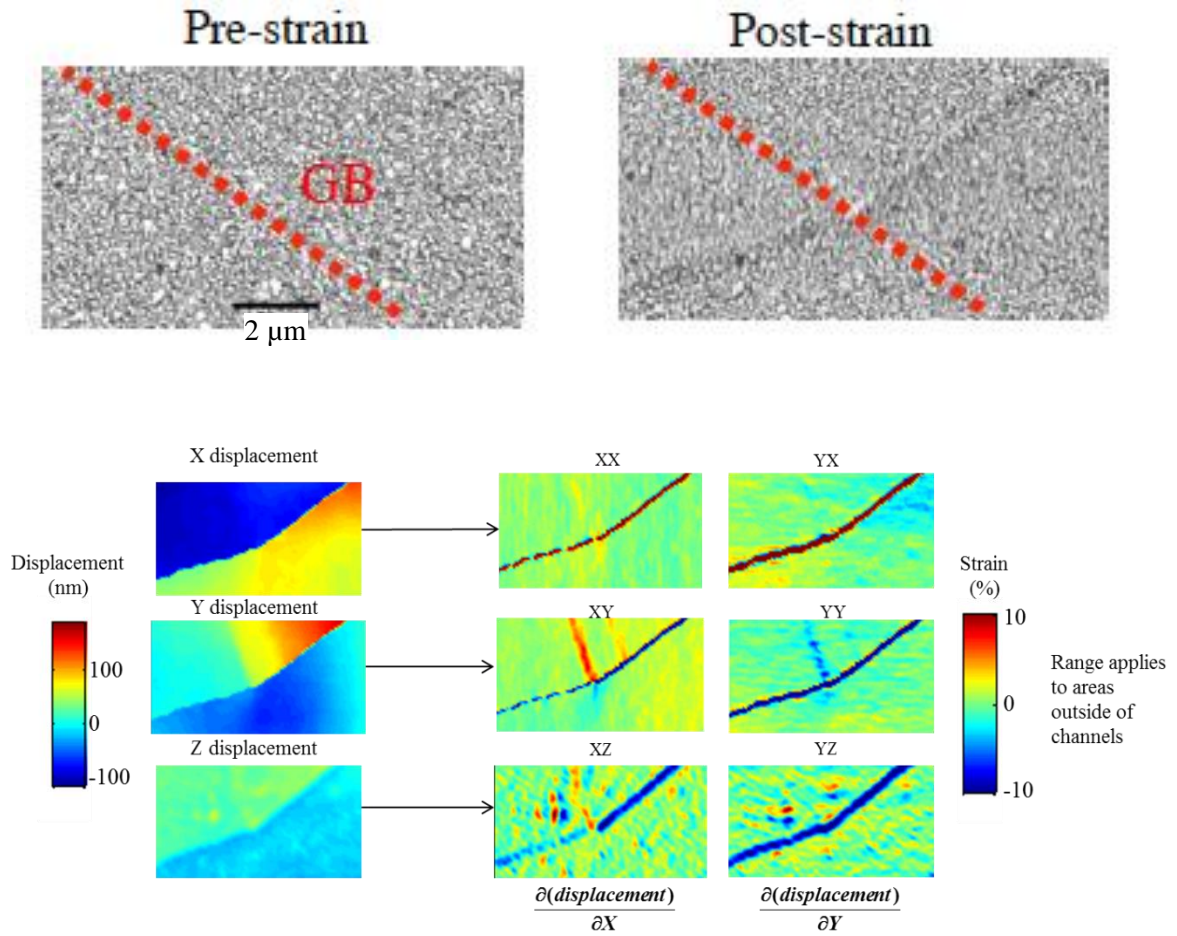
Figure 3.13. First row: SEM images before and after straining. Second row: X, Y and Z displacement maps on the left, as measured using DIC. On the right are the strain maps, as calculated from the measured displacement maps.
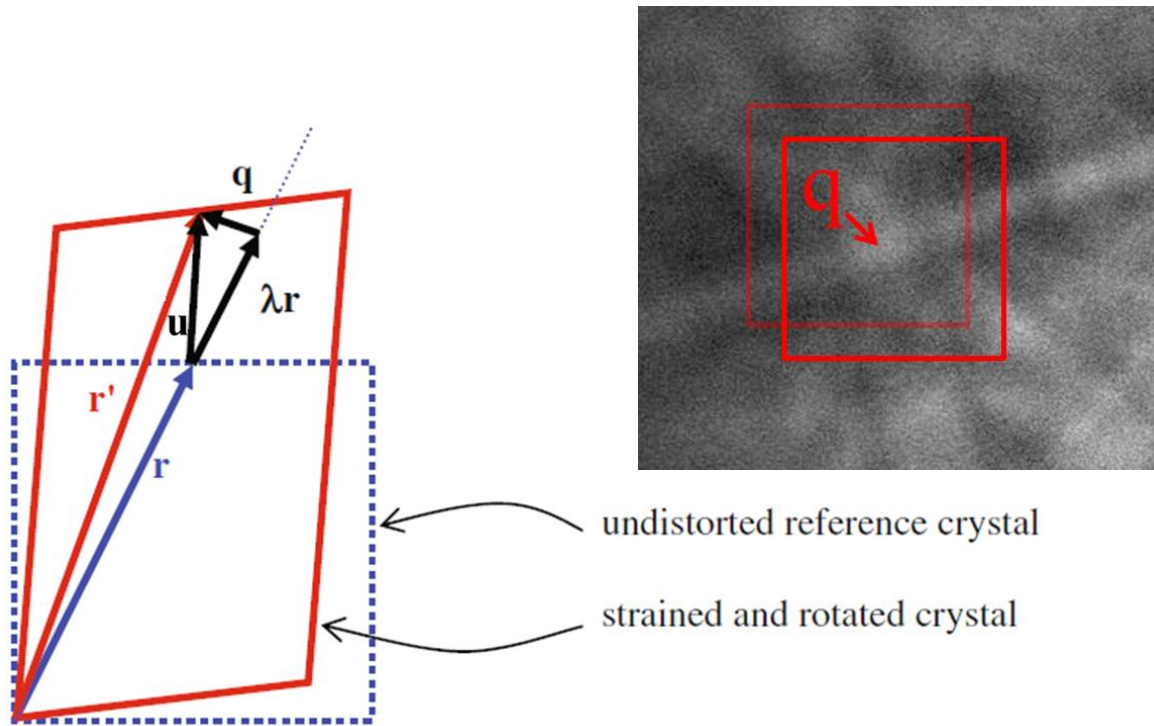
Figure 3.14   Graphical depiction showing the effects of lattice distortion on EBSD patterns. Changes in the lattice shape will cause features in the EBSD pattern to shift by some distance q, which may be related to the displacement vector (u) of the lattice.
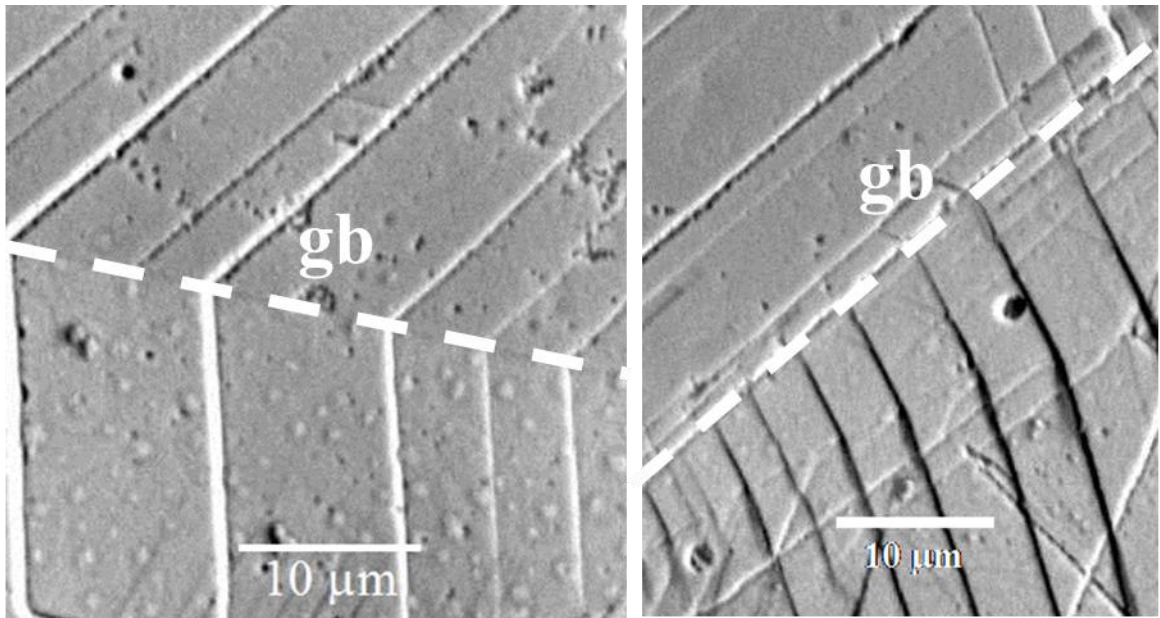
Figure 3.15 Examples of GBs with continuous (left) and discontinuous (right) slip, as determined solely by SEM images of the sample surface.

Chapter 4

RESULTS

In this chapter, results from the previously described experiments will be presented. An in depth discussion of the results will be given in the next chapter. The significant results include the characterization of cracked grain boundaries in irradiated stainless steel, the plastic deformation measurements of the dislocation channels and grain boundaries, the stress analysis of the DC-GB intersections, and the subsequent cracking study performed to understand the link between deformation and IASCC. In particular, the characterization of cracking based on DC-GB intersection types is important, as this type of characterization has not been performed previously in studies of IASCC initiation.

4.1 Alloy characterizations

4.1.1 Stacking fault energy

Stacking fault energy was calculated for each of the alloys, using the methods described in the background chapter of this Thesis. The three lab purity alloys, 21Cr32Ni, 13Cr15Ni and 16Cr12Ni were selected for their high, moderate and low SFE energy, respectively. The SFE may be seen in Table 4.1. These are calculated values that only estimate the actual stacking fault energies of the alloys. Measured values have

been shown to differ in absolute value from calculated values, but to show the same general trends between alloys [7].

4.1.2 Grain boundary character

As mentioned in the previous chapter, prior to straining, the tensile bars were characterized with EBSD to determine the orientation of the grains on the sample surface. In particular, grain boundary misorientation, Schmid factor, and Taylor factor are determined and reported in Table 4.2. Note that the majority of grain boundaries are random high angle boundaries, though in both alloys, there is a significant fraction of sigma three boundaries as well. The grains are divided somewhat evenly across the possible Taylor factors, with a small preference towards higher numbers. The Schmid factors of the grains are weighted heavily towards the higher possible values.

In addition to using Schmid factor and Taylor factor to describe the deformation properties of the grains, localized deformation was also characterized. It is possible for multiple slip systems to be activated within a single grain. The number of active slip systems was characterized for a number of grains and the results shown in Table 4.3. For this characterization, dislocation channel traces on the sample surface were examined, and channels at different angles were considered different slip systems. As such, only slip systems with distinct slip channels could be differentiated. Different slip directions on the same slip plane could not be differentiated with this method.

Slip transmission at grain boundaries was also examined in the SEM images. Grain boundaries were characterized as either allowing slip transmission, such that dislocation channels were continuous across the grain boundary, or as resulting in discontinuous slip of the channels at the grain boundaries. Boundaries where more than

50% of the channels were continuous were considered continuous slip boundaries, whereas less than 50% of the channels exhibiting slip continuity were considered discontinuous slip boundaries. General alloy characteristics for slip continuity were characterized after the second strain increment of the ASCC1, BSCC1 and HSCC1 samples on grain boundaries that did not crack. These are reported in Table 4.4 under the rows listed as uncracked grain boundaries.

4.2 Susceptible grain boundaries

After the characterization of the grain boundaries by using EBSD, as described in the previous section, the samples were cracked in a BWR NWC environment. The cracks were characterized based on the grain boundary properties, as described in the previous section, and shown in Figure 4.1. The characteristics of the grain boundary susceptible to cracking were compared to those of the general bulk material, and trends in the character of the cracked boundaries were examined.

Figure 4.2 shows the cracking susceptibility based on the grain boundary type. Random high angle boundaries were much more likely to initiate cracks. The Schmid and Taylor factors of the adjacent grains also was examined, as seen in Figure 4.3 and Figure 4.4, and low Schmid factor/high Taylor factors in the adjacent grains was found to cause the boundary to be more susceptible to cracking. Figure 4.5 shows the surface traces of the cracked grain boundaries, showing that boundaries closer to 90° to the tensile axis are more likely to crack. Finally, boundaries intersected by discontinuous channels were also found to be more susceptible to cracking, shown in Figure 4.6.

4.3 Deformation analysis

Initially, for the ASCC1, BSCC1, and HSCC1 samples, dislocation channels were either considered continuous (C) or discontinuous at the boundary, depending solely on whether the channel was connected to another in the adjacent grain across the boundary. A random sampling of grain boundaries was characterized for each of these samples, and reported in Table 4.8. Grain boundaries were considered continuous or discontinuous only if all the channels intersecting the boundary exhibited either continuous or discontinuous slip. With the addition of DIC measurements, GB slip was possible to measure, and the "discontinuous" category was divided into discontinuous slip (D), and discontinuous slip-inducing grain boundary slip (D/GB). These classifications are similar to those described by Shen et al. [45] in their work studying unirradiated materials using *in situ* TEM straining. Examples of these classifications are shown in Figure 4.7. Also in Table 4.8, the grain boundary misorientation data was taken on a number of each of these three categories on the ASCC3 sample, and reported. When collecting the DIC data, preference was given to RHABs, as these are the boundaries more likely to crack, and of more interest to this study. A number of special boundaries were still examined, however.

Slip transmission was readily identified by the continuation of a dislocation channel through the grain boundary. When two channels in adjacent grains coincided at the same point on a grain boundary, it was considered a result of slip transmission. In the cases of discontinuous slip, the intersection was considered to include grain boundary slip when there was a discernable amount of displacement in the grain boundary in either the

DIC or topographical measurements, which was clearly visible above any noise in the displacement map.

In some cases, slip in the grain boundary was observed to occur only along a segment of the grain boundary near the point of intersection with the discontinuous channel, resulting in the activation of a channel in the adjoining grain at a location away from the intersection site, as shown in Figure 4.7d. In order for the glissile dislocations in the grain boundary plane to be re-emitted into the adjoining grain, the dislocations would have to be screw dislocations and the grain boundary would have to be the conjugate plane of the slip plane in the lattice [115]. Alternatively, the passage of grain boundary dislocations may be blocked within the grain boundary such that a pile-up is formed, activating a grain boundary source and then ejecting dislocations into the grain. For the purpose of characterizing the strain field around the DC-GB intersection, the DC-GB cases were considered as discontinuous slip-inducing grain boundary slip, regardless of what happened elsewhere along the grain boundary

After 2.5% strain in the ASCC3 sample, an area of 1.1×10^5 μm^2 was examined to determine the general distribution of DC-GB intersection types. In this area, 42 GBs were found where DCs were continuous across the boundary, 37 GBs where DCs were discontinuous, of which 21 of the 37 induced GB slip. The densities of GBs that exhibit the different DC-GB intersection types are shown in Table 4.9. In order to collect statistically significant data on all three DC-GB intersection types, when choosing DC-GB intersections to measure with DIC and the confocal microscope, preference was given to the discontinuous channels, as continuous were more plentiful and easier to find

in order to collect slip data. The total number of channels characterized, classified by the type of DC-GB intersection, is given in Table 4.10.

The change in the DCs from 1.5% to 2.5% was examined for the ASCC3 sample. The number density of dislocation channels was examined by drawing lines (0.7 mm in length) and counting the number of dislocation channels that intersected the lines. Figure 4.8 shows the results of the DC density measurements at each of the strain increments (1.5%, 2.5%, 4.5% and 7.2%). Dislocation density was found to increase quickly at the lower strain increments, but level off during the higher strain increments. There was a large scatter in the DC density measurements, depending on the location of where the lines were drawn. Some grains had almost no slip channels form, whereas others contained a high density of channels. Ten measurements were taken for each strain increment, and the standard deviations are shown in Figure 4.8 as well.

As only in-plane slip was measured at 1.5%, all comparisons were made with only in-plane slip. 44 channels were measured at both 1.5% and 2.5%. Of these, 33 were continuous, and the strain in the channels increased, on average, 144% from their values at 1.5% strain. 7 discontinuous with GB slip were found to almost double their sizes, increasing 91% in size on average. 4 discontinuous channels increased, on average, 151% in size. The overall change in size distribution may be seen in Figure 4.9.

Two of the DC-GB intersections examined at 1.5% macroscopic tensile strain that had initially appeared discontinuous were found to change classification after the additional straining to 2.5% macroscopic tensile strain. The SEM images and horizontal (tensile axis) strain maps are shown in Figure 4.10 and Figure 4.11. In the case shown in Figure 4.10, the discontinuous channel overcomes the grain boundary barrier to

dislocation transmission, and becomes continuous across the grain boundary. At 1.5% macroscopic strain, small amounts of plastic strain are visible in the grain adjacent to the discontinuous channel, indicating a likely dislocation pileup occurring within the adjacent grain, prior to the formation of the channel within that grain, however this strain does not form a channel that moves across the entire grain, and the strain level is significantly less than the strain of the impinging dislocation channel. In Figure 4.11, grain boundary slip is observed, which was not initially visible at 1.5% macroscopic strain. These two channels that changed classification were not included in the 44 measurements shown in Figure 4.9.

4.3.2 Dislocation channel slip measurements

DIC measurements (in-plane slip) were taken after three CERT experiments. For ASCC2, in-plane slip was measured after the applied 3.5% macroscopic plastic strain. In-plane slip measurements were taken after both the 1.5% and 2.5% applied strain of ASCC3. After the 1.5% strain, it was noted that channel density was low, and so measurements were discontinued in favor of an application of an additional 1% strain to form more channels. Slip normal to the sample surface was measured only after the 3.5% strain for the ASCC2 and after the 2.5% strain of the ASCC3.

Measured slip in channels ranged from 14 nm to 660 nm of total displacement. The distribution of the channel sizes may be seen in Figure 4.12. This data shows the total slip, accounting for all directions, and therefore does not include the 1.5% strained ASCC3 measurements, as only in-plane displacement was measured for that case. The majority of the channels measured underwent a total amount of slip between 80-100 nm.

Figure 4.13 shows the slip measurements, divided by the type of DC-GB intersection. Figure 4.13a shows all of the total slip measurements for the ASCC2 and ASCC3 specimens, Figure 4.13b shows a detailed histogram of the distribution of the slip measurements for the ASCC3 specimen strained to 2.5%. It is noted that discontinuous undergoes significantly lower levels of slip than either continuous or discontinuous that induces GB slip. Slip measurements are also tabulated in Table 4.11 through Table 4.13, for continuous, discontinuous with GB slip, and discontinuous channels, respectively. Table 4.14 shows the average displacement values for the three DC-GB intersection types, as well as the maximum value measured for each case.

The data set includes the measurement of displacement in the grain boundaries at the D/GB intersections. Multiple channels may intersect the boundary and contribute to GB slip. Figure 4.14 shows an SEM image of multiple D/GB dislocation channels, each intersecting the GB and inducing GB slip, as seen in the XY shear strain map on the right-hand side. In order to report GB slip due to a single channel, as in Figure 4.13 and Table 4.14, displacement in the GB was measured along the grain boundary on both sides of a DC-GB intersection, and the difference in the two measurements was attributed to the GB slip caused by the dislocation channel.

Figure 4.15 shows the GB displacement from multiple channels intersecting the boundary, rather than that from a single DC, as in Figure 4.13. As displacement normal to the surface may play a role in rupturing the oxide layer and causing cracking, it is plotted against the total displacement. Displacement in the grain boundary occurs predominantly at D/GB intersections, though there were two cases where slip in the grain boundary was observed at the intersection with a continuous DC. These two cases were classified as continuous DC-GB intersections, and the associated GB slip (~120 and 30 nm displacement) was not included in GB displacement measurements in Figure 4.13b.

The direction of the slip was also measured. Table 4.15 shows the results of these measurements. The angle of the surface trace of the DC was measured with respect to the tensile axis. The displacement vector was also determined, using the X, Y and Z displacement measurements from DIC (X and Y) and the confocal microscope (Z).

4.4 Elastic stress measurements

Stress measurements were conducted on each of the three DC-GB classifications, resulting in stress tensor maps of varying quality, as seen in Figure 4.16. The stress tensor is summarized with the Von Mises stress (σ_V), which is calculated from each component of the stress tensor as

$$\sigma_V{}^2 = \frac{1}{2}\Big[\left(\sigma_X - \sigma_Y\right)^2 + \left(\sigma_Y - \sigma_Z\right)^2 + \left(\sigma_X - \sigma_Z\right)^2 + 6\left(\sigma_{XY}{}^2 + \sigma_{XZ}{}^2 + \sigma_{YZ}{}^2\right)\Big]. \tag{4.1}$$

In total, 48 intersections were measured, 20 from ASCC2 and 28 from ASCC3 after the 2.5% argon strain. Data was unavailable in large areas of the map, primarily due to the surface topography that resulted in blockage of the diffracted beam, resulting in poor pattern quality in the region where the data was sought. An example of the

dislocation channel blocking the diffracted beam is shown in Figure 4.17. In particular, it was difficult to obtain results near dislocation channels, as this resulted in a steep shift in topography created by the channel step in the surface. Figure 4.18 shows the results from the ASCC3 measurements, depicted in terms of average stress at increasing radii moving out from the DC-GB intersection. The different colored dots represent the different measured intersections.

In order to depict the level of stress reached by the intersections, in each case from the ASCC3 sample, the highest 3 values of stress near the DC-GB intersection (within 1 µm) was taken and averaged together with all others of its DC-GB type. The results are shown in Table 4.16. It is noted that while the continuous and disc. w/ GB slip are similar, the discontinuous intersections tend to create areas of higher stress, as determined from these measurements.

4.5 DC-GB interaction effect on cracking

The DIC measurements on the pre-argon strained samples were performed in order to determine the classification of the DC-GB intersection prior to the occurrence of a crack. Once determined, cracks were formed on the sample (ASCC3 and CSCC1) and the cracks characterized based on their prior DC-GB characterization. After the first strain increment, 89 cracks were characterized in ASCC3, and 7 cracks were found in the CSCC1 sample. After the second strain increments, 198 cracks were located on the ASCC3 and 10 on the CSCC1. The cracking densities are listed in Table 4.17. Examples of the three crack types are shown in Figure 4.19 through Figure 4.21.

In describing the prior local strain state of these cracks, aside from continuous, discontinuous with GB slip and discontinuous DC-GB intersections, two additional

classifications are needed. A large number of cracks occurred at triple junctions, and are counted separate from the cracks occurring mid-grain. Other cracks occurred in locations where there were no channels present after the prior-argon straining, and thus no information could be determined from the strain state. These classifications may be seen in Table 4.18, and example images of both may be found in Figure 4.22 and Figure 4.23. It is noted that many of the CSSC1 cracks occurred in locations where channels did not exist prior to the BWR straining. In the case of the 3.5% strain, 2 of these appeared to have discontinuous channels, and 1 in the case of the 7.5% strain. However, as the slip could not be characterized prior to the cracking in the BWR environment, it is not clear if there was grain boundary sliding in these cases. Bulk DC-GB characterization measurements, taken over an area ~0.1 mm^2, are also shown in Table 4.18, which show that continuous DC-GB intersections are the most common and discontinuous are the most rare.

Table 4.1 Calculated stacking fault energy for the alloys used in this study using Pickering [96], Schramm [97], and Rhodes [98] methods.

	CP304	18Cr12Ni1Si	13Cr15Ni	21Cr32Ni	16Cr12Ni
Pickering	40.6	41.1	49.0	73.8	38.2
Schramm	23.0	49.6	53.9	158.6	36.7
Rhodes	47.2	47.2	48.5	73.3	45.3

Table 4.2 Bulk properties of 13Cr15Ni and 18Cr12Ni1Si, as measured by EBSD.

		18Cr12Ni1Si	13Cr15Ni
Number of boundary types in random sampling	RHABs	67250	14576
	LABs	6549	934
	$\sum 3$	44488	12042
	$\sum 5$	963	169
	$\sum 9$	1546	771
	$\sum 27$	402	349
Number of grains with given Taylor factor	2.37	126	99
	2.5	267	217
	2.63	294	204
	2.76	189	92
	2.89	254	168
	3.02	366	252
	3.15	331	225
	3.28	351	206
	3.41	437	290
	3.54	349	284
	3.67	370	302
Number of grains with given Schmid factor	0.29	4	6
	0.32	24	45
	0.35	36	74
	0.38	4	119
	0.41	151	163
	0.44	410	421
	0.47	761	810
	0.5	897	757

Table 4.3 Number of active slip systems within grains after second strain increment.

Number of active slip systems	13Cr15Ni (5.6% strain)	18Cr12Ni1Si (8.8% strain)
0	1	5
1	59	72
2	58	43
3	6	2
Total	124	122

Table 4.4 Slip continuity of grain boundaries for both cracked and uncracked alloy boundaries after second strain increment of samples ASCC1, BSCC1 and HSCC1.

	Continuous slip	Discontinuous slip
Sum of all uncracked GBs	253	336
Sum of all cracked GBs	73	229
Uncracked 18Cr12Ni1Si (HSCC1) GBs	59	132
Cracked 18Cr12Ni1Si (HSCC1) GBs	33	131
Uncracked 13Cr15Ni (ASCC1) GBs	100	98
Cracked 13Cr15Ni (ASCC1) GBs	40	98
Uncracked 21Cr32Ni (BSCC1) GBs	94	106

Table 4.5 Crack densities for four tensile bars strained in a BWR NWC with no prior argon straining. All samples were irradiated to 5 dpa with 2 MeV protons at 360 °C.

Alloy	Sample designation	Strain %	Crack density	
			#/mm^2	μm/mm^2
13Cr15Ni	ASCC1	2.3	5.2	23.3
		5.6	15.0	50.6
21Cr32Ni	BSCC1	2.2	0	0
		5.2	0	0
18Cr12Ni1Si	HSCC1	2.5	50.0	164.3
		8.8	302.4	759.3
16Cr12Ni	CSCC2	1.0	1	2.3
		6.0	32	55.4
CP304	304SCC1	1.0	3.2	68.3

Table 4.6 Crack properties of the two strain increments of ASCC1 (13Cr15Ni) and HSCC1 (18Cr12Ni1Si).

		1st strain increment		2nd strain increment	
		18Cr12Ni1Si	13Cr15Ni	18Cr12Ni1Si	13Cr15Ni
Number of cracked boundaries by type	RHABs	50	45	192	128
	LABs	3	1	6	0
	$\sum 3$	6	1	30	6
	$\sum 5$	1	6	3	16
	$\sum 9$	0	1	11	5
	$\sum 27$	0	1	0	3
Number of cracked boundaries by angle with respect to the tensile axis	10	0	0	4	2
	20	0	0	3	1
	30	1	2	7	3
	40	2	2	9	10
	50	3	1	18	10
	60	3	5	14	17
	70	13	10	56	36
	80	17	16	55	42
	90	21	19	67	37
Number of cracked boundaries by Taylor factor pair types	LL	2	3	4	11
	LM	8	9	44	19
	LH	7	3	53	22
	MM	12	10	39	26
	MH	23	19	67	55
	HH	8	11	35	25
Number of cracked boundaries by Schmid factor pair types	LL	15	12	44	21
	LM	14	15	64	32
	LH	17	8	64	44
	MM	2	3	20	17
	MH	5	8	34	28
	HH	2	4	16	16

Table 4.7 Maximum bin values set for Schmid and Taylor factors, such that each bin contained approximately one third of the grains examined.

	Maximum bin value	
	Taylor factor	Schmid factor
L	2.89	0.44
M	3.32	0.47
H	3.67	0.5

Table 4.8 Continuous/Discontinuous measurements from 4 tested alloys. Note that only in 13Cr15Ni ASCC3 were disc. w/ GB slip and discontinuous distinguished. When selecting GBs to characterize for ASCC3, RHABs were given preference, so a larger number of them were characterized when compared to special boundaries. Boundaries were only considered continuous or discontinuous in cases where all channels acted the same upon impinging on the GB.

Sample	GB-DC type	RHAB	Special boundaries
13Cr15Ni ASCC1	Continuous	30	32
	Discontinuous	29	14
21Cr32Ni BSCC1	Continuous	22	36
	Discontinuous	37	15
18Cr12Ni1Si HSCC1	Continuous	21	20
	Discontinuous	49	13
13Cr15Ni ASCC3	Continuous	29	12
	D/GB	18	12
	Discontinuous	30	5

Table 4.9 Bulk classification of grain boundaries that exhibit the different DC-GB interaction classification. Data taken from the 13Cr15Ni (ASCC3) sample after 2.5% strain in 288°C argon.

DC-GB type	GB/mm^2	Error (GB/mm^2)
Continuous	385	±41
Disc. w/ GB slip	192	±36
Discontinuous	147	±33

Table 4.10. Number of dislocation channels characterized by DIC, separated into DC-GB classifications. The 13Cr15Ni ASCC2 channels, as well as the 2.5% strain 13Cr15Ni ASCC3 channels were also characterized using the confocal microscope to measure topography.

	ASCC2	ASCC3 (1.5% strain)	ASCC3 (2.5% strain)
Continuous	16	28	126
Disc. w/ GB slip	9	6	44
Discontinuous	5	6	43

Table 4.11 Continuous slip measurements (in nm) taken on 13Cr15Ni ASCC3, strained to 1.5% and then 2.5% total strain. X and Y are in-plane (X in the tensile direction), measured with DIC. Z is normal to the surface, measured with confocal microscopy.

Channel ID	1.5% strain		2.5% strain			Channel ID	2.5% strain		
	X	Y	X	Y	Z		X	Y	Z
1	136.2	30.5	256.1	41.2	240.7	38	121.7	22.9	92.0
2	112.3	27.7	242.2	34.8	156.7	39	26.4	2.1	31.0
3	143.3	55.8	197.8	50.8	286.3	40	71.9	9.4	93.7
4	55.4	48.0	158.3	126.7	57.7	41	46.2	36.3	99.3
5	41.0	19.8	126.4	32.3	41.0	42	238.5	343.1	192.7
6	29.6	12.9	66.1	26.5	34.7	43	234.9	356.2	141.7
7	34.1	23.3	67.4	57.5	30.7	44	26.2	28.4	49.0
8	200.5	184.8	409.0	382.8	249.7	45	49.7	10.5	19.3
9	206.4	194.2	417.1	367.1	214.0	46	94.2	59.9	132.3
10	58.9	60.6	237.8	245.6	157.7	47	206.0	37.6	219.8
11	52.7	45.0	226.2	167.5	50.7	48	219.9	50.1	174.1
12	208.9	128.2	366.1	228.6	259.0	49	50.0	23.3	71.7
13	177.4	34.9	272.7	48.9	229.0	50	23.4	9.5	31.7
14	25.1	36.6	77.0	120.7	61.7	51	45.2	3.5	40.6
15	178.6	125.2	259.8	186.1	153.3	52	27.8	27.8	11.7
16	40.3	7.2	69.7	12.6	100.3	53	14.6	23.4	15.5
17	80.0	127.7	122.0	195.1	510.0	54	8.7	30.9	15.6
18	77.5	43.9	187.7	120.0	130.7	55	92.9	136.9	54.8
19	18.5	2.4	61.0	33.9	37.0	56	27.0	103.1	82.5
20	16.9	3.9	157.8	9.1	NM	57	121.3	105.4	33.7
21	173.3	126.1	288.1	215.0	111.7	58	41.5	46.9	31.3
22	61.5	31.4	90.6	50.7	55.0	59	55.4	36.5	69.5
23	33.1	13.0	75.8	24.0	35.0	60	117.5	262.8	58.0
24	97.6	55.6	100.6	61.5	93.3	61	194.9	271.0	88.7
25	10.6	2.6	33.3	4.2	15.0	62	31.6	60.0	45.9
26	35.8	5.4	119.0	9.8	116.7	63	50.7	69.5	43.2
27	68.0	45.5	207.7	118.4	182.7	64	89.5	40.7	88.3
28	101.2	95.0	184.2	159.7	163.8	65	31.1	5.0	32.8
29	49.4	44.0	87.9	97.5	65.9	66	26.1	9.8	97.3
30	52.8	79.1	47.5	73.8	NM	67	137.7	19.0	88.9
31	71.4	94.9	67.4	86.4	NM	68	71.4	6.3	92.4
32	37.5	45.2	26.4	36.9	NM	69	27.1	11.0	42.1
33	47.5	23.8	37.2	20.5	NM	70	99.9	83.7	35.0
34			49.2	15.2	80.0	71	67.4	78.1	18.3
35			57.9	2.9	56.3	72	99.4	8.1	133.8
36			50.0	39.1	35.3	73	49.2	20.4	62.5
37			30.5	26.8	22.0	74	62.7	51.5	46.3

Channel	2.5% strain			Channel	2.5% strain		
ID	X	Y	Z	ID	X	Y	Z
75	150.0	79.6	15.7	110	54.7	66.0	28.1
76	138.7	72.8	17.3	111	8.9	19.6	14.0
77	19.6	12.8	35.7	112	78.9	48.5	177.0
78	54.0	7.3	63.7	113	38.1	1.2	86.3
79	45.3	10.2	43.3	114	112.8	88.5	113.7
80	120.7	15.5	80.7	115	110.0	82.4	103.0
81	256.2	105.0	259.2	116	139.1	87.5	30.4
82	318.7	16.2	338.4	117	314.1	179.4	273.8
83	101.6	94.9	49.3	118	91.7	48.0	70.8
84	82.9	80.5	65.3	119	48.1	25.5	16.0
85	137.3	121.4	95.3	120	61.8	39.4	15.1
86	133.4	194.6	101.0	121	45.3	16.9	21.5
87	154.9	194.2	200.5	122	22.8	13.3	7.6
88	122.2	188.8	202.8	123	156.2	21.2	211.5
89	176.8	132.8	77.8	124	246.7	10.7	232.2
90	97.5	79.7	77.4	125	20.9	27.3	75.4
91	40.2	21.1	41.6	126	7.9	8.2	25.8
92	33.4	43.5	24.3	127	498.5	23.0	422.6
93	38.6	42.5	46.4	128	498.8	42.1	430.2
94	45.4	38.7	64.3	129	54.1	67.6	66.7
95	57.2	15.4	53.3	130	18.3	26.1	15.0
96	56.8	28.8	50.9	131	96.6	48.0	97.5
97	109.3	54.8	57.7	132	16.9	11.7	73.5
98	107.4	28.5	36.9	133	212.4	209.5	197.3
99	47.0	27.7	31.2	134	280.0	162.8	164.3
100	67.6	50.1	46.6	135	184.9	57.4	142.9
101	180.7	152.8	92.8	136	13.8	21.7	22.3
102	60.7	75.3	18.6	137	50.9	34.0	38.3
103	223.2	109.4	193.3	138	16.9	9.3	9.6
104	118.7	137.7	221.0	139	13.6	23.4	30.5
105	102.4	5.6	70.0	140	25.9	71.2	10.8
106	117.5	5.9	120.0	141	153.2	129.5	204.4
107	131.0	7.8	139.0	142	115.9	39.7	31.1
108	86.6	43.2	60.7	143	48.3	49.4	37.0
109	73.9	17.3	47.7				

Table 4.12 Discontinuous with GB slip channel slip measurements performed on the 13Cr15Ni ASCC3 sample, strained to 1.5% and then 2.5% total strain. X and Y are in-plane (X in the tensile direction), measured with DIC. Z is normal to the surface, measured with confocal microscopy. All measurements are in nanometers.

Channel ID	1.5% strain		2.5% strain			Channel ID	2.5% strain		
	X	Y	X	Y	Z		X	Y	Z
1	13.6	11.8	19.9	23.6	84.7	24	143.4	146.2	84.6
2	121.2	112.1	213.3	205.6	238.0	25	42.8	74.2	110.0
3	109.8	111.9	114.0	120.3	53.0	26	120.1	63.1	168.7
4	23.1	63.4	84.0	133.4	191.7	27	191.2	128.8	165.7
5	102.0	18.6	75.2	26.9	NM	28	47.7	3.5	67.0
6	39.6	27.5	115.4	81.0	97.2	29	37.1	33.1	30.2
7	24.2	26.6	71.8	69.0	77.0	30	71.9	27.3	79.2
8			18.6	11.7	56.7	31	121.3	54.5	120.4
9			153.9	7.2	61.0	32	152.8	52.5	129.5
10			40.1	31.1	66.3	33	88.1	36.3	97.5
11			50.4	18.3	58.7	34	52.4	27.0	66.4
12			100.4	28.6	58.7	35	330.3	114.3	84.6
13			262.3	14.2	222.3	36	188.8	114.2	153.5
14			70.5	6.6	74.8	37	302.1	63.0	276.2
15			50.0	2.4	118.2	38	41.1	41.1	32.6
16			166.3	46.9	132.6	39	89.5	34.3	79.8
17			80.0	17.7	59.2	40	151.3	63.9	189.8
18			33.8	33.5	26.1	41	8.1	2.0	27.9
19			15.0	2.6	94.3	42	23.2	18.1	29.0
20			18.1	2.0	66.3	43	62.3	8.0	51.5
21			34.9	22.5	51.0	44	446.5	169.0	253.8
22			221.8	85.0	NM	45	314.2	57.0	300.1
23			195.9	204.2	133.0				

Table 4.13 Discontinuous channel slip measurements performed on the 13Cr15Ni ASCC3 sample, strained to 1.5% and then 2.5% total strain. X and Y are in-plane (X in the tensile direction), measured with DIC. Z is normal to the surface, measured with confocal microscopy. All measurements are in nanometers.

Channel ID	1.5% strain		2.5% strain			Channel ID	2.5% strain		
	X	Y	X	Y	Z		X	Y	Z
1	41.2	24.5	69.0	39.2	25.0	23.0	62.1	3.3	106.8
2	14.1	15.8	74.1	40.0	70.3	24	54.1	14.0	63.9
3	14.1	16.8	47.7	24.8	27.0	25	59.4	55.1	27.5
4	50.9	7.5	94.5	36.1	77.3	26	20.5	45.8	60.5
5			22.7	4.1	27.0	27	32.1	2.1	12.7
6			32.0	15.7	75.3	28	23.1	1.2	35.5
7			61.6	45.7	39.3	29	65.6	14.6	160.1
8			87.5	40.0	71.3	30	88.6	42.7	84.4
9			48.0	6.8	31.5	31	47.2	45.2	65.4
10			17.1	12.5	27.7	32	14.1	13.1	18.8
11			28.6	8.4	52.7	33	39.1	17.4	53.0
12			17.1	29.7	12.0	34	23.6	66.0	77.2
13			12.1	26.2	1.0	35	46.2	14.0	20.7
14			17.0	7.8	46.3	36	18.9	10.4	48.1
15			12.3	9.2	17.0	37	69.6	17.1	41.3
16			15.2	5.5	63.7	38	25.1	6.8	46.2
17			36.3	33.0	28.6	39	60.0	4.3	29.1
18			41.9	31.5	133.3	40	33.3	13.0	52.7
19			18.8	17.0	21.3	41	11.9	4.2	10.8
20			24.3	1.2	23.0	42	21.9	25.3	39.3
21			39.0	2.7	34.9	43	14.7	8.2	20.6
22			38.5	5.0	36.0	44	16.3	28.4	14.1

Table 4.14 Average and maximum displacement in dislocation channels for each DC-GB intersection type (including both DC and GB displacement measurements for D/GB). Measurements taken after 2.5% strain in an argon environment at 288°C.

	C	D/GB		D
		DC	GB	
Average displacement (nm)	185	172	132	66
Maximum displacement (nm)	660	541	320	174

Table 4.15 Measured slip plane surface trace angle with respect to the tensile direction and the measured displacement vector due to the DC slip. Slip plane angle was determined using SEM images of the surface. The displacement vector was determined by combining the DIC and confocal microscopy displacement measurements.

DC surface trace angle measured in SEM (degrees)	Normalized measured displacement vector		
	X	Y	Z
60	0.303	0.369	1.331
30	-0.558	1.052	-0.763
60	0.624	0.624	1.105
30	0.627	0.654	1.086
66	0.832	0.916	-0.685
83	-0.406	-1.121	0.761
58	-0.506	-0.831	1.027
83	0.685	1.127	-0.512
66	0.966	0.289	-0.992
66	1.288	0.511	-0.280
62	0.441	-0.128	1.338
60	0.945	0.762	0.725
46	-1.001	-0.859	0.510
37	1.173	0.666	-0.425
34	0.542	0.266	1.278
34	-1.010	0.750	-0.645
49	1.121	-0.583	0.635
77	-1.049	-0.401	0.859
34	-0.955	0.516	-0.907
34	1.174	0.165	-0.770
34	1.033	0.473	-0.842
66	-0.802	0.727	-0.910
84	1.027	0.051	0.971
37	-0.320	0.895	-1.047
55	0.506	0.280	-1.290
29	1.190	0.291	0.705
80	-1.270	0.090	0.615
65	-0.739	0.290	1.170
51	-1.013	0.361	-0.918

Table 4.16 Average of three highest values of stress taken near the DC-GB intersection (within 1 µm) from the 13Cr15Ni ASCC3 high resolution EBSD results.

	Stress (GPa)	Error (GPa)
Continuous	12.1	3.3
Disc. W/ GB slip	12.1	4.0
Discontinuous	14.4	2.3

Table 4.17 Cracking density of 13Cr15Ni ASCC3 and 16Cr12Ni CSCC1 after water straining. ASCC3 was strained in argon to 2.5%, and CSCC1 was strained in argon to 1.7% prior to straining in water.

Alloy	Sample designation	Total Strain %	Cracking #/mm^2
13Cr15Ni	ASCC3	4.5	51
		7.2	148
16Cr12Ni	CSCC1	3.5	23
		7.5	34

Table 4.18 Cracking results for 13Cr15Ni ASCC3 and 16Cr12Ni CSCC1. While most cracks in CSCC1 did not have channels after argon straining, 2 of those for the 3.5% and 1 for the 7.5% did have discontinuous channels after the water straining, however it is not clear if there was grain boundary slip or not, so the data could not be used.

	ASCC3 bulk (#/mm^2)	ASCC3 4.5% strain (# of cracks)	ASCC3 7.2% strain (# of cracks)	CSCC1 3.5% strain (# of cracks)	CSCC1 7.5% strain (# of cracks)
Continuous	385	11	27	0	1
Disc. w/ GB slip	192	8	14	0	0
Discontinuous	147	14	27	0	0
Unknown DC-GB intersection type	-	7	18	2	3
No visible DC		9	28	3	2
Triple junction	-	40	79	1	4
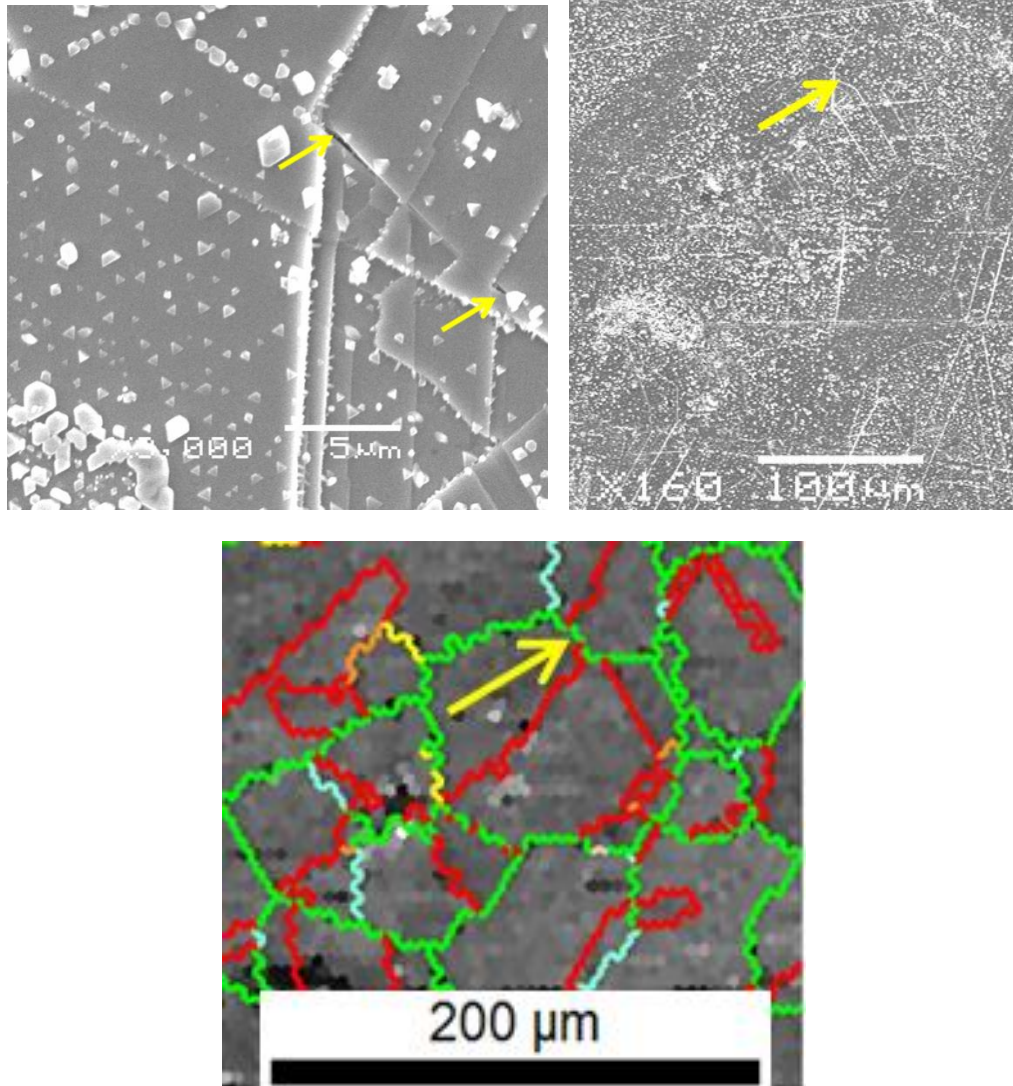
Figure 4.1 Image describes the process of characterizing cracked boundary, in clockwise order: First, cracked boundaries are located in a SEM. Cracked area is imaged at lower magnification to examine surrounding structure and matched to the EBSD scans taken prior to deformation (bottom image). Yellow arrows indicate location of cracked grain boundary. From the EBSD data, the cracked grain boundary character is determined.
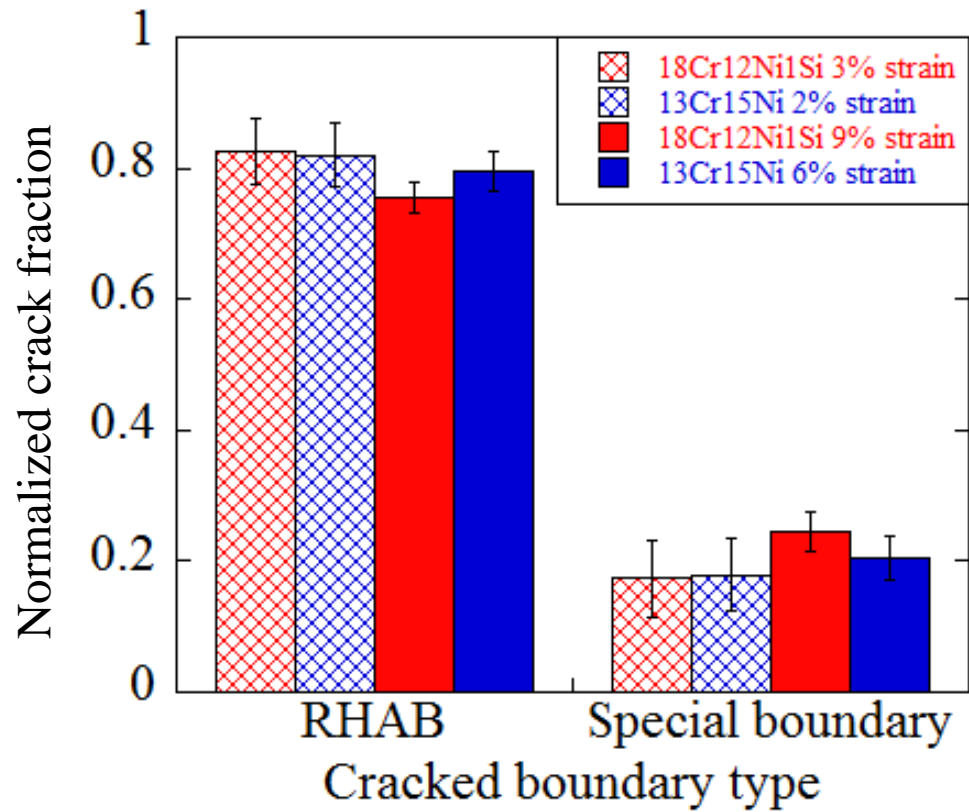
Figure 4.2 Cracking susceptibility based on grain boundary misorientation type of two austenitic steel alloys strained in two increments in a BWR NWC environment. Cracked boundary fractions were normalized to total boundary fractions.
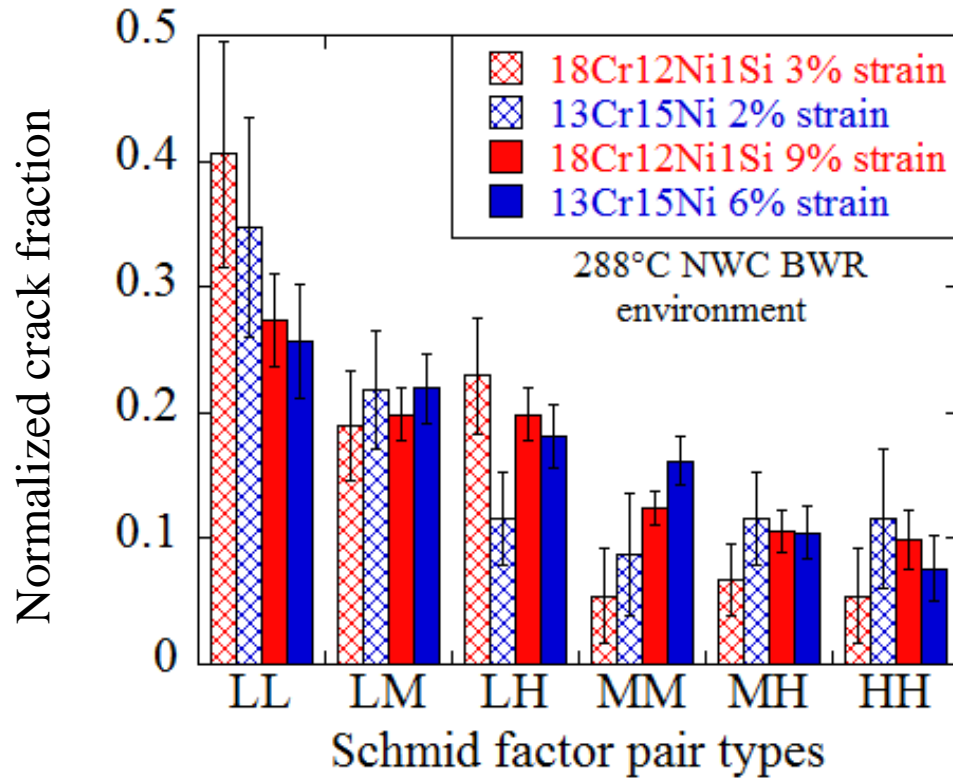
Figure 4.3 Cracking susceptibility based on SF pair types (the Schmid factors of the grain adjacent to the cracked GB). Samples include two austenitic stainless steel alloys strained in two increments in a BWR NWC environment.
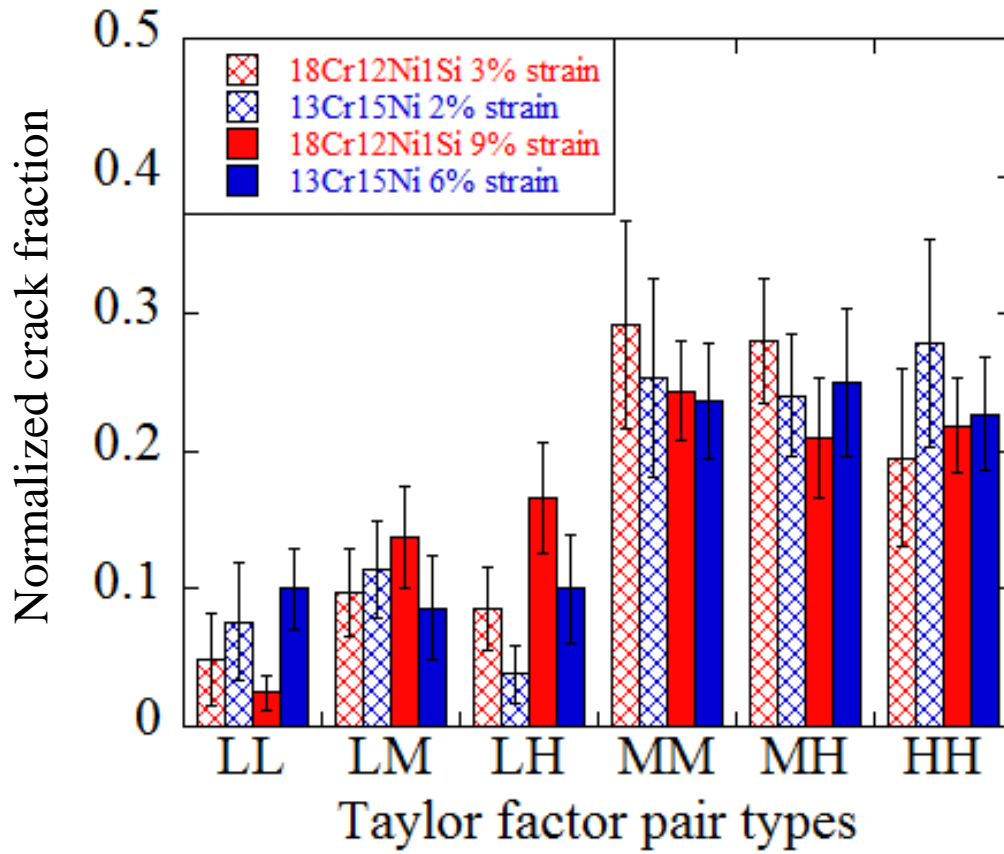
Figure 4.4 Cracking susceptibility based on TF pair types (the Taylor factors of the grain adjacent to the cracked GB). Samples include two austenitic stainless steel alloys strained in two increments in a BWR NWC environment.
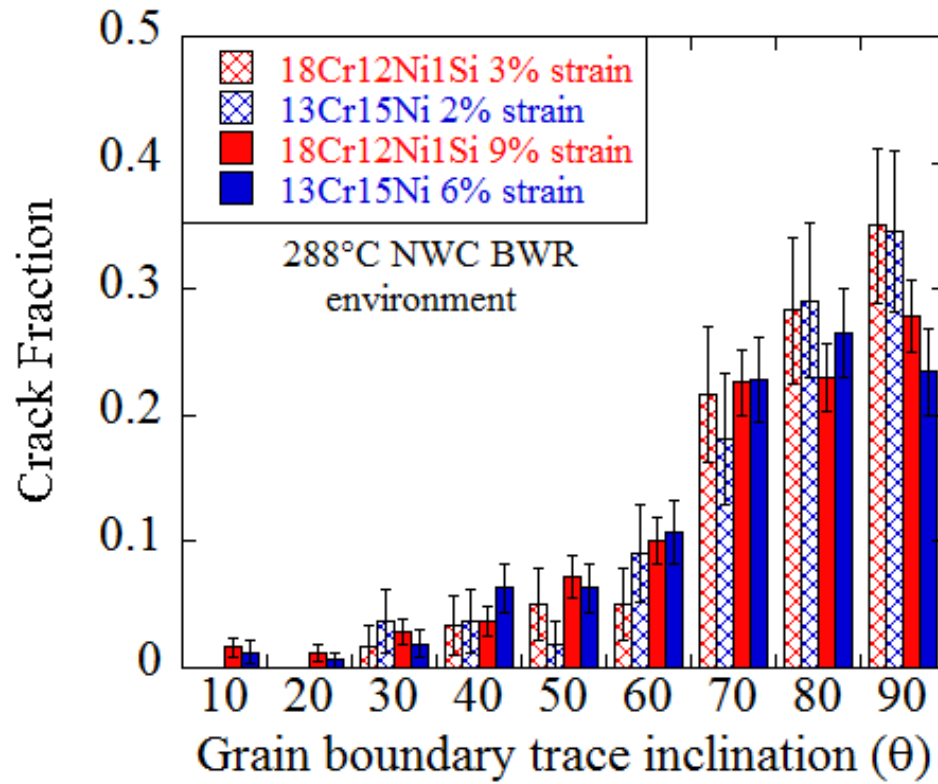
Figure 4.5 Cracking susceptibility based on grain boundary surface trace inclination, with respect to the tensile axis. Samples include two austenitic stainless steel alloys strained in two increments in a BWR NWC environment.
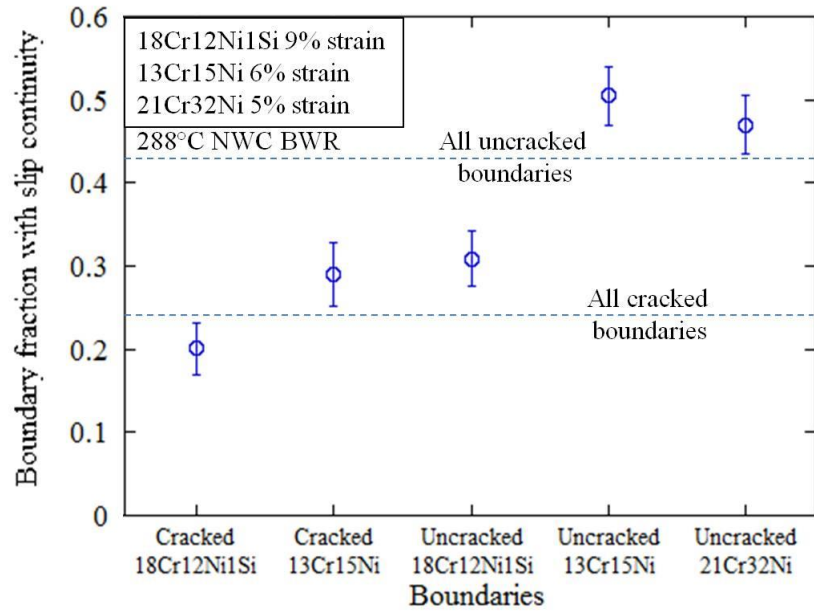
(a)
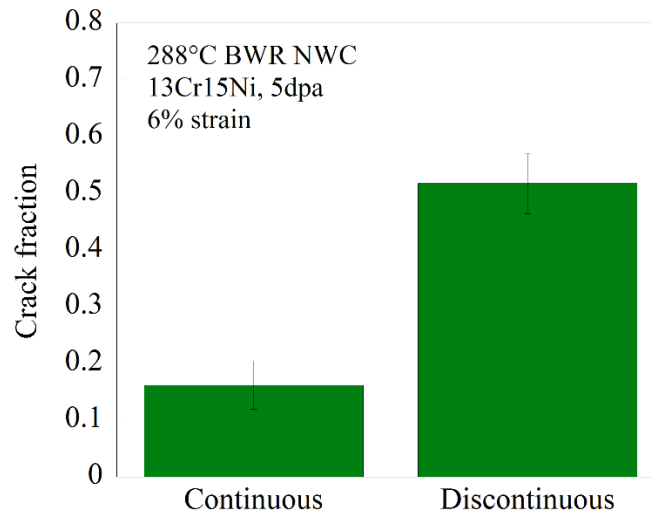
(b)

Figure 4.6 (a) Cracking susceptibility based on dislocation channel continuity at the grain boundary. Samples include three austenitic stainless steel alloys strained in two increments in a BWR NWC environment. Only two alloys underwent cracking during the straining. (b) Shows the cracking fraction of GBs that exhibit continuous and discontinuous slip for the 13Cr15Ni alloy.
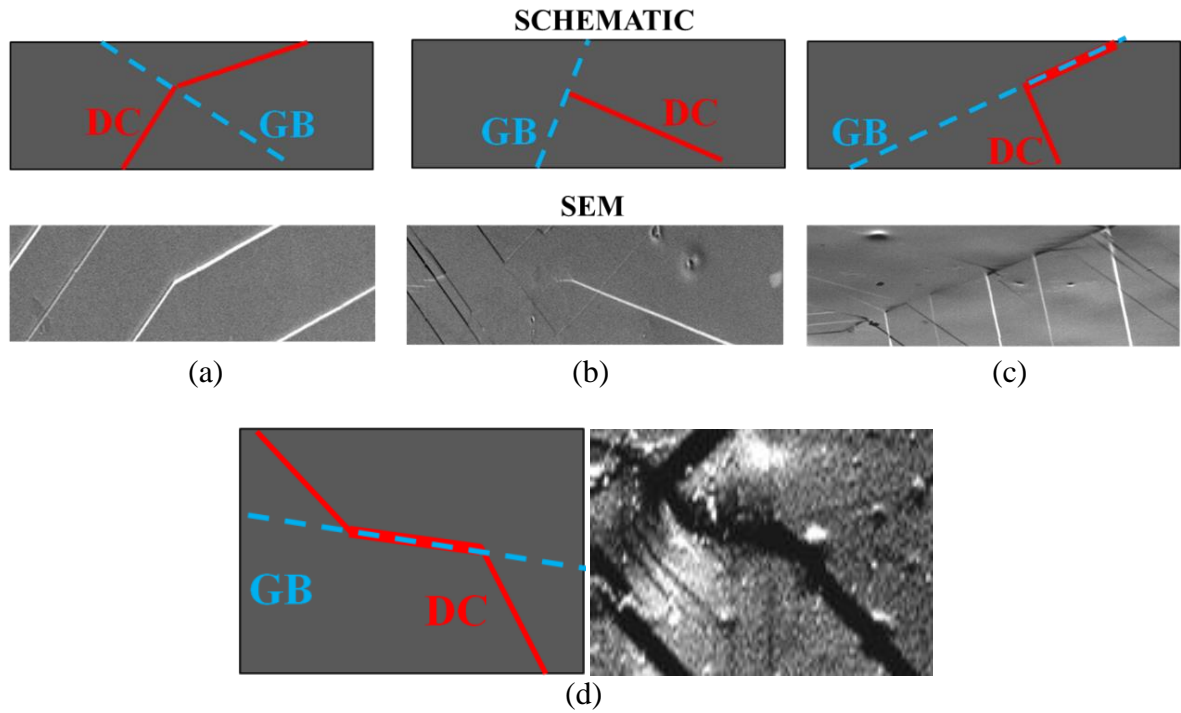
Figure 4.7 Schematic diagram and SEM images of the DC-GB intersection classifications of (a) continuous, (b) discontinuous, and (c) discontinuous-inducing GB slip. In (d), grain boundary absorption with subsequent re-emission is shown in schematic and SEM.
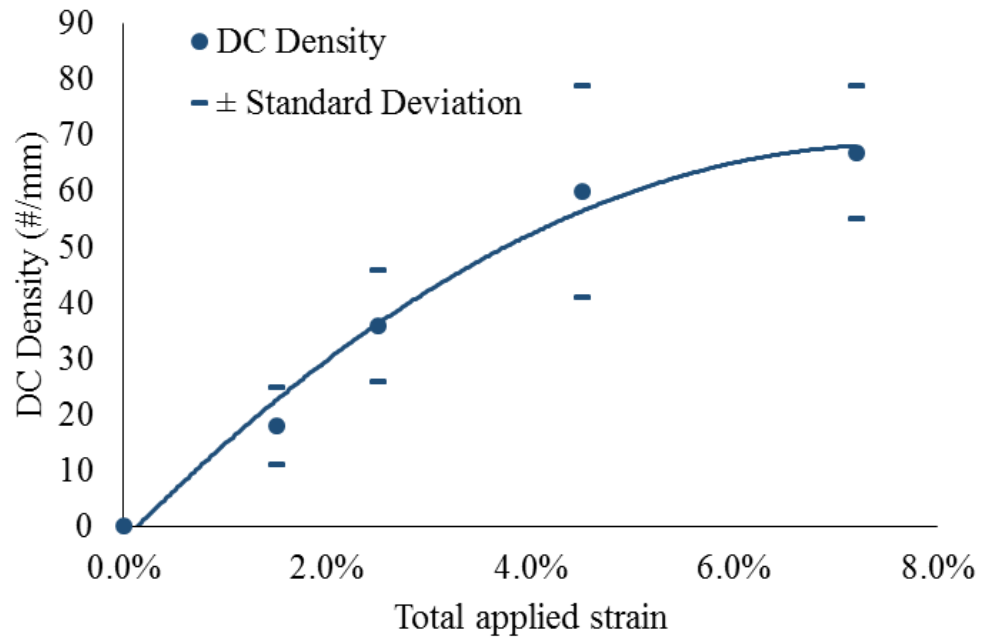
Figure 4.8 DC density vs applied strain. Measurements were taken by drawing lines across SEM images (0.7 mm long) and counting the number of dislocations that intersected the lines. Line shown to emphasize trend.
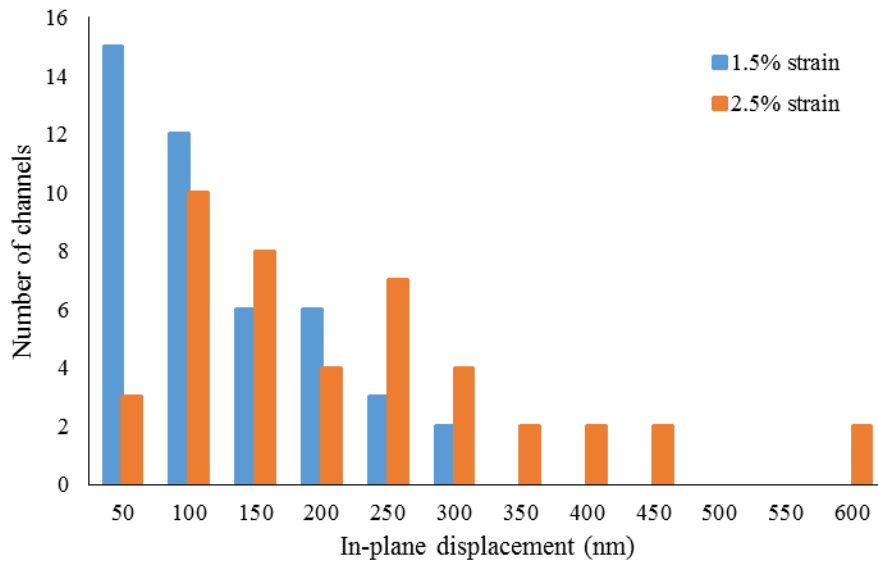
Figure 4.9 Change of in-plane slip of 44 DC when macroscopic strain of the ASCC3 sample increased from 1.5% to 2.5%.
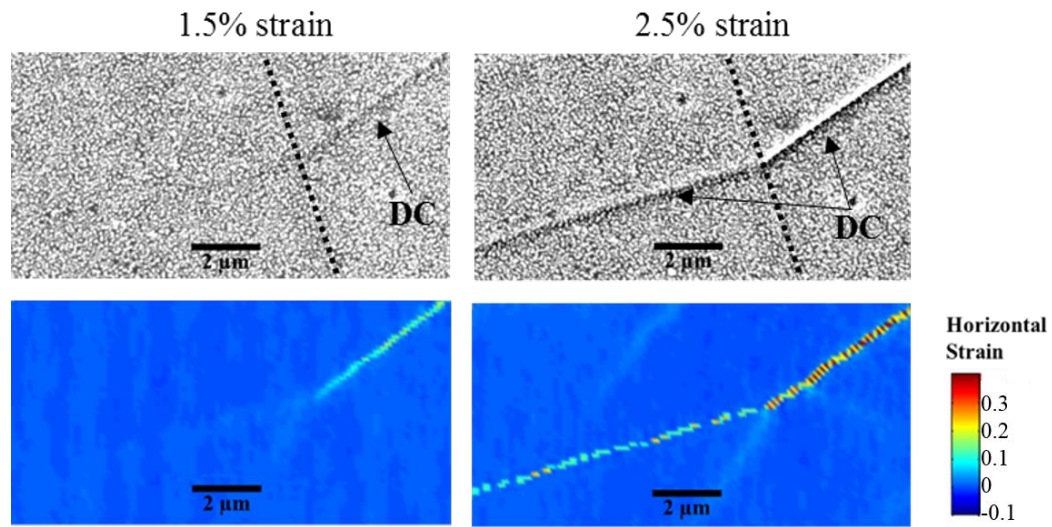
Figure 4.10 SEM image and horizontal (tensile axis) strain map of DC-GB intersection of the ASCC3 sample at 1.5% macroscopic strain (left) and 2.5% macroscopic strain (right). At 2.5%, the channel, which initially appeared discontinuous with small amounts of strain in the adjacent grain (but did not form a channel through the grain), was clearly continuous.
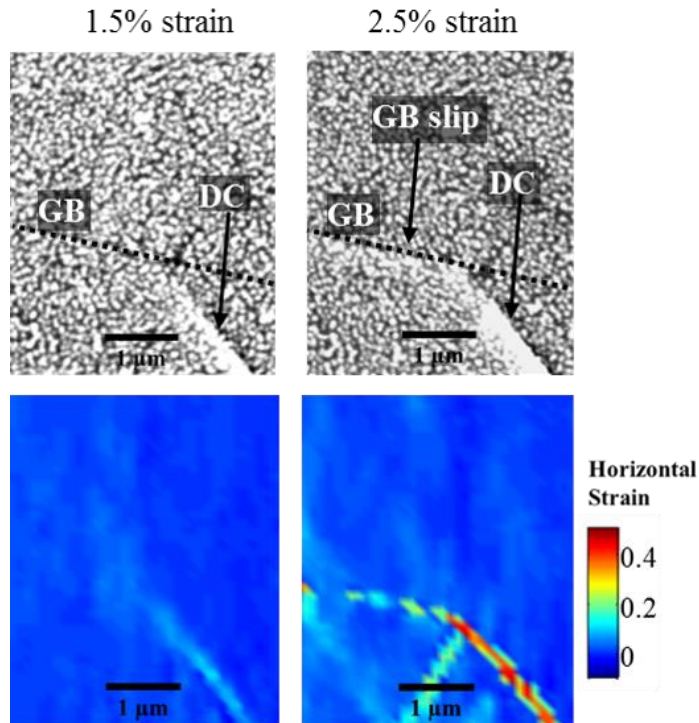
Figure 4.11 SEM image and horizontal (tensile axis) strain map of DC-GB intersection of the ASCC3 sample at 1.5% macroscopic strain (left) and 2.5% macroscopic strain (right). At 2.5%, the channel, which initially appeared purely discontinuous, initiated grain boundary slip.
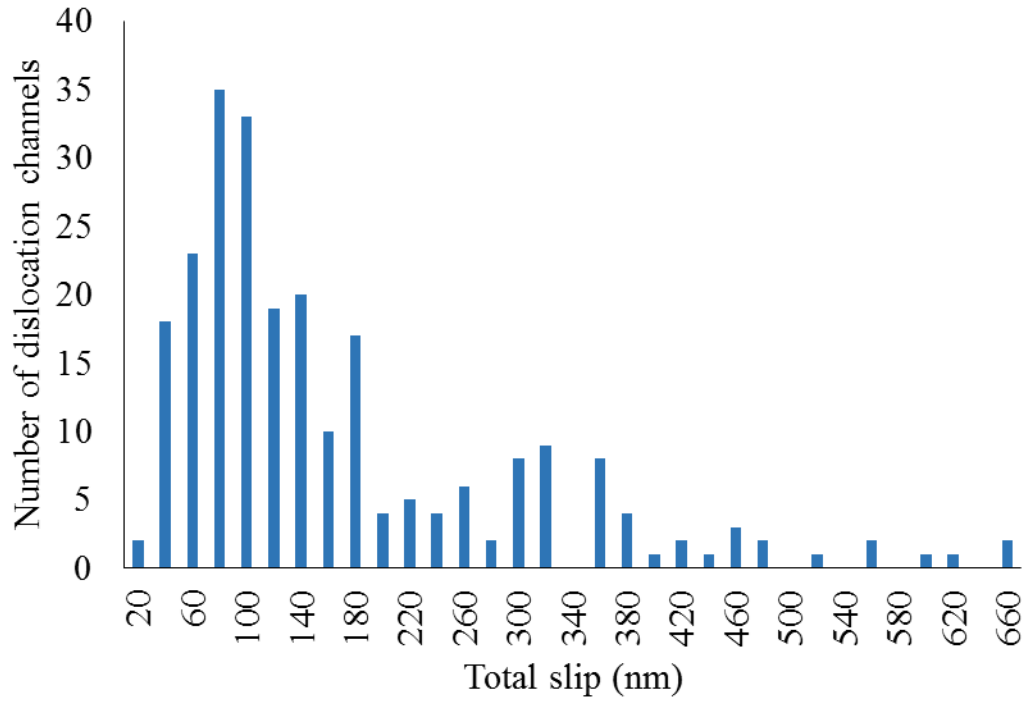
Figure 4.12 Distribution of the amount of displacement within the dislocation channels, determined by measuring the total amount of slip occurring across the channel. Data taken from 243 channels measured on two different irradiated 13Cr15Ni samples (ASCC2 and ASCC3), strained to 3.5% and 2.5% in 288 °C argon.
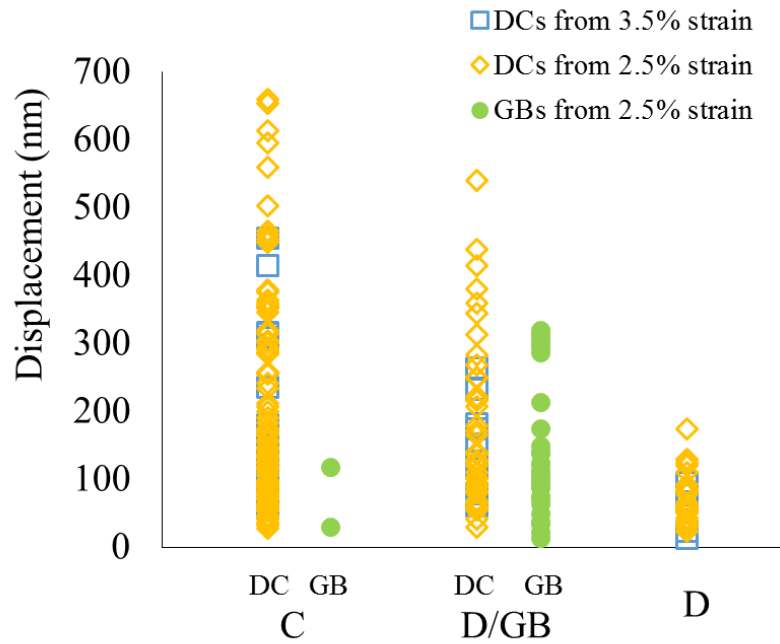
(a)
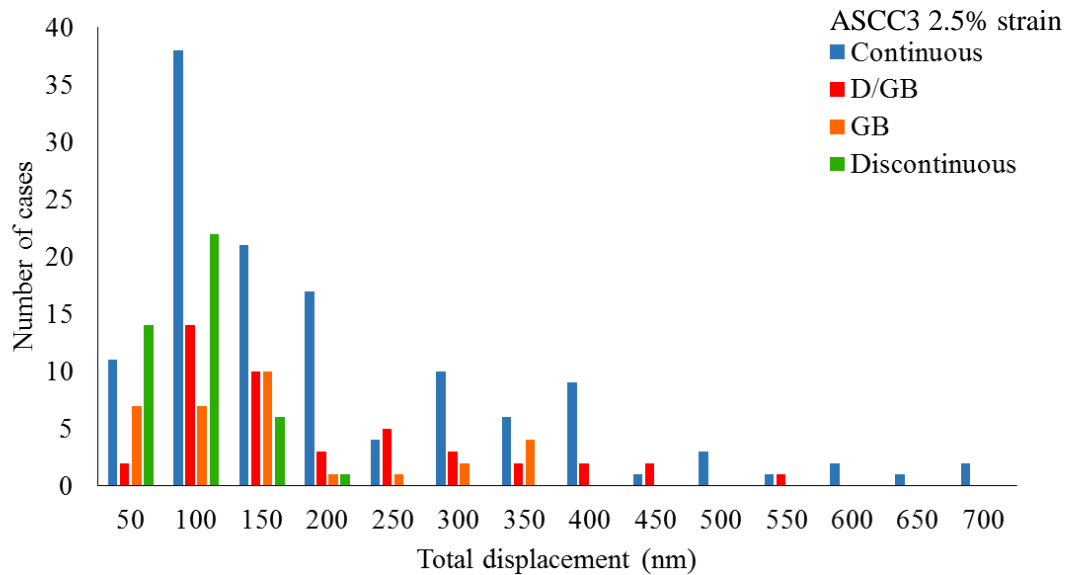
(b)

Figure 4.13. Total slip measurements based on DC-GB intersection type. Results for (a) both 3.5% strained ASCC2 and 2.5% strained ASCC3 specimen, as well as a (b) distribution of sizes for the 2.5% strained ASCC3 specimen.
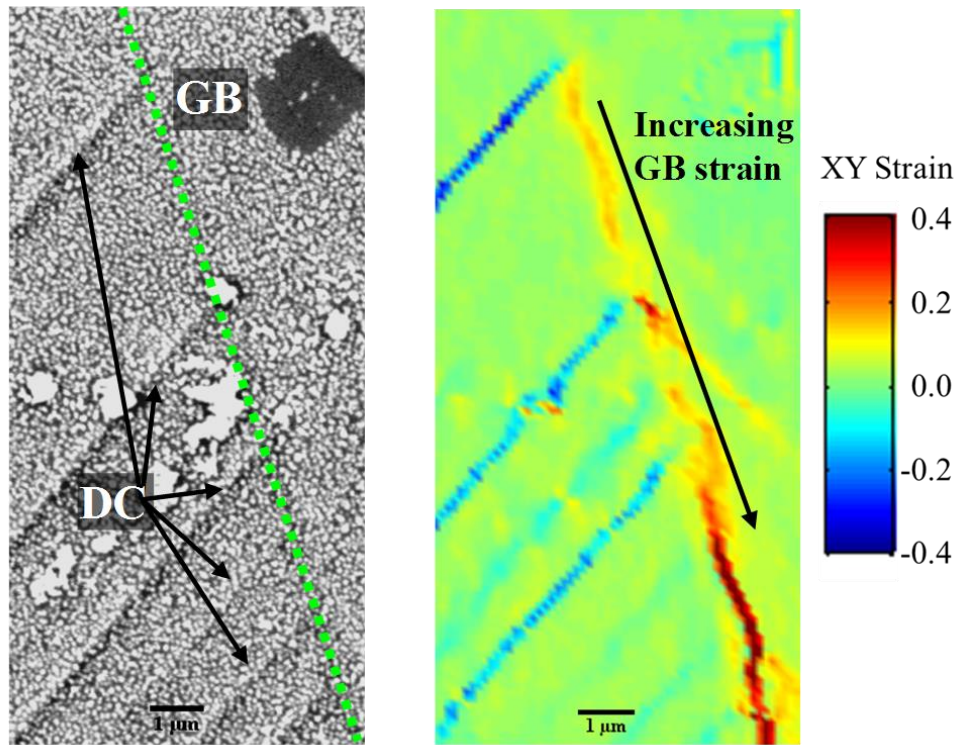
Figure 4.14 Example of D/GB intersection from the Fe-13Cr15Ni ASCC3 specimen strained to 2.5% in a high temperature argon environment, where each DC channel contributes to additional slip in the boundary. GB is labeled, as well as the DCs in the SEM image to the left. The right-side image shows the XY shear strain map.
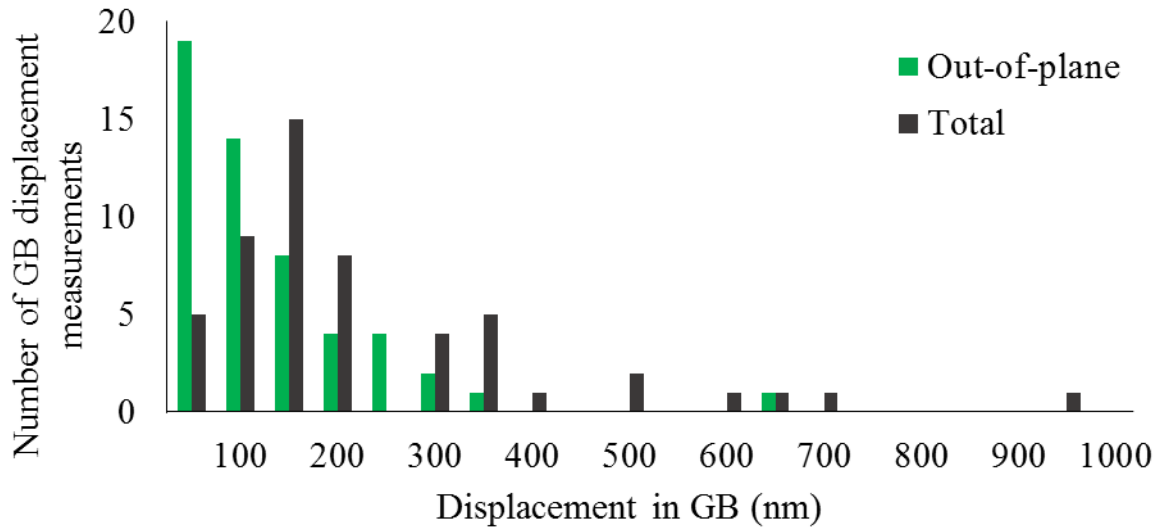
Figure 4.15 Displacement measurements taken at grain boundaries where slip occurred from the Fe-13Cr15Ni ASCC3 specimen after 2.5% strain in a high temperature argon environment. These measurements constitute the total amount of displacement measured, which in some cases was due to multiple DCs causing slip in the GB. Graph depicts out-of-plane (normal to the surface) and total displacement.
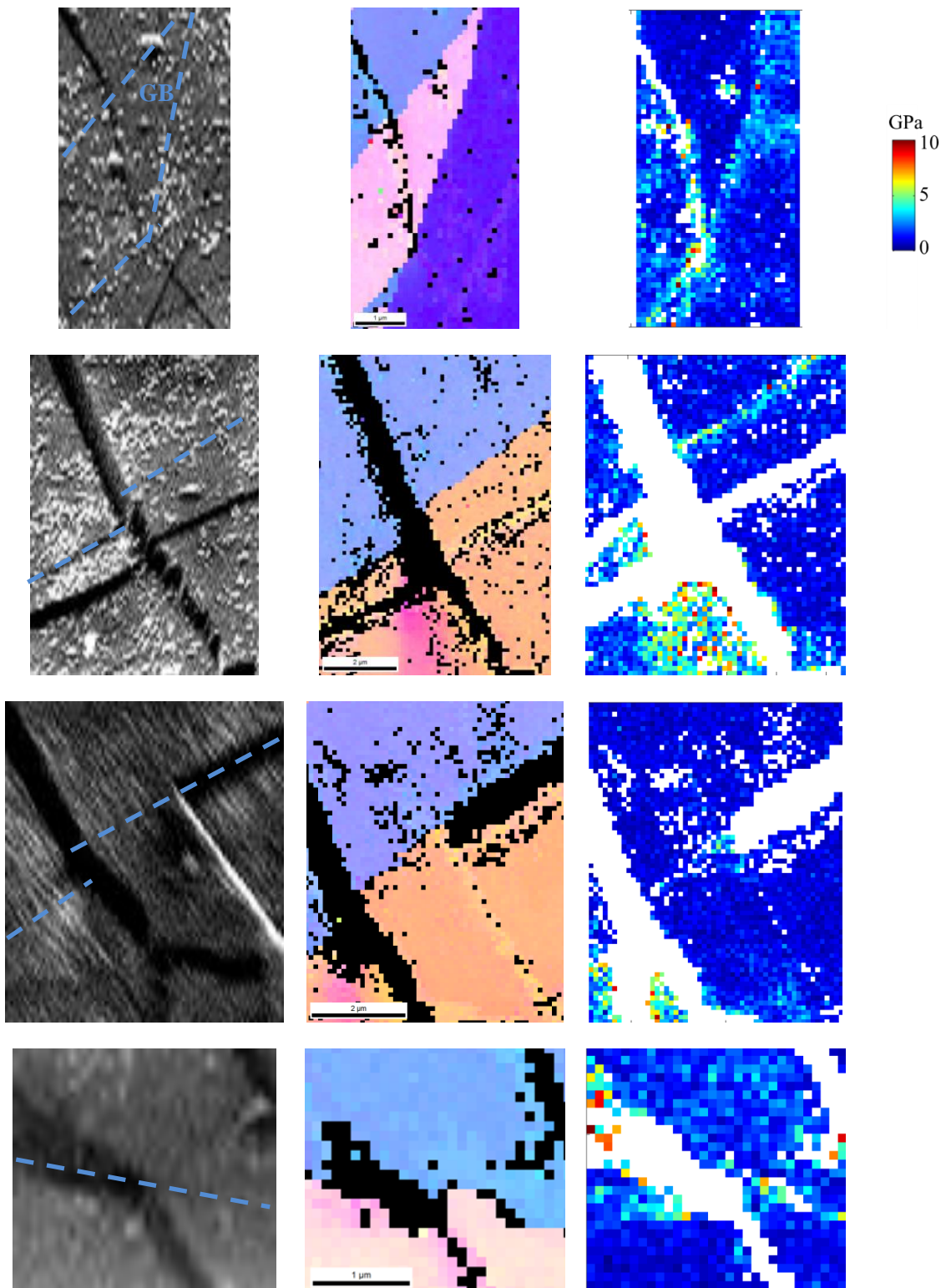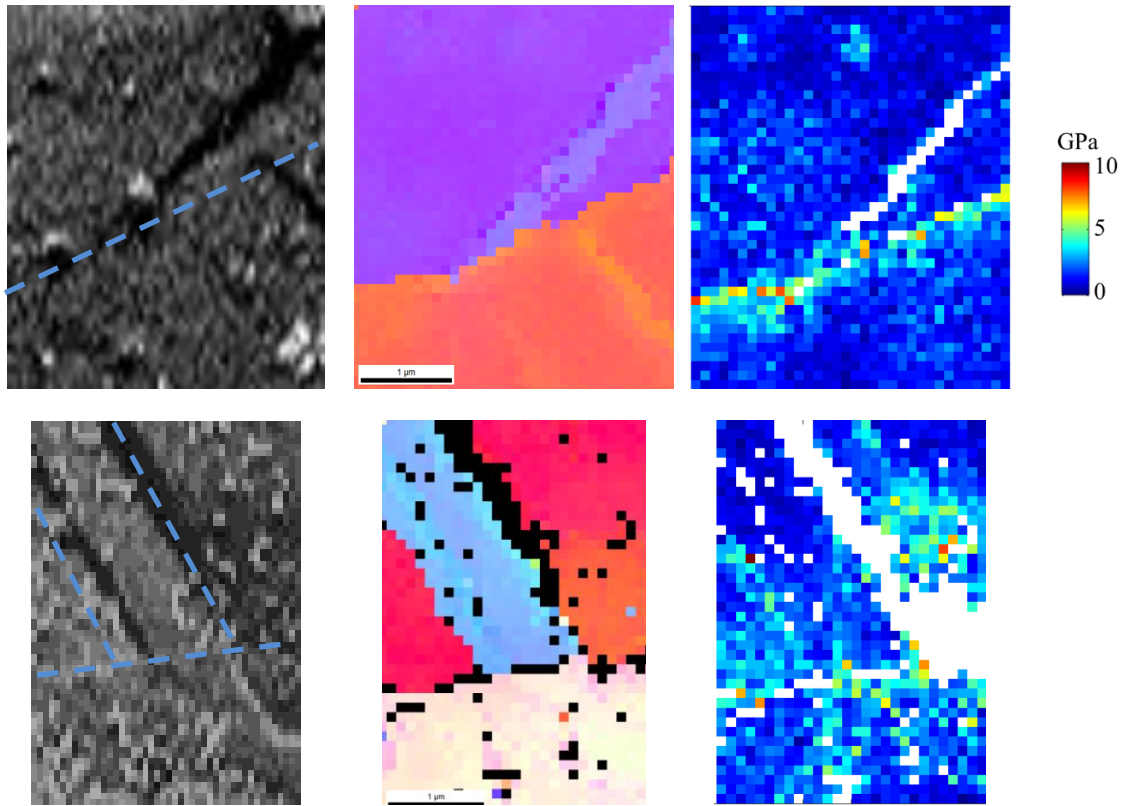
Figure 4.16 Elastic stress measurements. In the first column, an SEM image of the intersection is shown, with GBs marked in blue. The second column shows EBSD measurements depicting orientation (each grain is a distinct color). In this column, black pixels represent locations where orientation could not be determined. The third column shows the Von Mises stress results of the high resolution EBSD stress analysis. White pixels represent areas where data could not be collected. Depicted in this figure are, in order, two examples of continuous slip, two examples of discontinuous slip inducing GB slip, and two examples of discontinuous slip.
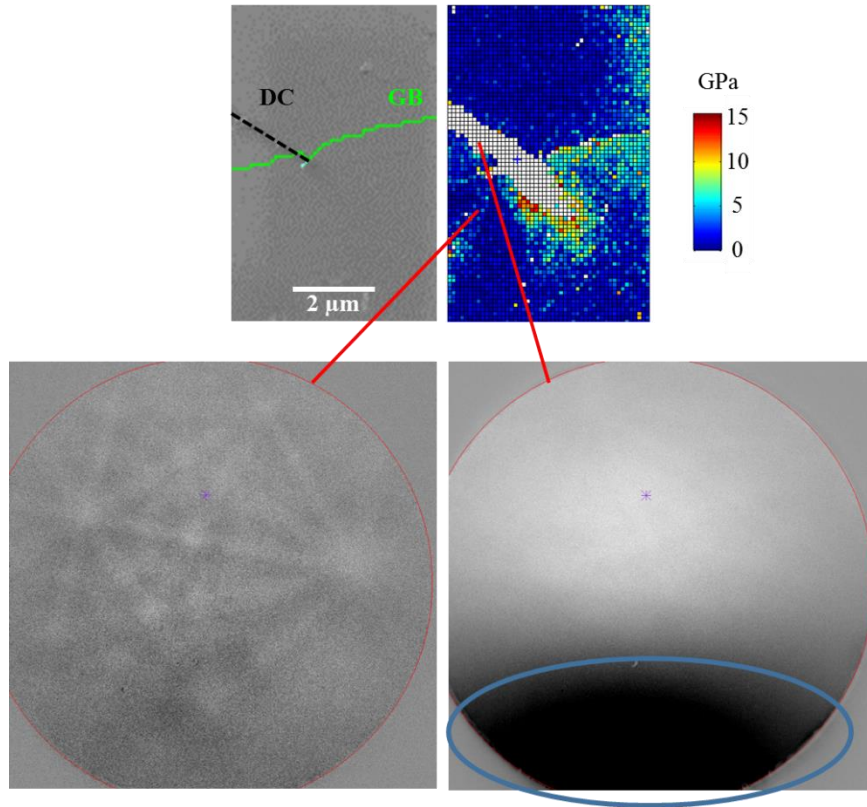
Figure 4.17 EBSD map of a discontinuous DC-GB intersection. The uncharacterized (white) area along the length of the channel was where the EBSD pattern was partially blocked by the emerging channel.
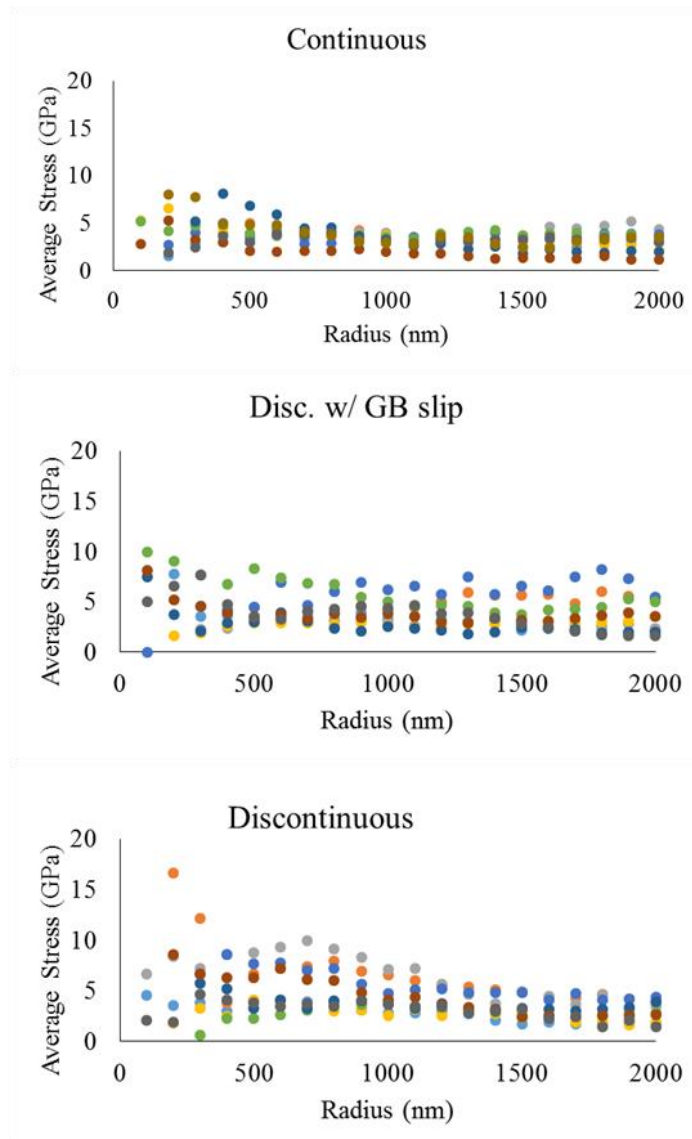
Figure 4.18 Results from 10 continuous, 9 disc. w/ GB slip, and 9 discontinuous DC-GB intersections. Stress at varying radii from the intersection points was averaged at 100 nm steps and reported. Each color represents a different DC-GB intersection that was characterized with high resolution EBSD.
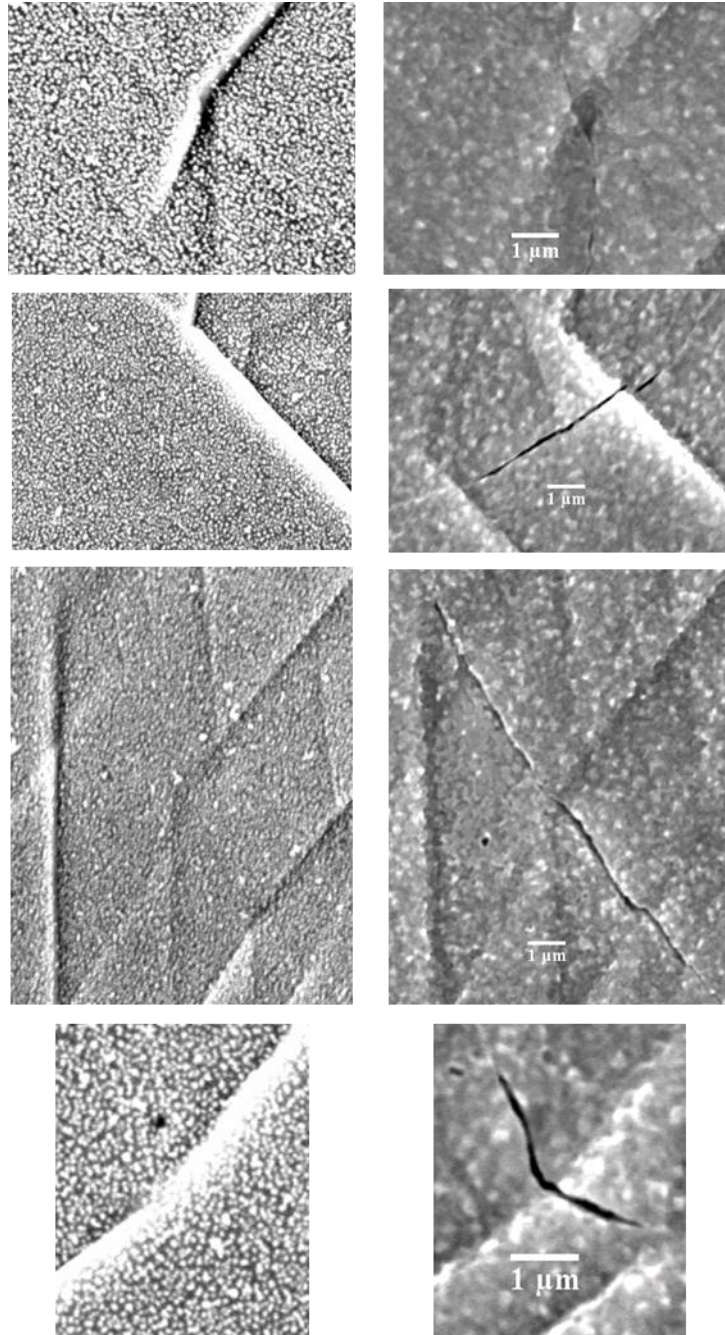
Figure 4.19 Examples of cracks that formed where continuous DC-GB intersections were observed in after the 2.5% argon strain step (left). SEM images of the cracks are shown on the right.
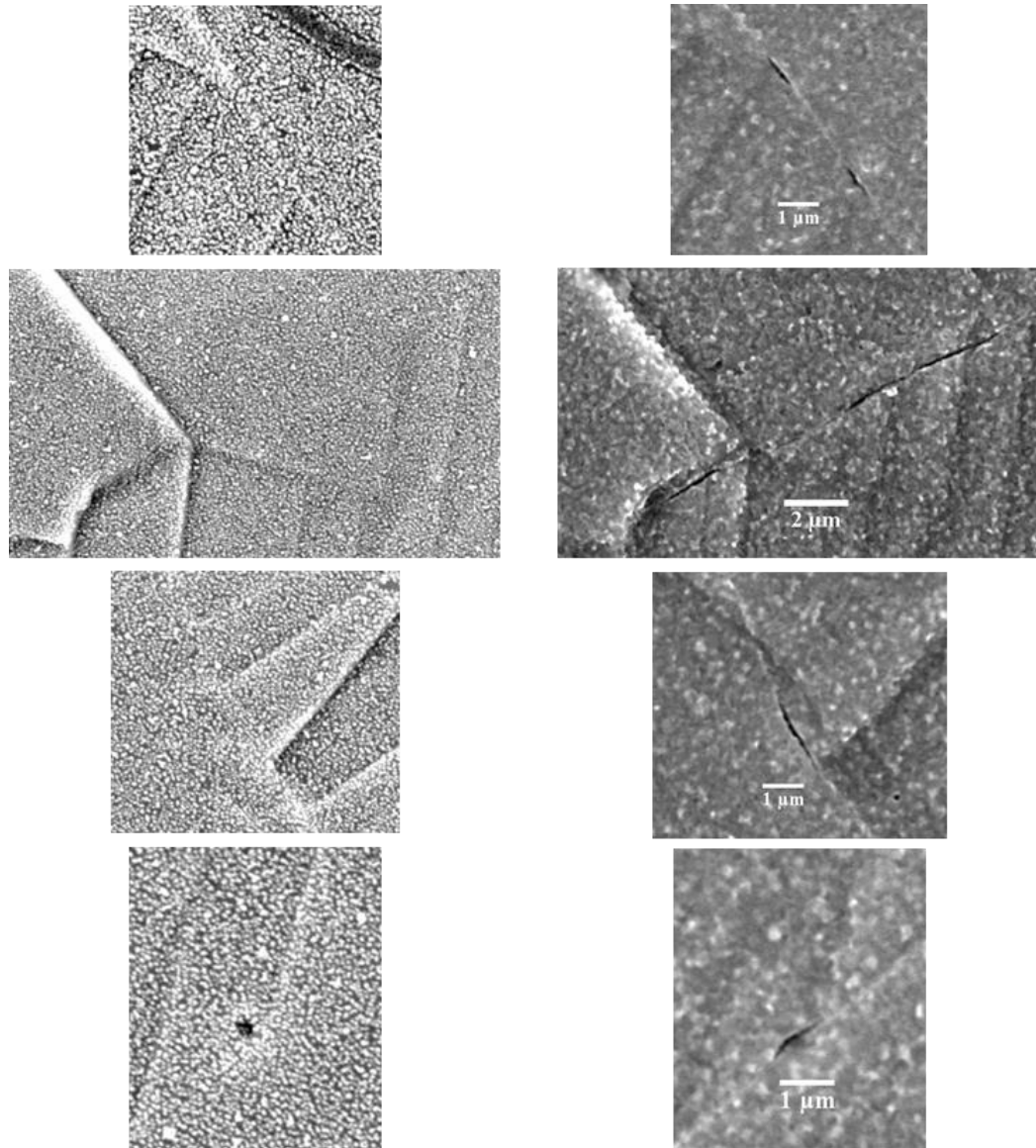
Figure 4.20 Examples of cracks that formed where discontinuous with induced GB slip DC-GB intersections were observed in after the 2.5% argon strain step (left). SEM images of the cracks are shown on the right.
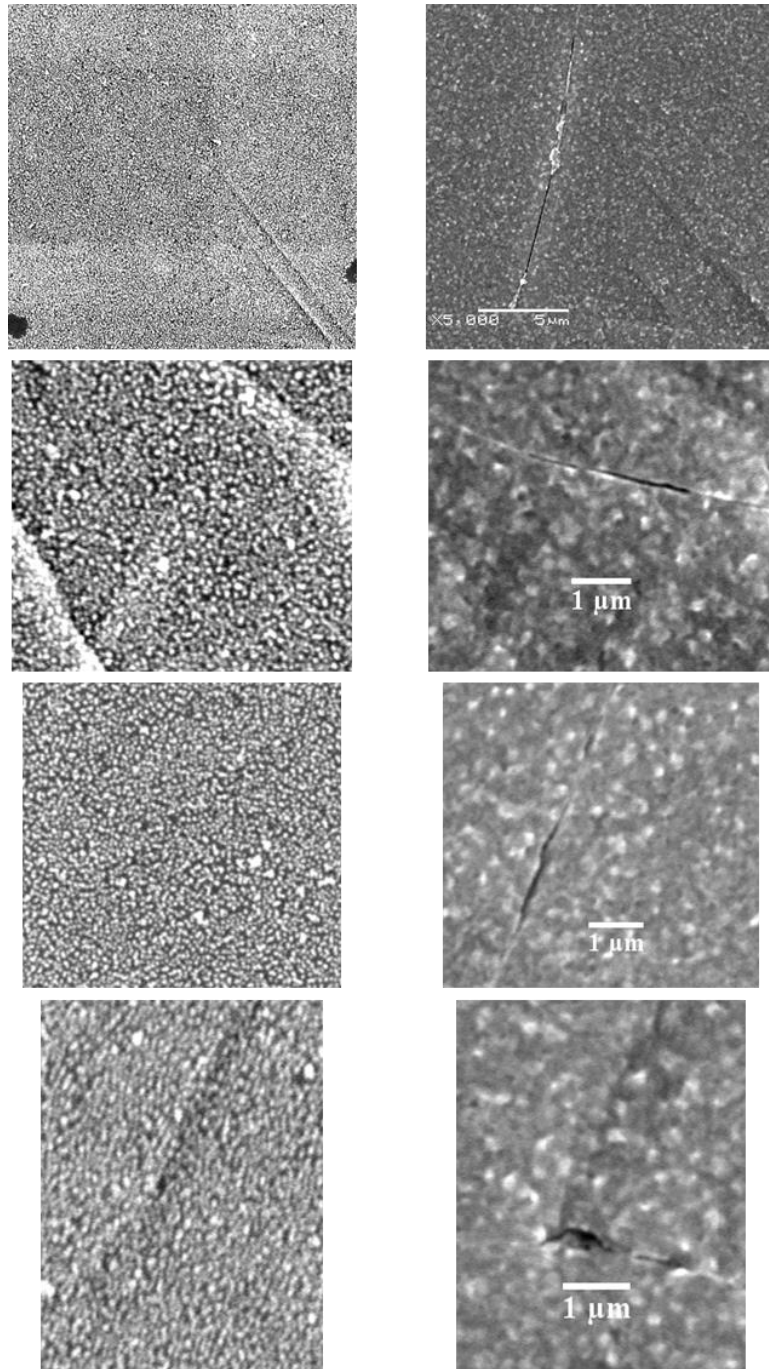
Figure 4.21 Examples of cracks that formed where discontinuous DC-GB intersections were observed in after the 2.5% argon strain step (left). SEM images of the cracks are shown on the right.
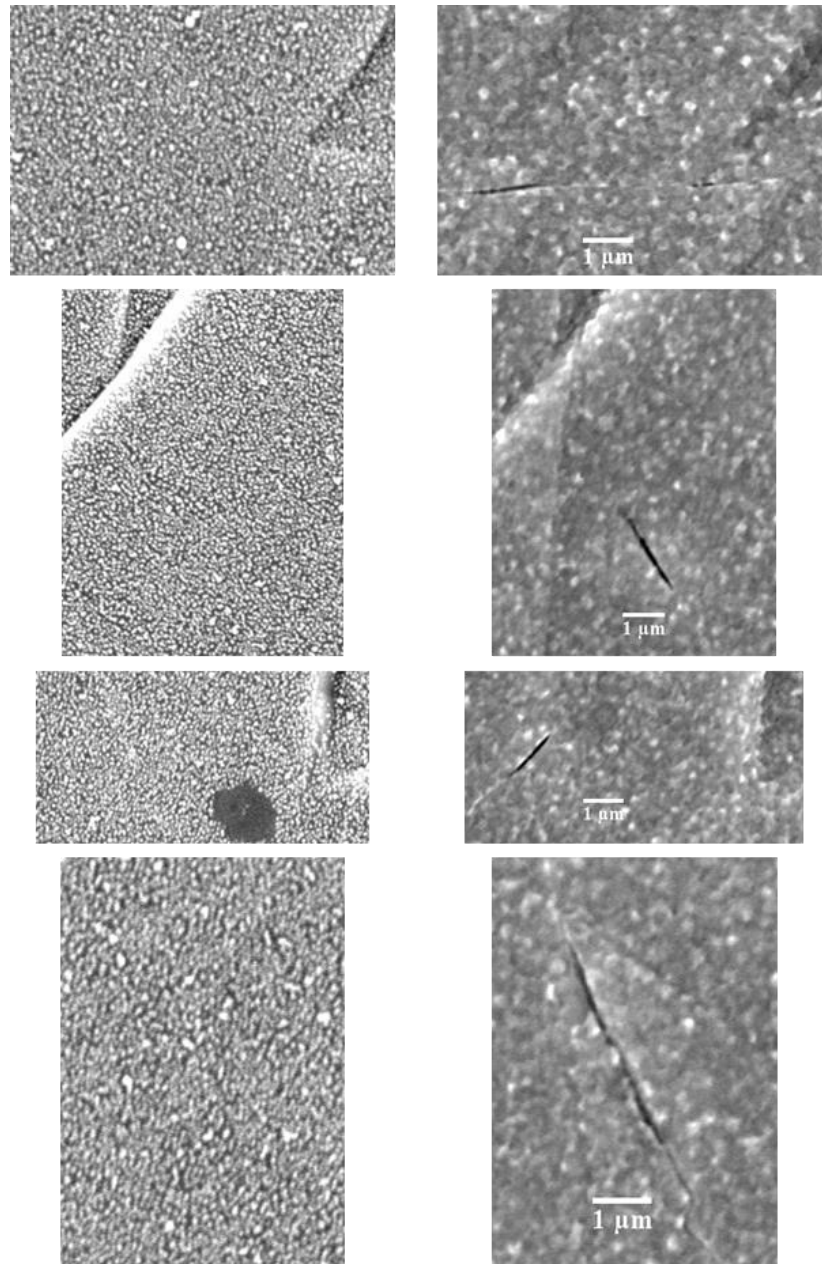
Figure 4.22 Examples of cracks that formed where no DC existed after the 2.5% argon strain step (left). SEM images of the cracks are shown on the right.
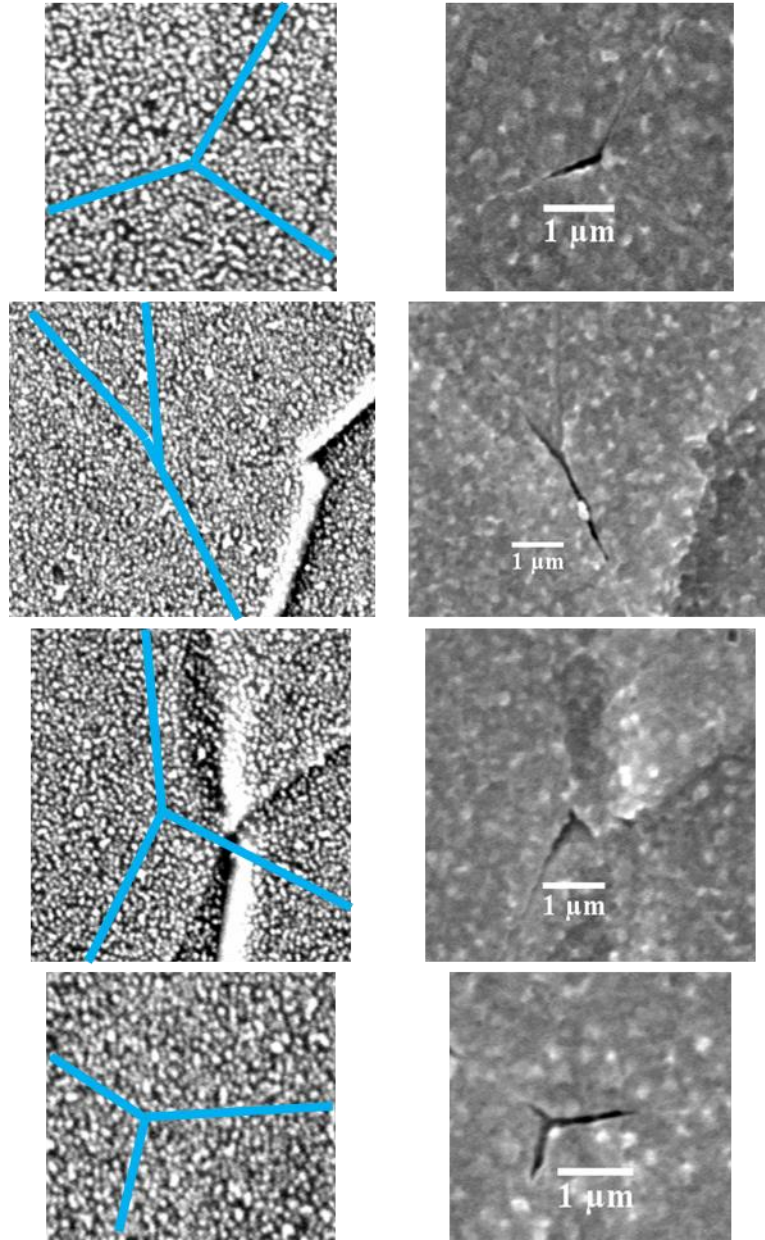
Figure 4.23 Examples of cracks that formed at triple junctions. Left images show after the 2.5% argon strain step, with the GBs marked with light blue lines. SEM images of the cracks are shown on the right.

Chapter 5

DISCUSSION

This chapter focuses on the effects of dislocation channeling on IASCC. It was shown in the previous chapter that there are three classifications of DC-GB intersections, which exhibit varying degrees of cracking susceptibility. This chapter will discuss the importance of the interaction of the channel with the grain boundary in the IASCC mechanism, in order to determine the mechanism linking dislocation channeling and IASCC.

First, the chapter will begin by discussing the general properties exhibited by grain boundaries that have a propensity to crack. Similar to previous results in other environments, the normal stress acting on the grain boundary has a large influence on cracking susceptibility of the grain boundaries. The second section will discuss the important characteristics of the dislocation channels. The effects of stress and grain boundary slip due to dislocation channels will be analyzed in the third section, showing the importance of stress in the IASCC mechanism. Past results will be discussed in the context of the worked contained within this thesis in the final section.

5.1 IASCC susceptible grain boundaries

Table 4.6 shows the properties of the grain boundaries found to crack for the 13Cr15Ni ASCC1 and 18Cr12Ni1Si HSCC1 samples. These data, as well as those in

Figure 4.2 through Figure 4.5, show evidence that certain types of boundaries are more susceptible to cracking, namely random high angle boundaries, boundaries adjacent to a low Schmid Factor (SF) or high Taylor Factor (TF) grain, and boundaries that are close to normal to the tensile axis. It is also evident, from Table 4.4 and Figure 4.6, that grain boundaries where slip is discontinuous are more prone to cracking. It is important to note that in these cases, discontinuous channels were observed after crack initiation, with no prior DIC measurements, and therefore there is no distinction between discontinuous w/ GB slip (D/GB) and discontinuous (D). Once the crack has formed, grain boundary slip can no longer be determined by future measurements. In this section, the importance of grain boundary properties on IASCC susceptibility will be discussed, and the sections that follow will focus on the importance discontinuous DCs.

It is important to understand why these particular characteristics of a grain boundary (RHAB, low SF or high TF adjacent grains, and near perpendicular angle to the tensile axis) cause the boundary to have a higher propensity for cracking. The high cracking susceptibility of RHABs is consistent with other intergranular corrosion studies of austenitic alloys [87,116,117]. RHABs tend to have higher grain boundary energies (see Figure 2.12) and lower cohesive strengths [35,39,118] than low angle or CSL boundaries. RHABs also tend not to transmit the strain from the dislocation channels, as seen in Table 4.8, where a high fraction of boundaries that exhibit discontinuous slip occur at RHABs. As mentioned previously, discontinuous slip has also been linked to cracking, and will be discussed in great detail later in this section. The combination of a lower cohesive strength and a decreased probability of accommodating strain through slip

transmission across the boundary likely contribute to a condition that is favorable for cracking.

The angle of the grain boundary with respect to the tensile axis, and the values of the SF and the TF are related to the amount of normal stress exhibited on the grain boundary from the applied tensile stress. Grain boundaries normal to the tensile axis have the highest resolved normal stress from the applied tensile stress. The relation between the applied tensile stress (σ_T) and resolved normal stress (σ_N) on the grain boundary is

$$\sigma_N = \sigma_T \cos^2 \alpha ,\tag{5.1}$$

where α is the angle between the tensile axis and the normal to the grain boundary plane. In this work, α is unknown, as the grain boundary inclination is not fully known. It is, however, related to the surface trace angle of the grain boundary (θ), which has been measured in this work. West [119] showed that the grain boundary trace may be estimated based on the surface trace. The angle α may be defined as the dot product of the vector normal to the grain boundary plane ($\boldsymbol{n}$) and the tensile direction ($\boldsymbol{x}$), resulting in

$$\alpha = \arccos\left(\frac{\boldsymbol{n} \cdot \boldsymbol{x}}{|\boldsymbol{n}||\boldsymbol{x}|}\right).\tag{5.2}$$

The grain boundary plane may be defined by two vectors, the grain boundary trace on the specimen surface ($\boldsymbol{v}_1$) and the trace of the grain boundary on the side of the specimen, perpendicular to the specimen surface ($\boldsymbol{v}_2$). These two vectors are defined as

$$v_1 = \left(\frac{1}{\tan(\theta)}, 1, 0 \right) \tag{5.3}$$

and

$$v_2 = \left(\frac{1}{\tan(\varphi)}, 0, 1 \right), \tag{5.4}$$

where φ is the angle between the tensile axis and the trace of the GB on the side surface of the specimen perpendicular to the specimen surface. These variables are depicted graphically in Figure 5.1. In these definitions, the X and Y axes are in-plane with the sample surface, with the X axis being parallel to the tensile axis. The Z axis is perpendicular to the sample surface. The normal vector to the grain boundary plane may be found taking the cross produce of these two vectors, resulting in

$$n = \begin{vmatrix} i & j & k \\ \dfrac{1}{\tan(\theta)} & 1 & 0 \\ \dfrac{1}{\tan(\varphi)} & 0 & 1 \end{vmatrix} = \left(1, -\frac{1}{\tan(\theta)}, \frac{1}{\tan(\varphi)} \right). \tag{5.5}$$

Returning to equation (5.2), and solving it with the definition of n in equation (5.5), α is defined in terms of θ (known) and φ (unknown),

$$\alpha = \arccos \left(\frac{(1,0,0) \cdot \left(1, -\frac{1}{\tan(\theta)}, \frac{1}{\tan(\varphi)} \right)}{\sqrt{1 + \frac{1}{\tan^2(\theta)} + \frac{1}{\tan^2(\varphi)}}} \right) = \arccos \left(\sqrt{1 + \frac{1}{\tan^2(\theta)} + \frac{1}{\tan^2(\varphi)}} \right)^{-1}. \tag{5.6}$$

The angle φ may vary from 0° to 90°, however, not all values are equally likely. Instead, the rotation around the grain boundary surface trace vector (v_1) are equally likely to exist within the specimen. The angle of rotation (β) may be defined in a similar way as α in equation (5.2), though the dot product defining β is between the normal vector (n) and the z direction (z), which is normal to the sample surface. This results in

$$\beta = \arccos\left(\frac{n \cdot z}{|n||z|}\right) = \arccos\left(\frac{\dfrac{1}{\tan(\varphi)}}{\sqrt{1 + \dfrac{1}{\tan^2(\theta)} + \dfrac{1}{\tan^2(\varphi)}}}\right), \tag{5.7}$$

which may be rewritten in terms of φ as

$$\varphi = \arctan\sqrt{\frac{\dfrac{1}{\cos^2(\beta)} - 1}{\dfrac{1}{\tan^2(\theta)} + 1}} \tag{5.8}$$

The angle φ may be solved at varying θ and β, using equations (5.2) and (5.5). Figure 5.2 shows that the lower values of φ are weighted more heavily, especially at low values of θ. With φ calculated, α may be determined using equations (5.2) and (5.5). Figure 5.3 shows values of α with respect to β at various angles of θ. At a θ of 90°, each angle α is equally likely, however at lower thetas, α is weighted more heavily towards higher values closer to 90°.

One final consideration to account for, however, is that while all angles of β are equally likely to exist within the specimen, they are not all equally likely to be observed (intersect the sample surface in a particular area being analyzed). An angle of 0 degrees, which is parallel to the surface cannot intersect the surface (two parallel planes do not

intersect), whereas an angle of 90 degrees is the most likely to intersect the surface, as it is perpendicular to the surface. This may be accounted for by the probability function $P(\beta)$:

$$P(\beta) = \frac{\int_a^b \sin(\beta)d\beta}{\int_{0°}^{90°} \sin(\beta)d\beta} \, , \qquad (5.9)$$

where a and b denote the range of angles of interest. Figure 5.4(a) shows the probability distribution of normal stress on the grain boundary for given angles of θ. Figure 5.4(b) shows that the normal stress at the GB may vary from 0 to the maximum possible stress for a given θ, but has a higher probability of existing at the maximum possible stress. Thus using the measured surface trace angle to estimate the normal stress on the grain boundary is justified. Had shear stress at the grain boundary been controlling crack initiation, it is expected that the GB angles near 45° to the tensile axis would have been more susceptible to cracking, as shown in Alexandreanu's studies on Ni-16Cr-9Fe in 360 °C primary water [120].

As SF and TF tend to be inversely related, the normal stress dependence on SF will be discussed here, with a similar line of reasoning applying inversely to TF. The increased susceptibility to IASCC of grain boundaries adjacent to at least one grain with a low Schmid factor has been observed by others [10,121]. West et al. [121] used the Schmid factor of the neighboring grains to determine the normal stress acting on the grain boundary, resulting in the Schmid-Modified Grain Boundary Stress (SMGBS) model. This model takes into account not only the orientation of the grain boundaries with respect to the tensile axis, but also the ease with which the grains adjacent to the

boundary are able to accommodate the applied stress, as grains with low Schmid factors were determined to have a low propensity to deform, and this resulted in higher stress at the grain boundary. This model showed that lower Schmid factors result in higher normal stress at the grain boundary, which correlated with cracking at the grain boundary.

These characteristics of GBs with higher susceptibility to IASCC show that normal stress acting on the grain boundary plays an important role in the cracking mechanism. This is observed from the orientation of the GB with respect to the tensile axis, as well as the propensity for deformation of the adjacent grains (SF and TF). It is also observed that the interaction of the DC at the GB is important in the cracking mechanism, and the remaining sections of this chapter will discuss the importance of these interactions.

5.2 The characteristics of dislocation channels

As stated previously, IASCC was found to be related to dislocation channeling (in particular discontinuous dislocation channels), though the exact manner of the relation is unclear. In order to understand the connection linking dislocation channels to IASCC, it is helpful to understand the characteristics of the channels.

5.2.1 Slip systems

The active slip planes were determined by matching the angle of the surface trace of the dislocation channel (Table 4.15) with the surface traces of the four possible {111} slip planes. The surface traces of the possible slip planes were determined using the Euler angles of the grain, as determined by EBSD. This was done for 28 channels, from

both ASCC2 and ASCC3 (after 2.5% strain), with good results, where the expected angle and the measured angle were within a few degrees of each other, as shown in Table 5.1.

Each slip plane has three different slip directions associated with it. The possible slip directions could be determined from the EBSD measurements, however, it was not possible to use SEM imaging to determine which directions were active in the slip plane. Instead, the measured displacement vectors (shown in Table 4.15, from measurements taken with DIC and confocal microscopy) were compared to the possible slip direction vectors determined with EBSD. As the slip directions are unit vectors and the displacement vectors are measured vectors of various magnitudes, the slip direction is multiplied by a normalizing constant (M). The difference of the vectors (V_{diff}) is defined as

$$V_{diff_i} = D_i - MS_i \ , \tag{5.10}$$

Where i represents the X, Y, and Z components of the measured displacement vector (D) and the EBSD determined slip direction (S). The fraction of slip unaccounted for by assuming a single slip system is calculated as the magnitude of V_{diff} over the magnitude of the displacement vector. The percentage of slip accounted for by a single slip system (%$_{SSS}$), therefore, was calculated as

$$\%_{SSS} = 100\% - \frac{\sqrt{V_{diff_X}^2 + V_{diff_Y}^2 + V_{diff_Z}^2}}{\sqrt{D_X^2 + D_Y^2 + D_Z^2}} 100\% \ . \tag{5.11}$$

The results for this study are shown in Table 5.2. On average, only 76% of the slip can be accounted for assuming one slip system is active. Some cases are as low as 40%, meaning a single active slip system is unlikely.

While the presence of multiple slip systems could not be confirmed with the SEM or DIC/confocal microscope measurements, it is possible to confirm in TEM. Work performed by Cui et al. [122], Figure 5.5, shows TEM images of cross slip between slip planes within the channel and multiple active slip systems have been seen to operate within a single channel. Cross slip events within a channel can contribute to obstacle bypass [123], widening of a channel, and creation of a new channel on the conjugate plane.

The TEM work by Cui confirms the EBSD/SEM analysis that deduced that multiple slip systems are active within the DCs. This limits some possible analyses that require a detailed understanding of the slip system active within the dislocation channel. Due to this complexity of the slip within the channel, it was not possible to determine which slip directions were active in the slip transmission process. As shown, this can be resolved by comparing the work performed by others using atomistic modeling and TEM work, with the results reported in this thesis.

5.2.2 Properties of dislocation channels based on interactions with grain boundaries

In the results chapter, it was shown that dislocation channels can be divided into three classifications based on their interactions with grain boundaries. These are shown in Figure 4.7. Not only do they vary based on their interactions with the grain boundary, but the results in Figure 4.13 and Table 4.11 through Table 4.13 also show that the degree of slip within the channels varies and is dependent on the classification. Notably, it is seen that channels where slip is accommodated in some form, as is the case with continuous and discontinuous w/ GB slip, undergo significantly more displacement than discontinuous channels, which have no means to resolve the slip at the grain boundary.

In this way, continuous DC-GB intersections are more similar to discontinuous w/ GB slip, in that both have a mechanism to relieve strain at the grain boundary. The back stresses caused by the dislocation pile-up at the grain boundary likely limit the activity of the dislocation source, such that discontinuous DC-GB intersections were not observed to increase in displacement past 200 nm. The stresses caused at discontinuous DC-GB intersections will be discussed further in the next section.

The other important property of the DC-GB classifications is the density, shown in Table 4.9 and defined here as the number of channels per unit length. Continuous DC-GB intersections are by far the most common type observed. When cracking results are discussed, it is important to understand that continuous DC-GB intersections are twice as likely to occur as either of the discontinuous cases. To account for this, cracking was normalized by dividing by the expected number of DC-GB intersections of that type in the examined area.

5.2.3 Effects of increasing strain on dislocation channels

A portion of the work in this thesis involved specimens strained in increments. DC-GB intersections were characterized after straining the specimen in an argon environment and cracking was induced by further straining in a simulated BRW NWC environment. As such, it is important to understand the evolution of slip as additional strain is imposed on the specimens in order to understand the changes in slip behavior during the straining in the BWR NWC environment.

A number of changes are observed in the DCs when strain in the entire specimen is increased. In order for the deformation of the specimen to be accommodated on the

microscopic level, slip must occur, either by the movement of additional dislocations through pre-existing channels, or through the formation of new dislocation channels. The formation of new channels may be determined through examination of the sample surface in an SEM. The results shown in Figure 4.8 show that new channels were forming with additional strain, in particular during the lower levels of strain. During these low levels of strain, slip was also found to increases in all the channels, as shown in Figure 4.9. Slip measurements require DIC, and so the measurements could not be taken at the higher strain levels, which were performed in the BWR NWC environment. However, in order for the deformation on the microscopic level to accommodate the imposed macroscopic strain, as the rate of new channel formation slowed down with increased applied strain, the amount of slip within the channels must have increased at a faster rate.

Due to the increase in the amount of displacement occurring within the channels with additional applied strain, the actual amount of slip within the channels is not known when cracking occurs. Figure 4.13a shows that the size of continuous channels is generally much larger than the size of discontinuous channels, with D/GB slip falling somewhere in-between, tending towards higher amounts of strain similar to continuous channels. The relative amount of displacement within the channels of the three DC-GB intersection types does not change with additional strain (continuous is always largest). What is affected is that actually amount of displacement in a given individual channel, but the amount of increase cannot be measured.

The change in DC-GB intersection classification with additional strain, such shown in Figure 4.10 and Figure 4.11, could be a concern, as the identification of DC-GB intersection type after the straining in argon was used to determine the DC-GB type that

induced cracking after straining in simulated BWR NWC. It is noted that not many DC-GB intersections were observed to change classifications, and as such, not many are expected to with additional strain. There were 44 DC-GB intersections classified after 1.5% strain in high temperature argon that did not change classification. Only two were found that did change classifications after the additional strain. The change in classification with increasing strain is not an unexpected result. As stress on the dislocation pile-up increases, and additional dislocations are pushed down the slip system, there exist levels of stress that, if exceeded, elicit a response in the system. The stress may overcome the barrier that is the grain boundary, and dislocations may either push through the grain boundary, or a new source may nucleate at the grain boundary and emit dislocations into the neighboring grain. This is the result of a discontinuous channel becoming continuous, as seen in Figure 4.10. The dislocation could also overcome the stress required to cause slip in the grain boundary, a result of the dislocation being incorporated into the boundary, as seen in Figure 4.11. Finally, in a corrosive environment, such as the simulated BWR NWC environment, the stress at the GB may be relieved by crack nucleation. This is not likely to occur, however, in the high temperature argon environment. As such, the DC-GB intersection may reach even higher stresses, eventually overcoming the barrier to either slip transmission or absorption into the GB.

The amount of stress required for any of these three options varies greatly, depending on the grain boundary character, as well as the dislocation character. It is difficult to determine the critical level of stress required for any of these three cases, due to the number of variables, however, these results were also observed in the process of

occurring through use of an atomistic model that examined the interactions of dislocations with GBs [122], as well as in-situ TEM straining experiments [124].

Farkas et al. [122] modeled austenitic fcc metal using the Mishin nickel potential. Polycrystalline digital samples were constructed, with a diameter of up to 50 nm. The simulated grain structure was based on actual grain orientations from SCC samples, as determined using EBSD. Uniaxial strain-controlled virtual tensile tests at constant temperature were performed using on the large-scale atomic/molecular massively parallel simulator (LAMMPS) code [125] and the Virginia Tech supercomputer infrastructure, at a strain rate of $3x10^8$ s^{-1}. Dislocations were observed to impinge on a boundary, and the buildup of stress in the boundary region was followed quantitatively. Figure 5.6 shows an example of this. The stress increased locally to values as high as 9 GPa, until the point when the dislocation overcame the barriers at the grain boundary and moved into the boundary and eventually transmitting across the boundary into the adjacent grain.

Additional observations of the interactions between DCs and GBs is provided by Robertson et al. [123,126], using in-situ TEM straining experiments performed on austenitic stainless steel samples irradiated in-situ at in the IVEM-Tandem microscope at Argonne National Laboratory at room temperature with 1 MeV Kr ions to a doses between 0.1-1 dpa [124,127]. The in-situ straining revealed dislocation pile-ups in an irradiated 304 stainless steel [124]. The TEM experiments performed by Cui et al. [127] revealed, in Figure 5.7a, over 60 dislocations in a pile-up at the grain boundary before the stress was relieved through slip transmission into the adjacent grain, first through the $1a_{out}$ slip system, and then, as shown in Figure 5.7b, in a second slip system, $1b_{out}$. Figure 5.7c shows that both slip system continued to be active with further straining until

184

eventually a large number of dislocations from the grain boundary were suddenly released and pushed through both slip systems. shown in Figure 5.7d. Unlike the atomistic model, stress could not be measured in the TEM experiments, but qualitative observations were made based on the observed dislocations in the pile-up and stress calculations.

Observations of in-situ TEM straining experiments of unirradiated and ion irradiated steels offer further understanding of the dislocation channel interactions with grain boundaries. For the case of continuous slip, Cui et al. [124] found slip transmission in irradiated metal to be governed by both minimization of the strain energy density in the grain boundary (by minimizing the Burgers vector of the residual grain boundary dislocation) and the resolved shear stress in the adjacent grain. Figure 5.8 shows an example of Cui's slip continuity study. In Figure 5.8a, two outgoing slip systems are seen to emit from the grain boundary where a dislocation pile-up has formed in the neighboring grain. Figure 5.8b shows the resolved shear stress and the residual Burgers vector for all possible outgoing slip systems, and it is seen that the two slip systems where slip occured maximized the shear stress and minimized the residual Burgers vectors. This is similar to the governing characteristics of slip transmission in unirradiated metal [45,46]. However, in the irradiated case, more importance is attributed to the shear stress on the outgoing slip system, as the dislocations must be able to propagate through the obstacles.

Dislocation slip within the grain boundary has been observed if the grain boundary is the conjugate slip system or if the incorporated dislocations dissociate into grain boundary dislocations that have a Burgers vector fully within the grain boundary

plane. That is, the impact of the incorporated dislocations is spread along the length of the grain boundary. The propensity for dislocation slip within the grain boundary has been shown to increase at higher deformation temperature [115]. This observation provides a possible explanation for the DC intersections with grain boundaries that is accompanied with a displacement along the grain boundary.

5.3 The link between dislocation channeling and IASCC

The corrosion resistance of austenitic steel is attributed to the passive oxide film that forms on the surface. The slip-oxidation model describes how cracking may occur in metals with protective films, and has been found to describe the SCC behavior of steel in BWR environments [128]. In general, the slip-oxidation process refers to a cycle where the oxide layer is ruptured, exposing metal to the corrosive environment, resulting in dissolution of the grain boundary. With time, the oxide layer reforms over the newly exposed metal, and the process repeats, with the crack growing larger with each cycle.

The effect of dislocation channel impingement on the grain boundary is examined in the context of oxide film rupture as a requirement for SCC. As the general category of discontinuous channels (i.e. including both discontinuous and discontinuous w/ GB slip) has been shown to induce cracking (Figure 4.6), the question remaining is whether the cracking is due to local strains or local stresses at the grain boundary. Either case could rupture the grain boundary, as shown in Figure 5.9. The role of slip at the grain boundary is to crack the oxide and move metal to the surface to be exposed to the environment. The role of stress is to open the crack tip through the oxide layer to the metal, for access by the bulk environment. It is important in this work to distinguish between the two

186

different forms of discontinuous DC-GB intersections in order to understand the mechanism linking channeling with IASCC.

5.3.1 Cracking propensity of the three DC-GB intersection classifications

Of the cracks that appeared at DC-GB intersections, it is observed that the majority occurred at continuous and discontinuous DC-GB intersections, in similar numbers. However, it is important to not only look at the number of cracks at each DC-GB intersection type, but also the number of those intersection types that exist. Using the bulk DC-GB characterization measurements from Table 4.18, the fraction of each DC-GB type to crack was estimated and the results are shown in Figure 5.10. It is clear that discontinuous DC-GB intersections are the most susceptible to cracking, with continuous and D/GB resulting in similar levels of cracking susceptibility.

It is also useful to examine the cracks characterized in Table 4.18 as "Unknown DC-GB intersection type". The straining in BWR NWC resulted in the creation of additional channels, as shown in Figure 4.8, where the channel density is seen to raise with applied strain. Though the DC-GB intersection type could not be characterized as one of the three types discussed in this paper (and was therefore designated as "Unknown DC-DB intersection type" in Table 4.18), it was possible to determine if was continuous or discontinuous. The discontinuous category would then contain both true discontinuous DCs and discontinuous w/ GB slip. Figure 5.11 shows that for the 25 cracks in the "Unknown DC-GB intersection type" category, cracking propensity at discontinuous boundaries was statistically higher than at continuous boundaries. This is similar to the results seen previously in Figure 4.6. Presumably if the two forms of discontinuous channels could be distinguished, they would follow the frequency reported in Table 4.18.

That is to say, more discontinuous DC-GBs causing cracking than discontinuous w/ GB slip. However, as this cannot be confirmed with the experiments, only those DC-GB intersections whose classification could be determined using DIC prior to cracking will be considered for the study of the mechanism connecting the channels to cracking.

It should also be noted that DCs with small amounts of out-of-plane slip are difficult to observe in the SEM after straining in BWR NWC as the oxide layer can obscure small surface steps. As such, it is difficult to determine how many cases classified as "No visible DC" in Table 4.18 actually are linked to unobserved DCs.

5.3.2 Grain boundary slip vs stress at the grain boundary in IASCC

According to the oxide film rupture model, cracking of the passive oxide layer is a critical step in promoting IGSCC [92,128,129]. Cracking could occur either by slip in grain boundaries below the oxide layer causing it to rupture, or by high local stress that overcomes the cohesive strength of the oxide layer at the grain boundary, opening the crack and exposing the metal to the bulk solution. In particular, the values of the stress and strain at the DC-GB intersection are important as this is the location at which cracking preferentially occurs. The high susceptibility to cracking exhibited by the discontinuous DC-GB intersections compared to D/GB shows that cracking is not caused by strain within the grain boundary. Previous experiments showed that exposure to 288°C BWR NWC for 5 days (about the length of the SCC tests in this study) results in an oxide layer of ~ 100 nm [130]. The amount of grain boundary slip exceeded this expected thickness, as shown in Figure 4.13, where grain boundary slip measurements reached as high as 320 nm, with an average slip displacement of 132 nm. This plot, however, only considers GB slip from a single dislocation channel intersecting a grain

188

boundary. As multiple dislocation channels impinge on the boundary, the amount of GB slip increases, as seen by comparing GB slip from single DC-GB interactions in Figure 4.13 with Figure 4.15 where total GB slip due to multiple channels is reported. Considering only the out-of-plane displacement shown in Figure 4.15, which would be more important in terms of cracking the oxide and exposing metal to the environment, the average out-of-plane displacement in the grain boundary was 104 nm and the largest was 618 nm. This is sufficient to rupture the oxide layer and cause IASCC if slip in the grain boundary were controlling. Furthermore, cracks initiating at a DC-GB intersection are not directly correlated to the amount of slip within the channel. Recall from Figure 4.13 that discontinuous channels had the smallest displacement of the three types by about a factor of 3. IASCC crack initiation does not occur as a direct result of strain, either in the grain boundary or the dislocation channel. Next the role of stress caused by the dislocation channel intersection with the grain boundary was considered.

As mentioned previously, grain boundaries where normal stress is highest were the most susceptible to IASCC, as evidence by the high propensity for cracking at GBs normal to the tensile axis, and GBs adjacent to low Schmid Factor grains. Due to unaccommodated strain, discontinuous channels caused regions of high local stress where they intersect the grain boundary. Strain in continuous channels was accommodated either by the transfer of slip to the neighboring grain or by activating a dislocation source in the neighboring grain. In a similar manner, discontinuous channels w/ GB slip accommodate strain by inducing slip in the grain boundary. In the case of discontinuous channels, the dislocations piled up at the grain boundary, causing an increase of stress locally. A discontinuous channel in a sample strained to 2.5% macroscopic strain may

contain up to ~200% strain within the channel, based on the displacement measurements shown in Figure 4.13, and assuming a channel width of 100 nm. This is equivalent to almost 800 dislocations in a channel. In work by Evrard and Sauzay [83], similar channels (100 nm wide) were modeled using finite element analysis, as this was believed to describe the deformation over multiple parallel slip planes better than simple dislocation pile-up models which assume a single slip plane. They found that the stress in the dislocation channels was higher than the applied stress on the grain boundary by a factor of ~4.

High resolution EBSD was used to make direct measurements of the local elastic strain field, from which the local stress field was determined according to Equations 3.9 and 3.10. As shown in Figure 4.17, the ledges created by out-of-plane dislocation channel and grain boundary slip interfered with the pattern collection by blocking the path from the backscattered beam to the detector. This created significant difficulty in collecting the data in the region around the DC-GB intersection. As a result, no data was collected within a 100 nm radius about the DC-GB intersection. It is also important to note that the values of the stresses as measured by EBSD were higher than realistic. Such high values of stress are believed to be due to distortion of the crystal lattice near the surface (from surface damage caused by irradiation and subsequent straining), resulting in blurred Kikuchi bands. The importance of these numbers is in the relative stress levels of the different types of DC-GB intersections, not the actual stress values shown in the maps of Figure 4.18. The results do show that stress tends to be elevated at the GB, and that there are cases where stress at discontinuous DC-GB intersections reach higher levels

(in one case, nearly double) than those observed at continuous or D/GB (Figure 4.18) intersections.

The likely role of the stress at the grain boundary in the IASCC mechanism was to rupture the oxide layer to expose the metal below, resulting in dissolution of the grain boundary until the passive oxide layer reforms, and the process repeats. The dislocation pile-up within the discontinuous channels cause regions of large stress in the underlying metal layer. Stress is transferred to the oxide from the metal substrate via the interface between the two as there is no evidence of delamination of the oxide layer [131]. The interface between the underlying metal and the oxide is not perfect due to mismatch between the lattices, as shown in Figure 5.12. Evans [132] states that if it were possible, the oxide would slip along the interface when the metal substrate deforms. However, interfacial slip is difficult and instead, the high stress in the oxide induced by the highly localized stress in the underlying metal in contact with the oxide causes the oxide to fracture.

The stress does not reach a level high enough to directly rupture the grain boundary in the underlying metal, otherwise cracking would not require an aggressive environment, and would occur even when the specimens were strained in high temperature argon (an inert environment). This mechanism results in cracking occurring where grain boundaries are under high levels of stress, in particular at discontinuous DC-GB intersections. These results explain the role of dislocation channeling in IASCC, which previously has not been well understood. It is important to note, however, there are other factors that play a role in the mechanism which are not explained in this work. As shown in Table 4.5, the various alloys tested resulted in a large range of cracking

susceptibilities, though all were found to undergo localized deformation. Future work is needed to fully understand the role of alloy composition, as well as other properties in IASCC.

5.3.3 Corroboration of theory by atomistic modeling and in-situ TEM straining

The observations of high stress due to discontinuous DC-GB intersections are supported by atomistic modeling that has examined the interactions of dislocations with GBs [122], as well as in-situ TEM straining experiments [124]. As stated previously, and shown in Figure 5.6, dislocations were observed [122] to impinge on a boundary and the stress was observed to increase locally to values as high as 9 GPa. This high level of stress then decreased when stress was relieved by the dislocations moving into the boundary and eventually transmitting across the boundary into the adjacent grain. Areas where dislocations were transmitted across the boundary generally exhibited lower stress levels than areas where transmission did not occur. Furthermore, if the dislocations were not transmitted into the neighboring grains, the high stresses typically resulted in crack initiation.

This was particularly true near triple junctions, which acted as stress concentrators and where a number of cracks were found to initiate. Kamaya et al. [40], using finite element modelling, have also observed the high levels of stress at triple junction, due to the deformation constraints caused by the adjacent grains. As the deformation constraints in an irradiated sample are significantly higher (due to the high density of defects in the microstructure) than those of an unirradiated sample, it is expected that the relative level of stress at the triple junctions will be even higher in the irradiated steel. It is believed that the cracks that occurred at triple junctions in the experiments (Table 4.17) were a

result of this elevated stress, similar to cracks that form at discontinuous DC-GB intersections. The simulations clearly showed that un-accommodated and un-transmitted slip arriving at the boundary created very high local stresses. Even though the simulations were carried out at the extremely high strain rates necessitated by the molecular dynamics technique, the results of high stress buildup and subsequent crack initiation provide clear support for the idea that it is these areas of high stresses at discontinuous DC-GB intersections that result in crack initiation.

In the in-situ TEM straining experiments [124], it was observed that when the pile-up could not be accommodated by slip transfer, stress was eventually relieved by crack nucleation. Observations were also made that are useful to understanding the cracking of triple junctions. Cui et al. [37] reported that dislocations within a channel once incorporated in the grain boundary caused dislocation emission from the vicinity of a triple junction to relieve stress. If dislocation propagation away from the sources in the vicinity of the triple junction was prohibited by the presence of the irradiation defects, the response may be for the triple junction to crack.

5.4 Understanding past results within the scope of the current work

Discontinuous slip is most important with regards to cracking, however it also results in channels containing the least amount of strain. Past results showed that increasing channel height (out-of-plane displacement) correlated with increasing cracking susceptibility (Figure 2.25). Channel height and total displacement are related, as the height measurements may be used to approximate total displacement, since the total displacement follows the same general trend as the channel height measurements, though larger in value, as shown in Figure 5.13. The correlation between channel height and

cracking susceptibility seems contradictory to the results of this thesis showing that discontinuous channels undergo cracking at the greatest frequency but they also have the smallest total channel displacement. However, as shown in Figure 4.9 that compares in-plane displacement after 1.5% strain and then with an additional 1.0% strain, all types of channels increased in size, including discontinuous channels. The average slip in the channels was higher after 2.5% strain, meaning that the slip within the discontinuous channels was also higher. Larger discontinuous channels would result in larger pile-ups and thus higher stress concentrations.

Table 5.1 Determination of the slip plane by measuring the angle of the surface trace of the DC with respect to the tensile direction, and comparing it to the angle made by the closest slip plane as determined by EBSD.

Angle measured in SEM (degrees)	Angle calculated based on EBSD (degrees)	Difference (degrees)
60	57	3
30	34	4
60	58	2
30	33	3
66	64	2
83	81	2
58	56	2
83	81	2
66	68	2
66	73	7
62	65	3
60	58	2
46	43	3
37	41	4
34	35	1
34	35	1
49	51	2
77	80	3
34	35	1
34	34	0
34	34	0
66	69	3
84	82	2
37	41	4
55	60	5
29	22	7
80	80	0
65	59	6
51	49	2
	Average	2.7

Table 5.2 Combination of DIC/confocal microscopy displacement measurements with EBSD to determine slip direction. Measured displacement vector is compared to best fit EBSD determined slip direction, and the percent of slip that can be accounted for assuming that slip direction is shown in the final column.

Normalized measured displacement vector			Closest slip direction (determined by EBSD)			% of slip accounted for with one slip system
X	Y	Z	X	Y	Z	
0.303	0.369	1.331	0.900	0.753	0.790	40.1
-0.558	1.052	-0.763	-0.903	1.065	-0.228	56.1
0.624	0.624	1.105	0.880	0.724	0.837	73.1
0.627	0.654	1.086	0.880	0.724	0.837	74.6
0.832	0.916	-0.685	0.926	0.533	-0.927	67.8
-0.406	-1.121	0.761	-0.498	-1.155	0.645	89.3
-0.506	-0.831	1.027	-0.498	-1.155	0.645	65.1
0.685	1.127	-0.512	0.843	1.121	-0.183	74.4
0.966	0.289	-0.992	0.906	0.597	-0.906	77.1
1.288	0.511	-0.280	1.023	0.888	-0.405	66.7
0.441	-0.128	1.338	0.304	-0.251	1.358	86.9
0.945	0.762	0.725	0.875	0.748	0.821	91.6
-1.001	-0.859	0.510	-1.041	-0.873	0.393	91.2
1.173	0.666	-0.425	1.176	0.748	-0.238	85.6
0.542	0.266	1.278	0.125	0.098	1.405	67.4
-1.010	0.750	-0.645	-1.067	0.650	-0.662	91.8
1.121	-0.583	0.635	0.954	-0.660	0.809	82.2
-1.049	-0.401	0.859	-0.965	-0.140	1.024	77.5
-0.955	0.516	-0.907	-1.067	0.650	-0.662	78.8
1.174	0.165	-0.770	1.147	0.476	-0.677	77.2
1.033	0.473	-0.842	1.147	0.476	-0.677	85.9
-0.802	0.727	-0.910	-0.961	0.772	-0.694	80.9
1.027	0.051	0.971	1.190	0.195	0.739	77.7
-0.320	0.895	-1.047	-0.539	0.953	-0.895	80.8
0.506	0.280	-1.290	0.143	0.165	-1.397	72.3
1.190	0.291	0.705	0.844	0.883	0.712	52.9
-1.270	0.090	0.615	-1.263	0.489	0.407	68.6
-0.739	0.290	1.170	-0.956	0.033	1.042	74.8
-1.013	0.361	-0.918	-0.870	0.493	-1.000	85.1
					Average	75.7%
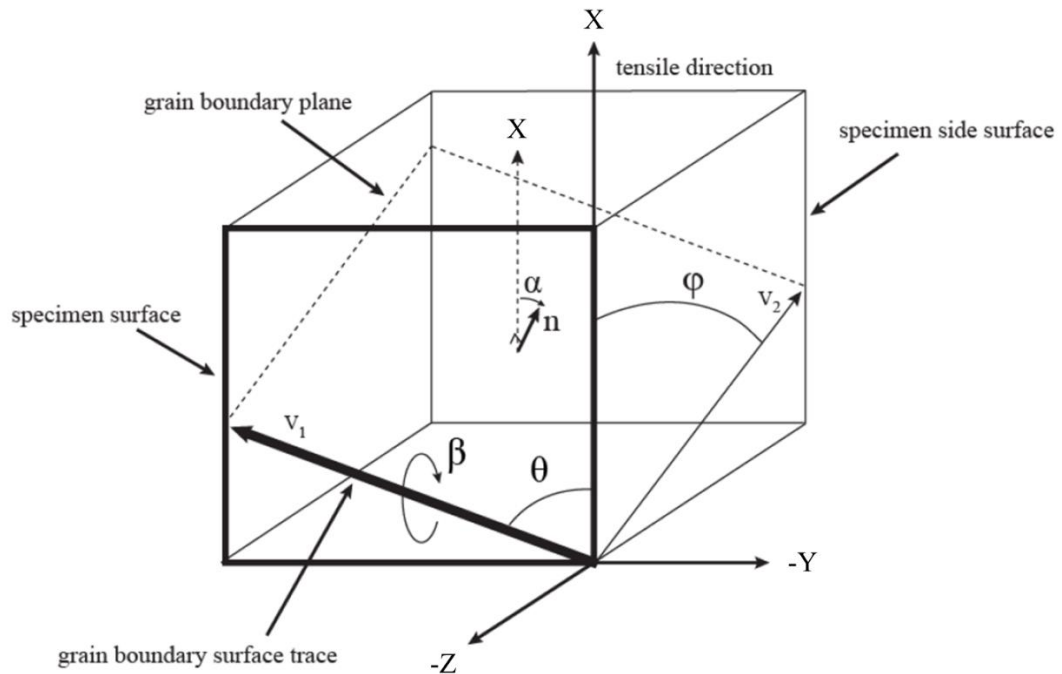
Figure 5.1 Graphic depicting grain boundary geometry and the respective angles and vectors associated with it. n is the vector normal to the grain boundary plane, α is the angle between n and the tensile axis (X), v_1 is the trace of the grain boundary plane on the specimen surface, whereas v_2 is the trace of the grain boundary plane on the specimen side surface, perpendicular to the gage surface. θ and φ are the angles between the tensile axis and the vectors v_1 and v_2, respectively, and β is the angle of rotation about the v_1 vector. Figure modified from West's thesis [119].
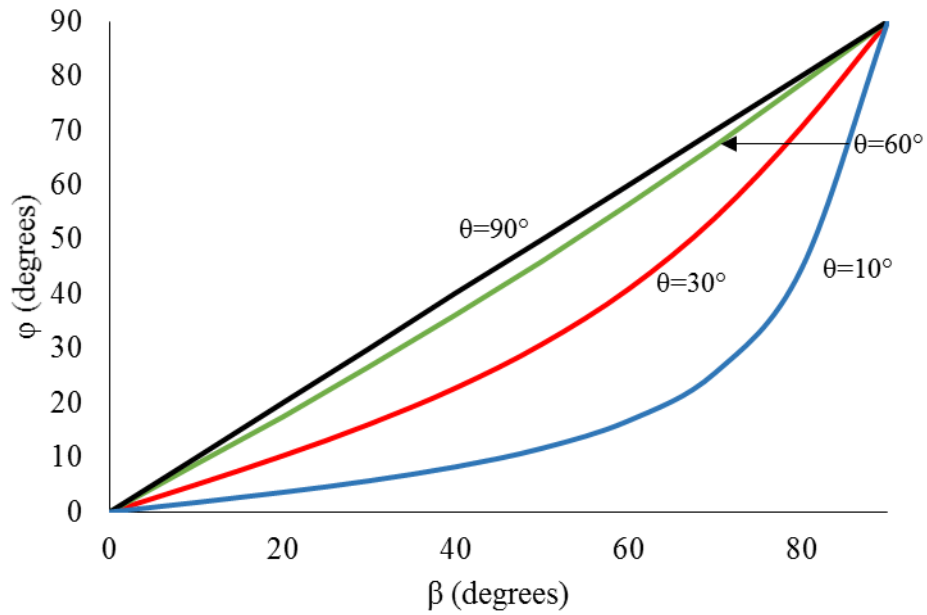
Figure 5.2 Calculated values of the angle between the tensile axis and the trace of the GB on the side surface of the specimen perpendicular to the specimen surface (φ), using equation (5.8) for varying angles of rotation about the GB surface trace vector (β) and angles of the grain boundary surface trace with respect to the tensile axis (θ). Each color represents a different set value of θ, as listed on the graph.
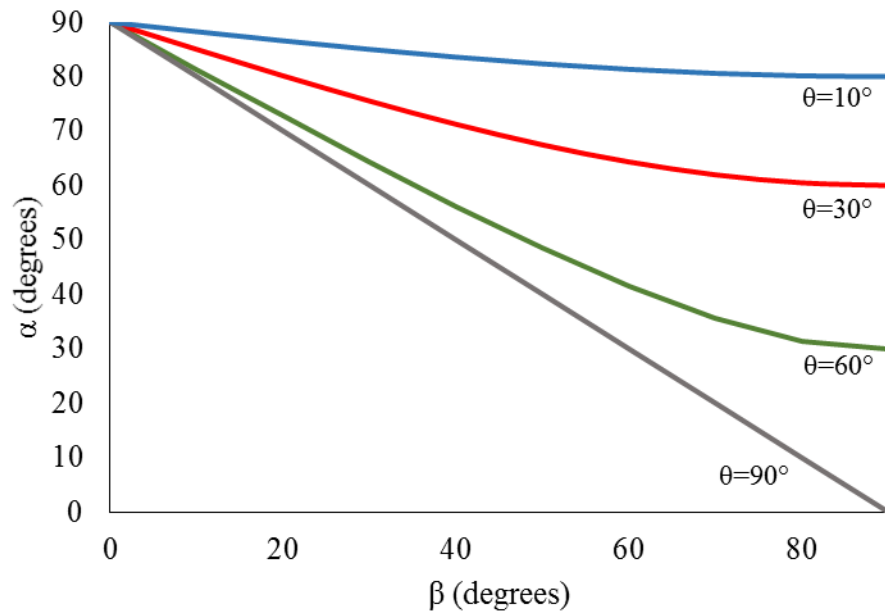
Figure 5.3 Calculated values of the angle between the normal to the grain boundary plane and the tensile axis (α) for varying angles of rotation about the GB surface trace vector (β) and angles of the grain boundary surface trace with respect to the tensile axis (θ). Each color represents a different set value of θ, as listed on the graph.
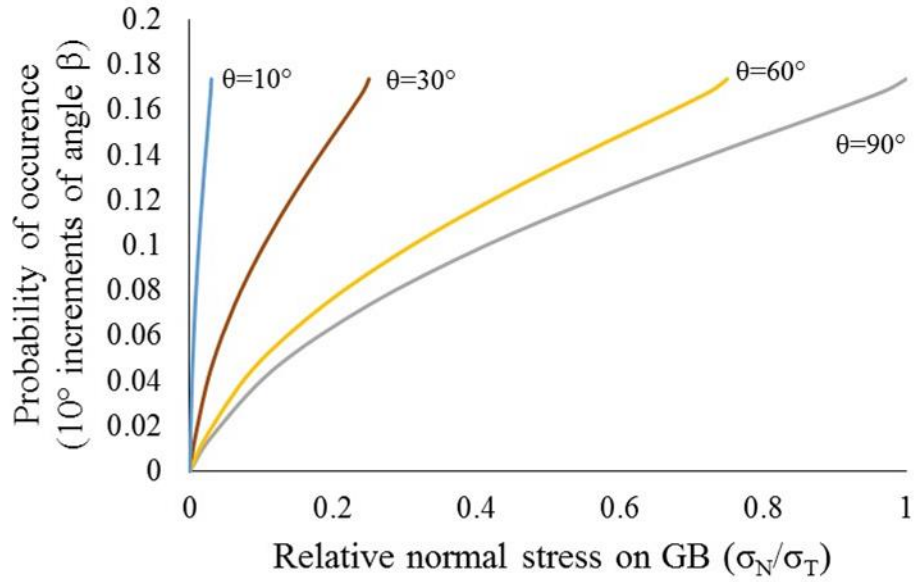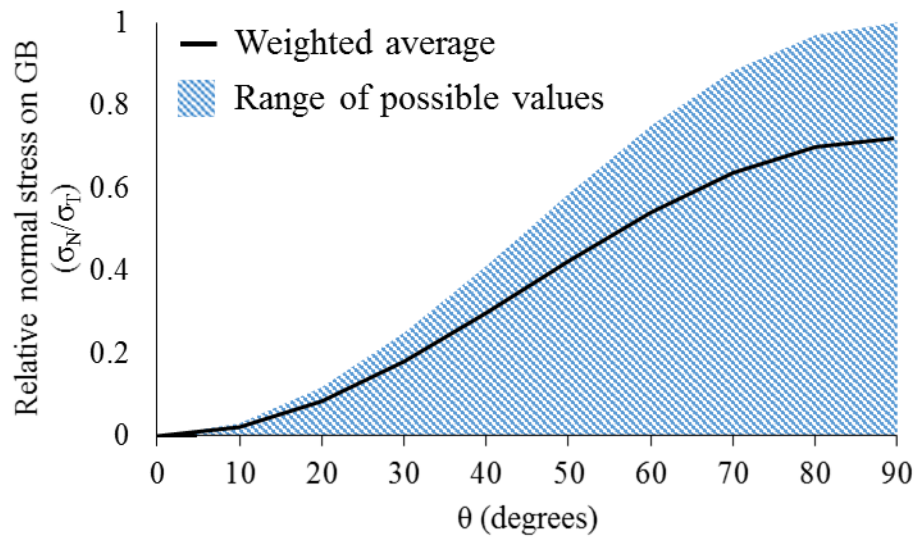
Figure 5.4 (a) Normal stress on the grain boundary [as determined by equation (5.1)], with varying angles of rotation about the GB surface trace vector (β) and angles of the grain boundary surface trace with respect to the tensile axis (θ). Each color represents a different set value of θ, as listed on the graph. (b) Weighted average stress over all possible angles of α, for each angle of θ. Stress values were weighted according to probability of occurrence.
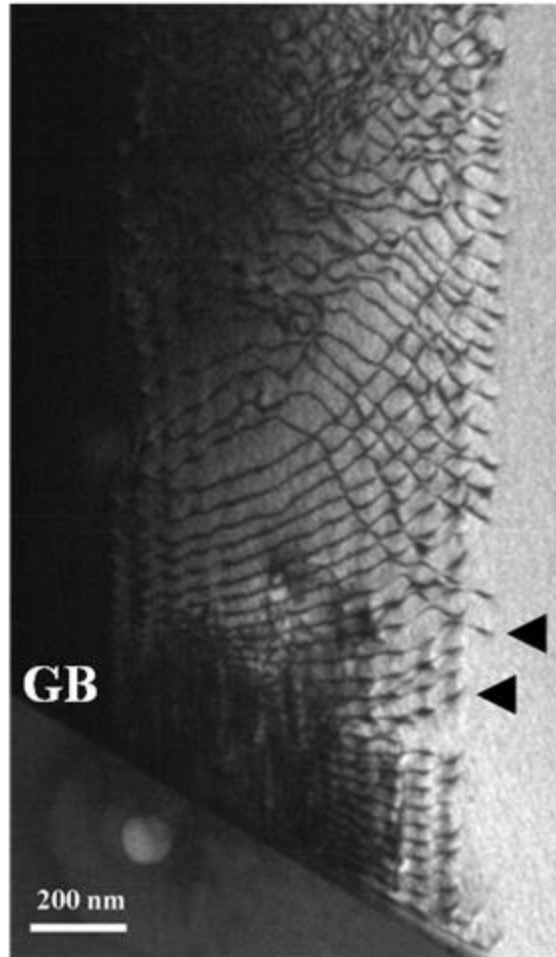
Figure 5.5 TEM image of a dislocation channel in a K$^+$ irradiated 304 stainless steel. The arrowheads indicate the dislocations in different slip systems (parallel slip planes but different slip directions) [122].
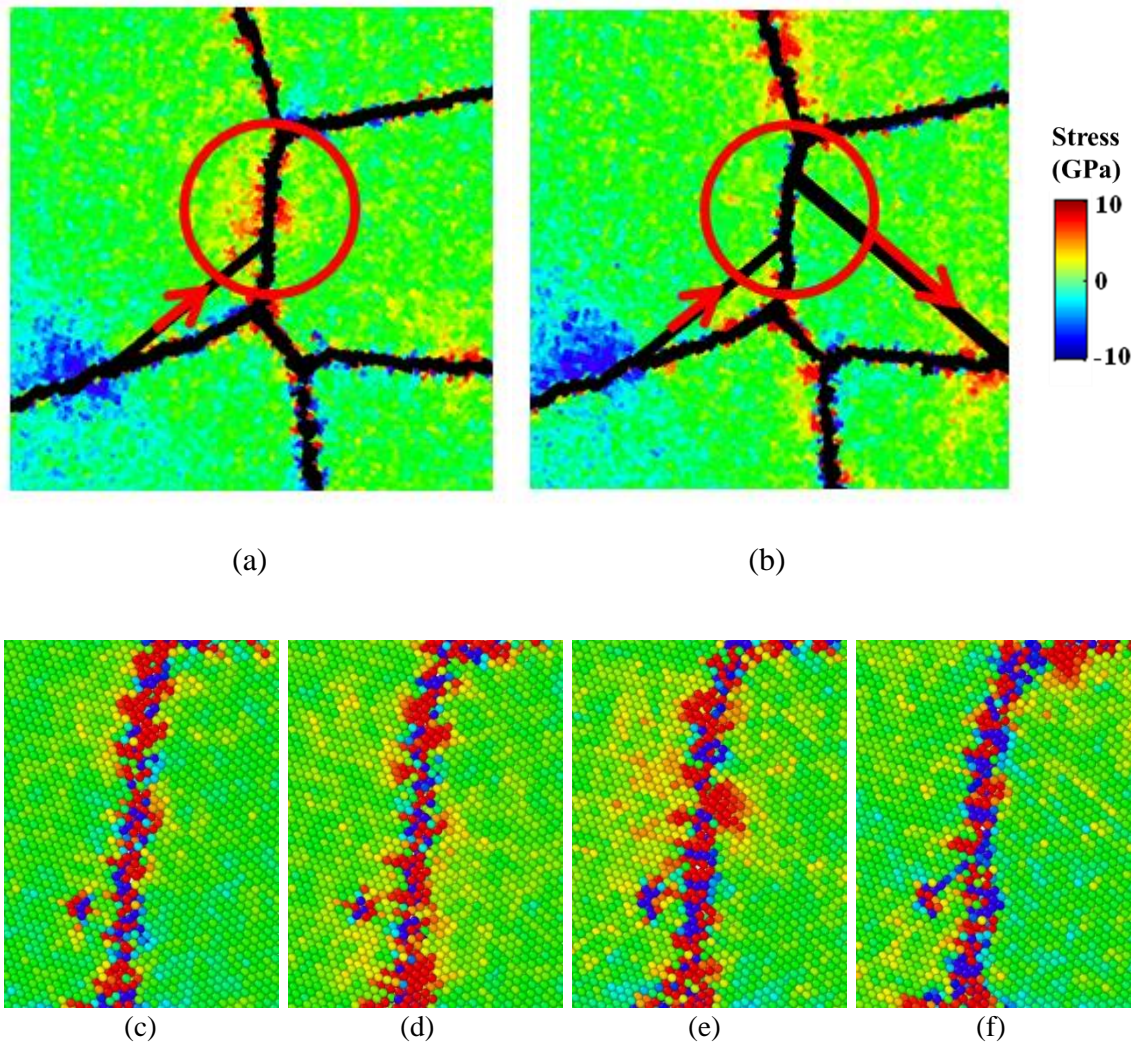
Figure 5.6 Simulated sample at (a) 4% and (b) 4.5% applied tensile strain. The black lines depict stacking faults created by the passage of partial dislocations. (c-f) show more detail of the circled area, at 3%, 3.5%, 4% and 4.5% applied tensile strain, respectively. At (e), the dislocation has intersected the grain boundary and a large buildup of stress is observed, which has been relieved in (f) as the dislocation moves through the boundary and passes into the adjacent grain. [122]
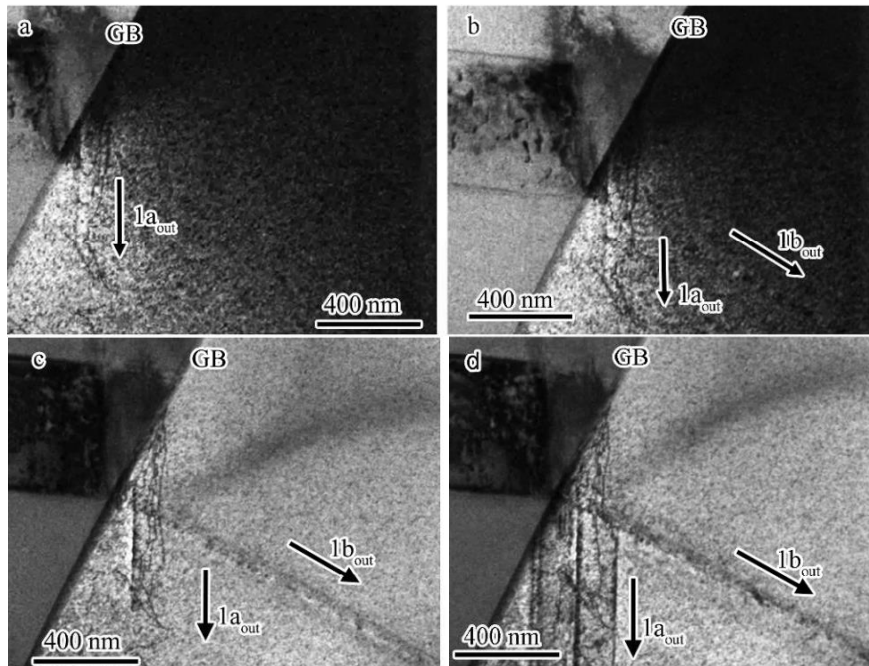
Figure 5.7 Dislocations interacting with a grain boundary in a 1 dpa Kr^+ irradiated 304 stainless steel. Time resolved snapshots from an in-situ TEM straining video show slip transmission occurring after a dislocation pile-up forms in the grain on the left-hand side. (a) Dislocations first transmitted into system $1a_{out}$. (b) Later, system $1b_{out}$ also activated and (c) both $1a_{out}$ and $1b_{out}$ continued to be active. (d) Further into the straining, an "avalanche" of dislocations were emitted from the boundary into parallel slip planes with systems $1a_{out}$ and $1b_{out}$. [127]
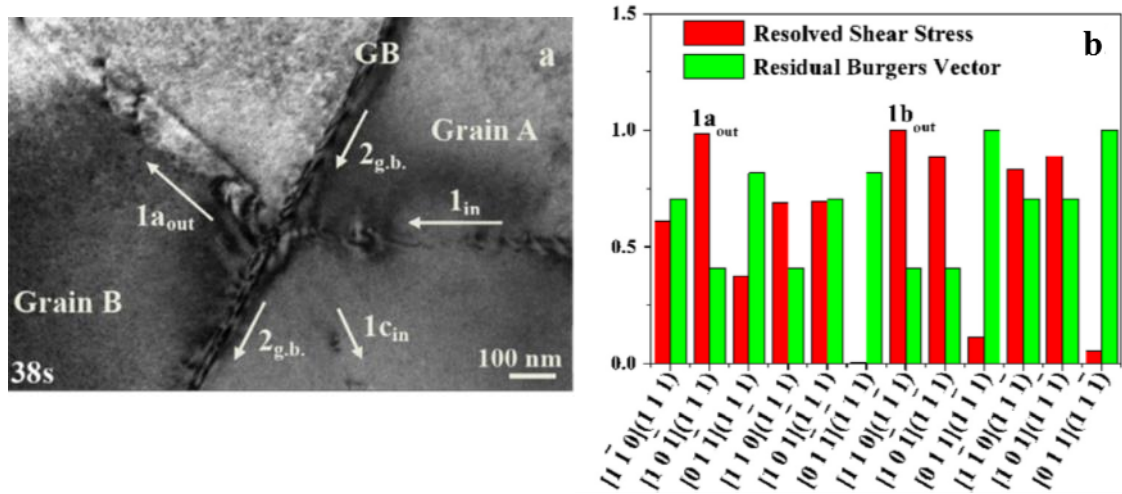
Figure 5.8 In-situ TEM straining of a Fe-13Cr15Ni specimen, irradiated with Kr+ ions. (a) Dislocations from Grain A on the right are seen to intersect the sigma 3 grain boundary, and two outgoing slip systems are activated in Grain B. (d) The resolved shear stresses are maximized in these two activated slip systems and residual Burgers Vectors left in the grain boundary are minimized. [127]
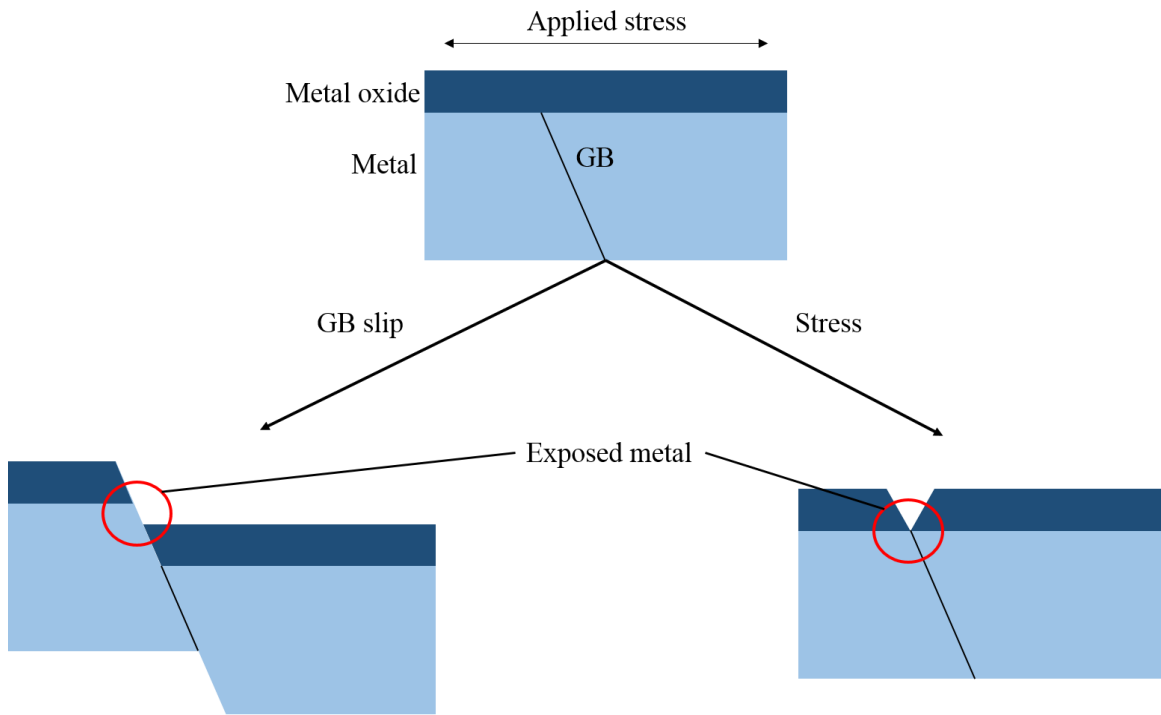
Figure 5.9 Slip-oxidation model for SCC. The oxide layer is ruptured, either through GB slip or stress at the GB. Following the rupture of the protective oxide layer, metal dissolution occurs prior to the reformation of an oxide layer over the cracked region. The process then repeats with the newly formed oxide layer rupturing, and further dissolution occurring.
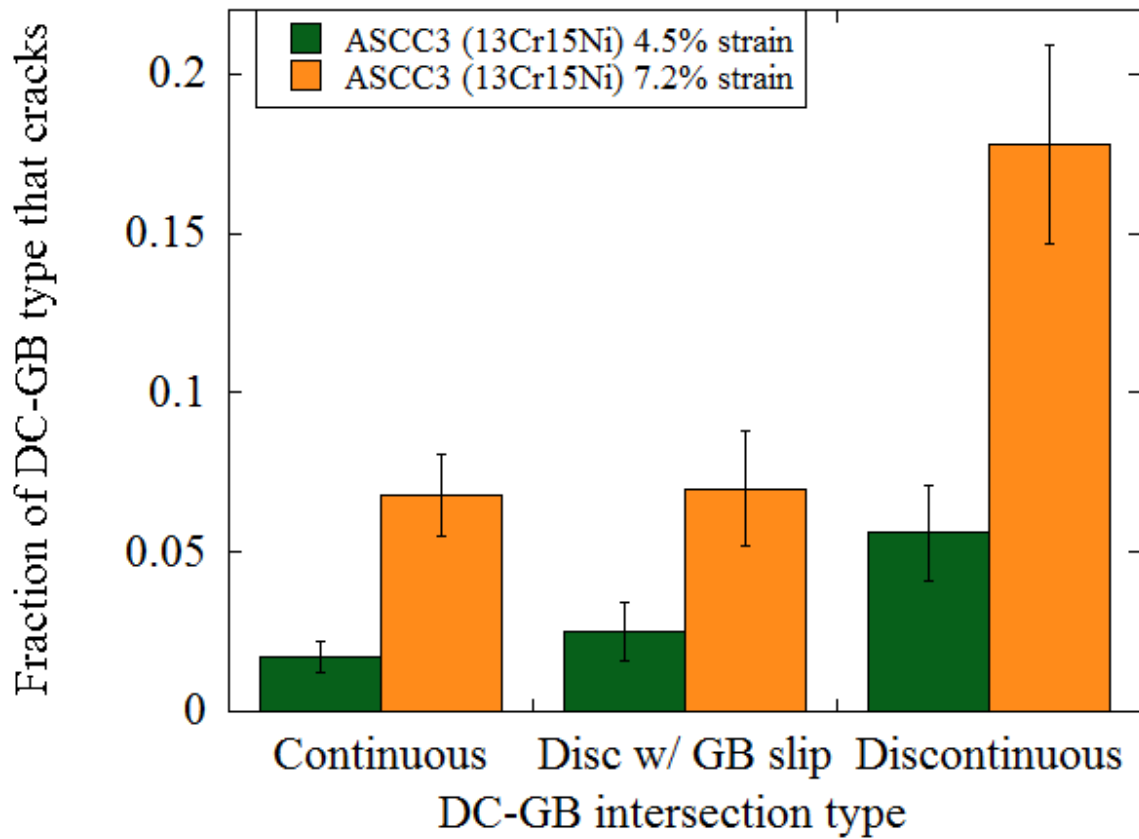
Figure 5.10 Fraction of DC-GB that induced cracking, based on relative amounts of the three DC-GB classifications.
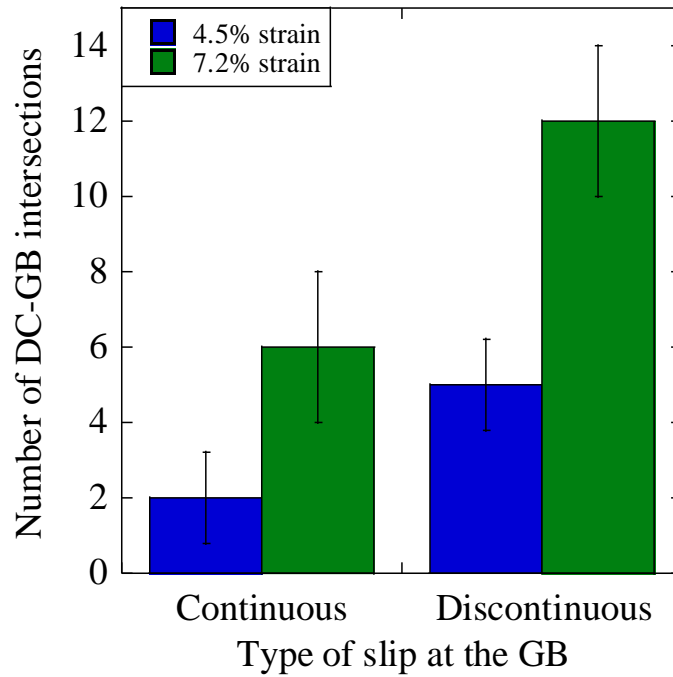
Figure 5.11 Fraction of DC-GB intersection from Table 4.18 in the classification "Unknown DC-GB intersection type" that exhibited IG cracking. Classifications are based on SEM observations, so D/BG and discontinuous cannot be distinguished and are combined into the "Discontinuous" category.
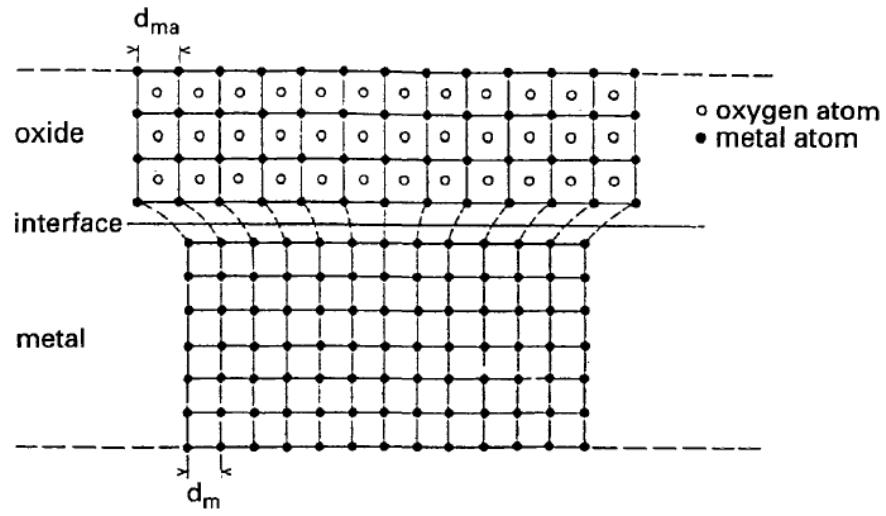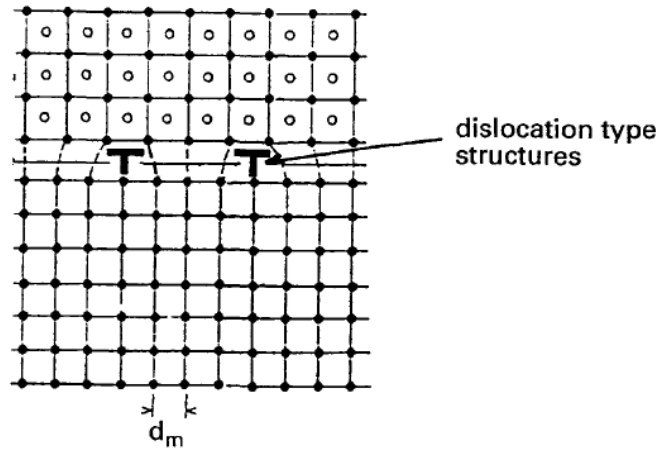
Figure 5.12 Possible models of oxide/metal interface [132].
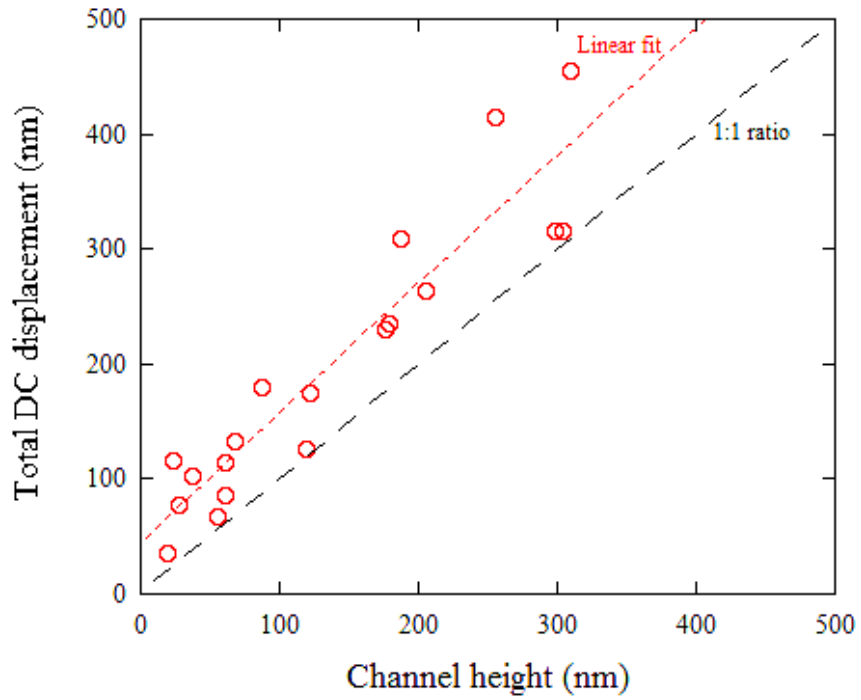
Figure 5.13   Total displacement in a dislocation channel compared to out-of-plane channel height measurements for the Fe-13Cr15Ni ASCC2 specimen strained to 3.5%. Dashed line shows 45° line where channel height is equal to total displacement.

Chapter 6

CONCLUSIONS AND FUTURE WORK

6.1 Conclusions

The major conclusions of this thesis are:

1. Grain boundaries with specific characteristics, namely those that are normal to the tensile axis, are high angle boundaries, are adjacent to low SF or high TF grains, and at which slip in an intersecting dislocation channel is discontinuous, have a higher propensity for IASCC in a BWR NWC environment.

2. Discontinuous channels may be subdivided into two classifications: Discontinuous with grain boundary slip, and discontinuous that terminates at the grain boundary with no additional slip. This, when combined with continuous DC-GB intersections, results in three possible interactions between channels and boundaries.

3. Channels where slip is accommodated at the GB (i.e. continuous and discontinuous w/ GB slip) undergo significantly more strain than those where slip is not accommodated in any form. Displacement measurements as high as 660 nm and 541 nm were recorded for continuous and discontinuous w/ GB slip, respectively. 174 nm was the largest displacement channel measurements for cases of discontinuous DC-GB intersections. These measurements were

determined using the combination of confocal microscopy with DIC and provide much more accurate strain measurements within the channels than previous work was able to provide using only channel height measurements.

4. As applied strain increases, the degree of displacement within the dislocation channels increases, regardless of the DC-GB intersection classification.

5. Discontinuous channels tend to cause areas of higher stress at the DC-GB intersection than the other two DC-GB intersection types. This is likely due to the dislocation pile-up pushing on the grain boundary, whereas for both continuous and discontinuous w/ GB slip, the dislocation pileup is at least partially resolved by slip transmission across the boundary or absorption into the boundary.

6. Of the three DC-GB intersection types, discontinuous DC-GB intersections are much more likely to induce cracking. This strongly suggests that the link between dislocation channeling and IASCC is the high level of stress that channeling causes at the grain boundary, showing that slip at the grain boundary in not needed to rupture the oxide layer, as had previously been assumed.

7. A number of cracks occurred mid-grain boundary but did not form at characterized DC-GB intersections. This is likely due to the formation of new channels during the subsequent straining in the simulated BWR-NWC environment (post-slip characterization). Of the cracks that formed where the new channels were visible in the SEM, the majority of the channels appeared discontinuous at the grain boundary.

8. A large number of cracks also was found to occur at triple junctions. Similar to discontinuous DC-GB intersections, this is believed to be due to the high levels of

stress that occurs at triple junctions, due to difficulties of accommodating strain across three grains.

9. While highly localized stress explains the link between dislocation channeling and IASCC, it does not fully describe the IASCC mechanism. This is clear from the large variation in cracking susceptibility in the tested austenitic stainless steels of varying compositions. In particular, the Fe-21Cr32Ni underwent localized deformation, but did not experience IASCC.

10. The findings discussed in this thesis are consistent with previous research that found IASCC to be related to dislocation channel height (or severity of deformation localization). The general increase in channel size will also translate into larger channels that result in discontinuous DC-GB intersections, which would create larger pile-ups and induce higher levels of stress at the GB.

6.2 Future work

Now that stress has been linked to IASCC, it will be important to understand the level of stress required to cause crack initiation. This was not possible in this work due to the inability to determine the level of strain within the channels at the point of cracking (as strain measurements were performed after straining in argon, and not during the BWR environment straining), and due to the qualitative results obtained from the EBSD stress analysis, where the quantitative results were exaggerated and unusable.

The plastic strain analysis portion of this work could be improved if it did not rely on taking deformation measurements during straining in an argon environment. If these measurements could be taken after straining in a BWR environment, the DC-GB intersections could be better characterized, including the size of the DCs after cracking

occurred. Additionally, if these measurements could be taken in-situ, then the displacement measurements at the exact time of crack initiation could be taken.

The elastic strain/stress measurements were only taken through their preliminary stages in this work. Further work will be needed to be determine, on a quantitative level, the amount of stress at DC-GB intersections. Very little quantitative, experimentally measured data exists for strained irradiated steel. By further developing the EBSD stress measurements, or some other stress measurement technique, it would be possible to determine the exact level of stress on the grain boundaries and compare this to the grain boundary character, as well as the displacement vector of the dislocation channel. Combined with accurate strain measurements, the level of stress at the grain boundary when cracking occurred, as well as the character of the dislocation channel inducing cracking, could be determined.

Finally, much more work needs to be performed in order to understand all of the variables in IASCC. This work focused on the role of localized deformation and IASCC, however, a wide range of cracking susceptibility was noted among the different alloys studied, suggesting the alloy composition also plays a role in the IASCC mechanism, which is not currently well understood.

APPENDICES

APPENDIX A

DIC Matlab® code

Portions of the DIC code were based off of code by Christoph Eberl and Robert Thompson, made available online at http://www.mathworks.com/matlabcentral/fileexchange/12413-digital-image-correlation-and-tracking. This code was heavily modified by the author of this thesis for use in this dissertation. This code uses the image processing toolbox available with Matlab.

The following functions are included in this appendix.

Grid_DIC
Filelist_generator
Grid_generator
Automate_image_DIC
CombineResults
Disp_GUI
SEM_correction

The base program calls several functions which will all be included in this appendix.

Grid_DIC:

Code start

```
function [displx disply]=grid_DIC;
msgboxwicon=msgbox('Select the prestrained image.')
waitfor(msgboxwicon)
filelist_generator;
msgboxwicon=msgbox('Define grid.  Either image may be selected for grid
definition')
waitfor(msgboxwicon)
[grid_x,grid_y]=grid_generator;
automate_image_DIC(grid_x,grid_y);
```


Code end

filelist_generator: This was program was used as is from Chris and Rob, as its primary purpose is just to create a list of the images used in the DIC analysis.

Code start

```
function [FileNameBase,PathNameBase,filenamelist]=filelist_generator

% Code to construct a list of 9999 or less filenames
% Programmed by Rob, changed by Chris. Automatic filelist generation
% and image time aquisition added by Chris.
% Last revision: 12/25/06

%filenamelistmode = menu(sprintf('How do you want to create the
filenamelist?'),...
%    'Manually','Automatically','Cancel');

filenamelistmode=2;  %Removed other option.  May be put back in by
removing this line and reinstating above comment.
if filenamelistmode==3
    return
end
if filenamelistmode==2
    [FileNameBase,PathNameBase,filenamelist]=automatically;
end
if filenamelistmode==1
    [FileNameBase,PathNameBase,filenamelist]=manually;
end

%[FileNameBase,PathNameBase,filenamelist]=imagetime(FileNameBase,PathNa
meBase,filenamelist);


% -----------------------------------------------------


function [Firstimagename,ImageFolder,filenamelist]=automatically

[Firstimagename ImageFolder]=uigetfile('*.tif','Open First Image');
if Firstimagename~~[];
    cd(ImageFolder);
end

if Firstimagename~~[];
    % Get the number of image name
    letters=isletter(Firstimagename);
    Pointposition=findstr(Firstimagename,'.');
    Firstimagenamesize=size(Firstimagename);
    counter=Pointposition-1;
    counterpos=1;
    letterstest=0;
    while letterstest==0
        letterstest=letters(counter);
        if letterstest==1
            break
        end
        Numberpos(counterpos)=counter;
```

```matlab
        counter=counter-1;
        counterpos=counterpos+1;
        if counter==0
            break
        end
    end

    Filename_first = Firstimagename(1:min(Numberpos)-1);
    Firstfilenumber=Firstimagename(min(Numberpos):max(Numberpos));
    Lastname_first =
Firstimagename(max(Numberpos)+1:Firstimagenamesize(1,2));
    Firstfilenumbersize=size(Firstfilenumber);
    onemore=10^(Firstfilenumbersize(1,2));
    filenamelist(1,:)=Firstimagename;

    Firstfilenumber=str2num(Firstfilenumber);
    u=1+onemore+Firstfilenumber;
    ustr=num2str(u);
    filenamelist(2,:)=[Filename_first
ustr(2:Firstfilenumbersize(1,2)+1) Lastname_first];
    numberofimages=2;

    counter=1;

    while exist(filenamelist((counter+1),:),'file') ==2;
        counter=counter+1;
        u=1+u;
        ustr=num2str(u);
        filenamelist(counter+1,:)=[Filename_first
ustr(2:Firstfilenumbersize(1,2)+1) Lastname_first];
        if exist(filenamelist((counter+1),:),'file') ==0;
            warning('Last image detected')
            filenamelist(counter+1,:)=[];
            break
        end
    end
end
%[FileNameBase,PathNameBase] = uiputfile('filenamelist.mat','Save as
"filenamelist" in image directory (recommended)');

FileNameBase='filenamelist.mat';
%cd(PathNameBase)    Don't need to change directories, saving in same
place
%as images.
save(FileNameBase,'filenamelist');


% -----------------------------------------------------
function [FileNameBase,PathNameBase,filenamelist]=manually;
% Prompt user for images to be used for analysis

prompt = {'Enter number of first image (i.e. "3" for PIC00003):','Enter
number of last image (i.e. "100" for PIC00100):'};
dlg_title = 'Input images to be used for the analysis';
num_lines= 1;
def     = {'1','100'};
```

```matlab
answer = inputdlg(prompt,dlg_title,num_lines,def);
F2 = str2num(cell2mat(answer(1,1)));
F = str2num(cell2mat(answer(2,1)));

if F >= 10000
    error0 = menu('!!!! ERROR - Code will only work properly for 9999 or
less picture files !!!','Restart');
    return
end

% Choose first name of images
G = 'PIC1';
prompt = {'Enter Image Name (first 4 letters):'};
dlg_title = 'Input images to be used for the analysis';
num_lines= 1;
def     = {'PIC1'};
answer = inputdlg(prompt,dlg_title,num_lines,def);
G = cell2mat(answer(1,1));


E='.tif';

namelist(1:F-F2+1,1)=G(1,1);
namelist(1:F-F2+1,2)=G(1,2);
namelist(1:F-F2+1,3)=G(1,3);
namelist(1:F-F2+1,4)=G(1,4);


% create the numberlist
num=((10000+F2):(10000+F))';

% Creation of final results
filenamelist=namelist;
str=num2str(num);
filenamelist(:,5:8)=str(:,2:5);

filenamelist(1:F-F2+1,9)=E(1,1);
filenamelist(1:F-F2+1,10)=E(1,2);
filenamelist(1:F-F2+1,11)=E(1,3);
filenamelist(1:F-F2+1,12)=E(1,4);



% Save results
[FileNameBase,PathNameBase] = uiputfile('filenamelist.mat','Save as
"filenamelist" in image directory (recommended)');
cd(PathNameBase)
save(FileNameBase,'filenamelist');



% ----------------------------------------
% Extract the time from images?

function
[FileNameBase,PathNameBase,filenamelist]=imagetime(FileNameBase,PathNam
eBase,filenamelist)
```

```matlab
selection_time_image = menu(sprintf('Do you also want to extract the
time from images to match stress and strain?'),'Yes','No');

if selection_time_image==1

    % Loop through all images in imagetimelist to get all image capture
times

    [ri,ci]=size(filenamelist);

    o=waitbar(0,'Extracting the image capture times...');

    for q=1:ri

        waitbar(q/ri);
        info=imfinfo(filenamelist(q,:));
        time=datevec(info.FileModDate,13);
        seconds(q)=time(1,4)*3600+time(1,5)*60+time(1,6);

    end

    close(o)

    % Configure and then save image number vs. image capture time text
file

    im_num_im_cap_time=[(1:ri)' seconds'];
    save time_image.txt im_num_im_cap_time -ascii -tabs

end
```

Code end

Grid_generator: This sub-function was originally coded by Daniel Gianola, then redone by Christoph Eberl. The author of this dissertation then modified the code further to work with his SEM images correctly. The purpose of this function is to create a grid of coordinates, which will be used for the DIC analysis. The displacement of each coordinate will be tracked from the base image to the post-strain image.

Code start

```
function [grid_x,grid_y]=grid_generator(FileNameBase,PathNameBase,
grid_x, grid_y)

% Code to generate the DIC analysis grid
% Changed by Sven
% Completely rewritten by Chris
% Programmed first by Dan and Rob
% Modified by Mike for use at UoM
%
% Last revision: Feb 2013


% The grid_generator function will help you create grids of markers.
% First you'll be asked for the base image that is used to define the
grid
% which is typically your first image. Then you'll be asked if you want
to
% modify an old grid or create a new one. The dialog has different
options
% allowing you to create a marker grid which is rectangular, circular,
a
% line or contains only of two markers or delet markers from created
grid.
% Every combination of them is also
% possible. You will be asked to click at the sites of interest and the
% markers will be plotted on top of your image. You can choose if you
want
% to keep these markers or if you want to try again. If you keep them
they
% will be saved and you'll come back to the main menu.
% It has to be noted that you can always generate your own marker
% positions. Therefore the marker position in pixel has to be saved as
a
% text based format where the x-position is saved as grid_x.dat and the
% y-position saved as grid_y.dat.
%

% Check if a grid is loaded if not new variables will be created
if exist('grid_x','var')~=1
    grid_x=[];
end

if exist('grid_y','var')~=1
    grid_y=[];
end
```

```matlab
% Prompt user for base image if no image already assigned
if exist('FileNameBase','var')~=1
    [FileNameBase,PathNameBase,FilterIndex] = uigetfile( ...
    {'*.bmp;*.tif;*.jpg;*.TIF;*.BMP;*.JPG','Image files
(*.bmp,*.tif,*.jpg)';'*.*',  'All Files (*.*)'}, ...
    'Open base image for grid creation');
else
    FilterIndex=1;
end
FilterIndex
FileNameBase
PathNameBase
% Check if an image is chosen, if true go to directory an load image,
call
% gridtypeselection, if false end program
if FilterIndex~=0
    cd(PathNameBase)
    im_grid = imread(FileNameBase);
    [grid_x,grid_y,FileNameBase,PathNameBase] =
gridtypeselection(FileNameBase, PathNameBase, im_grid, grid_x,
grid_y,1); %call gridselection
else %if not the empty variables will be cleared
    clear  FileNameBase PathNameBase FilterIndex
    disp('No Image is chosen')
end

close all
end
%% Main Menu
function [grid_x,grid_y,FileNameBase,PathNameBase] =
gridtypeselection(FileNameBase, PathNameBase, im_grid, grid_x,
grid_y,loadgrid)

% Code to select which kind of grid should be added and to display the
actual grid
% Decide which type of grid you want to create
% In this area you should select the type of grid you want to add or
create
%
hold off
imshow(im_grid,'InitialMagnification',100); %show chosen Image
%-----------------------
% Load old grid?
%
if loadgrid
loadoldgrid=menu(sprintf('Load old grid?'),...
'Yes','No');

if loadoldgrid==1
    [gridxname,Pathgridx] = uigetfile('*.dat','Open grid_x.dat'); %load
grid_x
        if gridxname==0
            disp('You did not select a file!')
        end
    cd(Pathgridx);
    grid_x=importdata(gridxname,'\t');
```

222

```matlab
    [gridyname,Pathgridy] = uigetfile('*.dat','Open grid_y.dat');%load
grid_y
        if gridyname==0
            disp('You did not select a file!')
        end
    cd(Pathgridy);
    grid_y=importdata(gridyname,'\t');
    hold on %plot old grid
    plot(grid_x, grid_y,'+r')
    hold off
    waitfor(msgbox('Grid successfully loaded.'));

else
    [grid_x,grid_y,FileNameBase,PathNameBase] = rect_grid(FileNameBase,
PathNameBase, im_grid, grid_x, grid_y);
end
end
end

%% Select a rect area
function [grid_x,grid_y,FileNameBase,PathNameBase] =
rect_grid(FileNameBase, PathNameBase, im_grid, grid_x, grid_y)

% Function to select a rectangular grid and to add these to an existing
one
% wirtten by Chris
%

title(sprintf('Define the region of interest.  Pick (single click) a
point in the LOWER LEFT region of the gage section.\n  Do the same for
a point in the UPPER RIGHT portion of the gage section.'))

[x(1,1),y(1,1)]=ginput(1);
hold on
plot(x(1,1),y(1,1),'+b')

[x(2,1),y(2,1)]=ginput(1);
hold on
plot(x(2,1),y(2,1),'+b')

drawnow

xmin = min(x);
xmax = max(x);
ymin = min(y);
ymax = max(y);

lowerline=[xmin ymin; xmax ymin];
upperline=[xmin ymax; xmax ymax];
leftline=[xmin ymin; xmin ymax];
rightline=[xmax ymin; xmax ymax];

plot(lowerline(:,1),lowerline(:,2),'-b')
plot(upperline(:,1),upperline(:,2),'-b')
```

```matlab
plot(leftline(:,1),leftline(:,2),'-b')
plot(rightline(:,1),rightline(:,2),'-b')

% closereq

cd(PathNameBase)

% Prompt user for grid spacing/resolution
prompt = {'Enter horizontal (x) resolution for image analysis
[pixels]:', ...
        'Enter vertical (y) resolution for image analysis [pixels]:'};
dlg_title = 'Input for grid creation';
num_lines= 1;
def      = {'50','50'};
answer = inputdlg(prompt,dlg_title,num_lines,def);
xspacing = str2double(cell2mat(answer(1,1)));
yspacing = str2double(cell2mat(answer(2,1)));

% Round xmin,xmax and ymin,ymax "up" based on selected spacing
numXelem = ceil((xmax-xmin)/xspacing)-1;
numYelem = ceil((ymax-ymin)/yspacing)-1;


xmin_new = (xmax+xmin)/2-((numXelem/2)*xspacing);
xmax_new = (xmax+xmin)/2+((numXelem/2)*xspacing);
ymin_new = (ymax+ymin)/2-((numYelem/2)*yspacing);
ymax_new = (ymax+ymin)/2+((numYelem/2)*yspacing);

% Create the analysis grid and show user
[x,y] =
meshgrid(xmin_new:xspacing:xmax_new,ymin_new:yspacing:ymax_new);
[rows columns] = size(x);
%zdummy = 200.*ones(rows,columns);
imshow(FileNameBase)
title(['Selected grid has ',num2str(rows*columns), ' rasterpoints'])
% plot a title onto the image
hold on;
plot(grid_x,grid_y,'+r')
plot(x,y,'+b')


% Do you want to keep/add the grid?
confirmselection = menu(sprintf('Do you want to use this grid?'),...
    'Yes','No, try again');

    % Yes
    if confirmselection==1
        % Save settings and grid files in the image directory for
visualization/plotting later
        x=reshape(x,[],1);
        y=reshape(y,[],1);
        grid_x=[grid_x;x];
        grid_y=[grid_y;y];
        save settings.dat xspacing yspacing xmin_new xmax_new ymin_new
ymax_new -ascii -tabs
        save grid_x.dat grid_x -ascii -tabs
```

```matlab
            save grid_y.dat grid_y -ascii -tabs
            close all
            hold off
            gridtypeselection(FileNameBase, PathNameBase, im_grid, grid_x,
grid_y,0);
     end


     % No, try again
     if confirmselection==2
            close all
            hold off
            imshow(im_grid,'InitialMagnification',100);
            hold on;
            plot(grid_x,grid_y,'+r');
            hold off;
            rect_grid(FileNameBase, PathNameBase, im_grid, grid_x, grid_y);
     end


     % Go back to Main Menu
end
%% Select a circular area
function [grid_x,grid_y,FileNameBase,PathNameBase] =
circ_grid(FileNameBase, PathNameBase, im_grid, grid_x, grid_y)

title(sprintf('Pick three points on the circle in clockwise order at
the upper boundary of the sample.') )

[x(1,1),y(1,1)]=ginput(1);
hold on
plot(x(1,1),y(1,1),'+g')

[x(2,1),y(2,1)]=ginput(1);
plot(x(2,1),y(2,1),'+g')

[x(3,1),y(3,1)]=ginput(1);
plot(x(3,1),y(3,1),'+g')

xnew=x;
ynew=y;

% Calculate center between the 3 sorted points and the normal slope of
the vectors
slope12=-1/((ynew(2,1)-ynew(1,1))/(xnew(2,1)-xnew(1,1)));
slope23=-1/((ynew(3,1)-ynew(2,1))/(xnew(3,1)-xnew(2,1)));
center12(1,1)=(xnew(2,1)-xnew(1,1))/2+xnew(1,1);
center12(1,2)=(ynew(2,1)-ynew(1,1))/2+ynew(1,1);
center23(1,1)=(xnew(3,1)-xnew(2,1))/2+xnew(2,1);
center23(1,2)=(ynew(3,1)-ynew(2,1))/2+ynew(2,1);



% Calculate the crossing point of the two vectors
achsenabschnitt1=center12(1,2)-center12(1,1)*slope12;
achsenabschnitt2=center23(1,2)-center23(1,1)*slope23;
xcross=(achsenabschnitt2-achsenabschnitt1)/(slope12-slope23);
```

225

```matlab
ycross=slope12*xcross+achsenabschnitt1;
plot(xcross,ycross,'or')

% Calculate radius
R=sqrt((xcross-xnew(1,1))*(xcross-xnew(1,1))+(ycross-
ynew(1,1))*(ycross-ynew(1,1)));

% Calculate angle between vectors
xvector=[1;0];
x1vec(1,1)=xnew(1,1)-xcross;x1vec(2,1)=ynew(1,1)-ycross;
x3vec(1,1)=xnew(3,1)-xcross;x3vec(2,1)=ynew(3,1)-ycross;
alpha13=acos((dot(x1vec,x3vec))/(sqrt(x1vec'*x1vec)*sqrt(x3vec'*x3vec))
)*180/pi;
alpha03=acos((dot(xvector,x3vec))/(sqrt(xvector'*xvector)*sqrt(x3vec'*x
3vec)))*180/pi;
totalangle=alpha13;
maxangle=alpha03;
angldiv=abs(round(totalangle))*10;
anglstep=(totalangle/angldiv);
anglall(1:angldiv+1)=maxangle+anglstep*(1:angldiv+1)-anglstep;
xcircle(1:angldiv+1)=xcross+R*cos(-anglall(1:angldiv+1)/180*pi);
ycircle(1:angldiv+1)=ycross+R*sin(-anglall(1:angldiv+1)/180*pi);
plot(xcircle,ycircle,'-b')
drawnow

title(['Segment of circle spreads over ',num2str(totalangle),'°'])


% Accept the chosen circle, try again or give up

confirmcircselection = menu(sprintf('Do you want to use this circle as
basis?'),...
    'Yes','No, try again','Go back to grid-type selection');

    % No, try again
    if confirmcircselection==2
        close all
        imshow(im_grid,'truesize');
        hold on
        plot(grid_x,grid_y,'+r');
        hold off
        circ_grid(FileNameBase, PathNameBase, im_grid, grid_x, grid_y);
    end

    % Go back to grid-type selection
    if confirmcircselection==3
        close all
        gridtypeselection(FileNameBase, PathNameBase, im_grid, grid_x,
grid_y,0);
    end

    % Yes
    if confirmcircselection==1
```

```matlab
        prompt = {'Enter the number of intersections between markers on
the circle:'};
        dlg_title = 'Input for grid creation';
        num_lines= 1;
        def    = {'30'};
        answer = inputdlg(prompt,dlg_title,num_lines,def);
        angldiv = str2double(cell2mat(answer(1,1)));

        anglstep=(totalangle/angldiv);
        anglall(1:angldiv+1)=maxangle+anglstep*(1:angldiv+1)-anglstep;

        markerxpos(1:angldiv+1)=xcross+R*cos(-
anglall(1:angldiv+1)/180*pi);
        markerypos(1:angldiv+1)=ycross+R*sin(-
anglall(1:angldiv+1)/180*pi);

        plot(markerxpos,markerypos,'ob');

        % Pick the lower bound in the image
        title(sprintf('Pick three points lying on the circle in
clockwise order. The first and last one define the width of the
raster') )

        [x(4,1),y(4,1)]=ginput(1);
        hold on
        plot(x(1,1),y(1,1),'+r')

        lowboundx=x(4,1);
        lowboundy=y(4,1);

        R2=sqrt((xcross-lowboundx(1,1))*(xcross-
lowboundx(1,1))+(ycross-lowboundy(1,1))*(ycross-lowboundy(1,1)));
        markerxposlb(1:angldiv+1)=xcross+R2*cos(-
anglall(1:angldiv+1)/180*pi);
        markeryposlb(1:angldiv+1)=ycross+R2*sin(-
anglall(1:angldiv+1)/180*pi);

        plot(markerxposlb,markeryposlb,'ob');

        prompt = {'Enter the number of intersections between the upper
and lower bound:'};
        dlg_title = 'Input for grid creation';
        num_lines= 1;
        def    = {'5'};
        answer = inputdlg(prompt,dlg_title,num_lines,def);
        Rdiv = str2double(cell2mat(answer(1,1)));

        Rstep=((R-R2)/Rdiv);
        Rall(1:Rdiv+1)=R2+Rstep*(1:Rdiv+1)-Rstep;

        x=ones(Rdiv+1,angldiv+1)*xcross;
        y=ones(Rdiv+1,angldiv+1)*ycross;
        x=x+Rall'*cos(-anglall(1:angldiv+1)/180*pi);
        y=y+Rall'*sin(-anglall(1:angldiv+1)/180*pi);
```

```matlab
        close all
        imshow(im_grid,'truesize');
        hold on
        plot(grid_x,grid_y,'+r')
        plot(x,y,'.b')

        title(['Selected grid has ',num2str(angldiv*Rdiv), '
rasterpoints'])    % plot a title onto the image


         % Do you want to keep/add the grid?
        confirmselection = menu(sprintf('Do you want to use this
grid?'),...
        'Yes','No, try again','Go back to Main Menu');

        % Yes
        if confirmselection==1
            % Save settings and grid files in the image directory for
visualization/plotting later
            x=reshape(x,[],1);
            y=reshape(y,[],1);
            grid_x=[grid_x;x];
            grid_y=[grid_y;y];
            save grid_x.dat grid_x -ascii -tabs
            save grid_y.dat grid_y -ascii -tabs
            close all
            hold off
            gridtypeselection(FileNameBase, PathNameBase, im_grid,
grid_x, grid_y,0);
        end

        % No, try again
        if confirmselection==2
            close all
            hold off
            imshow(im_grid,'truesize');
            hold on;
            plot(grid_x,grid_y,'+r');
            hold off;
            circ_grid(FileNameBase, PathNameBase, im_grid, grid_x,
grid_y);
        end

        % Go back to Main Menu
        if confirmselection==3
            close all
            hold off
            gridtypeselection(FileNameBase, PathNameBase, im_grid,
grid_x, grid_y,0);
        end
    end
end
%% Select 2 Points
```

```matlab
function [grid_x,grid_y,FileNameBase,PathNameBase] =
twop_grid(FileNameBase, PathNameBase, im_grid, grid_x, grid_y)


title(sprintf('Pick two points on the sample.') )


[x(1,1),y(1,1)]=ginput(1);
hold on
plot(x(1,1),y(1,1),'+b')

[x(2,1),y(2,1)]=ginput(1);
plot(x(2,1),y(2,1),'+b')

% Do you want to keep/add the grid?
confirmselection = menu(sprintf('Do you want to use this grid?'),...
    'Yes','No, try again','Go back to Main Menu');

    % Yes
    if confirmselection==1
        % Save settings and grid files in the image directory for
visualization/plotting later
        x=reshape(x,[],1);
        y=reshape(y,[],1);
        grid_x=[grid_x;x];
        grid_y=[grid_y;y];
        save grid_x.dat grid_x -ascii -tabs
        save grid_y.dat grid_y -ascii -tabs
        close all
        hold off
        gridtypeselection(FileNameBase, PathNameBase, im_grid, grid_x,
grid_y,0);
    end


    % No, try again
    if confirmselection==2
        close all
        hold off
        imshow(im_grid,'truesize');
        hold on;
        plot(grid_x,grid_y,'+r');
        hold off;
        twop_grid(FileNameBase, PathNameBase, im_grid, grid_x, grid_y);
    end


    % Go back to Main Menu
    if confirmselection==3
        close all
        hold off
        gridtypeselection(FileNameBase, PathNameBase, im_grid, grid_x,
grid_y,0);
    end
end
%% Select a line
function [grid_x,grid_y,FileNameBase,PathNameBase] =
line_grid(FileNameBase, PathNameBase, im_grid, grid_x, grid_y)
```

```matlab
title(sprintf('Pick two points on the sample.') )

[x(1,1),y(1,1)]=ginput(1);
hold on
plot(x(1,1),y(1,1),'+b')

[x(2,1),y(2,1)]=ginput(1);
plot(x(2,1),y(2,1),'+b')

lineslope=(y(2,1)-y(1,1))/(x(2,1)-x(1,1));
intersecty=y(1,1)-lineslope*x(1,1);
ycalc=zeros(2,1);
ycalc=lineslope*x+intersecty;
plot(x(:,1),ycalc(:,1),'-b')


prompt = {'Enter the number of intersections between markers on the
line:'};
dlg_title = 'Input for grid creation';
num_lines= 1;
def      = {'30'};
answer = inputdlg(prompt,dlg_title,num_lines,def);
linediv = str2num(cell2mat(answer(1,1)));
linestep=((max(x)-min(x))/linediv);
x(1:linediv+1)=min(x)+linestep*(1:linediv+1)-linestep;
y=lineslope*x+intersecty;

plot(x,y,'ob')
title(['Selected grid has ',num2str(linediv), ' rasterpoints'])    %
plot a title onto the image


% Do you want to keep/add the grid?
confirmselection = menu(sprintf('Do you want to use this grid?'),...
    'Yes','No, try again','Go back to Main Menu');

    % Yes
    if confirmselection==1
        % Save settings and grid files in the image directory for
visualization/plotting later
        x=reshape(x,[],1);
        y=reshape(y,[],1);
        grid_x=[grid_x;x];
        grid_y=[grid_y;y];
        save grid_x.dat grid_x -ascii -tabs
        save grid_y.dat grid_y -ascii -tabs
        close all
        hold off
        gridtypeselection(FileNameBase, PathNameBase, im_grid, grid_x,
grid_y,0);
    end

    % No, try again
    if confirmselection==2
        close all
```

```matlab
        hold off
        imshow(im_grid,'truesize');
        hold on;
        plot(grid_x,grid_y,'+r');
        hold off;
        line_grid(FileNameBase, PathNameBase, im_grid, grid_x, grid_y);
    end

    % Go back to Main Menu
    if confirmselection==3
        close all
        hold off
        gridtypeselection(FileNameBase, PathNameBase, im_grid, grid_x,
grid_y,0);
    end
end
%% Select Points to remove
function [grid_x,grid_y,FileNameBase,PathNameBase] =
removepoints(FileNameBase, PathNameBase, im_grid, grid_x, grid_y)

% Delete some markers
%

%create working copy of the grid

grid_xtemp=grid_x;
grid_ytemp=grid_y;

imshow(im_grid,'InitialMagnification',100); %show chosen Image
hold on
plot(grid_x, grid_y,'ob');
hold off,

title(sprintf('Define the region of interest.  \n  All points inside
that region will be deleted'))

        [xdel,ydel]=ginput(2);
        x(1,1) = xdel(1);
        x(1,2) = xdel(2);
        y(1,1) = ydel(2);
        y(1,2) = ydel(1);

        deletepoints=find(grid_x>min(x) & grid_x<max(x) & grid_y<max(y)
& grid_y>min(y));

        grid_xtemp(deletepoints,:)=[];
        grid_ytemp(deletepoints,:)=[];

        imshow(im_grid,'InitialMagnification',100); %show chosen Image
        hold on
        plot(grid_xtemp, grid_ytemp,'ob');
        hold off,
```

```matlab
        % delete point permanently?
        keepchanges = menu(sprintf('Do you want to delete these markers
permanently?'),'Yes','No');
        if keepchanges==1
            grid_x=grid_xtemp;
            grid_y=grid_ytemp;
            save grid_x.dat grid_x -ascii -tabs
            save grid_y.dat grid_y -ascii -tabs
            close all
            hold off
            gridtypeselection(FileNameBase, PathNameBase, im_grid,
grid_x, grid_y,0);
        end


        if keepchanges==2
            gridtypeselection(FileNameBase, PathNameBase, im_grid,
grid_x, grid_y,0);
        end


end
```

Code end

Automate_image_DIC: This is the main function where the DIC analysis of the images in the filelist created previously is performed, using the grid created in the grid_generator function. This function was originally programmed by Christoph and Rob, however, was modified from its original form by the author of this dissertation, so as to allow him to examine the results in a way meaningful to this thesis. The function relies on the CPcorr function, which is a Matlab prebuilt function, though it is important to note, minor modifications were made on CPcorr, as suggested by Christoph, to allow for greater accuracy in the DIC analysis.

Code start

```
function
[validx,validy]=automate_image_ld(grid_x,grid_y,filenamelist,validx,val
idy);


% Code to start actual image correlation for large displacements, needs
the
% displacementin x and y direction
% Original program by Chris and Rob
% Modified for use at UoM by Mike
% Last revision: Feb 2013


% The automation function is the central function and processes all
markers and
% images by the use of the matlab function cpcorr.m.
% Therefore the Current directory in matlab has to be the folder where
%  automate_image.m finds the filenamelist.mat, grid_x.dat and
grid_y.dat as well
% as the images specified in filenamelist.mat. Just type
automate_image; and
% press ENTER at the command line of matlab.
% At first, automate_image.m will open the first image in the
filenamelist.mat and
% plot the grid as green crosses on top. The next step will need some
time since
% all markers in that image have to be processed for the first image.
After correlating
% image one and two the new raster positions will be plotted as red
crosses. On top
% of the image and the green crosses. The next dialog will ask you if
you want to
% continue with this correlation or cancel. If you press continue,
automate_image.m
% will process all images in the filenamelist.mat. The time it will
take to process
% all images will be plotted on the figure but can easily be estimated
by knowing the
% raster point processing speed (see processing speed).
% Depending on the number of images and markers you are tracking, this
process
% can take between seconds and days. For 100 images and 200 markers a
decent
% computer should need 200 seconds. To get a better resolution you can
always
```

```
% run jobs overnight (e.g. 6000 markers in 1000 images) with higher
resolutions.
% Keep in mind that CORRSIZE which you changed in cpcorr.m will limit
your
% resolution. If you chose to use the 15 pixel as suggested a marker
distance of
% 30 pixel will lead to a full cover of the strain field. Choosing
smaller marker
% distances will lead to an interpolation since two neighboring markers
share
% pixels. Nevertheless a higher marker density can reduce the noise of
the strain field.
% When all images are processed, automate_image will write the files
validx.mat,
% validy.mat, validx.txt and validy.txt. The text files are meant to
store the result in a
% format which can be accessed by other programs also in the future.

% Load necessary files

if exist('grid_x')==0
    load('grid_x.dat')              % file with x position, created by
grid_generator.m
end
if exist('grid_y')==0
    load('grid_y.dat')              % file with y position, created by
grid_generator.m
end
if exist('filenamelist')==0
    load('filenamelist')            % file with the list of filenames
to be processed
end
resume=0;
if exist('validx')==1
    if exist('validy')==1
        resume=1;
        [Rasternum Imagenum]=size(validx);
    end
end


% Initialize variables
input_points_x=grid_x;
base_points_x=grid_x;

input_points_y=grid_y;
base_points_y=grid_y;

if resume==1
    input_points_x=validx(:,Imagenum);
    input_points_y=validy(:,Imagenum);
    inputpoints=1;
end
```

```matlab
[row,col]=size(base_points_x);          % this will determine the number of
rasterpoints we have to run through
[r,c]=size(filenamelist);                       % this will determine the
number of images we have to loop through

% Open new figure so previous ones (if open) are not overwritten
h=figure;
imshow(filenamelist(1,:))               % show the first image
title('Initial Grid For Image Correlation (Note green crosses)')
% put a title
hold on
plot(grid_x,grid_y,'g+')                % plot the grid onto the image
hold off

% Start image correlation using cpcorr.m
g = waitbar(0,sprintf('Processing images'));        % initialize the
waitbar
set(g,'Position',[275,50,275,50])                               % set
the position of the waitbar [left bottom width height]
firstimage=1;

if resume==1
    firstimage=Imagenum+1
end
    base = uint8(mean(double(imread(filenamelist(1,:))),3));
% read in the base image ( which is always  image number one. You might
want to change that to improve correlation results in case the light
conditions are changing during the experiment
for i=firstimage:(r-1)                  % run through all images


    tic             % start the timer
    input = uint8(mean(double(imread(filenamelist((i+1),:))),3));
% read in the image which has to be correlated

    input_points_for(:,1)=reshape(input_points_x,[],1);         % we
reshape the input points to one row of values since this is the shape
cpcorr will accept
    input_points_for(:,2)=reshape(input_points_y,[],1);
    base_points_for(:,1)=reshape(base_points_x,[],1);
    base_points_for(:,2)=reshape(base_points_y,[],1);
    input_correl(:,:)=cpcorr(round(input_points_for),
round(base_points_for), input, base);           % here we go and give
all the markers and images to process to cpcorr.m which ic a function
provided by the matlab image processing toolbox
    input_correl_x=input_correl(:,1);
% the results we get from cpcorr for the x-direction
    input_correl_y=input_correl(:,2);
% the results we get from cpcorr for the y-direction

    if exist ('valid_points.dat')==1
        load('valid_points.dat')        %MIKE edit
    end
    load('invalid_x.dat')
    load('invalid_y.dat')
```

```matlab
    validx(:,i)=input_correl_x;
% lets save the data
    savelinex=input_correl_x';
    dlmwrite('resultsimcorrx.txt', savelinex , 'delimiter', '\t', '-
append');        % Here we save the result from each image; if you are
desperately want to run this function with e.g. matlab 6.5 then you
should comment this line out. If you do that the data will be saved at
the end of the correlation step - good luck ;-)

    validy(:,i)=input_correl_y;
    saveliney=input_correl_y';
    dlmwrite('resultsimcorry.txt', saveliney , 'delimiter', '\t', '-
append');

    waitbar(i/(r-1))
% update the waitbar

    % Update base and input points for cpcorr.m
    base_points_x=grid_x;
    base_points_y=grid_y;
    input_points_x=input_correl_x;
    input_points_y=input_correl_y;

    imshow(filenamelist(i+1,:))                        % update image
    hold on
    plot(grid_x,grid_y,'c+')                           % plot
start position of raster
    plot(input_correl_x,input_correl_y,'g+')        % plot actual
postition of raster
    plot(invalid_x,invalid_y,'r+')      %MIKE edit
    hold off
    drawnow
    time(i)=toc;                                       % take
time
    estimatedtime=sum(time)/i*(r-1);        % estimate time to
process
    title(['# Im.: ', num2str((r-1)),'; Proc. Im. #: ', num2str((i)),';
# Rasterp.:',num2str(row*col), '; Est. Time [s] ',
num2str(round(estimatedtime)), ';  Elapsed Time [s] ',
num2str(round(sum(time)))]);    % plot a title onto the image
    drawnow

end
saveas(h,['grid.tif']);    %Mike Edit
%msgboxwicon=msgbox('Finished.')
%waitfor(msgboxwicon)

close(g)
close all

% save

save time.dat time -ascii -tabs
save validx.dat validx -ascii -tabs
```

```
save validy.dat validy -ascii -tabs
```

Code end

CombineResults: This program was used to combine the X and Y DIC results with the Z component which was measured with confocal microscopy. High resolution EBSD stress results are also combined here to allow plastic and elastic strain to be analyzed together, though this portion of the function was not used in the final results of this thesis. This code was created solely by the author of this thesis.

Code start

```matlab
function varargout = CombineResults(varargin)

% First load the data file containing the topographical information. This
% should be a tab delimited text file.


%LOADING DIC DATA FROM .MAT FILE
[FileNameBase,PathNameBase,FilterIndex] = uigetfile('*.mat', 'Open
matlab file with DIC data');
cd(PathNameBase)
load(FileNameBase);
minstrain=-0.1;
maxstrain=0.1;
minZ=min([min(ZIx) min(ZIy)]);
maxZ=max([max(ZIx) max(ZIy)]);
epsxy=0.5*(exy+eyx);
epsyx=epsxy;
rotxy=0.5*(exy-eyx);
rotyx=0.5*(eyx-exy);

h=figure;
pcolor(XI,YI,epsxx)
axis('equal')
shading('interp')
straincolor(1)=minstrain;
straincolor(2)=maxstrain;
caxis(straincolor)
h1 = colorbar;
%rectangle('Position',[sb_posx, sb_posy, scale_pixels, 10],
'FaceColor','k');
set(h1, 'PlotBoxAspectRatio',[2.0 10 8.0])
set(h1, 'FontSize', 12);
title('X Strain');
print(h,'-dtiff','X_strain_2D.tif');
hhandle=get(h);
haxes=hhandle.CurrentAxes;

[FileNameBase,PathNameBase,FilterIndex] = uigetfile('*.txt', 'Open text
file with topography data');
cd(PathNameBase)
BaseImage=load(FileNameBase);
%input information used to convert pixels to microns.
prompt = 'Enter spacing (µm) between data points';
dlg_title = 'Pixel to µm conversion';
num_lines= 1;
```

```matlab
def     = {'0.250'};
answer = inputdlg(prompt,dlg_title,num_lines,def);
AOIum_pixel=str2double(cell2mat(answer(1,1)));

%display entire data set
[BaseSizeX,BaseSizeY]=size(BaseImage);
[BaseX,BaseY]=meshgrid(1:BaseSizeY,1:BaseSizeX);
YTemp=BaseY;  %Y is always reversed in matlab
sizeY=size(BaseY,1);
for reverseY=1:sizeY
    from_end=sizeY+1-reverseY;
    BaseY(reverseY,:)=YTemp(from_end,:);
end
BaseX=double(BaseX)*AOIum_pixel;
BaseY=double(BaseY)*AOIum_pixel;

AOIConfirm=2;
while AOIConfirm==2
    BaseFigure=figure
    pcolor(BaseX,BaseY,BaseImage)
    shading('interp')
    axis('equal')
    %select a smaller area of the figure to work with
    waitfor(msgbox('select area of interest (click in lower left corner
and upper right corner of AOI)'))
    [AOISizeX(1,1),AOISizeY(1,1)]=ginput(1);
    hold on
    plot(AOISizeX(1,1),AOISizeY(1,1),'+k')
    [AOISizeX(2,1),AOISizeY(2,1)]=ginput(1);
    plot(AOISizeX(2,1),AOISizeY(2,1),'+k')
    AOISizeX=round(AOISizeX);
    AOISizeY=round(AOISizeY);
    hold off

    xmin=min(AOISizeX);
    xmax=max(AOISizeX);
    ymin=min(AOISizeY);
    ymax=max(AOISizeY);
    start_x=1;
    start_y=1;
    while BaseX(1,start_x)<xmin
        start_x=start_x+1;
    end
    while BaseY(start_y,1)>ymax
        start_y=start_y+1;
    end
    end_x=start_x+1;
    end_y=start_y+1;
    while BaseX(1,end_x+1)<xmax  %NEEDS TO BE MODIFIED IN CASE USER
SELECTS EDGE
        end_x=end_x+1;
    end
    while BaseY(end_y+1,1)>ymin
        end_y=end_y+1;
    end
    AOISizeX(1,1)=start_x;
```

```matlab
    AOISizeX(2,1)=end_x;
    AOISizeY(1,1)=start_y;
    AOISizeY(2,1)=end_y;




AOI=BaseImage(AOISizeY(1,1):AOISizeY(2,1),AOISizeX(1,1):AOISizeX(2,1));

AOIX=BaseX(AOISizeY(1,1):AOISizeY(2,1),AOISizeX(1,1):AOISizeX(2,1));

AOIY=BaseY(AOISizeY(1,1):AOISizeY(2,1),AOISizeX(1,1):AOISizeX(2,1));


gridsizex=round(min(min(AOIX))*10)/10:.1:round(max(max(AOIX))*10)/10;
%Make pixels 100 nm.

gridsizey=round(min(min(AOIY))*10)/10:.1:round(max(max(AOIY))*10)/10;
    [AOIXn,AOIYn]=meshgrid(gridsizex,gridsizey);
    AOIn=griddata(AOIX,AOIY,AOI,AOIXn,AOIYn,'cubic');      %MIKE edit
change displx to displtot
    AOI=AOIn;
    AOIX=AOIXn;
    AOIY=AOIYn;

    AOIFigure=figure;
    pcolor(AOIX,AOIY,AOI)
    axis('equal')
    shading('interp')
    AOIConfirm = menu(sprintf('Is this the area you are interested
in?'),'Yes','No');
    if AOIConfirm==2
        close(AOIFigure)
    end
    close(BaseFigure)
end

YTemp=AOIY;  %Y is always reversed in matlab, when new grid at 100 nm
was made, Y inverted, and needs to be put back to be consistent
YTemp2=AOI;
sizeY=size(AOI,1);
for reverseY=1:sizeY
    from_end=sizeY+1-reverseY;
    AOIY(reverseY,:)=YTemp(from_end,:);
    AOI(reverseY,:)=YTemp2(from_end,:);
end
close(AOIFigure)
AOIFigure=figure;
    pcolor(AOIX,AOIY,AOI)
    axis('equal')
    shading('interp')




%LOADING EBSD DATA FROM CROSSCOURT MATLAB FILE
```

```matlab
[FileNameBase,PathNameBase,FilterIndex] = uigetfile('*.mat', 'Open
matlab file with EBSD data');
cd(PathNameBase)
BaseEBSD=load(FileNameBase);
EBSDum_pixel=BaseEBSD.xstep;
EBSDXsize=BaseEBSD.numrows;
EBSDYsize=BaseEBSD.numcols;
EBSDShearsSep=BaseEBSD.datashears;  %Data for separate grains
EBSDNormalsSep=BaseEBSD.datanorms;
EBSDRotationsSep=BaseEBSD.datarots;
EBSDStressSep=BaseEBSD.datastress;
EBSDErrorSep=BaseEBSD.dataerror;
NumGrains=size(EBSDShearsSep,4);
EBSDShears=zeros(size(EBSDShearsSep(:,:,:,1)));
EBSDNormals=zeros(size(EBSDNormalsSep(:,:,:,1)));
EBSDRotations=zeros(size(EBSDRotationsSep(:,:,:,1)));
EBSDStress=zeros(size(EBSDStressSep(:,:,:,1)));
EBSDError=zeros(size(EBSDErrorSep(:,:,:,1)));
for CombineRows=1:EBSDXsize  %Combining grain data into one map
    for CombineCols=1:EBSDYsize
        checkNaN=1;
        for CombineGrains=1:NumGrains
            if
~isnan(EBSDShearsSep(CombineRows,CombineCols,:,CombineGrains))
                checkNaN=0;

EBSDShears(CombineRows,CombineCols,:)=EBSDShearsSep(CombineRows,Combine
Cols,:,CombineGrains);

EBSDNormals(CombineRows,CombineCols,:)=EBSDNormalsSep(CombineRows,Combi
neCols,:,CombineGrains);

EBSDRotations(CombineRows,CombineCols,:)=EBSDRotationsSep(CombineRows,C
ombineCols,:,CombineGrains);

EBSDStress(CombineRows,CombineCols,:)=EBSDStressSep(CombineRows,Combine
Cols,:,CombineGrains);

EBSDError(CombineRows,CombineCols)=EBSDErrorSep(CombineRows,CombineCols
,CombineGrains);
                if EBSDError(CombineRows,CombineCols)>0.01  %Remove
points with error >.01 radians
                    checkNaN=1;
                end
            end
        end
        if checkNaN==1
            EBSDShears(CombineRows,CombineCols,:)=NaN;
            EBSDNormals(CombineRows,CombineCols,:)=NaN;
            EBSDRotations(CombineRows,CombineCols,:)=NaN;
            EBSDStress(CombineRows,CombineCols,:)=NaN;
            EBSDError(CombineRows,CombineCols,:)=NaN;
        end
    end
end
[EBSDX,EBSDY]=meshgrid(1:EBSDYsize,1:EBSDXsize); %Map 1 is XY, 2 is YZ,
3 is ZX
```

```
EBSDStepSize=BaseEBSD.xstep;
YTemp=EBSDY;  %Y is always reversed in matlab
sizeY=size(EBSDY,1);
for reverseY=1:sizeY
    from_end=sizeY+1-reverseY;
    EBSDY(reverseY,:)=YTemp(from_end,:);
end
EBSDX=double(EBSDX)*EBSDStepSize;
EBSDY=double(EBSDY)*EBSDStepSize;
EBSDFigure=figure
pcolor(EBSDX,EBSDY,EBSDNormals(:,:,1))
axis('equal')

figure(h)
waitfor(msgbox('Select a point that all maps have in common. Then
select a second feature to help align rotation of the maps'))
[DICX1,DICY1]=ginput(1);
hold on
plot(DICX1,DICY1,'+k')
[DICX2,DICY2]=ginput(1);
plot(DICX2,DICY2,'+k')
drawnow
hold off
DICL=DICX1-XI(1,1);
DICR=XI(1,size(XI,2))-DICX1;
DICU=YI(1,1)-DICY1;
DICD=DICY1-YI(size(YI,1),1);

figure(AOIFigure)
waitfor(msgbox('Select the same point that all maps have in common.
Then select the same second feature to help align rotation of the
maps'))
[AOIX1,AOIY1]=ginput(1);
hold on
plot(AOIX1,AOIY1,'+k')
[AOIX2,AOIY2]=ginput(1);
plot(AOIX2,AOIY2,'+k')
drawnow
hold off
AOIL=AOIX1-AOIX(1,1);
AOIR=AOIX(1,size(AOIX,2))-AOIX1;
AOIU=abs(AOIY(1,1)-AOIY1);     %********************AOI Y doesn't go
same way as others. Should change to be consistent
AOID=abs(AOIY1-AOIY(size(AOIY,1),1));

figure(EBSDFigure)
waitfor(msgbox('Select the same point that all maps have in common.
Then select the same second feature to help align rotation of the
maps'))
[EBSDX1,EBSDY1]=ginput(1);
hold on
plot(EBSDX1,EBSDY1,'+k')
[EBSDX2,EBSDY2]=ginput(1);
plot(EBSDX2,EBSDY2,'+k')
drawnow
hold off
```

```
EBSDL=EBSDX1-EBSDX(1,1);
EBSDR=EBSDX(1,size(EBSDX,2))-EBSDX1;
EBSDU=EBSDY(1,1)-EBSDY1;
EBSDD=EBSDY1-EBSDY(size(EBSDY,1),1);

%Need to rotate so that all images are in the same orientation.  Will
use
%EBSD as set orientation and rotate DIC/Topographical data.




LeftMin=min(min(EBSDL, AOIL), DICL);
RightMin=min(min(EBSDR, AOIR), DICR);
UpMin=min(min(EBSDU,AOIU),DICU);
DownMin=min(min(EBSDD,AOID),DICD);

%DIC Cropping
start_x=1;
start_y=1;
xmin=DICX1-LeftMin;
xmax=DICX1+RightMin;
ymax=DICY1+UpMin;
ymin=DICY1-DownMin;
while XI(1,start_x)<xmin
    start_x=start_x+1;
end
while YI(start_y,1)>ymax
    start_y=start_y+1;
end
end_x=start_x+1;
end_y=start_y+1;
while XI(1,end_x+1)<xmax  %NEEDS TO BE MODIFIED IN CASE USER SELECTS
EDGE
    end_x=end_x+1;
end
while YI(end_y+1,1)>ymin
    end_y=end_y+1;
end
epsxxC=epsxx(start_y:end_y,start_x:end_x);  %Cropped data so all is
same size
epsyyC=epsyy(start_y:end_y,start_x:end_x);
epsxyC=epsxy(start_y:end_y,start_x:end_x);
exyC=exy(start_y:end_y,start_x:end_x);
eyxC=eyx(start_y:end_y,start_x:end_x);
rotyxC=rotyx(start_y:end_y,start_x:end_x);
rotxyC=rotxy(start_y:end_y,start_x:end_x);
ZIC=ZI(start_y:end_y,start_x:end_x);
ZIxC=ZIx(start_y:end_y,start_x:end_x);
ZIyC=ZIy(start_y:end_y,start_x:end_x);
XIC=XI(start_y:end_y,start_x:end_x);
YIC=YI(start_y:end_y,start_x:end_x);
DICFigure=figure;
pcolor(XIC,YIC,epsxxC)
axis('equal')
shading('interp')
```

```matlab
%Topographical cropping
start_x=1;
start_y=1;
xmin=AOIX1-LeftMin;
xmax=AOIX1+RightMin;
ymax=AOIY1+UpMin;
ymin=AOIY1-DownMin;
while AOIX(1,start_x)<xmin
    start_x=start_x+1;
end
while AOIY(start_y,1)>ymax
    start_y=start_y+1;
end
end_x=start_x+1;
end_y=start_y+1;
while AOIX(1,end_x+1)<xmax  %NEEDS TO BE MODIFIED IN CASE USER SELECTS
EDGE
    end_x=end_x+1;
end
while AOIY(end_y+1,1)>ymin
    end_y=end_y+1;
end
AOIC=AOI(start_y:end_y,start_x:end_x);  %Cropped data so all is same
size
AOIX=AOIX(start_y:end_y,start_x:end_x);
AOIY=AOIY(start_y:end_y,start_x:end_x);
AOIFigure1=figure;
pcolor(AOIX,AOIY,AOIC)
axis('equal')
shading('interp')
[ezx, ezy] = gradient(AOIC,.1,.1);
AOIFigure2=figure;
pcolor(AOIX,AOIY,ezx)
axis('equal')
shading('interp')
AOIFigure2=figure;
pcolor(AOIX,AOIY,ezy)
axis('equal')
shading('interp')

%EBSD cropping
start_x=1;
start_y=1;
xmin=EBSDX1-LeftMin;
xmax=EBSDX1+RightMin;
ymax=EBSDY1+UpMin;
ymin=EBSDY1-DownMin;
while EBSDX(1,start_x)<xmin
    start_x=start_x+1;
end
while EBSDY(start_y,1)>ymax
    start_y=start_y+1;
end
end_x=start_x+1;
end_y=start_y+1;
```

```matlab
while EBSDX(1,end_x+1)<xmax  %NEEDS TO BE MODIFIED IN CASE USER SELECTS
EDGE
    end_x=end_x+1;
end
while EBSDY(end_y+1,1)>ymin
    end_y=end_y+1;
end
EBSDNormalsC=EBSDNormals(start_y:end_y,start_x:end_x,:);  %Cropped data
so all is same size
EBSDShearsC=EBSDShears(start_y:end_y,start_x:end_x,:);
EBSDRotationsC=EBSDRotations(start_y:end_y,start_x:end_x,:);
EBSDStressC=EBSDStress(start_y:end_y,start_x:end_x,:);

EBSDX=EBSDX(start_y:end_y,start_x:end_x);
EBSDY=EBSDY(start_y:end_y,start_x:end_x);
EBSDFigure=figure;
pcolor(EBSDX,EBSDY,EBSDNormalsC(:,:,1))
axis('equal')

DirCount=1;  %Create folder for results
DirName=[pwd '\Combined Results'];
DirNameTemp=DirName;
while exist(DirName, 'dir')
    DirCount=DirCount+1;
    DirStr=num2str(DirCount);
    DirName=[DirNameTemp DirStr];
end
mkdir(DirName);
cd(DirName);

save('PlasticStrainResults.mat','ZIxC','ZIyC','ZIC','epsxxC','epsxyC','
exyC','eyxC','epsyyC','rotyxC','rotxyC','XIC','YIC','AOIC','AOIX','AOIY
','ezx','ezy');
save('ElasticStrainResults.mat','EBSDNormalsC','EBSDShearsC','EBSDRotat
ionsC','EBSDStressC','EBSDX','EBSDY');

% Displacement
AOICave=sum(sum(AOIC))/(size(AOIC,1)*size(AOIC,2));  %Normalizing Z
displacement (should this be moved up in the program?)
AOIC=AOIC-AOICave
minstrain=min([min(min(ZIxC)) min(min(ZIyC)) min(min(AOIC))]);
maxstrain=max([max(max(ZIxC)) max(max(ZIyC)) max(max(AOIC))]);

plastic=figure
subplot(1,3,1)
pcolor(XIC,YIC,ZIxC)
axis('equal')
shading('interp')
straincolor(1)=minstrain;
straincolor(2)=maxstrain;
caxis(straincolor)
h1 = colorbar;
set(h1, 'PlotBoxAspectRatio',[2.0 10 8.0])
set(h1, 'FontSize', 12);
title('X disp');
```

```
subplot(1,3,2)
pcolor(XIC,YIC,ZIyC)
axis('equal')
shading('interp')
caxis(straincolor)
h1 = colorbar;
set(h1, 'PlotBoxAspectRatio',[2.0 10 8.0])
set(h1, 'FontSize', 12);
title('Y disp');

subplot(1,3,3)
pcolor(AOIX,AOIY,AOIC)
axis('equal')
shading('interp')
caxis(straincolor)
h1 = colorbar;
set(h1, 'PlotBoxAspectRatio',[2.0 10 8.0])
set(h1, 'FontSize', 12);
title('Z disp');

print(plastic,'-dtiff','Plastic displ.tif');

%PLASTIC TENSOR
minstrain=-0.035
maxstrain=0.035;
plastic=figure
subplot(3,3,1)
pcolor(XIC,YIC,epsxxC)
axis('equal')
shading('interp')
straincolor(1)=minstrain;
straincolor(2)=maxstrain;
caxis(straincolor)
h1 = colorbar;
set(h1, 'PlotBoxAspectRatio',[2.0 10 8.0])
set(h1, 'FontSize', 12);
title('X Strain');

subplot(3,3,2)
pcolor(XIC,YIC,exyC)
axis('equal')
shading('interp')
caxis(straincolor)
h1 = colorbar;
set(h1, 'PlotBoxAspectRatio',[2.0 10 8.0])
set(h1, 'FontSize', 12);
title('XY Strain');

subplot(3,3,4)
pcolor(XIC,YIC,eyxC)
axis('equal')
shading('interp')
caxis(straincolor)
h1 = colorbar;
set(h1, 'PlotBoxAspectRatio',[2.0 10 8.0])
set(h1, 'FontSize', 12);
```

```matlab
title('YX Strain');

subplot(3,3,5)
pcolor(XIC,YIC,epsyyC)
axis('equal')
shading('interp')
caxis(straincolor)
h1 = colorbar;
set(h1, 'PlotBoxAspectRatio',[2.0 10 8.0])
set(h1, 'FontSize', 12);
title('Y Strain');

subplot(3,3,7)
pcolor(AOIX,AOIY,ezx)
axis('equal')
shading('interp')
caxis(straincolor)
h1 = colorbar;
set(h1, 'PlotBoxAspectRatio',[2.0 10 8.0])
set(h1, 'FontSize', 12);
title('ZX Strain');

subplot(3,3,8)
pcolor(AOIX,AOIY,ezy)
axis('equal')
shading('interp')
caxis(straincolor)
h1 = colorbar;
set(h1, 'PlotBoxAspectRatio',[2.0 10 8.0])
set(h1, 'FontSize', 12);
title('ZY Strain');
print(plastic,'-dtiff','Plastic strain.tif');

%PLASTIC TENSOR
minstrain=-0.1;
maxstrain=0.1;
plastic=figure
subplot(3,3,1)
pcolor(XIC,YIC,epsxxC)
axis('equal')
shading('interp')
straincolor(1)=minstrain;
straincolor(2)=maxstrain;
caxis(straincolor)
h1 = colorbar;
set(h1, 'PlotBoxAspectRatio',[2.0 10 8.0])
set(h1, 'FontSize', 12);
title('X Strain');

subplot(3,3,2)
pcolor(XIC,YIC,exyC)
axis('equal')
shading('interp')
caxis(straincolor)
h1 = colorbar;
set(h1, 'PlotBoxAspectRatio',[2.0 10 8.0])
```

247

```matlab
set(h1, 'FontSize', 12);
title('XY Strain');

subplot(3,3,4)
pcolor(XIC,YIC,eyxC)
axis('equal')
shading('interp')
caxis(straincolor)
h1 = colorbar;
set(h1, 'PlotBoxAspectRatio',[2.0 10 8.0])
set(h1, 'FontSize', 12);
title('YX Strain');

subplot(3,3,5)
pcolor(XIC,YIC,epsyyC)
axis('equal')
shading('interp')
caxis(straincolor)
h1 = colorbar;
set(h1, 'PlotBoxAspectRatio',[2.0 10 8.0])
set(h1, 'FontSize', 12);
title('Y Strain');

subplot(3,3,7)
pcolor(AOIX,AOIY,ezx)
axis('equal')
shading('interp')
caxis(straincolor)
h1 = colorbar;
set(h1, 'PlotBoxAspectRatio',[2.0 10 8.0])
set(h1, 'FontSize', 12);
title('ZX Strain');

subplot(3,3,8)
pcolor(AOIX,AOIY,ezy)
axis('equal')
shading('interp')
caxis(straincolor)
h1 = colorbar;
set(h1, 'PlotBoxAspectRatio',[2.0 10 8.0])
set(h1, 'FontSize', 12);
title('ZY Strain');
print(plastic,'-dtiff','Plastic strain2.tif');

%ELASTIC TENSOR
minstrain=-0.035;
maxstrain=0.035;
elastic=figure
subplot(3,3,1)
set(pcolor(EBSDX,EBSDY,EBSDNormalsC(:,:,1)),'EdgeColor','none')
axis('equal')
straincolor(1)=minstrain;
straincolor(2)=maxstrain;
caxis(straincolor)
h1 = colorbar;
set(h1, 'PlotBoxAspectRatio',[2.0 10 8.0])
```

```matlab
set(h1, 'FontSize', 12);
title('X Strain');

subplot(3,3,2)
set(pcolor(EBSDX,EBSDY,EBSDShearsC(:,:,1)),'EdgeColor','none')
axis('equal')
caxis(straincolor)
h1 = colorbar;
set(h1, 'PlotBoxAspectRatio',[2.0 10 8.0])
set(h1, 'FontSize', 12);
title('XY Strain');

subplot(3,3,3)
set(pcolor(EBSDX,EBSDY,EBSDShearsC(:,:,3)),'EdgeColor','none')
axis('equal')
caxis(straincolor)
h1 = colorbar;
set(h1, 'PlotBoxAspectRatio',[2.0 10 8.0])
set(h1, 'FontSize', 12);
title('XZ Strain');

subplot(3,3,4)
set(pcolor(EBSDX,EBSDY,EBSDShearsC(:,:,1)),'EdgeColor','none')
axis('equal')
caxis(straincolor)
h1 = colorbar;
set(h1, 'PlotBoxAspectRatio',[2.0 10 8.0])
set(h1, 'FontSize', 12);
title('YX Strain');

subplot(3,3,5)
set(pcolor(EBSDX,EBSDY,EBSDNormalsC(:,:,2)),'EdgeColor','none')
axis('equal')
caxis(straincolor)
h1 = colorbar;
set(h1, 'PlotBoxAspectRatio',[2.0 10 8.0])
set(h1, 'FontSize', 12);
title('Y Strain');

subplot(3,3,6)
set(pcolor(EBSDX,EBSDY,EBSDShearsC(:,:,2)),'EdgeColor','none')
axis('equal')
caxis(straincolor)
h1 = colorbar;
set(h1, 'PlotBoxAspectRatio',[2.0 10 8.0])
set(h1, 'FontSize', 12);
title('YZ Strain');

subplot(3,3,7)
set(pcolor(EBSDX,EBSDY,EBSDShearsC(:,:,3)),'EdgeColor','none')
axis('equal')
caxis(straincolor)
h1 = colorbar;
set(h1, 'PlotBoxAspectRatio',[2.0 10 8.0])
set(h1, 'FontSize', 12);
title('ZX Strain');
```

```matlab
subplot(3,3,8)
set(pcolor(EBSDX,EBSDY,EBSDShearsC(:,:,2)),'EdgeColor','none')
axis('equal')
caxis(straincolor)
h1 = colorbar;
set(h1, 'PlotBoxAspectRatio',[2.0 10 8.0])
set(h1, 'FontSize', 12);
title('ZY Strain');

subplot(3,3,9)
set(pcolor(EBSDX,EBSDY,EBSDNormalsC(:,:,3)),'EdgeColor','none')
axis('equal')
caxis(straincolor)
h1 = colorbar;
set(h1, 'PlotBoxAspectRatio',[2.0 10 8.0])
set(h1, 'FontSize', 12);
title('Z Strain');

print(elastic,'-dtiff','Elastic Strain.tif');

minstrain=-10;
maxstrain=10;
StressFigure=figure;
subplot(3,3,1)
set(pcolor(EBSDX,EBSDY,EBSDStressC(:,:,1)),'EdgeColor','none')
axis('equal')
straincolor(1)=minstrain;
straincolor(2)=maxstrain;
caxis(straincolor)
h1 = colorbar;
set(h1, 'PlotBoxAspectRatio',[2.0 10 8.0])
set(h1, 'FontSize', 12);
title('X Stress');

subplot(3,3,2)
set(pcolor(EBSDX,EBSDY,EBSDStressC(:,:,4)),'EdgeColor','none')
axis('equal')
caxis(straincolor)
h1 = colorbar;
set(h1, 'PlotBoxAspectRatio',[2.0 10 8.0])
set(h1, 'FontSize', 12);
title('XY Stress');

subplot(3,3,3)
set(pcolor(EBSDX,EBSDY,EBSDStressC(:,:,6)),'EdgeColor','none')
axis('equal')
caxis(straincolor)
h1 = colorbar;
set(h1, 'PlotBoxAspectRatio',[2.0 10 8.0])
set(h1, 'FontSize', 12);
title('XZ Stress');

subplot(3,3,4)
set(pcolor(EBSDX,EBSDY,EBSDStressC(:,:,4)),'EdgeColor','none')
```

```matlab
axis('equal')
caxis(straincolor)
h1 = colorbar;
set(h1, 'PlotBoxAspectRatio',[2.0 10 8.0])
set(h1, 'FontSize', 12);
title('YX Stress');

subplot(3,3,5)
set(pcolor(EBSDX,EBSDY,EBSDStressC(:,:,2)),'EdgeColor','none')
axis('equal')
caxis(straincolor)
h1 = colorbar;
set(h1, 'PlotBoxAspectRatio',[2.0 10 8.0])
set(h1, 'FontSize', 12);
title('Y Stress');

subplot(3,3,6)
set(pcolor(EBSDX,EBSDY,EBSDStressC(:,:,5)),'EdgeColor','none')
axis('equal')
caxis(straincolor)
h1 = colorbar;
set(h1, 'PlotBoxAspectRatio',[2.0 10 8.0])
set(h1, 'FontSize', 12);
title('YZ Stress');

subplot(3,3,7)
set(pcolor(EBSDX,EBSDY,EBSDStressC(:,:,6)),'EdgeColor','none')
axis('equal')
caxis(straincolor)
h1 = colorbar;
set(h1, 'PlotBoxAspectRatio',[2.0 10 8.0])
set(h1, 'FontSize', 12);
title('ZX Stress');

subplot(3,3,8)
set(pcolor(EBSDX,EBSDY,EBSDStressC(:,:,5)),'EdgeColor','none')
axis('equal')
caxis(straincolor)
h1 = colorbar;
set(h1, 'PlotBoxAspectRatio',[2.0 10 8.0])
set(h1, 'FontSize', 12);
title('ZY Stress');

subplot(3,3,9)
set(pcolor(EBSDX,EBSDY,EBSDStressC(:,:,3)),'EdgeColor','none')
axis('equal')
caxis(straincolor)
h1 = colorbar;
set(h1, 'PlotBoxAspectRatio',[2.0 10 8.0])
set(h1, 'FontSize', 12);
title('Z Stress');

print(StressFigure,'-dtiff','Elastic Stress.tif');

VonMises=figure;
```

```matlab
set(pcolor(EBSDX,EBSDY,EBSDStressC(:,:,7)),'EdgeColor','none')
axis('equal')
straincolor(1)=0;
caxis(straincolor)
h1 = colorbar;
set(h1, 'PlotBoxAspectRatio',[2.0 10 8.0])
set(h1, 'FontSize', 12);
title('Von Mises Stress');

print(VonMises,'-dtiff','Von Mises Stress.tif');

estrainXX=EBSDNormalsC(:,:,1);
estrainYY=EBSDNormalsC(:,:,2);
estrainZZ=EBSDNormalsC(:,:,3);
estrainXY=EBSDShearsC(:,:,1);
estrainXZ=EBSDShearsC(:,:,3);
estrainYZ=EBSDShearsC(:,:,2);

estressXX=EBSDStressC(:,:,1);
estressYY=EBSDStressC(:,:,2);
estressZZ=EBSDStressC(:,:,3);
estressXY=EBSDStressC(:,:,4);
estressXZ=EBSDStressC(:,:,6);
estressYZ=EBSDStressC(:,:,5);

U=0.5*(estrainXX.*estressXX+estrainYY.*estressYY+estrainZZ.*estressZZ+e
strainXY.*estressXY+estrainXZ.*estressXZ+estrainYZ.*estressYZ);

straincolor(1)=-0;
straincolor(2)=.1;
h=figure;
set(pcolor(EBSDX,EBSDY,U),'EdgeColor','none')
axis('equal')
caxis(straincolor)
h1 = colorbar;
set(h1, 'PlotBoxAspectRatio',[2.0 10 8.0])
set(h1, 'FontSize', 12);
title('Elastic Stored Energy');
print(h,'-dtiff','Stored Elastic Energy.tif');


close all

end
```

Code end

Disp_GUI: This is the program used to analyze the results obtained previously through automated_image_DIC. This is used only for the DIC results, and does not incorporate the topographical data. The graphical user interface is shown below, and the code that interfaces it follows the image. This code was created by the author of this thesis
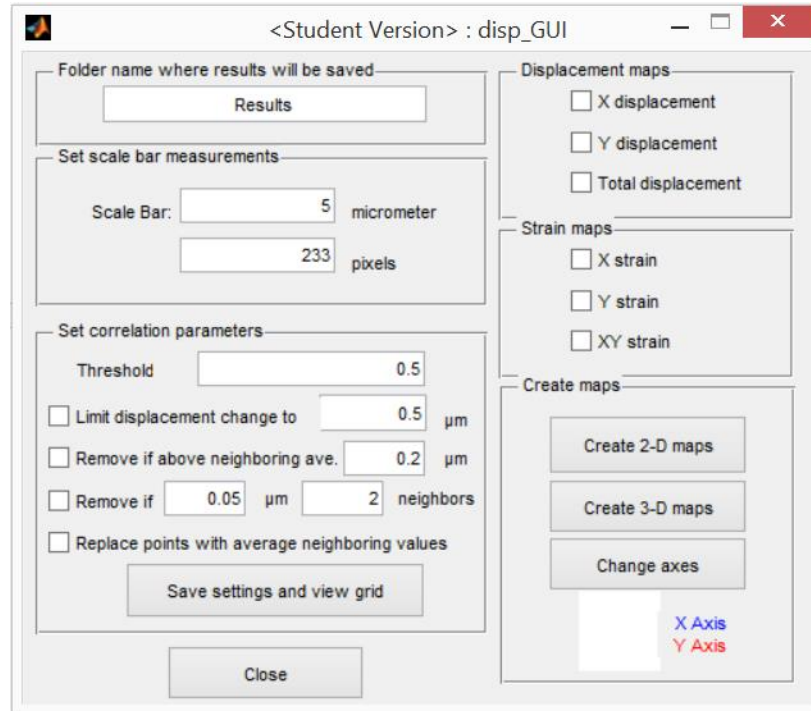
Figure A.1 The graphical user interface used by the Disp_GUI code.

Code starts

```
function varargout = disp_GUI(varargin)
% DISP_GUI MATLAB code for disp_GUI.fig
%       DISP_GUI, by itself, creates a new DISP_GUI or raises the existing
%       singleton*.
%
%       H = DISP_GUI returns the handle to a new DISP_GUI or the handle to
%       the existing singleton*.
%
%       DISP_GUI('CALLBACK',hObject,eventData,handles,...) calls the local
%       function named CALLBACK in DISP_GUI.M with the given input arguments.
%
%       DISP_GUI('Property','Value',...) creates a new DISP_GUI or raises the
%       existing singleton*.  Starting from the left, property value pairs are
%       applied to the GUI before disp_GUI_OpeningFcn gets called.  An
%       unrecognized property name or invalid value makes property application
```

```
%       stop.  All inputs are passed to disp_GUI_OpeningFcn via
varargin.
%
%       *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only
one
%       instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help disp_GUI

% Last Modified by GUIDE v2.5 03-Dec-2012 11:24:05

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @disp_GUI_OpeningFcn, ...
                   'gui_OutputFcn',  @disp_GUI_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end


% End initialization code - DO NOT EDIT


% --- Executes just before disp_GUI is made visible.
function disp_GUI_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to disp_GUI (see VARARGIN)

if exist('handles.validx')==0
    [validxname,Pathvalidx] = uigetfile('*.dat','Open validx.dat');
    if validxname==0
        disp('You did not select a file!')
        return
    end
    cd(Pathvalidx);
    validx=importdata(validxname,'\t');
end
if exist('handles.validy')==0
    %[validyname,Pathvalidy] = uigetfile('*.dat','Open validy.dat');
    %if validyname==0
```

```matlab
%     disp('You did not select a file!')
%       return
%end
%cd(Pathvalidy);
    validy=load('validy.dat');
end
grid_x=load('grid_x.dat');
grid_y=load('grid_y.dat');
valid_points=load('valid_points.dat');
threshold_values=load('threshold_values.dat');
validx=[grid_x validx];
validy=[grid_y validy];

%define the size of the data set
sizevalidx=size(validx);
sizevalidy=size(validy);
row_num=1;
while grid_x(row_num)==grid_x(row_num+1)
    row_num=row_num+1;
end

looppoints=sizevalidx(1,1);
loopimages=sizevalidx(1,2);

%calculate the displacement relative to the first image in x and y
%direction
clear displx;
validxfirst=zeros(size(validx));
validxfirst=mean(validx(:,1),2)*ones(1,sizevalidx(1,2));
displx=validx-validxfirst;
clear disply;
validyfirst=zeros(size(validy));
validyfirst=mean(validy(:,1),2)*ones(1,sizevalidy(1,2));
disply=validy-validyfirst;
displx(:,2)=displx(:,2)-mean(displx(:,2));  %Bring displacements to a
baseline, since pictures were not exactly lined up.
disply(:,2)=disply(:,2)-mean(disply(:,2));
validx=validxfirst+displx;
validy=validyfirst+disply;
clear validyfirst;
clear validxfirst;
handles.validx=validx;
handles.validy=validy;
handles.row_num=row_num;
handles.column_num=sizevalidx(1)/handles.row_num;
handles.displx=displx;
handles.disply=disply;
handles.original_validx=validx;
handles.original_validy=validy;
handles.original_displx=displx;
handles.original_disply=disply;
handles.threshold_values=threshold_values;
handles.original_threshold_values=threshold_values;
handles.grid_filename=[];
handles.grid_file_dir=[];
handles.invalid_x=[];
```

```matlab
handles.invalid_y=[];
handles.Xaxis=[1 0]; %All axis start from 0,0. First column gives end X
vector, second column gives end Y vector.
handles.Yaxis=[0 1];

% Choose default command line output for disp_GUI
handles.output = hObject;
set(handles.X_strain,'Value',1);
set(handles.X_dis,'Value',1);
set(handles.Y_strain,'Value',1);
set(handles.Y_dis,'Value',1);
set(handles.XY_strain,'Value',1);
set(handles.XY_dis,'Value',1);
set(handles.Max_dis_box,'Value',1);
set(handles.neighbors,'Value',1);


plot(handles.axes1,[0 handles.Xaxis(1)],[0 -handles.Xaxis(2)],'-b',
'LineWidth',2);
hold on;
plot(handles.axes1,[0 handles.Yaxis(1)],[0 -handles.Yaxis(2)],'-r',
'LineWidth',2);
xlim([-1 1]);
ylim([-1 1]);
axis off;
hold off;
% Update handles structure
guidata(hObject, handles);
apply_threshold(hObject, handles);

% UIWAIT makes disp_GUI wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = disp_GUI_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;


% --- Executes on button press in Close_button.
function Close_button_Callback(hObject, eventdata, handles)
% hObject    handle to Close_button (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
close all;

% --- Executes on button press in maps_2D.
function maps_2D_Callback(hObject, eventdata, handles)
% hObject    handle to maps_2D (see GCBO)
```

```matlab
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
plot_2D(handles);


% --- Executes on button press in maps_3D.
function maps_3D_Callback(hObject, eventdata, handles)
% hObject    handle to maps_3D (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
plot_3D(handles);


% --- Executes during object creation, after setting all properties.
function Threshold_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Threshold (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called


% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



% --- Executes on button press in View_corrected_grid.
function View_corrected_grid_Callback(hObject, eventdata, handles)
% hObject    handle to View_corrected_grid (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
apply_threshold(hObject, handles);
show_grid(hObject, handles);
%guidata(handles_in.output,handles_in);


% --- Executes during object creation, after setting all properties.
function SB_microns_CreateFcn(hObject, eventdata, handles)
% hObject    handle to SB_microns (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called


% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```matlab
% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes during object creation, after setting all properties.
function num_neighbors_CreateFcn(hObject, eventdata, handles)
% hObject    handle to num_neighbors (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in Change_axes.
function Change_axes_Callback(hObject, eventdata, handles)
% hObject    handle to Change_axes (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
Define_axes(handles);
handles=guidata(handles.output);
```

```matlab
plot(handles.axes1,[0 handles.Xaxis(1)],[0 -handles.Xaxis(2)],'-b',
'LineWidth',2);
hold on;
plot(handles.axes1,[0 handles.Yaxis(1)],[0 -handles.Yaxis(2)],'-r',
'LineWidth',2);
xlim([-1 1]);
ylim([-1 1]);
axis off;
hold off;




%*******************************************************************
%************************END OF GUI BUTTONS************************
%*******************************************************************

function plot_3D(handles)

scalebar_pix_str=get(handles.SB_pixels,'String');   %Conversion of
pixels to μm using scalebar
scalebar_pixels=str2double(scalebar_pix_str);
scalebar_mic_str=get(handles.SB_microns,'String');
scalebar_microns=str2double(scalebar_mic_str);
um_pixels=scalebar_microns/scalebar_pixels;

sizevalidx=size(handles.validx);
looppoints=sizevalidx(1,1);
loopimages=sizevalidx(1,2);
displtot=(handles.displx.^2+handles.disply.^2).^0.5;
gridsizex=10*round(min(min(handles.validx))/10):10:10*round(max(max(han
dles.validx))/10);
gridsizey=10*round(min(min(handles.validy))/10):10:10*round(max(max(han
dles.validy))/10);
[XI,YI]=meshgrid(gridsizex,gridsizey);
%ZI=griddata(handles.validx(:,1),handles.validy(:,1),displtot(:,1),XI,Y
I,'cubic');     %MIKE edit
%ZIsize=size(ZI);

displcolor = [-20 20];
straincolor = [-0.03 0.03];

maxminusminvalidx=(max(max(handles.validx))-min(min(handles.validx)));
maxminusminvalidy=(max(max(handles.validy))-min(min(handles.validy)));
%MIKE correction, used to be max(validx)

sizevalidx=size(handles.validx);
sizevalidy=size(handles.validy);
looppoints=sizevalidx(1,1);
loopimages=sizevalidx(1,2);

gridsizex=10*round(min(min(handles.validx(:,1)))/10):10:10*round(max(ma
x(handles.validx(:,1)))/10);
gridsizey=10*round(min(min(handles.validy(:,1)))/10):10:10*round(max(ma
x(handles.validy(:,1)))/10);
minminvalidx=min(min(handles.validx));
```

```matlab
maxmaxvalidx=max(max(handles.validx));
minminvalidy=min(min(handles.validy));
maxmaxvalidy=max(max(handles.validy));
minmindisplx=min(min(handles.displx));
maxmaxdisplx=max(max(handles.displx));
minmindisply=min(min(handles.disply));
maxmaxdisply=max(max(handles.disply));

    %ZI, ZIx, and ZIy are the displacement values (total, X and Y) over
the map area

ZI=griddata(handles.validx(:,1),handles.validy(:,1),displtot(:,2),XI,YI
,'cubic');       %MIKE edit change displx to displtot
    ZIsize=size(ZI);

ZIx=griddata(handles.validx(:,1),handles.validy(:,1),handles.displx(:,2
),XI,YI,'cubic');
    ZIxsize=size(ZIx);

ZIy=griddata(handles.validx(:,1),handles.validy(:,1),handles.disply(:,2
),XI,YI,'cubic');
    ZIysize=size(ZIy);


ZIx2=ZIx;  %Dot product of displacement with axes
ZIy2=ZIy;
ZIx=ZIx2*handles.Xaxis(1)+ZIy2*handles.Xaxis(2);
ZIy=ZIx2*handles.Yaxis(1)+ZIy2*handles.Yaxis(2);

    [epsxx, exy] =
gradient(ZIx,(maxminusminvalidx/ZIxsize(1,1)),(maxminusminvalidy/ZIxsiz
e(1,2)));  %MIKE edit
    [eyx, epsyy] =
gradient(ZIy,(maxminusminvalidx/ZIysize(1,1)),(maxminusminvalidy/ZIysiz
e(1,2)));
   % [epsxx, exy] = gradient(ZIx,.1,.1); %100 nm channel size
   %  [eyx, epsyy] = gradient(ZIy,.1,.1);
epsxx2=epsxx;  %Dot product of strain with axes
epsyy2=epsyy;
exy2=exy;
eyx2=eyx;
epsxx=epsxx2*handles.Xaxis(1)+exy*handles.Xaxis(2);
epsyy=eyx*handles.Yaxis(1)+epsyy2*handles.Yaxis(2);
exy=epsxx2*handles.Yaxis(1)+exy*handles.Yaxis(2);
eyx=eyx*handles.Xaxis(1)+epsyy2*handles.Xaxis(2);


    %Shear strain XY and YX should be equal, but due to rotation, they
may
    %not be.  This seperates the symetrical shear strain from the
    %anti-symetrical rotation.
    epsxy=0.5*(exy+eyx);
    epsyx=epsxy;
    rotxy=0.5*(exy-eyx);
    rotyx=0.5*(eyx-exy);
```

```matlab
    %The top left corner of the image is considered 0,0, and so when
the
    %data is graphed, the top left point of the image becomes the
bottom
    %left point of the graph (point 0,0 of the graph).  To fix this, YI
    %is inverted.
    YI_temp=YI;
    sizeYI=size(YI,1);
    for reverseYI=1:sizeYI
        from_end=sizeYI+1-reverseYI;
        YI(reverseYI,:)=YI_temp(from_end,:);
    end
    YI=YI*um_pixels;    %Converting X Y Z values to pixels
    XI=XI*um_pixels;
    ZI=ZI*um_pixels;
    ZIx=ZIx*um_pixels;
    ZIy=ZIy*um_pixels;

minstrain=min([min(epsxx) min(epsyy) min(epsxy)]);
maxstrain=max([max(epsxx) max(epsyy) max(epsxy)]);
minZ=min([min(ZIx) min(ZIy)]);
maxZ=max([max(ZIx) max(ZIy)]);


Warning_message=1;
if exist(get(handles.folder_name,'String'), 'dir')
    Warning_message = menu(sprintf('The folder %s already exists!
Similar map images will be overwritten',
get(handles.folder_name,'String')),'Continue','Cancel');
else
    mkdir(get(handles.folder_name,'String'));
end
if Warning_message==1
cd(get(handles.folder_name,'String'));
if get(handles.X_strain,'Value')==1
    h=figure;
    surf(XI,YI,epsxx); hold on
    axis('equal')
    %shading('interp')
    xlabel('x-position [µm]')
    ylabel('y-position [µm]')
    zlabel('strain xx')
    hold off
    title('X strain');
    print(h,'-dtiff','X_strain_3D.tif');
end
if get(handles.X_dis,'Value')==1
    h2=figure;
    surf(XI,YI,ZIx); hold on
    axis('equal')
    %shading('interp')
    xlabel('x-position [µm]')
    ylabel('y-position [µm]')
    zlabel('displacement x [µm]')
    hold off
    title('X displacement');
    print(h2,'-dtiff','X_dis_3D.tif');
```

```matlab
end
if get(handles.Y_strain,'Value')==1
    h3=figure;
    surf(XI,YI,epsyy); hold on
    axis('equal')
    %shading('interp')
    xlabel('x-position [µm]')
    ylabel('y-position [µm]')
    zlabel('strain yy')
    hold off
    title('Y strain');
    print(h3,'-dtiff','Y_strain_3D.tif');
end
if get(handles.Y_dis,'Value')==1
    h4=figure;
    surf(XI,YI,ZIy); hold on
    axis('equal')
    %shading('interp')
    xlabel('x-position [µm]')
    ylabel('y-position [µm]')
    zlabel('displacement y [µm]')
    hold off
    title('Y displacement');
    print(h4,'-dtiff','Y_dis_3D.tif');
end
if get(handles.XY_strain,'Value')==1
    h5=figure;
    surf(XI,YI,epsxy); hold on
    axis('equal')
    %shading('interp')
    xlabel('x-position [µm]')
    ylabel('y-position [µm]')
    zlabel('strain xy')
    hold off
    title('XY strain');
    print(h5,'-dtiff','XY_strain_3D.tif');
end
if get(handles.XY_dis,'Value')==1
    h6=figure;
    surf(XI,YI,ZI); hold on
    axis('equal')
    %shading('interp')
    xlabel('x-position [µm]')
    ylabel('y-position [µm]')
    zlabel('displacement total [µm]')
    hold off
    title('Total displacement');
    print(h6,'-dtiff','Total_dis_3D.tif');
end

Threshold_limit_str=get(handles.Threshold,'String'); %Create data file
dis_limit=[];
neigh_limit=[];
neigh_ave_limit=[];
if get(handles.Max_dis_box,'Value')==1
    dis_limit_str=get(handles.Max_dis,'String');
    dis_limit=['\r\nDisplacement limit is ' dis_limit_str];
```

```matlab
end
if get(handles.neighbors,'Value')==1
    neigh_limit_str=get(handles.dis_neighbors,'String');
    neigh_num_limit_str=get(handles.num_neighbors,'String');
    neigh_limit=['\r\nDisplacement limit is ' neigh_limit_str ' for '
neigh_num_limit_str ' neighbors'];
end
if get(handles.High_dis_box,'Value')==1
    neigh_ave_limit_str=get(handles.High_dis,'String');
    neigh_ave_limit=['\r\nDisplacement limit is ' neigh_ave_limit_str '
for average of neighboring values'];
end
if get(handles.Replace_bad,'Value')==1
    cleanup_type='Poorly correlated points were set to the average of
the nearest points that were correlated well.';
else
    cleanup_type='Poorly correlated points were removed.';
end
max_x=['\r\nMaximum X strain ' num2str(max(max(epsxx)))];
max_y=['\r\nMaximum Y strain ' num2str(max(max(epsyy)))];
max_xy=['\r\nMaximum XY strain ' num2str(max(max(epsxy)))];
min_x=['\r\nMinimum X strain ' num2str(min(min(epsxx)))];
min_y=['\r\nMinimum Y strain ' num2str(min(min(epsyy)))];
min_xy=['\r\nMinimum XY strain ' num2str(min(min(epsxy)))];
cleanup_info=[cleanup_type '\r\nThreshold limit is '
Threshold_limit_str dis_limit neigh_ave_limit neigh_limit '\r\n' max_x
min_x '\r\n' max_y min_y '\r\n' max_xy min_xy];
fileID=fopen('data.txt','w');
fprintf(fileID, cleanup_info);
fclose(fileID);

cd ..;
end


%*************************** 2D Graphs
**********************************
function plot_2D(handles)
scalebar_pix_str=get(handles.SB_pixels,'String');  %Conversion of
pixels to µm using scalebar
scalebar_pixels=str2double(scalebar_pix_str);
scalebar_mic_str=get(handles.SB_microns,'String');
scalebar_microns=str2double(scalebar_mic_str);
um_pixels=scalebar_microns/scalebar_pixels;

sizevalidx=size(handles.validx);
looppoints=sizevalidx(1,1);
loopimages=sizevalidx(1,2);
displtot=(handles.displx.^2+handles.disply.^2).^0.5;

spacingx=handles.validx(handles.row_num+1,1)-handles.validx(1,1);
spacingy=handles.validy(2,1)-handles.validy(1,1);
gridsizex=spacingx*round(min(min(handles.validx(:,1)))/spacingx):spacin
gx:spacingx*round(max(max(handles.validx(:,1)))/spacingx);
gridsizey=spacingy*round(min(min(handles.validy(:,1)))/spacingy):spacin
gy:spacingy*round(max(max(handles.validy(:,1)))/spacingy);
```

```matlab
%gridsizex=round(min(min(handles.validx(:,1)))):1:round(max(max(handles
.validx(:,1))));
%gridsizey=round(min(min(handles.validy(:,1)))):1:round(max(max(handles
.validy(:,1))));

[XI,YI]=meshgrid(gridsizex,gridsizey);
%ZI=griddata(handles.validx(:,1),handles.validy(:,1),displtot(:,1),XI,Y
I,'cubic');
%ZIsize=size(ZI);
displcolor = [-20 20];
straincolor = [-0.03 0.03];

maxminusminvalidx=(max(max(handles.validx))-min(min(handles.validx)));
%Is this wrong? could be causing it to get the spacing wrong, as it's
taking max of both columns, and min of both columns and subracting
maxminusminvalidy=(max(max(handles.validy))-min(min(handles.validy)));

ZI=griddata(handles.validx(:,1),handles.validy(:,1),displtot(:,2),XI,YI
,'cubic');
ZIsize=size(ZI);
ZIx=griddata(handles.validx(:,1),handles.validy(:,1),handles.displx(:,2
),XI,YI,'cubic');
ZIxsize=size(ZIx);
ZIy=griddata(handles.validx(:,1),handles.validy(:,1),handles.disply(:,2
),XI,YI,'cubic');
ZIysize=size(ZIy);

ZIx2=ZIx;    %Dot product of displacement with axis
ZIy2=ZIy;
ZIx=ZIx2*handles.Xaxis(1)+ZIy2*handles.Xaxis(2);
ZIy=ZIx2*handles.Yaxis(1)+ZIy2*handles.Yaxis(2);




[epsxx, exy] =
gradient(ZIx,(maxminusminvalidx/ZIxsize(1,1)),(maxminusminvalidy/ZIxsiz
e(1,2)));   %MIKE edit
[eyx, epsyy] =
gradient(ZIy,(maxminusminvalidx/ZIysize(1,1)),(maxminusminvalidy/ZIysiz
e(1,2)));
%nm=.05;   %uses 3 points when calculating slope, so it will use double
the spacing. To get a spacing of 100 nm, need to feed it 50 nm.
%    [epsxx, exy] = gradient(ZIx,nm,nm); %100 nm channel size
%    [eyx, epsyy] = gradient(ZIy,nm,nm);


YI_temp=YI;
sizeYI=size(YI,1);
for reverseYI=1:sizeYI
    from_end=sizeYI+1-reverseYI;
    YI(reverseYI,:)=YI_temp(from_end,:);
end
YI=YI*um_pixels;      %Converting X Y Z values from pixels into
micrometers
```

```matlab
XI=XI*um_pixels;
ZI=ZI*um_pixels;
ZIx=ZIx*um_pixels;
ZIy=ZIy*um_pixels;
X_columns=XI(1,:);
Y_rows=YI(:,1);


epsxx2=epsxx;    %Dot product of strain with axis
epsyy2=epsyy;
exy2=exy;
eyx2=eyx;
epsxx=epsxx2*handles.Xaxis(1)+exy*handles.Xaxis(2);
epsyy=eyx*handles.Yaxis(1)+epsyy2*handles.Yaxis(2);
exy=epsxx2*handles.Yaxis(1)+exy*handles.Yaxis(2);
eyx=eyx*handles.Xaxis(1)+epsyy2*handles.Xaxis(2);


epsxy=0.5*(exy+eyx);
epsyx=epsxy;
rotxy=0.5*(exy-eyx);
rotyx=0.5*(eyx-exy);


    %The top left corner of the image is considered 0,0, and so when
the
    %data is graphed, the top left point of the image becomes the
bottom
    %left point of the graph (point 0,0 of the graph).  To fix this, YI
    %is inverted.



minstrain=min([min(epsxx) min(epsyy) min(epsxy)]);
maxstrain=max([max(epsxx) max(epsyy) max(epsxy)]);
minZ=min([min(ZIx) min(ZIy)]);
maxZ=max([max(ZIx) max(ZIy)]);

Warning_message=1;
if exist(get(handles.folder_name,'String'), 'dir')
    Warning_message = menu(sprintf('The folder %s already exists!
Similar map images will be overwritten',
get(handles.folder_name,'String')),'Continue','Cancel');
else
    mkdir(get(handles.folder_name,'String'));
end
if Warning_message==1
cd(get(handles.folder_name,'String'));
if get(handles.X_strain,'Value')==1
    h=figure;
    pcolor(XI,YI,epsxx)
    axis('equal')
    shading('interp')
    straincolor(1)=minstrain;
    straincolor(2)=maxstrain;
    caxis(straincolor)
    h1 = colorbar;
    %rectangle('Position',[sb_posx, sb_posy, scale_pixels, 10],
'FaceColor','k');
    set(h1, 'PlotBoxAspectRatio',[2.0 10 8.0])
```

```matlab
    set(h1, 'FontSize', 12);
    title('X Strain');
    print(h,'-dtiff','X_strain_2D.tif');
    hhandle=get(h);
    haxes=hhandle.CurrentAxes;
end
if get(handles.X_dis,'Value')==1
    h2=figure;
    pcolor(XI,YI,ZIx)
    axis('equal')
    shading('interp')
    displcolor(1)=minZ;
    displcolor(2)=maxZ;
    caxis(displcolor)
    h1 = colorbar;
    %rectangle('Position',[sb_posx, sb_posy, scale_pixels, 10],
'FaceColor','k');
    set(h1, 'PlotBoxAspectRatio',[2.0 10 8.0])
    set(h1, 'FontSize', 12);
    title('X Displacement');
    print(h2,'-dtiff','X_dis_2D.tif');
end
if get(handles.Y_strain,'Value')==1
    h3=figure;
    pcolor(XI,YI,epsyy)
    axis('equal')
    shading('interp')
    straincolor(1)=minstrain;
    straincolor(2)=maxstrain;
    caxis(straincolor)
    h1 = colorbar;
    %rectangle('Position',[sb_posx, sb_posy, scale_pixels, 10],
'FaceColor','k');
    set(h1, 'PlotBoxAspectRatio',[2.0 10 8.0])
    set(h1, 'FontSize', 12);
    title('Y Strain');
    print(h3,'-dtiff','Y_strain_2D.tif');
    h3handle=get(h3);
    h3axes=h3handle.CurrentAxes;
end
if get(handles.Y_dis,'Value')==1
    h4=figure;
    pcolor(XI,YI,ZIy)
    axis('equal')
    shading('interp')
    displcolor(1)=minZ;
    displcolor(2)=maxZ;
    caxis(displcolor)
    h1 = colorbar;
    %rectangle('Position',[sb_posx, sb_posy, scale_pixels, 10],
'FaceColor','k');
    set(h1, 'PlotBoxAspectRatio',[2.0 10 8.0])
    set(h1, 'FontSize', 12);
    title('Y Displacement');
    print(h4,'-dtiff','Y_dis_2D.tif');
end
if get(handles.XY_strain,'Value')==1
```

```matlab
    h5=figure;
    pcolor(XI,YI,epsxy)
    axis('equal')
    shading('interp')
    straincolor(1)=minstrain;
    straincolor(2)=maxstrain;
    caxis(straincolor)
    h1 = colorbar;
    %rectangle('Position',[sb_posx, sb_posy, scale_pixels, 10],
'FaceColor','k');
    set(h1, 'PlotBoxAspectRatio',[2.0 10 8.0])
    set(h1, 'FontSize', 12);
    title('XY Shear Strain');
    print(h5,'-dtiff','XY_strain_2D.tif');
    h5handle=get(h5);
    h5axes=h5handle.CurrentAxes;
end
if get(handles.XY_dis,'Value')==1
    h6=figure;
    pcolor(XI,YI,ZI)
    axis('equal')
    shading('interp')
    displcolor(1)=min(min(ZI));
    displcolor(2)=max(max(ZI));
    caxis(displcolor)
    h1 = colorbar;
    %rectangle('Position',[sb_posx, sb_posy, scale_pixels, 10],
'FaceColor','k');
    set(h1, 'PlotBoxAspectRatio',[2.0 10 8.0])
    set(h1, 'FontSize', 12);
    title('Total Displacement');
    print(h6,'-dtiff','Total_dis_2D.tif');
end
save('DICData.mat','ZIx','ZIy','ZI','epsxx','exy','eyx','epsyy','XI','Y
I');
h7=figure('visible','off');
plot([0 handles.Xaxis(1)],[0 -handles.Xaxis(2)],'-b', 'LineWidth',2);
hold on;
plot([0 handles.Yaxis(1)],[0 -handles.Yaxis(2)],'-r', 'LineWidth',2);
xlim([-1 1]);
ylim([-1 1]);
axis('equal');
axis off;
hold off;
print(h7,'-dtiff','axes.tif');
Threshold_limit_str=get(handles.Threshold,'String'); %Create data file
dis_limit=[];
neigh_limit=[];
neigh_ave_limit=[];
if get(handles.Max_dis_box,'Value')==1
    dis_limit_str=get(handles.Max_dis,'String');
    dis_limit=['\r\nDisplacement limit is ' dis_limit_str];
end
if get(handles.neighbors,'Value')==1
    neigh_limit_str=get(handles.dis_neighbors,'String');
    neigh_num_limit_str=get(handles.num_neighbors,'String');
```

```matlab
    neigh_limit=['\r\nDisplacement limit is ' neigh_limit_str ' for '
neigh_num_limit_str ' neighbors'];
end
if get(handles.High_dis_box,'Value')==1
    neigh_ave_limit_str=get(handles.High_dis,'String');
    neigh_ave_limit=['\r\nDisplacement limit is ' neigh_ave_limit_str '
for average of neighboring values'];
end
if get(handles.Replace_bad,'Value')==1
    cleanup_type='Poorly correlated points were set to the average of
the nearest points that were correlated well.';
else
    cleanup_type='Poorly correlated points were removed.';
end
strain_q=0;
while strain_q~=4
strain_q=menu(sprintf('Examine strain in a section of the maps?'),...
%get numerical strain values
'Examine average strain','Examine strain (manual)', 'Examine strain at
intersection (polar coordinates)', 'End');
data_num=0;
if strain_q==1
    if get(handles.X_strain,'Value')==1
        figure(h)
    end
    if get(handles.Y_strain,'Value')==1
        figure(h3)
    end
    if get(handles.XY_strain,'Value')==1
        figure(h5)
    end
data_num=data_num+1;
[x(1,1),y(1,1)]=ginput(1);
hold on
plot(x(1,1),y(1,1),'+b')


[x(2,1),y(2,1)]=ginput(1);
hold on
plot(x(2,1),y(2,1),'+b')


drawnow

xmin = min(x);
xmax = max(x);
ymin = min(y);
ymax = max(y);

lowerline=[xmin ymin; xmax ymin];
upperline=[xmin ymax; xmax ymax];
leftline=[xmin ymin; xmin ymax];
rightline=[xmax ymin; xmax ymax];

plot(lowerline(:,1),lowerline(:,2),'-k')
plot(upperline(:,1),upperline(:,2),'-k')
plot(leftline(:,1),leftline(:,2),'-k')
plot(rightline(:,1),rightline(:,2),'-k')
```

```
start_x=1;
start_y=1;

while X_columns(start_x)<xmin
    start_x=start_x+1;
end
while Y_rows(start_y)>ymax
    start_y=start_y+1;
end
end_x=start_x+1;
end_y=start_y+1;
while X_columns(end_x+1)<xmax  %NEEDS TO BE MODIFIED IN CASE USER
SELECTS EDGE
    end_x=end_x+1;
end
while Y_rows(end_y+1)>ymin
    end_y=end_y+1;
end
vector_size=(end_x-start_x+1)*(end_y-start_y+1);
epsxxvector=reshape(epsxx(start_y:end_y,start_x:end_x),vector_size,1);
epsyyvector=reshape(epsyy(start_y:end_y,start_x:end_x),vector_size,1);
epsxyvector=reshape(epsxy(start_y:end_y,start_x:end_x),vector_size,1);
limit_x=abs(mean(epsxxvector))+abs(std(epsxxvector));  %if abs(strain)
is greater than the limit, it is considered part of the channel when
calculating channel average strain
limit_y=abs(mean(epsyyvector))+abs(std(epsyyvector));
limit_xy=abs(mean(epsxyvector))+abs(std(epsxyvector));
num_avex=0;
ave_totx=0;
num_avey=0;
ave_toty=0;
num_avexy=0;
ave_totxy=0;
num_data=[epsxx(start_y,start_x) epsxx(start_y,start_x)
epsyy(start_y,start_x) epsyy(start_y,start_x) epsxy(start_y,start_x)
epsxy(start_y,start_x) ZIx(start_y,start_x) ZIx(start_y,start_x)
ZIy(start_y,start_x) ZIy(start_y,start_x)];
for x_check=start_x:end_x   %num_data: 1,2: max,min x strain; 3,4
max,min y strain; 5,6 max,min xy strain 7,8 max,min x disp; 9,10
max,min y disp
    for y_check=start_y+1:end_y
        if epsxx(y_check,x_check)>num_data(1)  %In future could add
markers to show where max and min strain were found
            num_data(1)=epsxx(y_check,x_check);
        end
        if epsxx(y_check,x_check)<num_data(2)
            num_data(2)=epsxx(y_check,x_check);
        end
        if epsyy(y_check,x_check)>num_data(3)
            num_data(3)=epsyy(y_check,x_check);
        end
        if epsyy(y_check,x_check)<num_data(4)
            num_data(4)=epsyy(y_check,x_check);
        end
        if epsxy(y_check,x_check)>num_data(5)
```

```matlab
                num_data(5)=epsxy(y_check,x_check);
            end
            if epsxy(y_check,x_check)<num_data(6)
                num_data(6)=epsxy(y_check,x_check);
            end
            if ZIx(y_check,x_check)>num_data(7)
                num_data(7)=ZIx(y_check,x_check);
            end
            if ZIx(y_check,x_check)<num_data(8)
                num_data(8)=ZIx(y_check,x_check);
            end
            if ZIy(y_check,x_check)>num_data(9)
                num_data(9)=ZIy(y_check,x_check);
            end
            if ZIy(y_check,x_check)<num_data(10)
                num_data(10)=ZIy(y_check,x_check);
            end

            if abs(epsxx(y_check,x_check))>limit_x   %Average strain in
channel
                num_avex=num_avex+1;
                ave_totx=ave_totx+epsxx(y_check,x_check);
                if get(handles.X_strain,'Value')==1
                    %figure(h);
                    hold(haxes, 'on');
                    plot(haxes,XI(1,x_check),YI(y_check),'*k')
                    %drawnow;
                end
            end
            if abs(epsyy(y_check,x_check))>limit_y
                num_avey=num_avey+1;
                ave_toty=ave_toty+epsyy(y_check,x_check);
                if get(handles.Y_strain,'Value')==1
                    %figure(h3);
                    hold(h3axes, 'on');
                    plot(h3axes,XI(1,x_check),YI(y_check),'*k')
                    %drawnow;
                end
            end
            if abs(epsxy(y_check,x_check))>limit_xy
                num_avexy=num_avexy+1;
                ave_totxy=ave_totxy+epsxy(y_check,x_check);
                if get(handles.XY_strain,'Value')==1
                    %figure(h5)
                    hold(h5axes, 'on');
                    plot(h5axes,XI(1,x_check),YI(y_check),'*k')
                    %drawnow;
                end
            end
        end
    end
end

average_x=ave_totx/num_avex;
average_y=ave_toty/num_avey;
average_xy=ave_totxy/num_avexy;
```

```matlab
average_x_result=['\r\nAverage X strain in area '
num2str(mean(epsxxvector))];   %Save data to txt file
average_y_result=['\r\nAverage Y strain in area '
num2str(mean(epsyyvector))];
average_xy_result=['\r\nAverage XY strain in area '
num2str(mean(epsxyvector))];
stdv_x_result=['\r\nSt dev of X strain in area '
num2str(std(epsxxvector))];
stdv_y_result=['\r\nSt dev of Y strain in area '
num2str(std(epsyyvector))];
stdv_xy_result=['\r\nSt dev of XY strain in area '
num2str(std(epsxyvector))];
max_x=['\r\nMaximum X strain ' num2str(num_data(1))];
max_y=['\r\nMaximum Y strain ' num2str(num_data(3))];
max_xy=['\r\nMaximum XY strain ' num2str(num_data(5))];
min_x=['\r\nMinimum X strain ' num2str(num_data(2))];
min_y=['\r\nMinimum Y strain ' num2str(num_data(4))];
min_xy=['\r\nMinimum XY strain ' num2str(num_data(6))];
ave_xstr=['\r\nAverage X strainin channel ' num2str(average_x)];
ave_ystr=['\r\nAvereage Y strain in channel ' num2str(average_y)];
ave_xystr=['\r\nAverage XY strain in channel ' num2str(average_xy)];
cleanup_info=[cleanup_type '\r\nThreshold limit is '
Threshold_limit_str dis_limit neigh_ave_limit neigh_limit '\r\n'
average_x_result stdv_x_result max_x min_x ave_xstr '\r\n'
average_y_result stdv_y_result max_y min_y ave_ystr '\r\n'
average_xy_result stdv_xy_result max_xy min_xy ave_xystr];
file_name=['data ' num2str(data_num)];
prompt = {'Enter filename (do not include extensions, will be saved as
a .txt)'};
dlg_title = 'File name for data storage';
num_lines= 1;
def = {file_name};
file_name = char(inputdlg(prompt,dlg_title,num_lines,def));
file_name1=[file_name ' summary.txt'];
fileID=fopen(file_name1,'w');
fprintf(fileID, cleanup_info);
fclose(fileID);
file_name2=[file_name ' X strain.txt'];
%fileID=fopen(file_name2,'w');
saveepsxx=epsxx(start_y:end_y,start_x:end_x);
saveepsyy=epsyy(start_y:end_y,start_x:end_x);
saveepsxy=epsxy(start_y:end_y,start_x:end_x);
save(file_name2, 'saveepsxx','-ASCII','-double','-tabs');
%fclose(fileID);
file_name3=[file_name ' Y strain.txt'];
%fileID=fopen(file_name3,'w');
save(file_name3, 'saveepsyy','-ASCII','-double','-tabs');
%fclose(fileID);
file_name4=[file_name ' XY strain.txt'];
%fileID=fopen(file_name4,'w');
save(file_name4, 'saveepsxy','-ASCII','-double','-tabs');
%fclose(fileID);
end

if strain_q==2
    strainmap=figure;
    pcolor(XI,YI,ZI)
```

```matlab
    axis('equal')
    displcolor(1)=min(min(ZI));
    displcolor(2)=max(max(ZI));
    caxis(displcolor)
    h1 = colorbar;
    set(h1, 'PlotBoxAspectRatio',[2.0 10 8.0])
    set(h1, 'FontSize', 12);
    title('Total Displacement');

    for i=1:3    %3 pairs of points are taken, and change of
displacement determined.  The average of the 3 is used to determine
strain
    [x(i,1),y(i,1)]=ginput(1);
    hold on
    plot(x(i,1),y(i,1),'+b')
    [x(i,2),y(i,2)]=ginput(1);
    hold on
    plot(x(i,2),y(i,2),'+b')
    drawnow
    rangex=(XI(1,2)-XI(1,1))/2;
    rangey=(YI(1,1)-YI(2,1))/2;
    [toss,findx(i,1)]=find((x(i,1)-rangex)<XI & XI<(x(i,1)+rangex),1);
    [findy(i,1),toss]=find((y(i,1)-rangey)<YI & YI<(y(i,1)+rangey),1);
    [toss,findx(i,2)]=find((x(i,2)-rangex)<XI & XI<(x(i,2)+rangex),1);
    [findy(i,2),toss]=find((y(i,2)-rangey)<YI & YI<(y(i,2)+rangey),1);
    end
    strain3=[(ZI(findy(1,2),findx(1,2))-ZI(findy(1,1),findx(1,1)))
(ZI(findy(2,2),findx(2,2))-ZI(findy(2,1),findx(2,1)))
(ZI(findy(3,2),findx(3,2))-ZI(findy(3,1),findx(3,1)))]/.1;
    strain=(strain3(1)+strain3(2)+strain3(3))/3;
    displacement_x=[(ZIx(findy(1,2),findx(1,2))-
ZIx(findy(1,1),findx(1,1)))  (ZIx(findy(2,2),findx(2,2))-
ZIx(findy(2,1),findx(2,1)))  (ZIx(findy(3,2),findx(3,2))-
ZIx(findy(3,1),findx(3,1)))];
    displacement_y=[(ZIy(findy(1,2),findx(1,2))-
ZIy(findy(1,1),findx(1,1)))  (ZIy(findy(2,2),findx(2,2))-
ZIy(findy(2,1),findx(2,1)))  (ZIy(findy(3,2),findx(3,2))-
ZIy(findy(3,1),findx(3,1)))];
    prompt = {'Enter filename (do not include extensions, will be saved
as a .txt)'};
    dlg_title = 'File name for data storage';
    num_lines= 1;
    def = {'strain'};
    file_name = char(inputdlg(prompt,dlg_title,num_lines,def));
    file_name1=[file_name 'strain.txt'];
    fileID=fopen(file_name1,'w');
    fprintf(fileID, '%f', strain);
    fclose(fileID);
    file_name2=[file_name 'displacementX.txt'];
    fileID2=fopen(file_name2,'w');
    fprintf(fileID2, '%f\r\n', displacement_x);
    fclose(fileID2);
    file_name3=[file_name 'displacementY.txt'];
    fileID3=fopen(file_name3,'w');
    fprintf(fileID3, '%f\r\n', displacement_y);
    fclose(fileID3);
end
```

```matlab
if strain_q==3
    if get(handles.X_strain,'Value')==1
        figure(h)
    end
    if get(handles.Y_strain,'Value')==1
        figure(h3)
    end
    if get(handles.XY_strain,'Value')==1
        figure(h5)
    end
    [x,y]=ginput(1);
    hold on
    plot(x,y,'+b')
    drawnow

[theta,rho]=meshgrid(0:1:359,0:0.05:2);
theta=theta*pi()/180;
x_polar=x+cos(theta).*rho;
y_polar=y+sin(theta).*rho;
epsxx_polar=griddata(XI,YI,epsxx,x_polar,y_polar,'cubic');
epsyy_polar=griddata(XI,YI,epsyy,x_polar,y_polar,'cubic');
epsxy_polar=griddata(XI,YI,epsxy,x_polar,y_polar,'cubic');
theta=theta*180/pi();

p1=figure;
p2=figure;
p3=figure;
removal1=0;
while removal1~=4
minstrain_polar=min([min(epsxx_polar) min(epsyy_polar)
min(epsxy_polar)]);
maxstrain_polar=max([max(epsxx_polar) max(epsyy_polar)
max(epsxy_polar)]);
straincolor_polar(1)=minstrain_polar;
straincolor_polar(2)=maxstrain_polar;
figure(p1)
pcolor(theta,rho,epsxx_polar)
shading('interp')
caxis(straincolor_polar)
c1 = colorbar;
set(c1, 'PlotBoxAspectRatio',[2.0 10 8.0])
set(c1, 'FontSize', 12);
title('X strain polar coordinates');
figure(p2)
pcolor(theta,rho,epsyy_polar)
shading('interp')
caxis(straincolor_polar)
c1 = colorbar;
set(c1, 'PlotBoxAspectRatio',[2.0 10 8.0])
set(c1, 'FontSize', 12);
title('Y strain polar coordinates');
figure(p3)
pcolor(theta,rho,epsxy_polar)
shading('interp')
caxis(straincolor_polar)
c1 = colorbar;
```

```matlab
set(c1, 'PlotBoxAspectRatio',[2.0 10 8.0])
set(c1, 'FontSize', 12);
title('XY strain polar coordinates');

removal1=menu(sprintf('Remove areas from the polar graph? (If you do
not want to include DC or GB)'),...
'Remove entire theta columns','Remove entire rho rows', 'Remove a
section', 'End');
%data_num=0;
if removal1<4    %all three of these start the same way
    figure(p1)
    [x(1,1),y(1,1)]=ginput(1);
    hold on
    plot(x(1,1),y(1,1),'+b')
    [x(2,1),y(2,1)]=ginput(1);
    hold on
    plot(x(2,1),y(2,1),'+b')
    drawnow
    hold off;
    if removal1==1
        [toss y1]=find(theta>min(x) & theta<max(x));
        epsxx_polar(:,min(y1):max(y1))=NaN;
        epsyy_polar(:,min(y1):max(y1))=NaN;
        epsxy_polar(:,min(y1):max(y1))=NaN;
    end
    if removal1==2
        [x1 toss]=find(rho>min(y) & rho<max(y));
        epsxx_polar(min(x1):max(x1),:)=NaN;
        epsyy_polar(min(x1):max(x1),:)=NaN;
        epsxy_polar(min(x1):max(x1),:)=NaN;
    end
    if removal1==3
        [x1 y1]=find(rho>min(y) & rho<max(y) & theta>min(x) &
theta<max(x));
        epsxx_polar(min(x1):max(x1),min(y1):max(y1))=NaN;
        epsyy_polar(min(x1):max(x1),min(y1):max(y1))=NaN;
        epsxy_polar(min(x1):max(x1),min(y1):max(y1))=NaN;
    end
end


end
[max_epsxx_r,I1]=max(abs(epsxx_polar),[],2);
Isub1=sub2ind(size(epsxx_polar),(1:size(rho,1))',I1);    %Need to make
sure if negative strain was largest, that strain value is set to the
negative value
max_epsxx_r=epsxx_polar(Isub1);
thetax_max=theta(Isub1);
[max_epsyy_r,I2]=max(abs(epsyy_polar),[],2);
Isub2=sub2ind(size(epsyy_polar),(1:size(rho,1))',I2);
max_epsyy_r=epsyy_polar(Isub2);
thetay_max=theta(Isub2);
[max_epsxy_r,I3]=max(abs(epsxy_polar),[],2);
Isub3=sub2ind(size(epsxy_polar),(1:size(rho,1))',I3);
max_epsxy_r=epsxy_polar(Isub3);
thetaxy_max=theta(Isub3);
```

```matlab
p4=figure;
hold on
scatter(rho(:,1),max_epsxx_r);
title('Max X strain vs rho (µm)');
p5=figure;
hold on
scatter(rho(:,1),max_epsyy_r);
title('Max Y strain vs rho (µm)');
p6=figure;
hold on
scatter(rho(:,1),max_epsxy_r);
title('Max XY strain vs rho (µm)');
p7=figure;
hold on
scatter(thetax_max,rho(:,1));
xlim([0 360]);
title('rho (µm) vs theta of max X strain');
p8=figure;
hold on
scatter(thetay_max,rho(:,1));
xlim([0 360]);
title('rho (µm) vs theta of max Y strain');
p9=figure;
hold on
scatter(thetaxy_max,rho(:,1));
xlim([0 360]);
title('rho (µm) vs theta of max XY strain');

fileID=fopen('Exx_vs_r.txt','w');          %Saving maximum strain from
each rho
fprintf(fileID, '%f\r\n', max_epsxx_r);
fclose(fileID);
fileID=fopen('Eyy_vs_r.txt','w');
fprintf(fileID, '%f\r\n',max_epsyy_r);
fclose(fileID);
fileID=fopen('Exy_vs_r.txt','w');
fprintf(fileID, '%f\r\n',max_epsxy_r);
fclose(fileID);
fileID=fopen('Thetax_vs_r.txt','w');        %Saving angle of maximum
strain from each rho
fprintf(fileID, '%f\r\n', thetax_max);
fclose(fileID);
fileID=fopen('Thetay_vs_r.txt','w');
fprintf(fileID, '%f\r\n',thetay_max);
fclose(fileID);
fileID=fopen('Thetaxy_vs_r.txt','w');
fprintf(fileID, '%f\r\n',thetaxy_max);
fclose(fileID);
print(p4,'-dtiff','Exx_vs_r.tif');
print(p5,'-dtiff','Eyy_vs_r.tif');
print(p6,'-dtiff','Exy_vs_r.tif');
print(p7,'-dtiff','thetax_vs_r.tif');
print(p8,'-dtiff','thetay_vs_r.tif');
print(p9,'-dtiff','thetaxy_vs_r.tif');
end
end
cd ..
```

```matlab
    end


function apply_threshold(hObject, handles)
handles=guidata(handles.output);
rem_count=1;
removal=[];
Threshold_limit_str=get(handles.Threshold,'String');
Threshold_limit=str2double(Threshold_limit_str);
handles.validx=handles.original_validx;
handles.validy=handles.original_validy;
handles.displx=handles.original_displx;
handles.disply=handles.original_disply;
handles.threshold_values=handles.original_threshold_values;
displx_grid=reshape(handles.displx(:,2),handles.row_num,handles.column_
num);
disply_grid=reshape(handles.disply(:,2),handles.row_num,handles.column_
num);
temp_displx=handles.displx;
temp_disply=handles.disply;
rem_ind=0;
handles.invalid_x=[];
handles.invalid_y=[];
    scalebar_pix_str=get(handles.SB_pixels,'String');  %Conversion of
pixels to µm using scalebar
    scalebar_pixels=str2double(scalebar_pix_str);
    scalebar_mic_str=get(handles.SB_microns,'String');
    scalebar_microns=str2double(scalebar_mic_str);
    um_pixels=scalebar_microns/scalebar_pixels;
    displtot=(handles.displx(:,2).^2+handles.disply(:,2).^2).^0.5;
if get(handles.Max_dis_box,'Value')==1
    dis_limit_str=get(handles.Max_dis,'String');
    dis_limit=str2double(dis_limit_str);
    dis_limit=dis_limit/um_pixels;      %Change limit from µm to pixels
    for dis_count=1:size(displtot,1)
        if displtot(dis_count)>dis_limit
            handles.threshold_values(dis_count)=-1;
        end
    end

end

if (get(handles.High_dis_box,'Value')==1) ||
(get(handles.neighbors,'Value')==1)
    highdis_count=0;
    disp_total_grid=(displx_grid.^2+disply_grid.^2).^0.5;
    High_dis_limit_str=get(handles.High_dis,'String');
    High_dis_limit=str2double(High_dis_limit_str);
    neighbor_limit_str=get(handles.num_neighbors,'String');
    neighbor_limit=str2double(neighbor_limit_str);
    neighbor_value_limit_str=get(handles.dis_neighbors,'String');
    neighbor_value_limit=str2double(neighbor_value_limit_str);
    High_dis_limit=High_dis_limit/um_pixels;      %Change limit from µm
to pixels
    neighbor_value_limit=neighbor_value_limit/um_pixels;
    for col_check=1:handles.column_num
```

276

```matlab
        for row_check=1:handles.row_num
            num_high_dis=0;
            ave_disp=0;
            num_neigh=0;
            highdis_count=highdis_count+1;
            if col_check>1  %Check left
                if get(handles.neighbors,'Value')==1
                if abs(disp_total_grid(row_check,col_check)-
disp_total_grid(row_check,col_check-1))>neighbor_value_limit
                    num_high_dis=num_high_dis+1;
                end
                end
                num_neigh=num_neigh+1;
                ave_disp=ave_disp+disp_total_grid(row_check,col_check-
1);
            end
            if col_check<handles.column_num      %Check right
                if get(handles.neighbors,'Value')==1
                if abs(disp_total_grid(row_check,col_check)-
disp_total_grid(row_check,col_check+1))>neighbor_value_limit
                    num_high_dis=num_high_dis+1;
                end
                end
                num_neigh=num_neigh+1;

ave_disp=ave_disp+disp_total_grid(row_check,col_check+1);
            end
            if row_check>1  %Check above
                if get(handles.neighbors,'Value')==1
                if abs(disp_total_grid(row_check,col_check)-
disp_total_grid(row_check-1,col_check))>neighbor_value_limit
                    num_high_dis=num_high_dis+1;
                end
                end
                num_neigh=num_neigh+1;
                ave_disp=ave_disp+disp_total_grid(row_check-
1,col_check);
            end
            if row_check<handles.row_num      %Check below
                if get(handles.neighbors,'Value')==1
                if abs(disp_total_grid(row_check,col_check)-
disp_total_grid(row_check+1,col_check))>neighbor_value_limit
                    num_high_dis=num_high_dis+1;
                end
                end
                num_neigh=num_neigh+1;

ave_disp=ave_disp+disp_total_grid(row_check+1,col_check);
            end

            if (get(handles.neighbors,'Value')==1)
                if num_high_dis>=neighbor_limit
                    handles.threshold_values(highdis_count)=-1;
                end
            end
            if (get(handles.High_dis_box,'Value')==1)
                ave_disp=ave_disp/num_neigh;
```

```matlab
                if abs(disp_total_grid(row_check,col_check)-
ave_disp)>High_dis_limit
                    handles.threshold_values(highdis_count)=-1;
                end
            end
        end
    end

end

threshold_grid=reshape(handles.threshold_values,handles.row_num,handles
.column_num);
for col_check=1:handles.column_num
    for row_check=1:handles.row_num
        num_neighbors=0;
        displx_neighbors=0;
        disply_neighbors=0;
        min_dist=0;
        top=0;
        bottom=0;
        left=0;
        right=0;
        rem_ind=rem_ind+1;
        if (handles.threshold_values(rem_ind) < Threshold_limit)
%Removes points that did not correlate well
            if get(handles.Replace_bad,'Value')~=1
                removal(rem_count)=rem_ind;
                handles.invalid_x(rem_count)=handles.validx(rem_ind,1);
                handles.invalid_y(rem_count)=handles.validy(rem_ind,1);
                rem_count=rem_count+1;
            else
            current_col=col_check;
            current_row=row_check;
            while (current_row>1) &&
(threshold_grid(current_row,current_col)<Threshold_limit) %Find nearest
good point above current point.
                current_row=current_row-1;
                top=top+1;
            end
            if threshold_grid(current_row,current_col)>=Threshold_limit
                min_dist=top;
            else
                top=0;
            end
            current_row=row_check;

            while (current_row<handles.row_num) &&
(threshold_grid(current_row,current_col)<Threshold_limit) %Find nearest
good point below current point.
                current_row=current_row+1;
                bottom=bottom+1;
            end
            if threshold_grid(current_row,current_col)>=Threshold_limit
                if min_dist~=0
                    min_dist=min([bottom min_dist]);
                else
```

```
                    min_dist=bottom;
                end
            else
                bottom=0;
            end
            current_row=row_check;

            while (current_col<handles.column_num) &&
(threshold_grid(current_row,current_col)<Threshold_limit) %Find nearest
good point right of current point.
                current_col=current_col+1;
                right=right+1;
            end
            if threshold_grid(current_row,current_col)>=Threshold_limit
                if min_dist~=0
                    min_dist=min([right min_dist]);
                else
                    min_dist=right;
                end
            else
                right=0;
            end
            current_col=col_check;

            while (current_col>1) &&
(threshold_grid(current_row,current_col)<Threshold_limit) %Find nearest
good point left of current point.
                current_col=current_col-1;
                left=left+1;
            end
            if threshold_grid(current_row,current_col)>=Threshold_limit
                if min_dist~=0
                    min_dist=min([left min_dist]);
                else
                    min_dist=left;
                end
            else
                left=0;
            end
            current_col=col_check;

            if min_dist~=0
            if top==min_dist
                num_neighbors=num_neighbors+1;

displx_neighbors=displx_neighbors+displx_grid(row_check-
min_dist,col_check);

disply_neighbors=disply_neighbors+disply_grid(row_check-
min_dist,col_check);
            end
            if bottom==min_dist
                num_neighbors=num_neighbors+1;

displx_neighbors=displx_neighbors+displx_grid(row_check+min_dist,col_ch
eck);
```

```matlab
disply_neighbors=disply_neighbors+disply_grid(row_check+min_dist,col_ch
eck);
            end
            if left==min_dist
                num_neighbors=num_neighbors+1;

displx_neighbors=displx_neighbors+displx_grid(row_check,col_check-
min_dist);

disply_neighbors=disply_neighbors+disply_grid(row_check,col_check-
min_dist);
            end
            if right==min_dist
                num_neighbors=num_neighbors+1;

displx_neighbors=displx_neighbors+displx_grid(row_check,col_check+min_d
ist);

disply_neighbors=disply_neighbors+disply_grid(row_check,col_check+min_d
ist);
            end
            else
                warning('No good value was determined for bad point');
            end

            if num_neighbors~=0
                displx_neighbors=displx_neighbors/num_neighbors;
                disply_neighbors=disply_neighbors/num_neighbors;
                temp_displx(rem_ind,2)=displx_neighbors;
                temp_disply(rem_ind,2)=disply_neighbors;
                if col_check==17
                    if row_check==2
                    end
                end

            else
                warning('No good value was determined for bad point');
            end
            end
        end
    end
end

if get(handles.Replace_bad,'Value')==1
    handles.displx=temp_displx;
    handles.disply=temp_disply;
    handles.validx(:,2)=handles.validx(:,1)+handles.displx(:,2);
    handles.validy(:,2)=handles.validy(:,1)+handles.disply(:,2);
else
    handles.validx(removal(:),:)=[];
    handles.validy(removal(:),:)=[];
    handles.displx(removal(:),:)=[];
    handles.disply(removal(:),:)=[];
end
guidata(handles.output, handles);
```

```
function show_grid(hObject, handles)
handles=guidata(handles.output);
if exist(handles.grid_file_dir, 'dir')==0
    [handles.grid_filename,handles.grid_file_dir,FilterIndex] =
uigetfile( ...
    {'*.bmp;*.tif;*.jpg;*.TIF;*.BMP;*.JPG','Image files
(*.bmp,*.tif,*.jpg)';'*.*',   'All Files (*.*)'}, ...
    'Open base image for grid creation');
end
old_folder=cd(handles.grid_file_dir);
h=figure;
imshow(handles.grid_filename);
cd(old_folder);

hold on
plot(handles.validx(:,1),handles.validy(:,1),'c+')
plot(handles.validx(:,2),handles.validy(:,2),'g+')
plot(handles.invalid_x,handles.invalid_y,'r+')
hold off
drawnow;
if get(handles.Replace_bad,'Value')==1
    saveas(h,['grid replaced.tif'])
else
    saveas(h,['grid removed.tif'])
end
guidata(handles.output, handles);

function Define_axes(handles)
handles=guidata(handles.output);
[axes_filename,axes_file_dir,FilterIndex] = uigetfile( ...
    {'*.bmp;*.tif;*.jpg;*.TIF;*.BMP;*.JPG','Image files
(*.bmp,*.tif,*.jpg)';'*.*',   'All Files (*.*)'}, ...
    'Open image to use to define the new axes');
old_folder=cd(axes_file_dir);
h=figure;
imshow(axes_filename);
cd(old_folder);
[x(1,1),y(1,1)]=ginput(1);
hold on
plot(x(1,1),y(1,1),'+b')
[x(2,1),y(2,1)]=ginput(1);
hold on
plot(x(2,1),y(2,1),'+b')
Xd=x(2,1)-x(1,1);
Yd=y(2,1)-y(1,1);
mag=(Xd^2+Yd^2)^0.5;
X=Xd/mag;
Y=Yd/mag;
handles.Xaxis=[X Y];
handles.Yaxis=[-Y X];
close(h);
guidata(handles.output, handles);
function find_strain(handles)
```

Code end

SEM_correction: This stand alone program was used to correct the SEM images, using a calibrated grid as input. The code was created by the author of this thesis, using the curcularHough_Grd function created by Tao Peng (available at http://www.mathworks.com/matlabcentral/fileexchange/9168-detect-circles-with-various-radii-in-grayscale-image-via-hough-transform).

```matlab
function [ Corrected_Image ] = SEM_correction( input_args )
corner_selection=2;
circle_selection=2;
[Image_name Image_Folder]=uigetfile('*.tif','Select calibration grid
image');
cd(Image_Folder);
Image=imread(Image_name);

grid_type = menu(sprintf('Is grid composed of squares or
circles?'),'Squares','Circles');

if grid_type==1
while corner_selection==2
    corner_quality=0;
    while corner_quality <= 0 | corner_quality>= 1
        prompt = {'Enter quality level for grid detection (0-1,
exclusive):'};
        dlg_title = 'Input quality for corner analysis';
        num_lines= 1;
        def     = {'0.3'};
        answer = inputdlg(prompt,dlg_title,num_lines,def);
        corner_quality = str2double(cell2mat(answer));
    end
    C=corner(Image, 'QualityLevel', corner_quality);
    imshow(Image);
    hold on
    plot(C(:,1), C(:,2),'r*');
    corner_selection = menu(sprintf('Were the corners consistently
detected (Multiple detections/corner ok, if consistent)?'),'Yes','No');
end
else
    while circle_selection==2

        prompt={'Enter circle radius in pixels:'};
        dlg_title='Circle radius';
        num_lines= 1;
        def={'30'};
        answer=inputdlg(prompt,dlg_title,num_lines,def);
        cir_rad=str2double(cell2mat(answer));
    rad_min=cir_rad-cir_rad*.1;
    rad_max=cir_rad+cir_rad*.1;
  tic;
  [accum, circen, cirrad] = CircularHough_Grd(Image, [rad_min rad_max],
10, 8, 1);
  toc;
```

```matlab
    figure(1); imagesc(accum); axis image;
    title('Accumulation Array from Circular Hough Transform');
    figure(2); imagesc(Image); colormap('gray'); axis image;
    hold on;
    plot(circen(:,1), circen(:,2), 'r+');
    for k = 1 : size(circen, 1),
        DrawCircle(circen(k,1), circen(k,2), cirrad(k), 32, 'b-');
    end
    hold off;
    title(['Raw Image with Circles Detected ', ...
        '(center positions and radii marked)']);

    C=circen;
    imshow(Image);
    hold on
    plot(C(:,1), C(:,2),'r*');
    circle_selection = menu(sprintf('Were the circles consistently
detected?'),'Yes','No');
    end
end

remove_points = menu(sprintf('Are there any extra points that need to
be removed?'),'Yes','No');
while remove_points==1
    [remove_x,remove_y]=ginput(1);

    remove_index=1;
    current_remove_point=((remove_x-C(1,1))^2+(remove_y-C(1,2))^2)^0.5;
    for remove_loop=2:size(C,1)
        check_remove_point=((remove_x-C(remove_loop,1))^2+(remove_y-
C(remove_loop,2))^2)^0.5;
        if check_remove_point<current_remove_point
            current_remove_point=check_remove_point;
            remove_index=remove_loop;
        end
    end
plot(C(:,1), C(:,2),'k*');
C(remove_index,:)=[];
plot(C(:,1), C(:,2),'r*');
remove_points = menu(sprintf('Are there any extra points that need to
be removed?'),'Yes','No');
end

n=size(C,1);
corner_correction=2;
while corner_correction==2
imshow(Image);
hold on
plot(C(:,1), C(:,2),'r*');
drawnow;
calibration=[];
C_temp=C;
C_temp(:,3)=1;

prompt = {'Enter the length of the scale bar (micrometers):', ...
        'Enter the length of the scale bar (pixels):'};
```

```matlab
dlg_title = 'Input for pixel to µm conversion';
num_lines= 1;
def3 = {'10','200'};
answer3 = inputdlg(prompt,dlg_title,num_lines,def3);
scale_length = str2double(cell2mat(answer3(1,1)));
scale_pixels = str2double(cell2mat(answer3(2,1)));
len_pix=scale_length/scale_pixels;


prompt = {'Distance between grid points (micrometeres):'};
dlg_title = 'Input grid information';
num_lines= 1;
def4 = {'3'};
answer4 = inputdlg(prompt,dlg_title,num_lines,def4);
grid_dist = str2double(cell2mat(answer4(1,1)));


spacing=grid_dist/len_pix;


Xcheck=spacing/4;
Ycheck=spacing/4;
title(sprintf('Define the region of interest.  Pick (single click) a
point in the LOWER LEFT region of the gage section.\n  Do the same for
a point in the UPPER RIGHT portion of the gage section.'))


[x(1,1),y(1,1)]=ginput(1);
hold on
plot(x(1,1),y(1,1),'+b')


[x(2,1),y(2,1)]=ginput(1);
hold on
plot(x(2,1),y(2,1),'+b')


xmin = min(x);
xmax = max(x);
ymin = min(y);
ymax = max(y);


if (C_temp(n,1)<xmin | C_temp(n,1)>xmax | C_temp(n,2)<ymin |
C_temp(n,2)>ymax)
        C_temp(n,3)=0;
end


for a=1:(n-1)    %Averaging multiple points that make up a single grid
intersection
    if (C_temp(a,1)<xmin | C_temp(a,1)>xmax | C_temp(a,2)<ymin |
C_temp(a,2)>ymax)
        C_temp(a,3)=0;
    end
    if C_temp(a,3)~=0
    for b=(a+1):n
        if (abs(C_temp(a,1)-C_temp(b,1))<Xcheck)
            if (abs(C_temp(a,2)-C_temp(b,2))<Ycheck)

C_temp(a,1)=(C_temp(a,1)*C_temp(a,3)+C_temp(b,1))/(C_temp(a,3)+1);

C_temp(a,2)=(C_temp(a,2)*C_temp(a,3)+C_temp(b,2))/(C_temp(a,3)+1);
```

```matlab
                C_temp(a,3)=C_temp(a,3)+1;
                C_temp(b,3)=0;
            end
        end
    end
    end
end

counter=1;
for c=1:n
    if C_temp(c,3)>0
        calibration(counter,1)=C_temp(c,1);
        calibration(counter,2)=C_temp(c,2);
        counter=counter+1;
    end
end
imshow(Image);
hold on
plot(calibration(:,1), calibration(:,2),'c*');
corner_correction = menu(sprintf('Were the grid points consistently
detected (Multiple detections/corner ok, if consistent)?'),'Yes','No');
end

[orderedX_calibration, x_index]=sort(calibration(:,1));
orderedX_calibration(:,2)=calibration(x_index,2);

sort_loop_size=size(calibration);
columns_found=0;
column_end=[];

for sort_loop=1:sort_loop_size(1)-1
    if abs(orderedX_calibration(sort_loop+1,1)-
orderedX_calibration(sort_loop,1))>(spacing/4)
        columns_found=columns_found+1;
        column_end(columns_found)=sort_loop;
    end
end
columns_found=columns_found+1;
column_end(columns_found)=sort_loop_size(1);

[sort_portion, index]=sort(orderedX_calibration(1:column_end(1),2));
sorted_calibration(1:column_end(1),1)=orderedX_calibration(index,1);
sorted_calibration(1:column_end(1),2)=orderedX_calibration(index,2);

for ysort_loop=2:columns_found
    range_min=1+column_end(ysort_loop-1);
    range_max=column_end(ysort_loop);
    [sort_portion,
index]=sort(orderedX_calibration(range_min:range_max,2));
    range_index=index+range_min-1;

sorted_calibration(range_min:range_max,1)=orderedX_calibration(range_in
dex,1);

sorted_calibration(range_min:range_max,2)=orderedX_calibration(range_in
dex,2);
```

```matlab
end

if abs(sorted_calibration(1,2)-
sorted_calibration(column_end(1)+1,2))<(spacing/4)
    grid_type=1;              %Grid type 1:    + + +
                              %                + + +
else                         %                + + +

    grid_type=2;             %Grid type 2:    + + +
end                          %               + + + +
                             %                + + +
center_point=1;
abs_center_x=round(size(Image,2)/2);
abs_center_y=round(size(Image,1)/2);
current_dif=((sorted_calibration(center_point,1)-
abs_center_x)^2+(sorted_calibration(center_point,2)-
abs_center_y)^2)^0.5;
for center_loop=2:sort_loop_size(1)
    test_dif=((sorted_calibration(center_loop,1)-
abs_center_x)^2+(sorted_calibration(center_loop,2)-abs_center_y)^2)^0.5
    if test_dif<current_dif
        current_dif=test_dif;
        center_point=center_loop;
    end
end
center_column=1;
col_left=0
col_right=columns_found-1;
rows_above=center_point-1;
rows_below=column_end(1)-center_point;

for center_loc=2:columns_found
if (center_point>column_end(center_loc-1) &&
center_point<column_end(center_loc))
    center_column=center_loc;
    col_left=center_column-1;
    col_right=columns_found-center_column
    rows_above=center_point-1-column_end(center_loc-1);
    rows_below=column_end(center_loc)-center_point;
end
end

if grid_type==1
    count=0;
    height=column_end(1);
for x_cor=1:columns_found
    for y_cor=1:height
        count=count+1;
        corrected_grid(count,1)=sorted_calibration(center_point,1)-
spacing*((col_left+1)-x_cor);
        corrected_grid(count,2)=sorted_calibration(center_point,2)-
spacing*((rows_above+1)-y_cor);

        if (corrected_grid(count,1)-
sorted_calibration(center_point,1))<0
```

```matlab
            theta(count,1)=pi+(atan((corrected_grid(count,2)-
sorted_calibration(center_point,2))./(corrected_grid(count,1)-
sorted_calibration(center_point,1))));
        else
            theta(count,1)=(atan((corrected_grid(count,2)-
sorted_calibration(center_point,2))./(corrected_grid(count,1)-
sorted_calibration(center_point,1))));
        end


    end
    if x_cor~=columns_found
        height=column_end(x_cor+1)-column_end(x_cor);
    end
end

else
    count=0;
    height=column_end(1);
    for x_cor=1:columns_found
        for y_cor=1:height
            count=count+1;
            corrected_grid(count,1)=sorted_calibration(center_point,1)-
(spacing/2)*((col_left+1)-x_cor);

            if sorted_calibration(column_end(center_column-1)+1,2)-
sorted_calibration(column_end(center_column-2)+1,2)>spacing/4     % If
columns next to center are higher by half a step + +
                offset=-spacing/2;
%                                                                 +
            else
%                                                                 + +
                offset=spacing/2;
            end
            if mod(abs(center_column-x_cor),2)==0

corrected_grid(count,2)=sorted_calibration(center_point,2)-
spacing*((rows_above+1)-y_cor);
            else

corrected_grid(count,2)=sorted_calibration(center_point,2)-
spacing*((rows_above+1)-y_cor)+offset;
            end

            if (corrected_grid(count,1)-
sorted_calibration(center_point,1))<0
                theta(count,1)=pi+(atan((corrected_grid(count,2)-
sorted_calibration(center_point,2))./(corrected_grid(count,1)-
sorted_calibration(center_point,1))));
            else
                theta(count,1)=(atan((corrected_grid(count,2)-
sorted_calibration(center_point,2))./(corrected_grid(count,1)-
sorted_calibration(center_point,1))));
            end
```

```matlab
        end
        if x_cor~=columns_found
            height=column_end(x_cor+1)-column_end(x_cor);
        end
    end

end

%calibration is detected grid, corrected_grid is expected grid
angle=0;
fit_angle=sum(((corrected_grid(:,1)-
sorted_calibration(:,1)).^2+(corrected_grid(:,2)-
sorted_calibration(:,2)).^2).^0.5);    %measured the total difference in
distance between the expected grid and the detected grid
best_grid=corrected_grid;
length=((corrected_grid(:,1)-
sorted_calibration(center_point,1)).^2+(corrected_grid(:,2)-
sorted_calibration(center_point,2)).^2).^0.5;
%theta=(atan((corrected_grid(:,2)-
sorted_calibration(sorted_center,2))./(corrected_grid(:,1)-
sorted_calibration(sorted_center,1))));
theta(center_point)=0;



for angle_loop=1:101
    test_angle=theta+(angle_loop*0.001745329251994-0.089011791851711);
%radians, equivalent to testing every quarter a degree. starting at -10
degrees.



test_grid(:,1)=corrected_grid(center_point,1)+length.*cos(test_angle)

test_grid(:,2)=corrected_grid(center_point,2)+length.*sin(test_angle)

    test_grid(center_point,:)=corrected_grid(center_point,:);
    fit_test=sum(((test_grid(:,1)-
sorted_calibration(:,1)).^2+(test_grid(:,2)-
sorted_calibration(:,2)).^2).^0.5);
    %throwaway=(angle_loop*0.00875-0.1838)/(2*pi)*360
    hold on
    %plot(test_grid(:,1), test_grid(:,2),'g+');
    if fit_test<fit_angle
        fit_test=fit_angle;
        angle=(angle_loop*0.001745329251994-0.089011791851711);
        best_grid=test_grid;
    end
end



corrected_grid=best_grid;
hold on
plot(corrected_grid(:,1), corrected_grid(:,2),'r+');
```

```matlab
plot(sorted_calibration(center_point,1),
sorted_calibration(center_point,2),'go');

displ_matrix(1:column_end(1)+2,1:columns_found)=0;
displ_matrix_x(1:column_end(1)+2,1:columns_found)=0;
displ_matrix_y(1:column_end(1)+2,1:columns_found)=0;
displ_matrix(1:column_end(1),1)=((corrected_grid(1:column_end(1),1)-
sorted_calibration(1:column_end(1),1)).^2+(corrected_grid(1:column_end(
1),2)-sorted_calibration(1:column_end(1),2)).^2).^0.5;
displ_matrix_x(1:column_end(1),1)=(corrected_grid(1:column_end(1),1)-
sorted_calibration(1:column_end(1),1));
displ_matrix_y(1:column_end(1),1)=(corrected_grid(1:column_end(1),2)-
sorted_calibration(1:column_end(1),2));
for displ_loop=2:columns_found
    column_length=column_end(displ_loop)-column_end(displ_loop-1);

displ_matrix(1:column_length,displ_loop)=((corrected_grid(column_end(di
spl_loop-1)+1:column_end(displ_loop),1)-
sorted_calibration(column_end(displ_loop-
1)+1:column_end(displ_loop),1)).^2+(corrected_grid(column_end(displ_loo
p-1)+1:column_end(displ_loop),2)-
sorted_calibration(column_end(displ_loop-
1)+1:column_end(displ_loop),2)).^2).^0.5;

displ_matrix_x(1:column_length,displ_loop)=(corrected_grid(column_end(d
ispl_loop-1)+1:column_end(displ_loop),1)-
sorted_calibration(column_end(displ_loop-
1)+1:column_end(displ_loop),1));

displ_matrix_y(1:column_length,displ_loop)=(corrected_grid(column_end(d
ispl_loop-1)+1:column_end(displ_loop),2)-
sorted_calibration(column_end(displ_loop-
1)+1:column_end(displ_loop),2));
end
save displ_matrix.dat displ_matrix -ascii -tabs
save displ_matrix_x.dat displ_matrix_x -ascii -tabs
save displ_matrix_y.dat displ_matrix_y -ascii -tabs

degrees_num=angle/(2*pi)*360;
degrees=num2str(degrees_num);
rotation_menu = menu(sprintf(['Rotation angle of ' degrees ' degrees
detected.']),'ok');

input_points=sorted_calibration;
base_points=corrected_grid;

tform = cp2tform(input_points,base_points,'lwm');
I=Image;
J = imtransform(I,tform);
figure, imshow(I), figure, imshow(J)

done=1;
while done==1

    [Image_name Image_Folder]=uigetfile('*.tif','Select uncorrected
image');
```

```matlab
    cd(Image_Folder);
    new_Image=imread(Image_name);
    K = imtransform(new_Image,tform);
    %figure, imshow(new_Image), figure, imshow(K)

    imwrite(K,['Corrected' Image_name '.tif'])

    done = menu(sprintf('Correct another image?'),'Yes','No');
end

end
```

Code end

APPENDIX B

DIC Results

All DIC results are given in the following fashion:
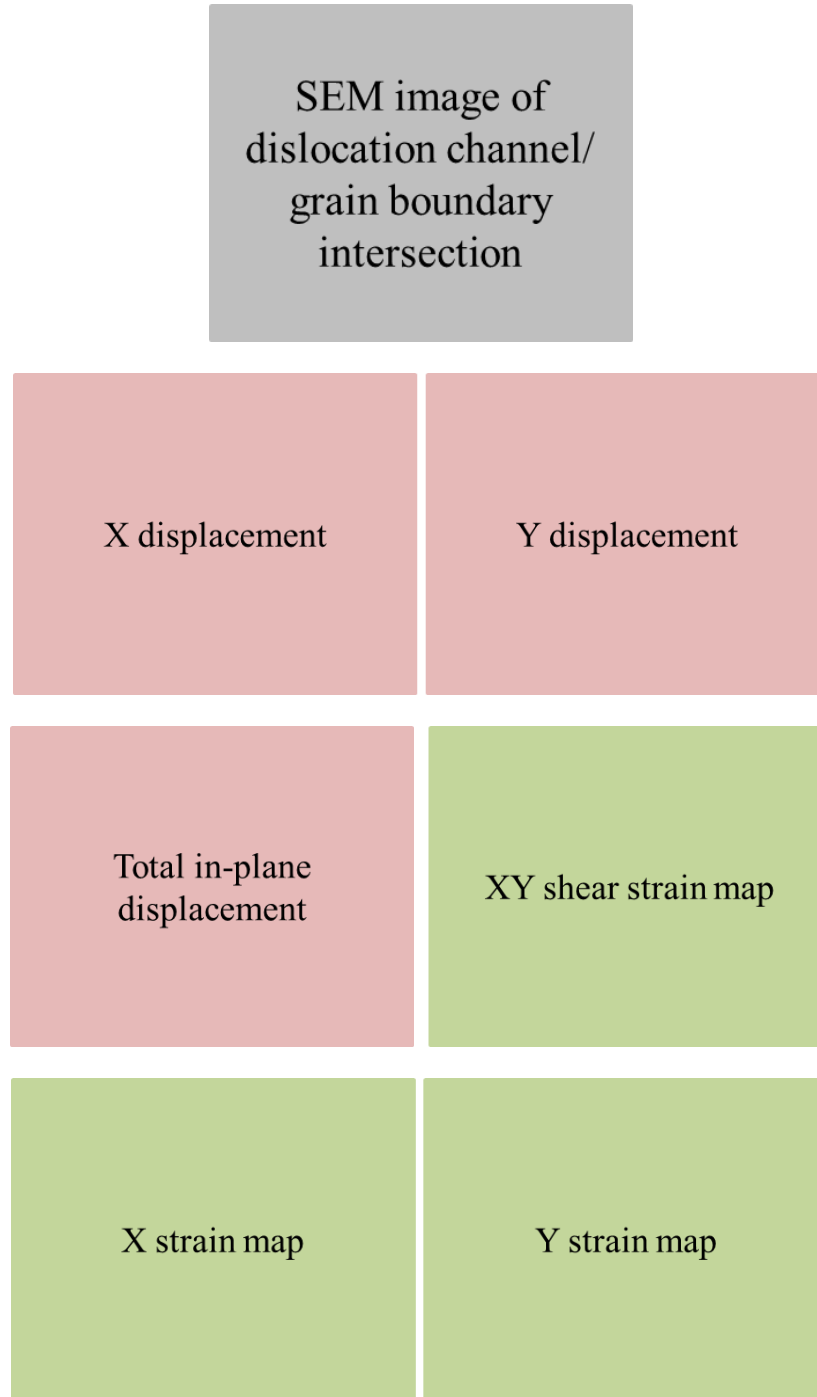
Figure B.1 Schematic showing how DIC results will be presented in this appendix. GB is represented by a blue dashed line in the SEM image. X and Y axis of strain/displacement maps are in micrometers. Strain/Displacement(μm) color bar is to the right of each map.

Figure B.2

X Displacement

Y Displacement

Total Displacement

XY Shear Strain

X Strain

Y Strain

X Displacement

Y Displacement

Total Displacement

XY Shear Strain

X Strain

Y Strain

X Displacement

Y Displacement

Total Displacement

XY Shear Strain

X Strain

Y Strain

X Displacement

Y Displacement

Total Displacement

XY Shear Strain

X Strain

Y Strain

X Displacement

Y Displacement

Total Displacement

XY Shear Strain

X Strain

Y Strain

X Displacement

Y Displacement

Total Displacement

XY Shear Strain

X Strain

Y Strain

X Displacement

Y Displacement

Total Displacement

XY Shear Strain

X Strain

Y Strain

X Displacement

Y Displacement

Total Displacement

XY Shear Strain

X Strain

Y Strain

X Displacement

Y Displacement

Total Displacement

XY Shear Strain

X Strain

Y Strain

2

X Displacement

Y Displacement

Total Displacement

XY Shear Strain

X Strain

Y Strain

322

X Displacement

Y Displacement

Total Displacement

XY Shear Strain

X Strain

Y Strain

X Displacement

Y Displacement

Total Displacement

XY Shear Strain

X Strain

Y Strain

342

X Displacement

Y Displacement

Total Displacement

XY Shear Strain

X Strain

Y Strain

X Displacement

Y Displacement

Total Displacement

XY Shear Strain

X Strain

Y Strain

X Displacement

Y Displacement

Total Displacement

XY Shear Strain

X Strain

Y Strain

X Displacement

Y Displacement

Total Displacement

XY Shear Strain

X Strain

Y Strain

X Displacement

Y Displacement

Total Displacement

XY Shear Strain

X Strain

Y Strain

355

X Displacement

Y Displacement

Total Displacement

XY Shear Strain

X Strain

Y Strain

Figure B.2. SEM images, displacement maps and strain maps of DC-GB intersection.
See Figure B.1 for key and description of figure layout.

APPENDIX C

LEXT Confocal microscope topographical maps

Figure C.1
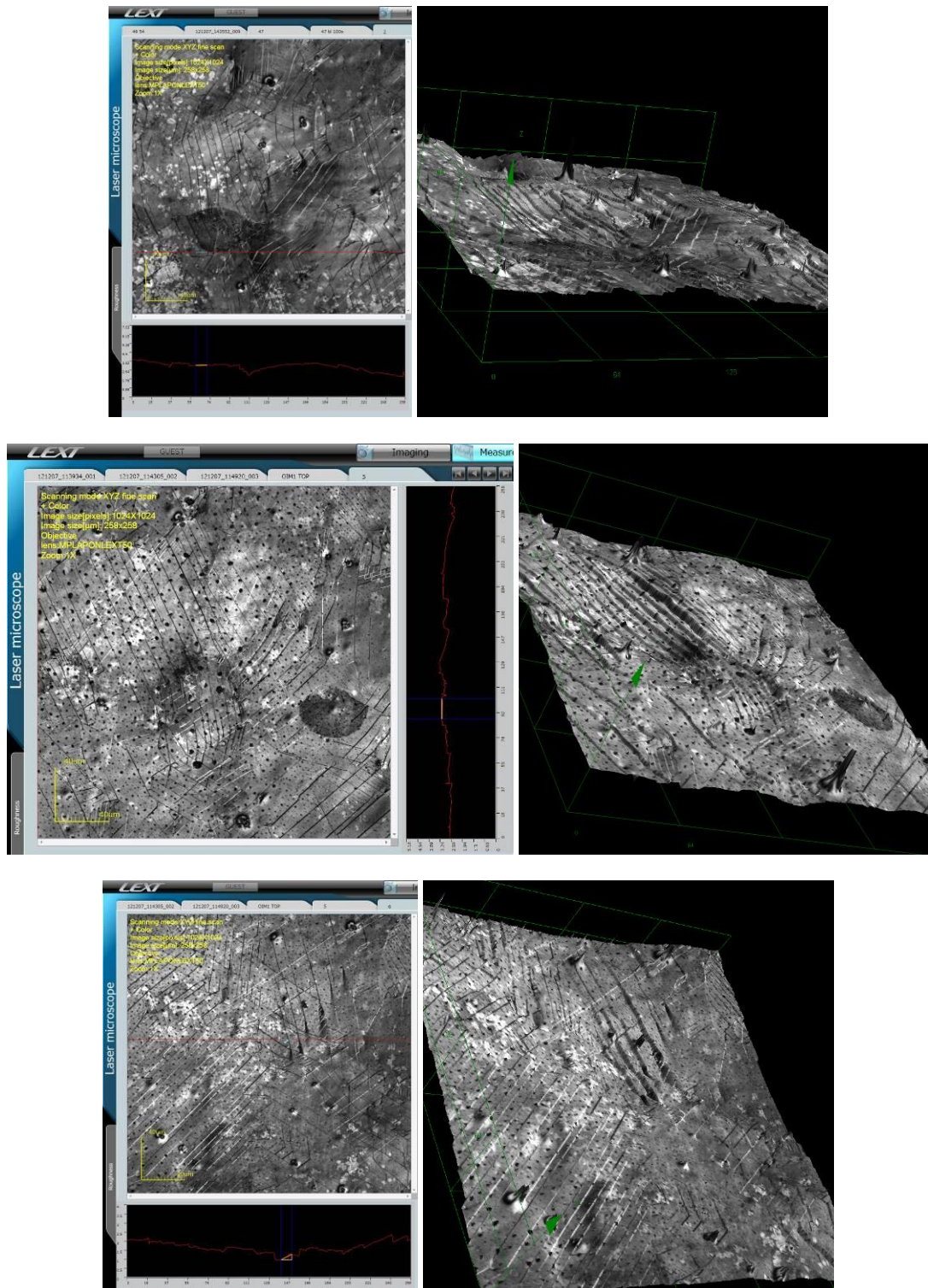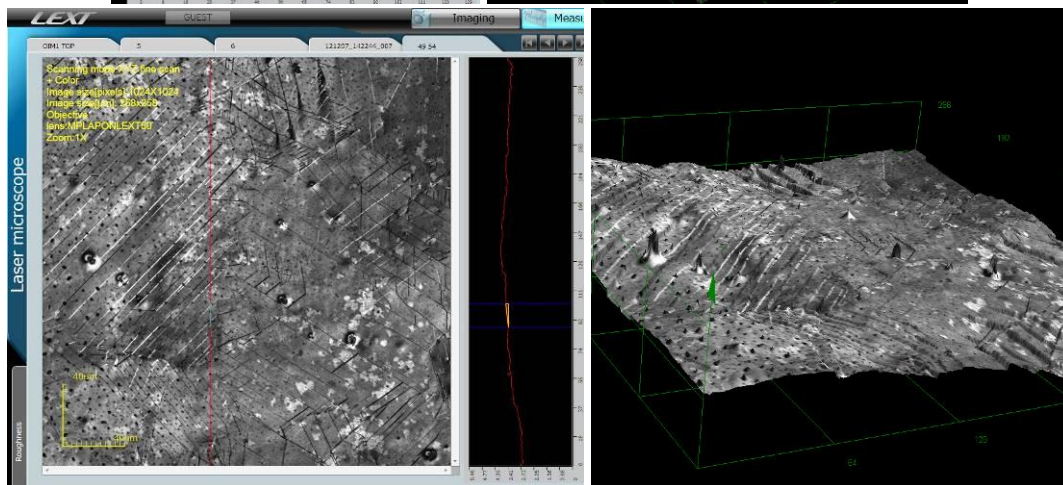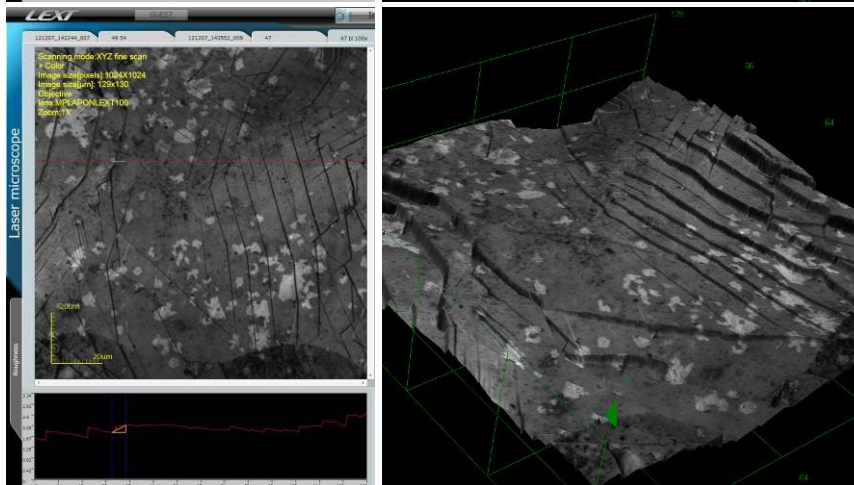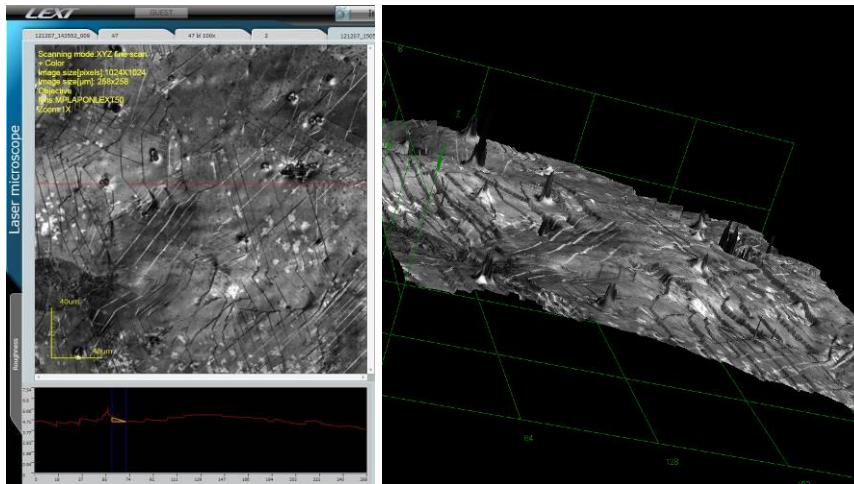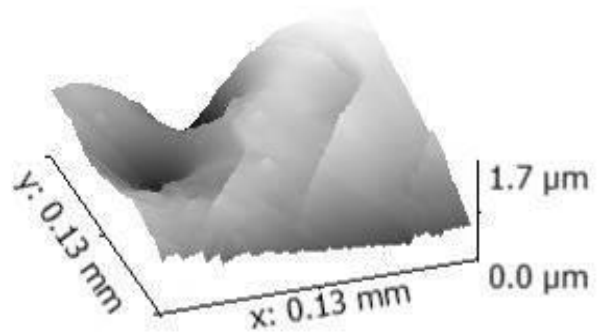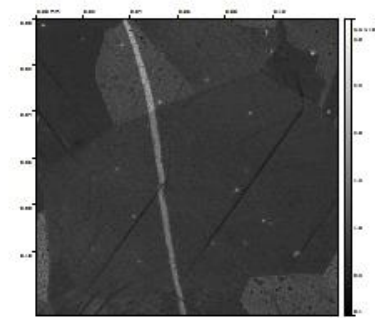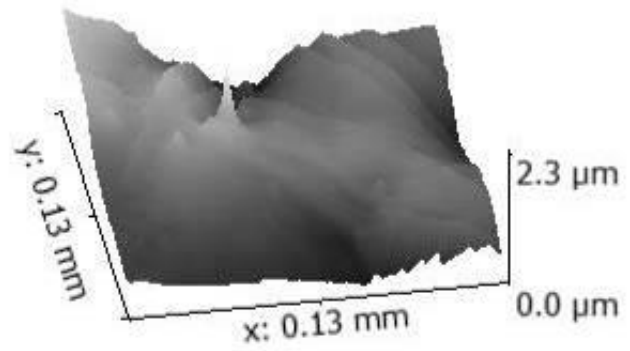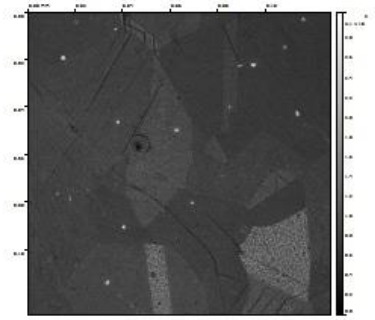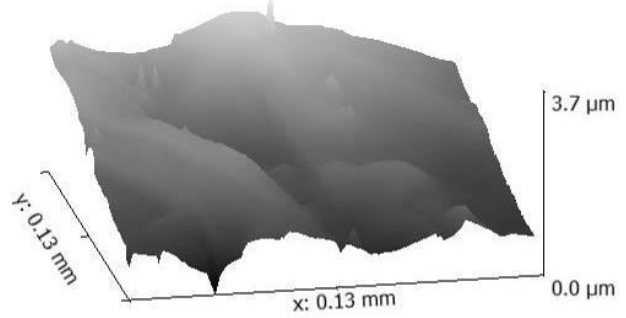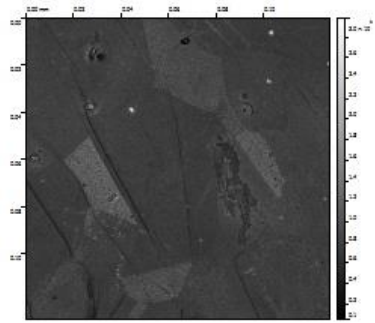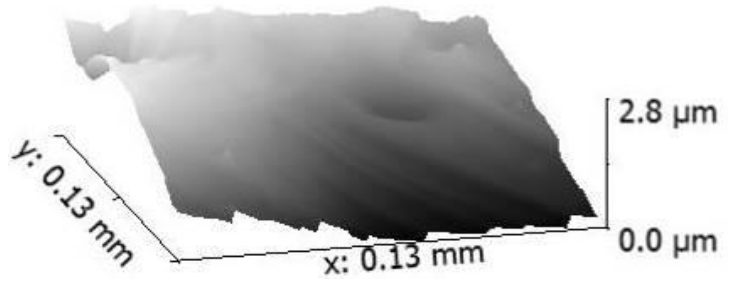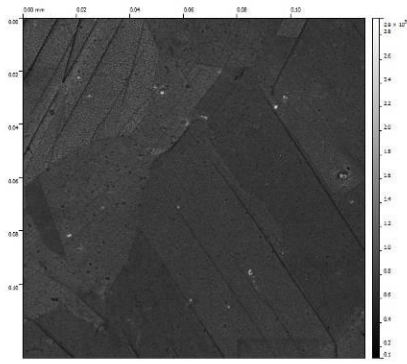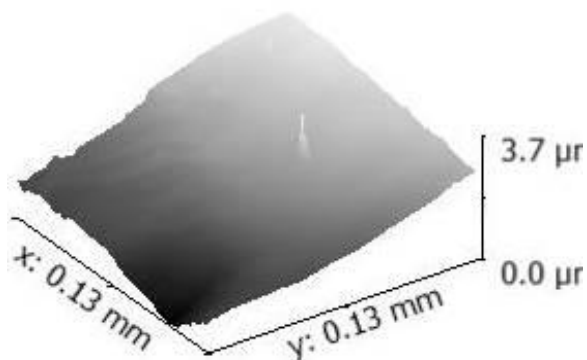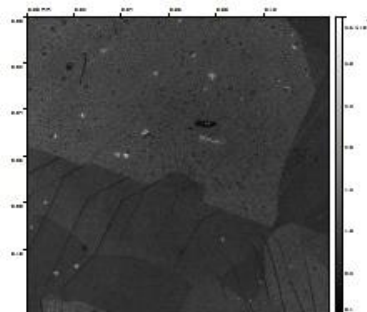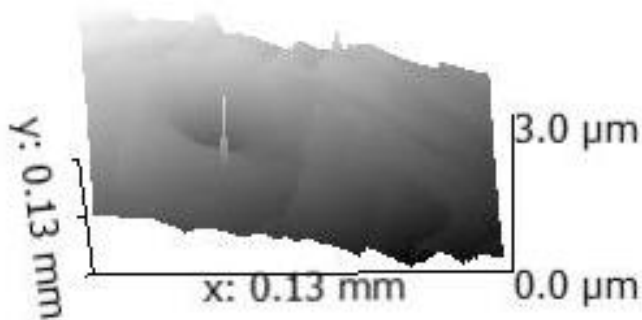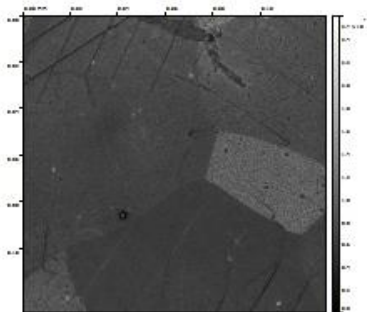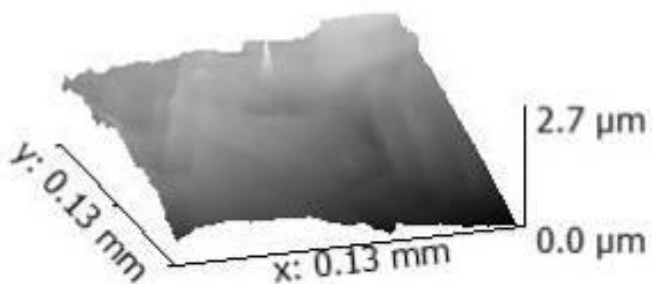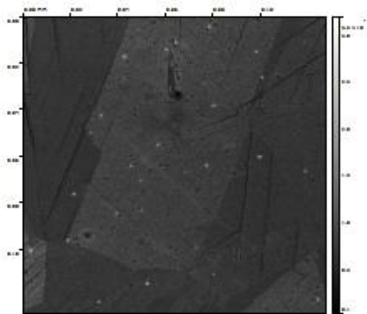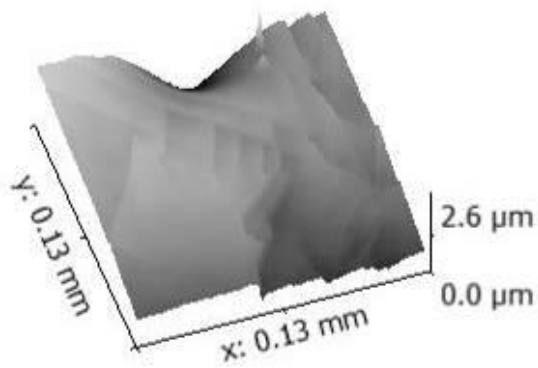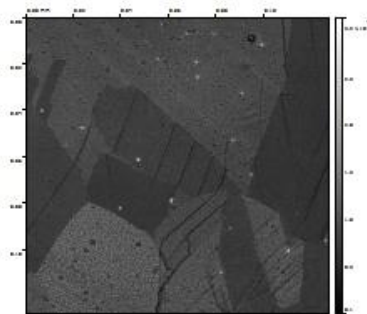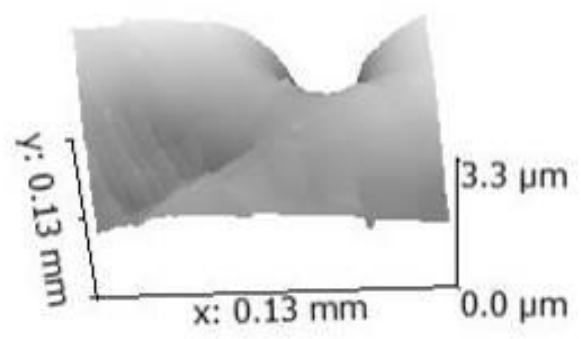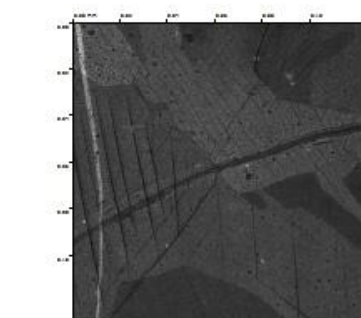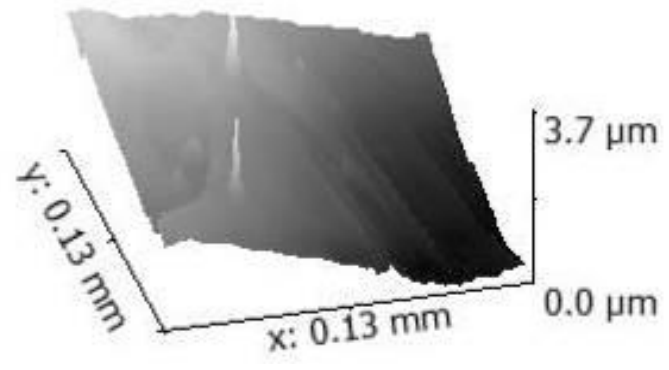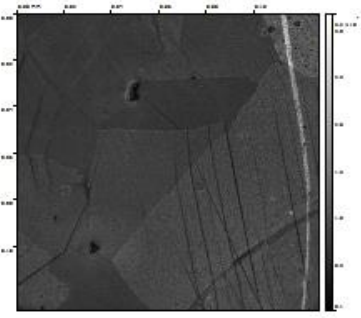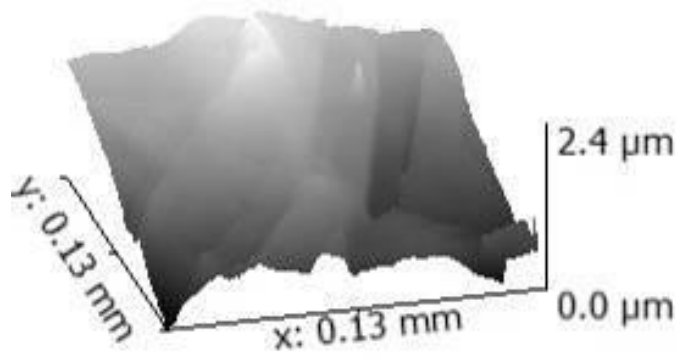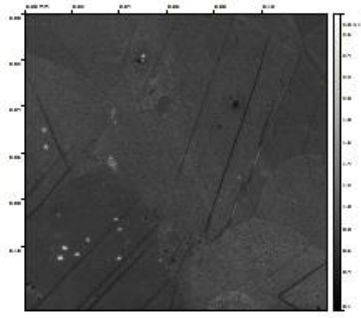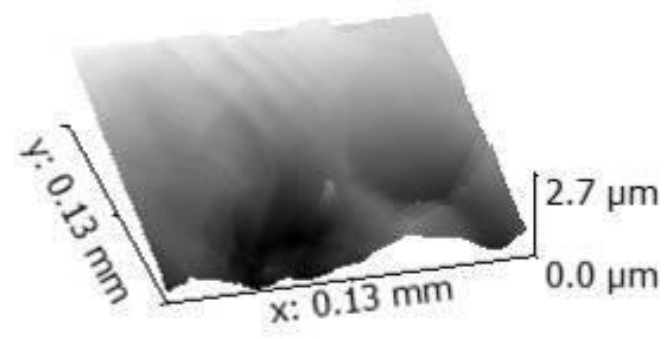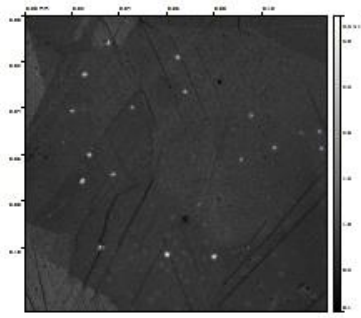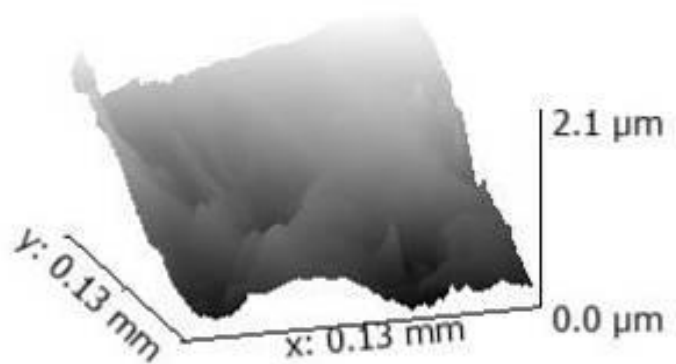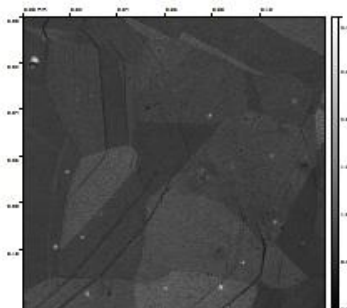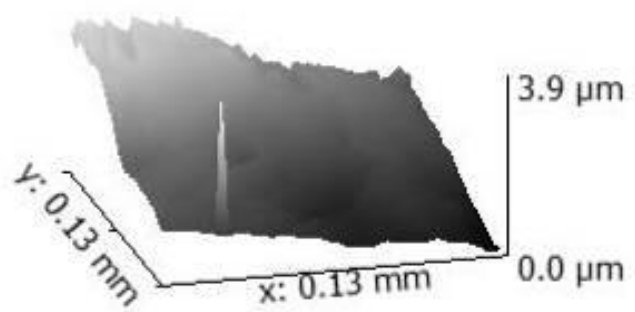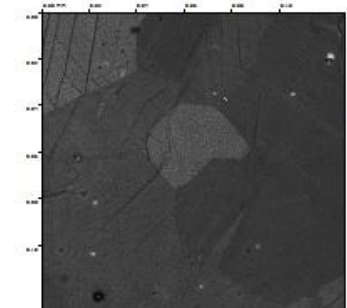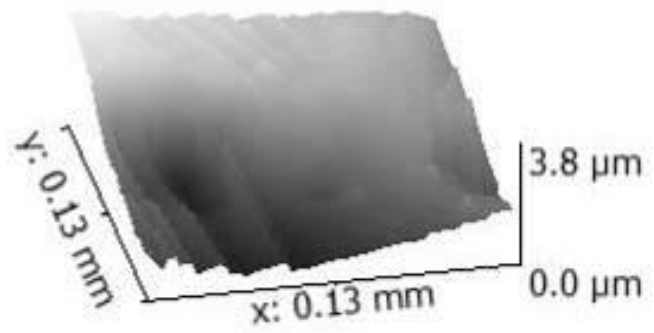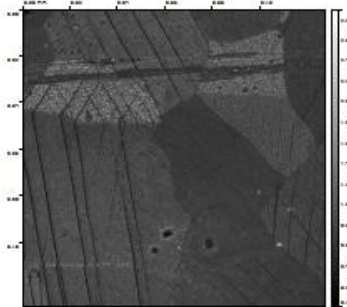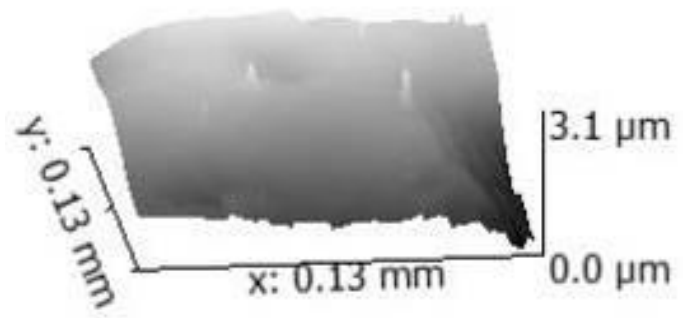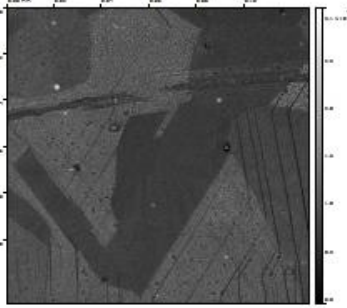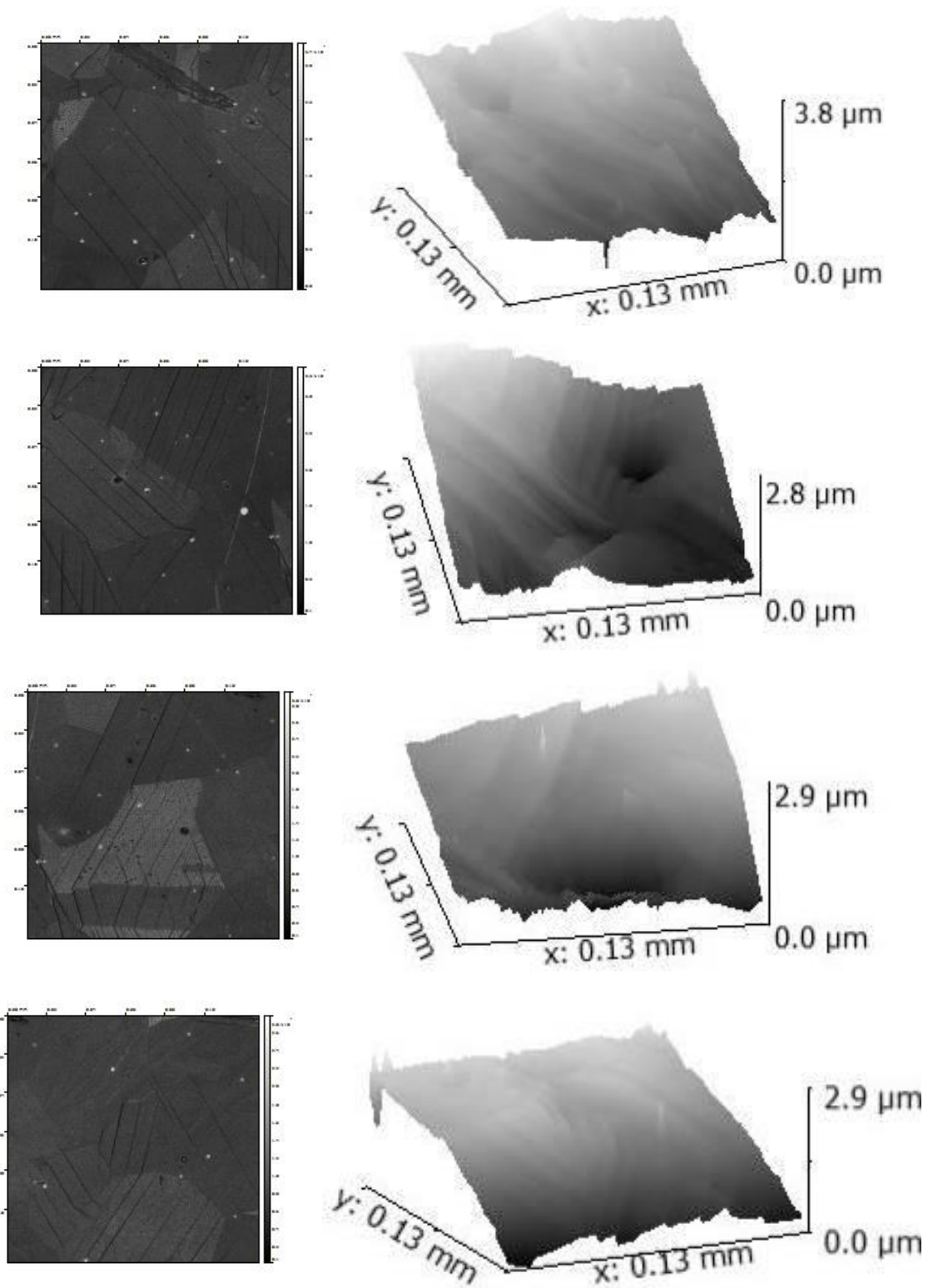
Figure C.1. Confocal microscope topography maps. Left hand images are viewed normal to the surface, right hand images are viewed at an angle to better show topography.
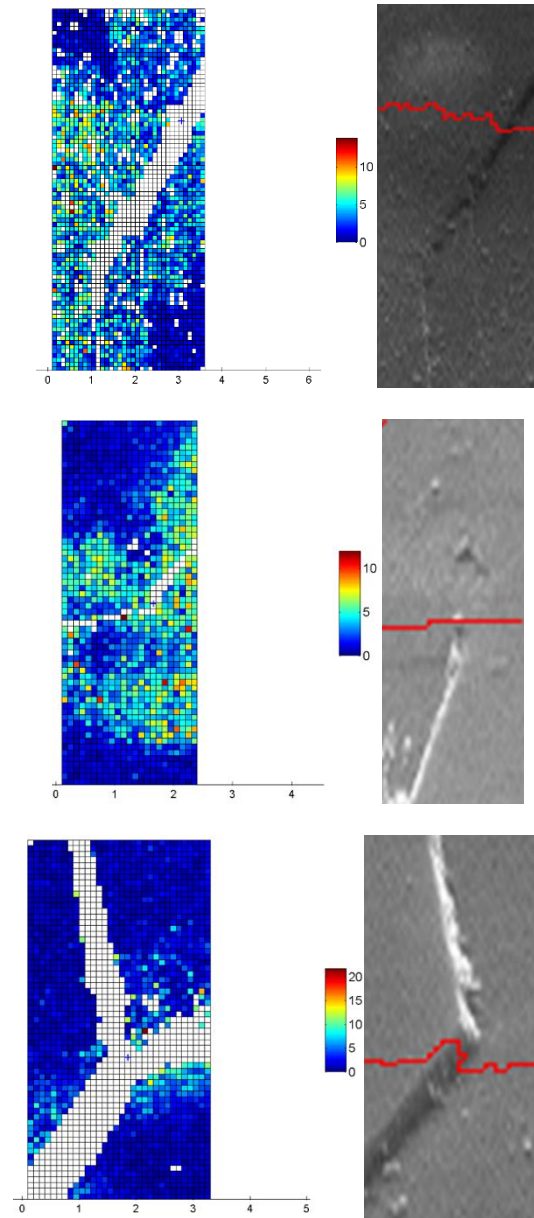
APPENDIX D

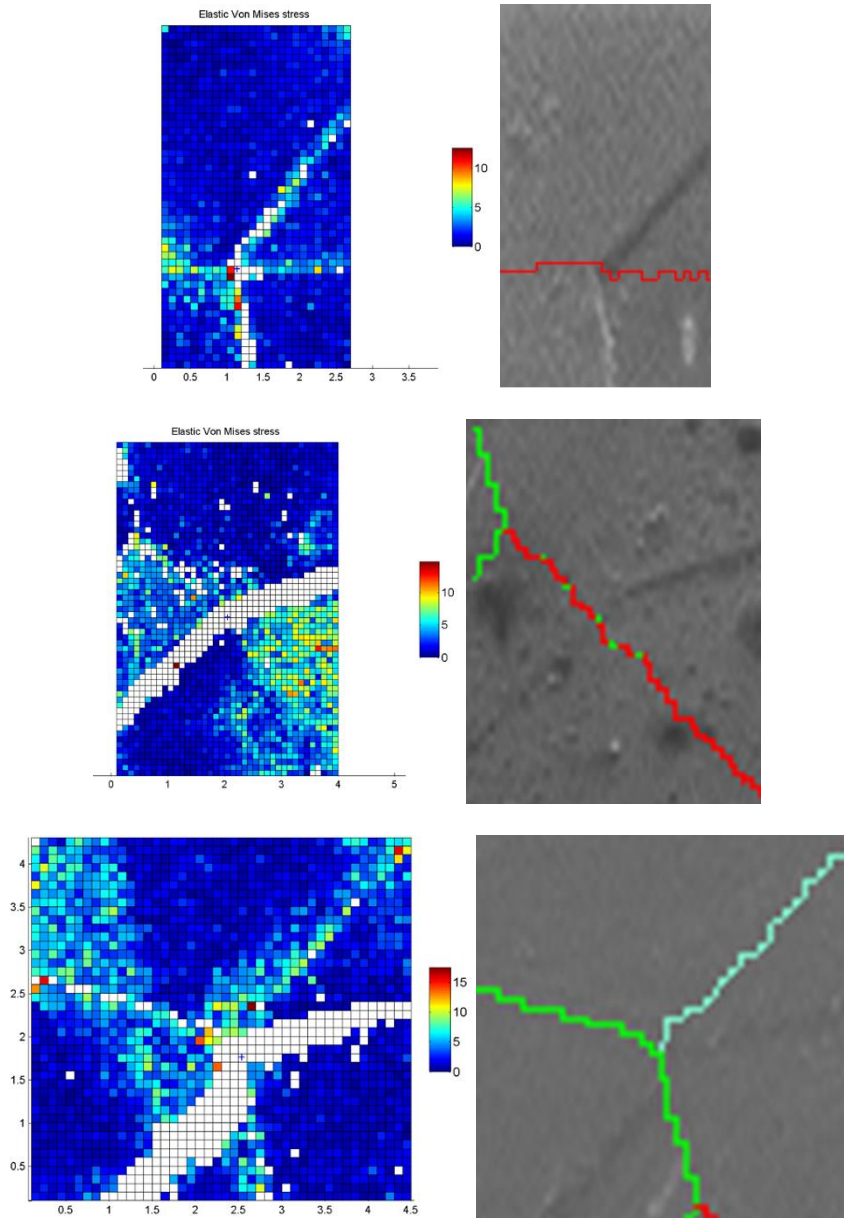**EBSD stress analysis results using CrossCourt 3**

Figure D.1

Color scale in maps is in GPa (white is undefined), axes are in μm. GBs are shown in SEM images (Green-RHAB, blue-LAB, Red-Sigma 3)

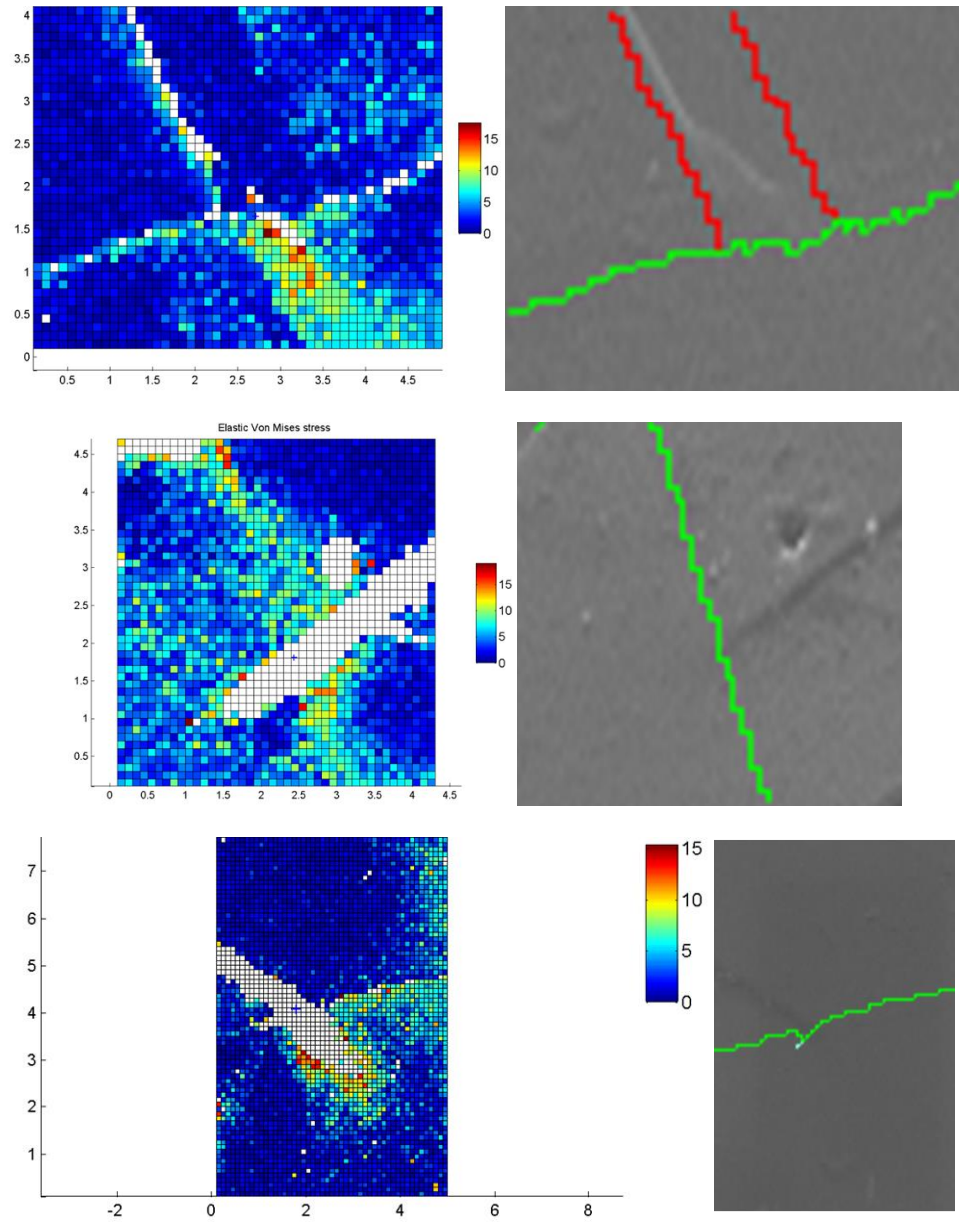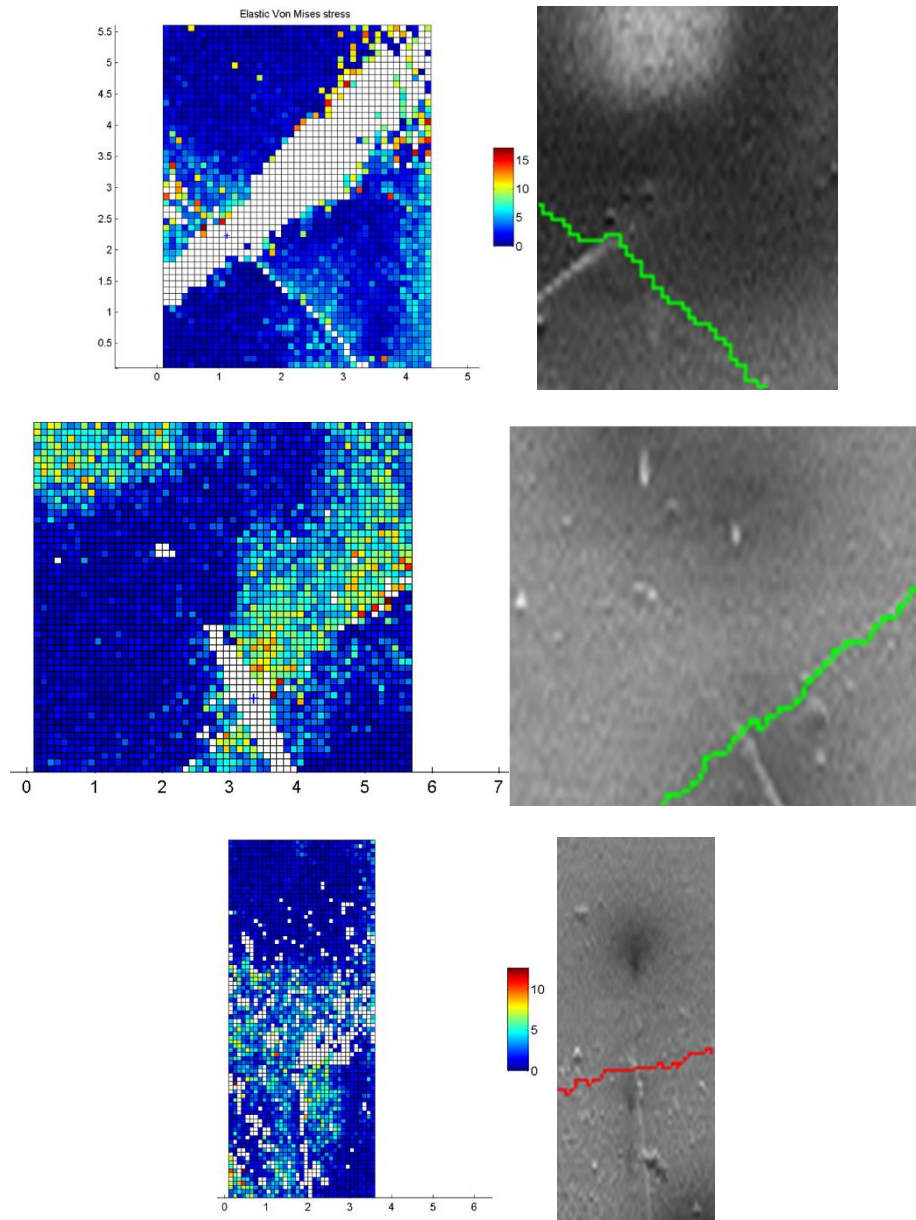Color scale in maps is in GPa (white is undefined), axes are in μm. GBs are shown in SEM images (Green-RHAB, blue-LAB, Red-Sigma 3)

Color scale in maps is in GPa (white is undefined), axes are in µm. GBs are shown in SEM images (Green-RHAB, blue-LAB, Red-Sigma 3)

Color scale in maps is in GPa (white is undefined), axes are in μm. GBs are shown in SEM images (Green-RHAB, blue-LAB, Red-Sigma 3)

Color scale in maps is in GPa (white is undefined), axes are in µm. GBs are shown in SEM images (Green-RHAB, blue-LAB, Red-Sigma 3)

Color scale in maps is in GPa (white is undefined), axes are in μm. GBs are shown in SEM images (Green-RHAB, blue-LAB, Red-Sigma 3)
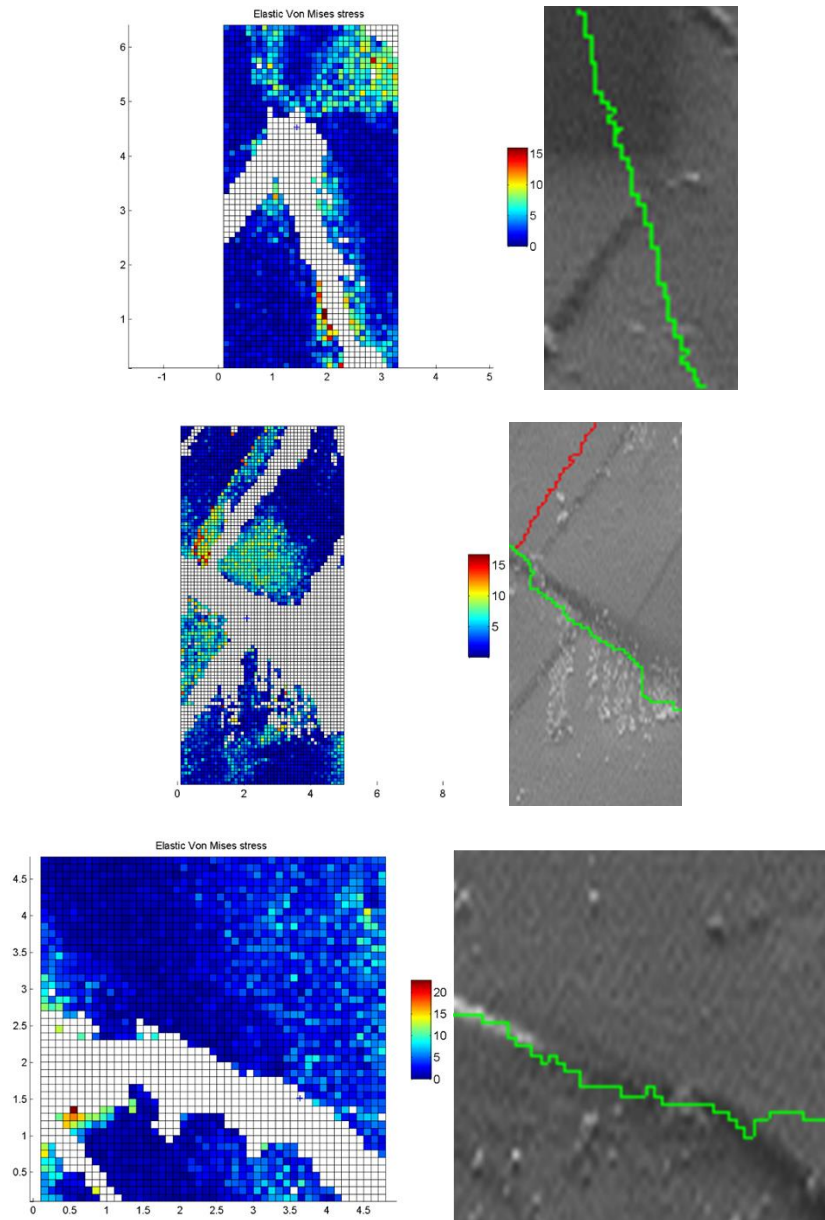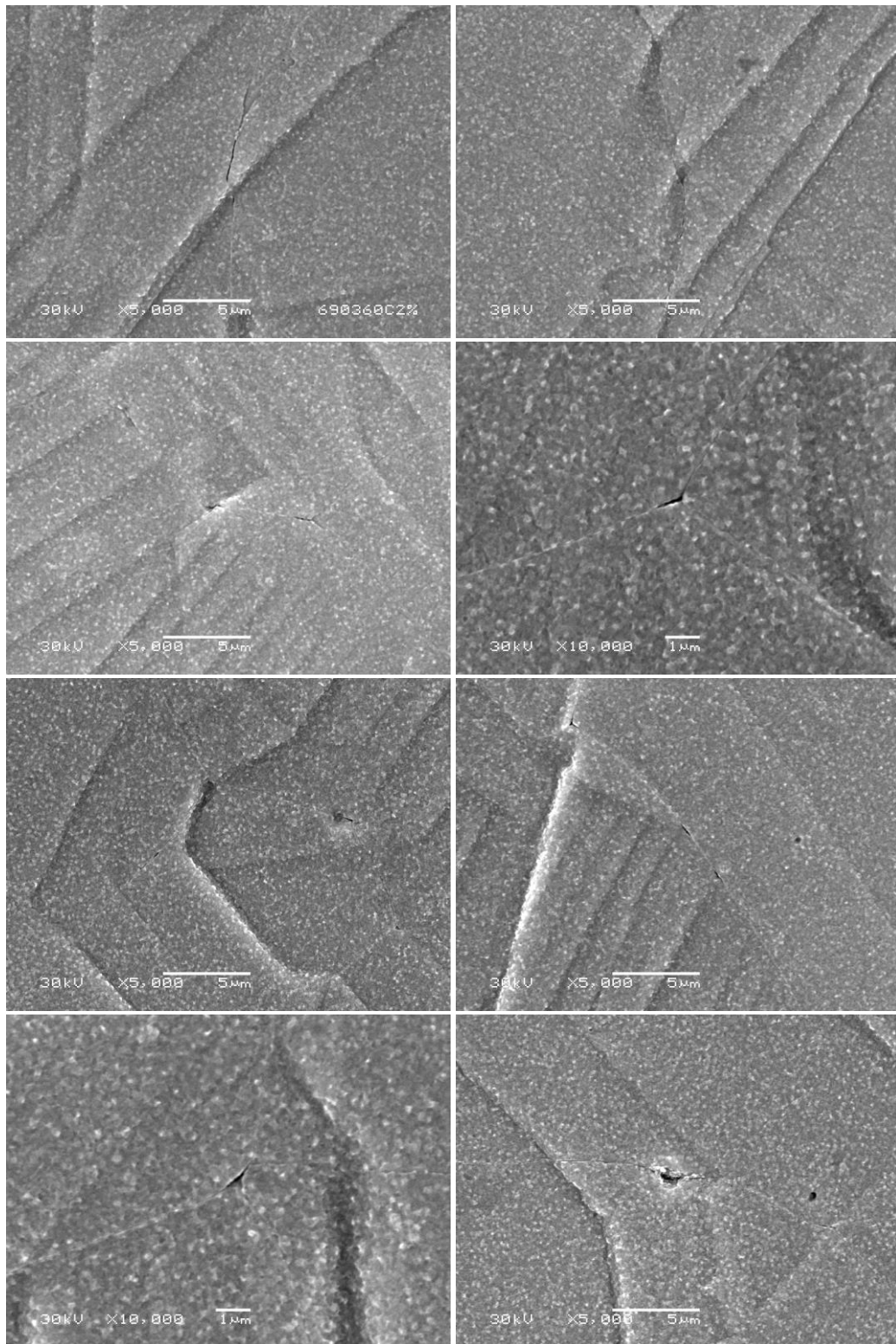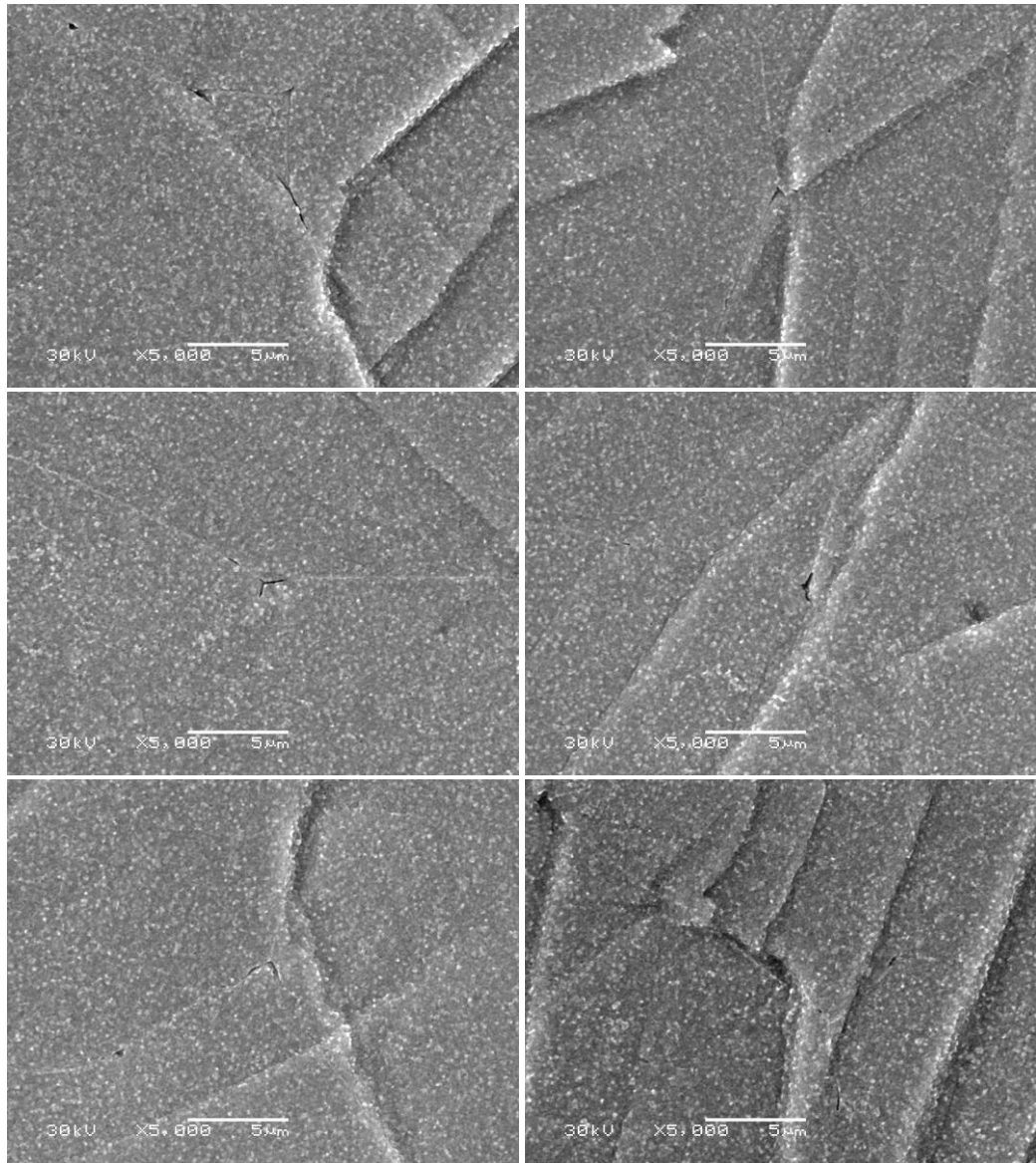

Figure D.1 Left hand side shows the stress maps for the area shown in the SEM images to the right. Stress maps were determined based on elastic stress measurements using a high resolution analysis of EBSD patterns.
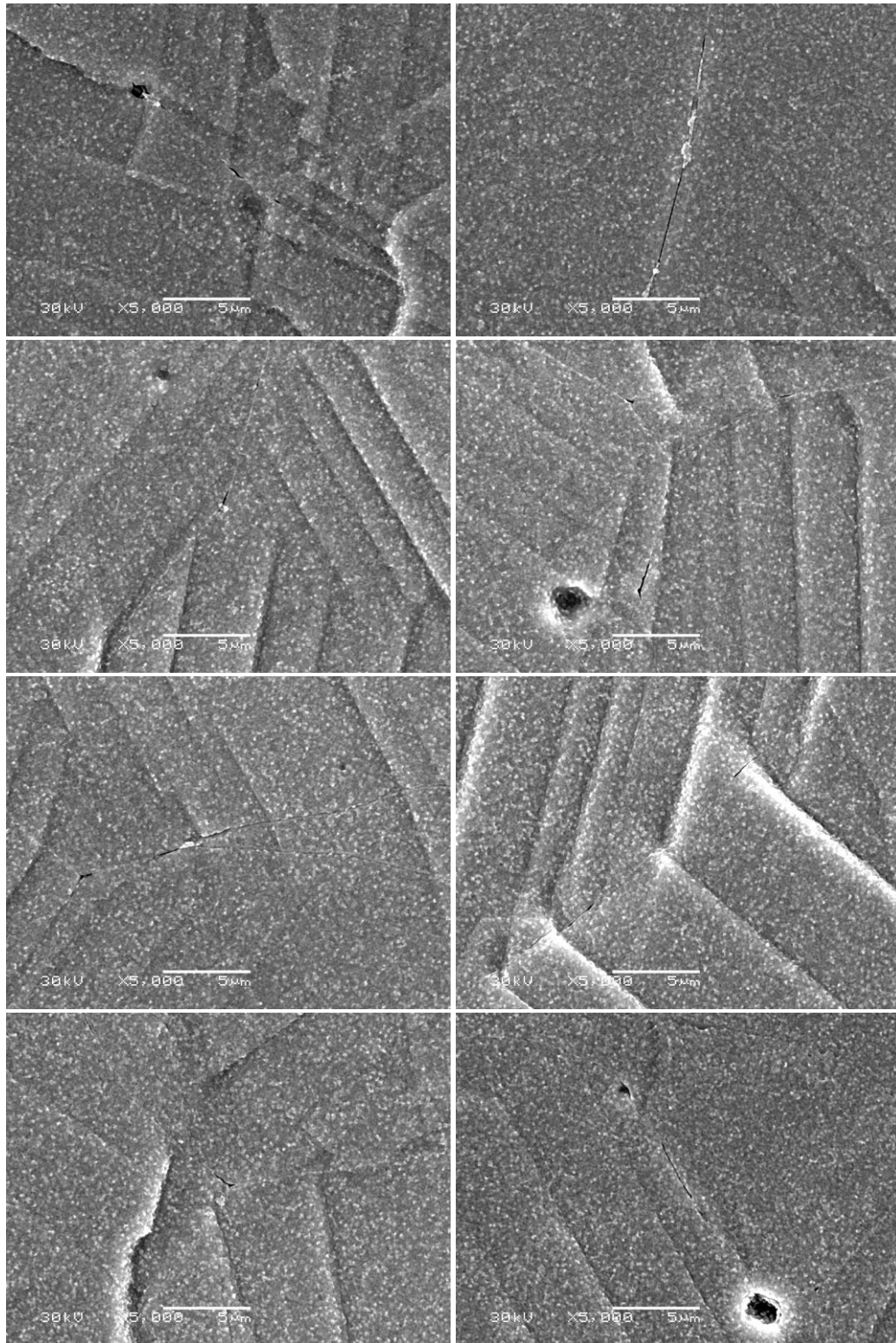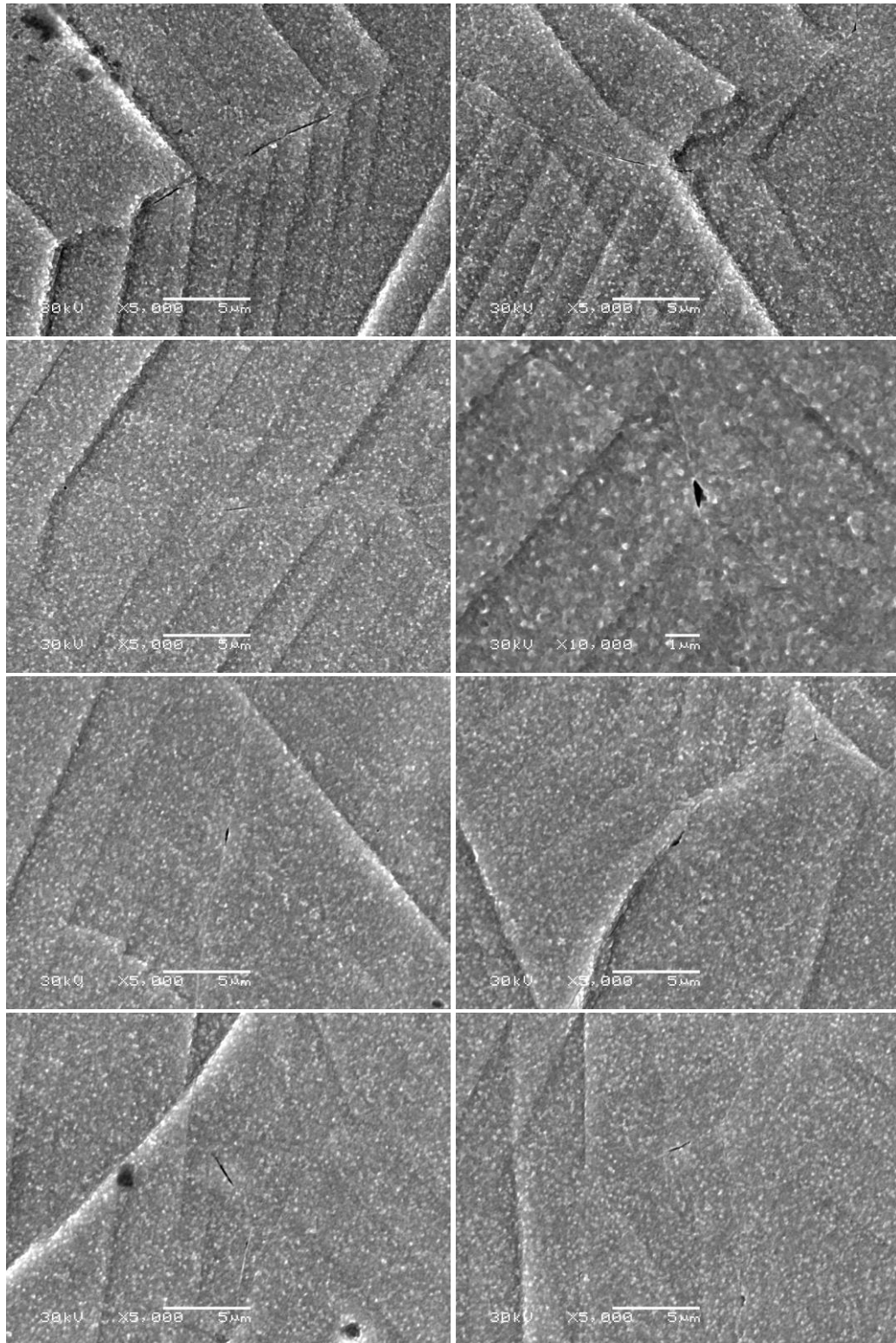
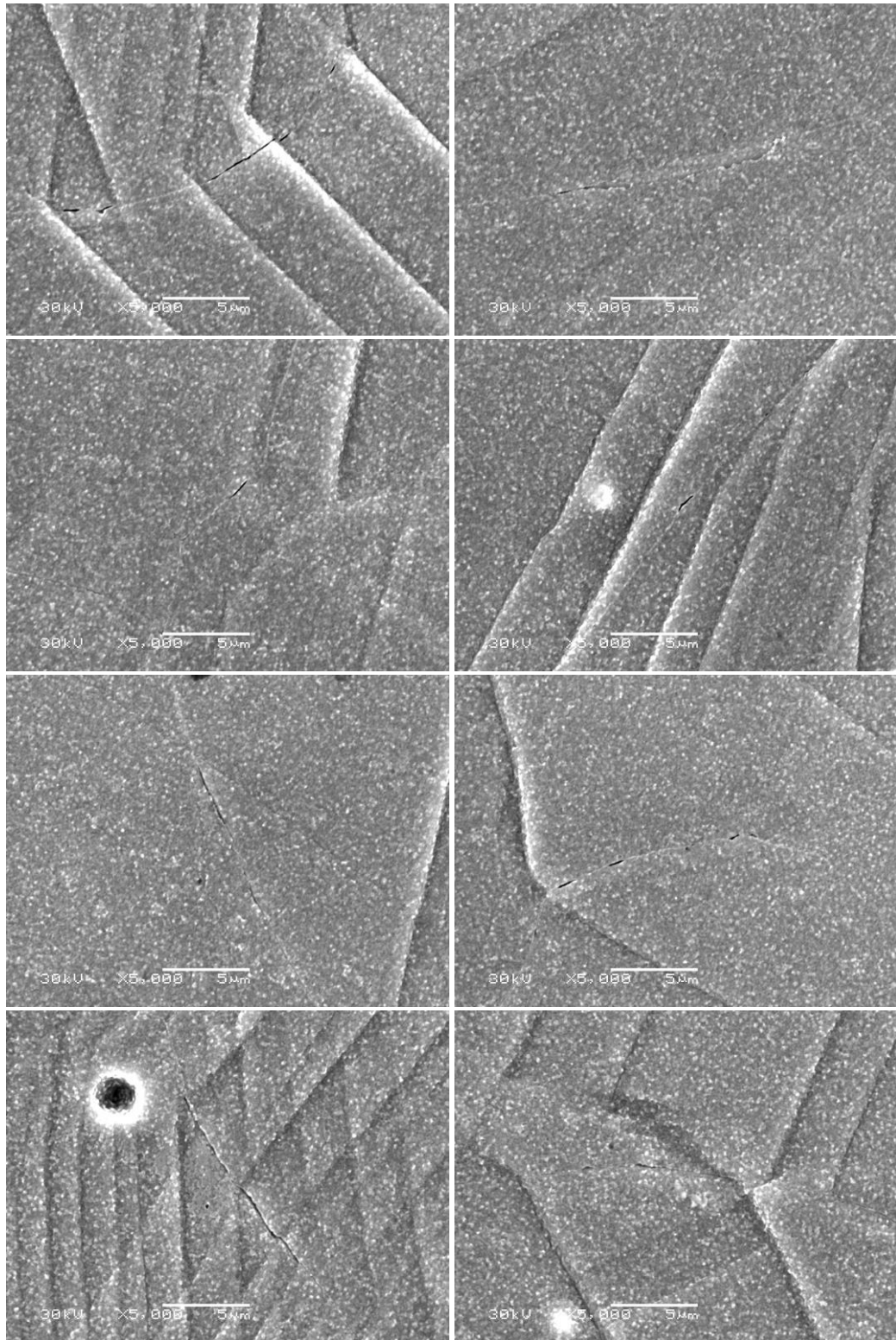# APPENDIX E
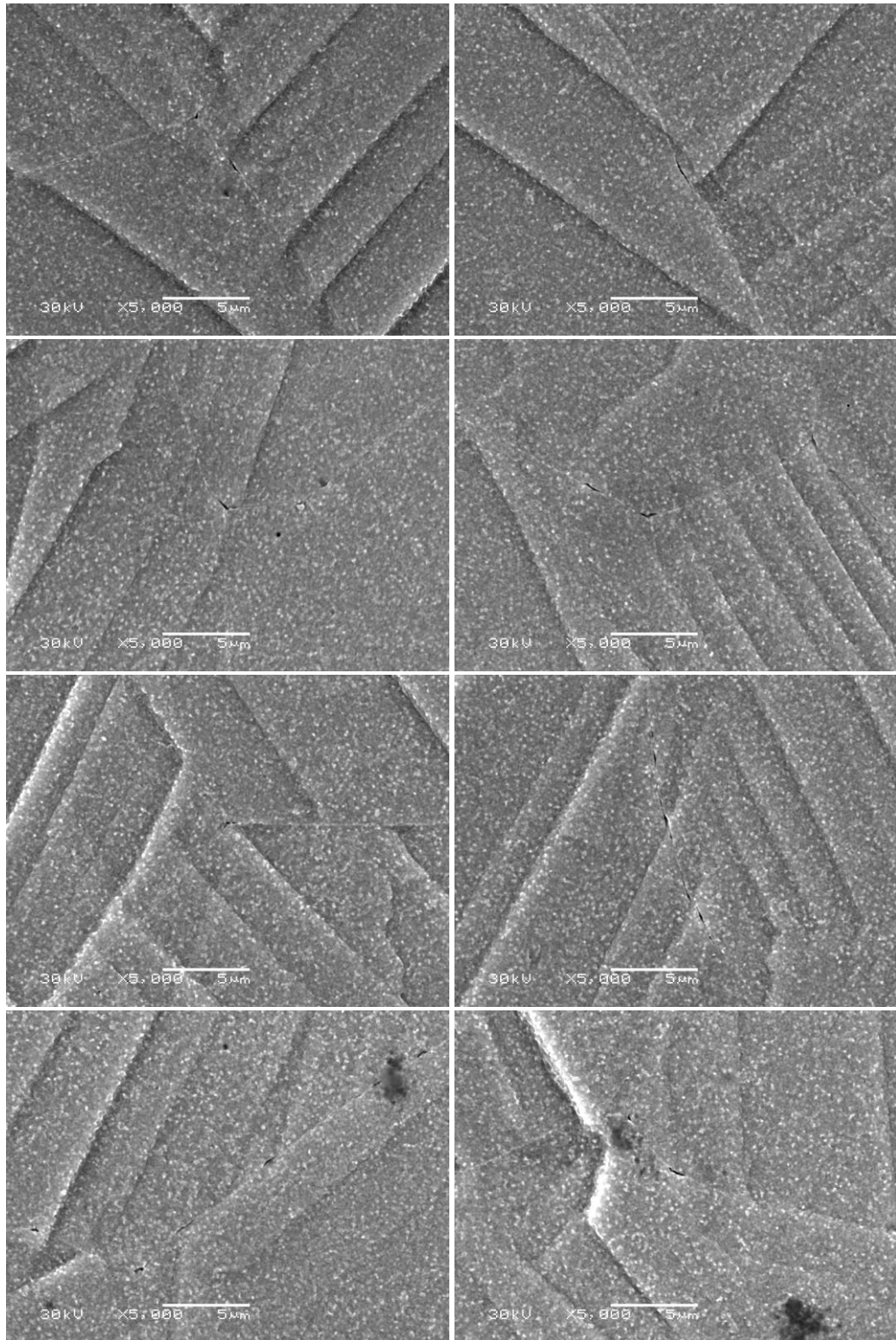
SEM images of crack initation sites

Figure E.1
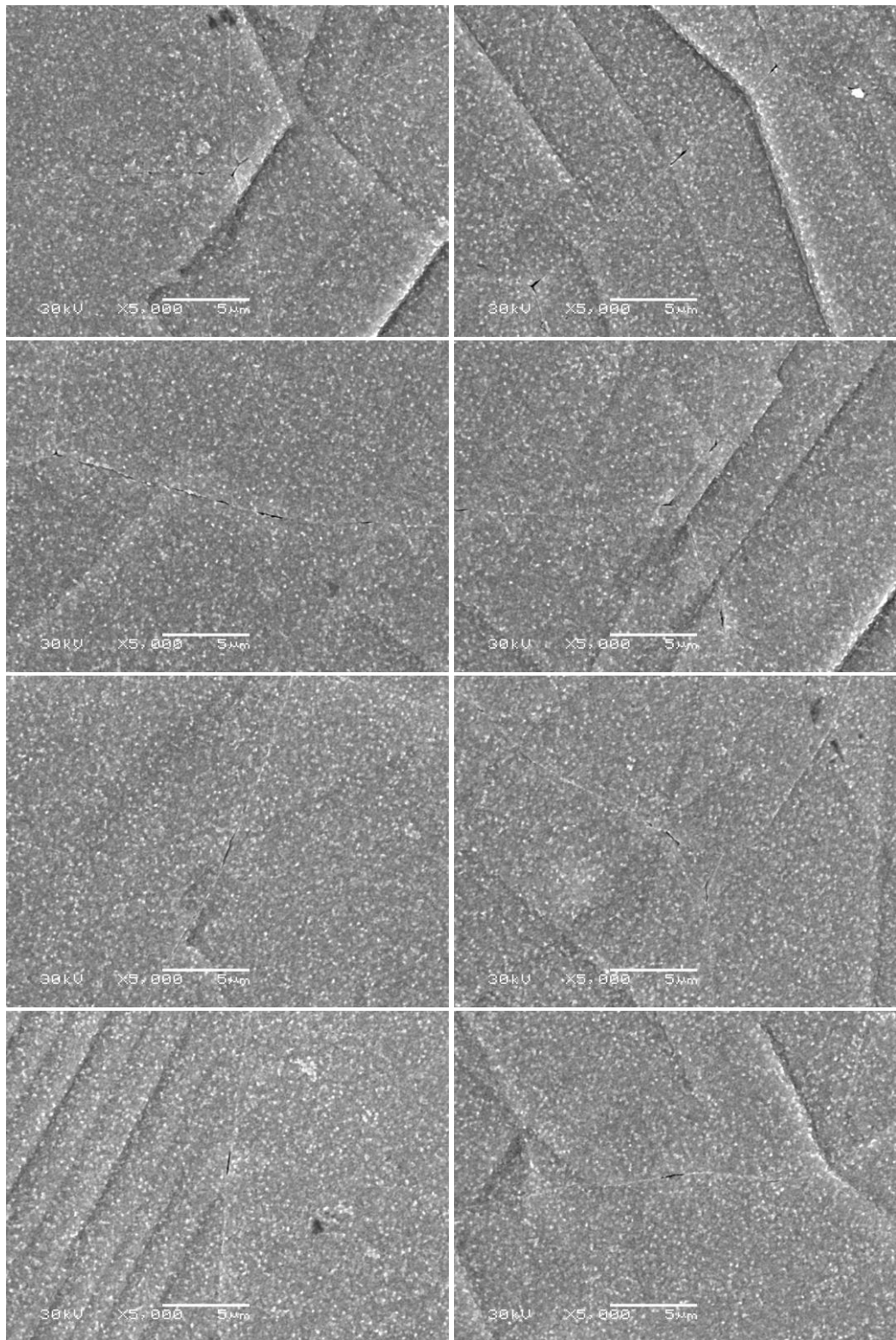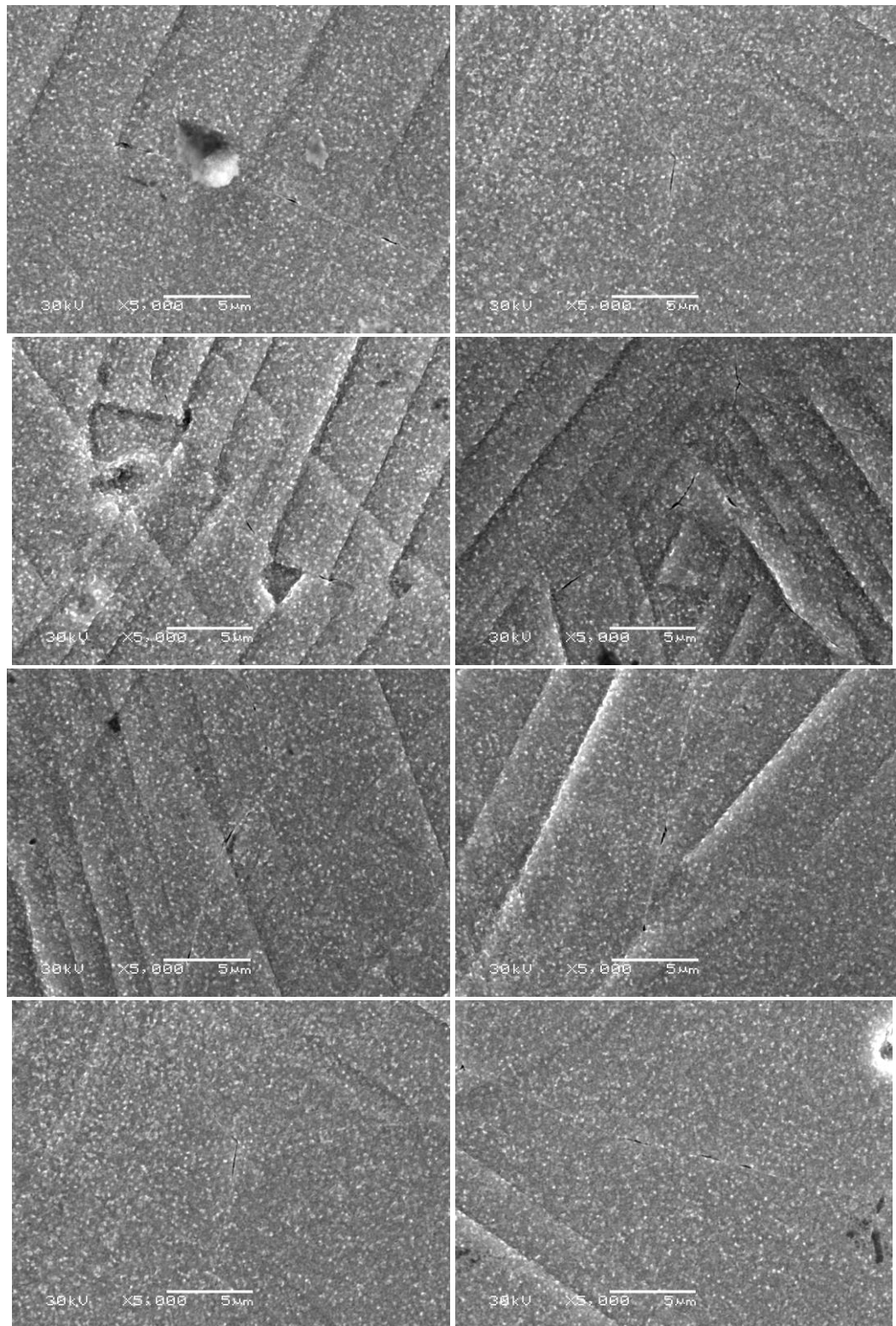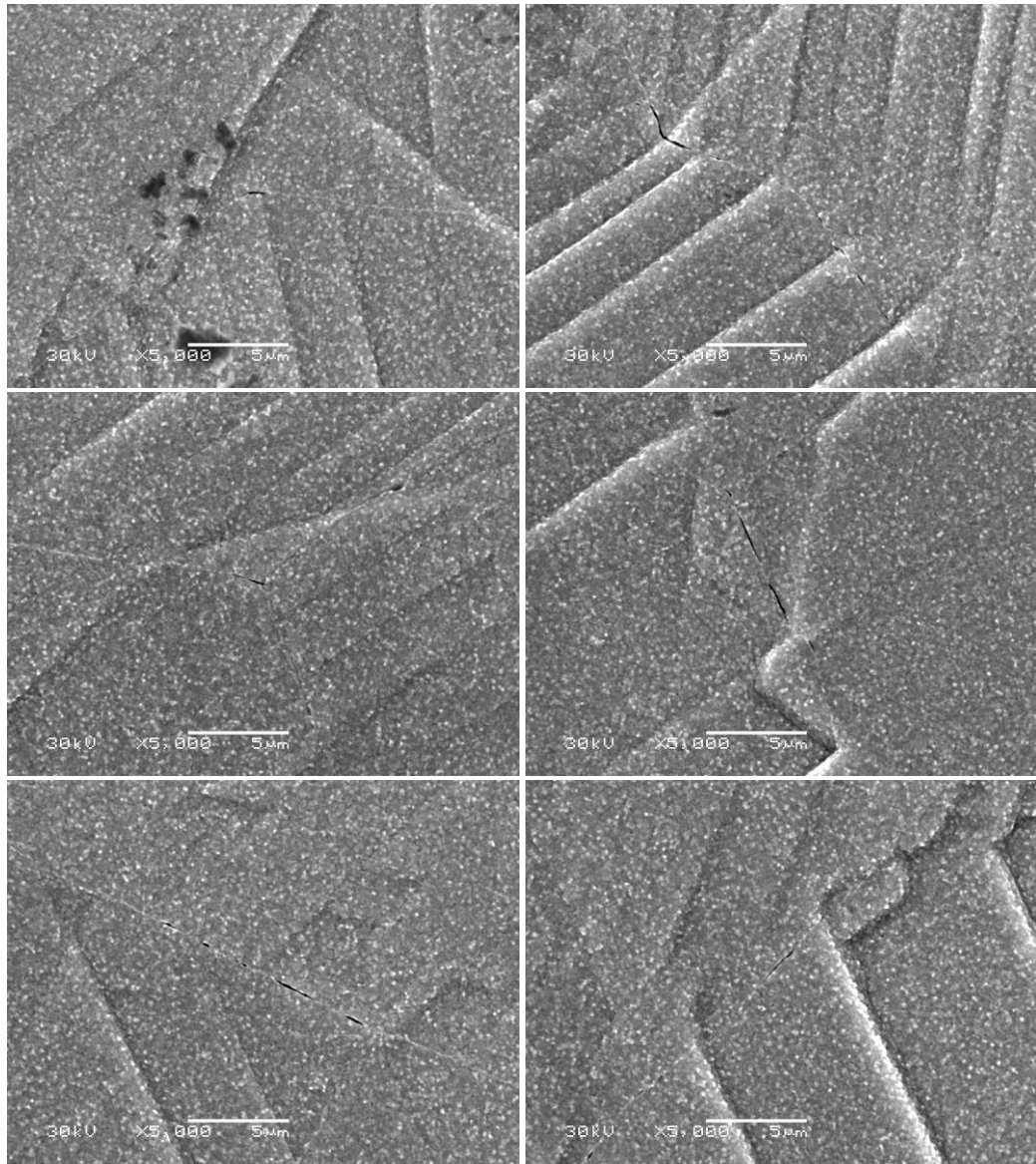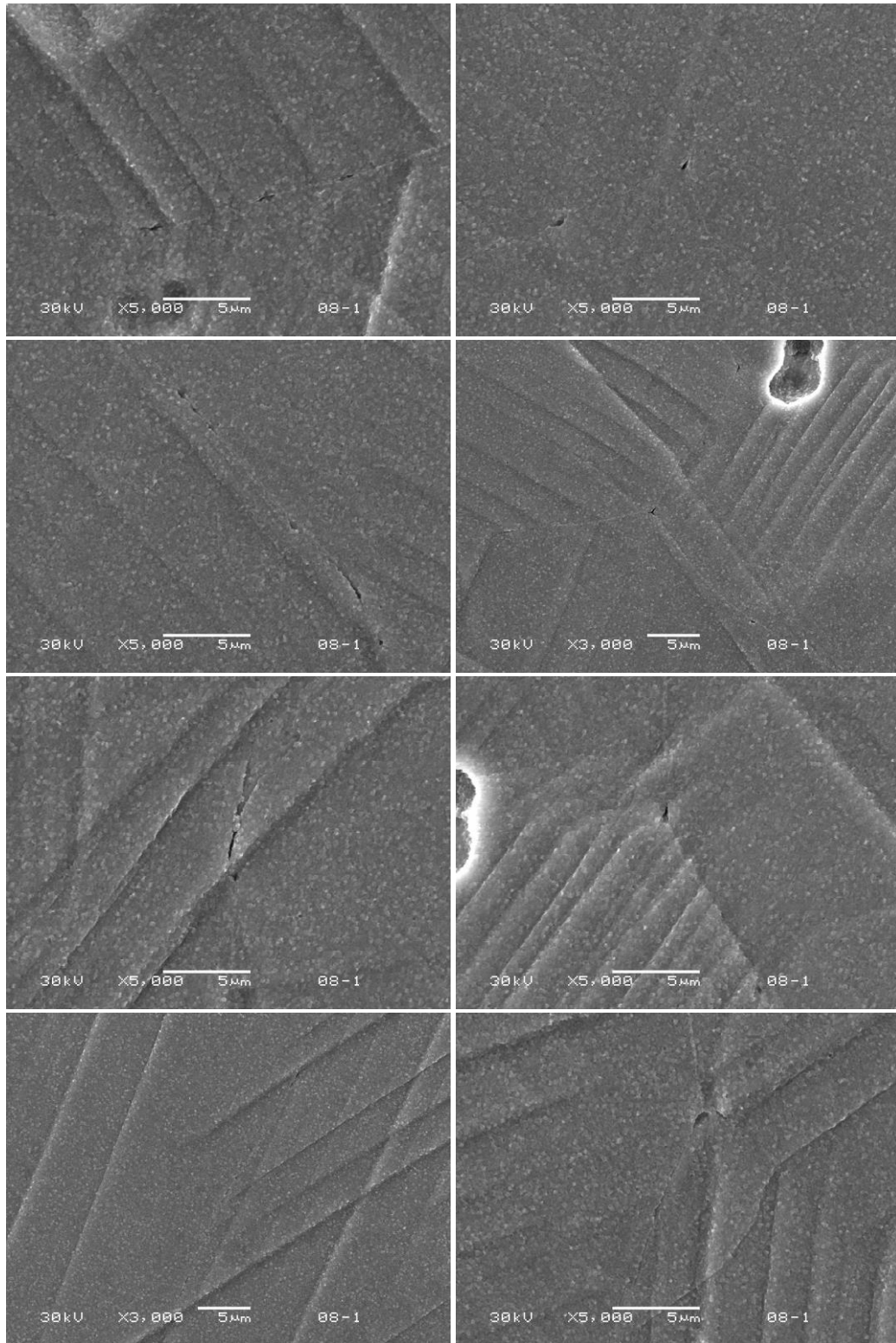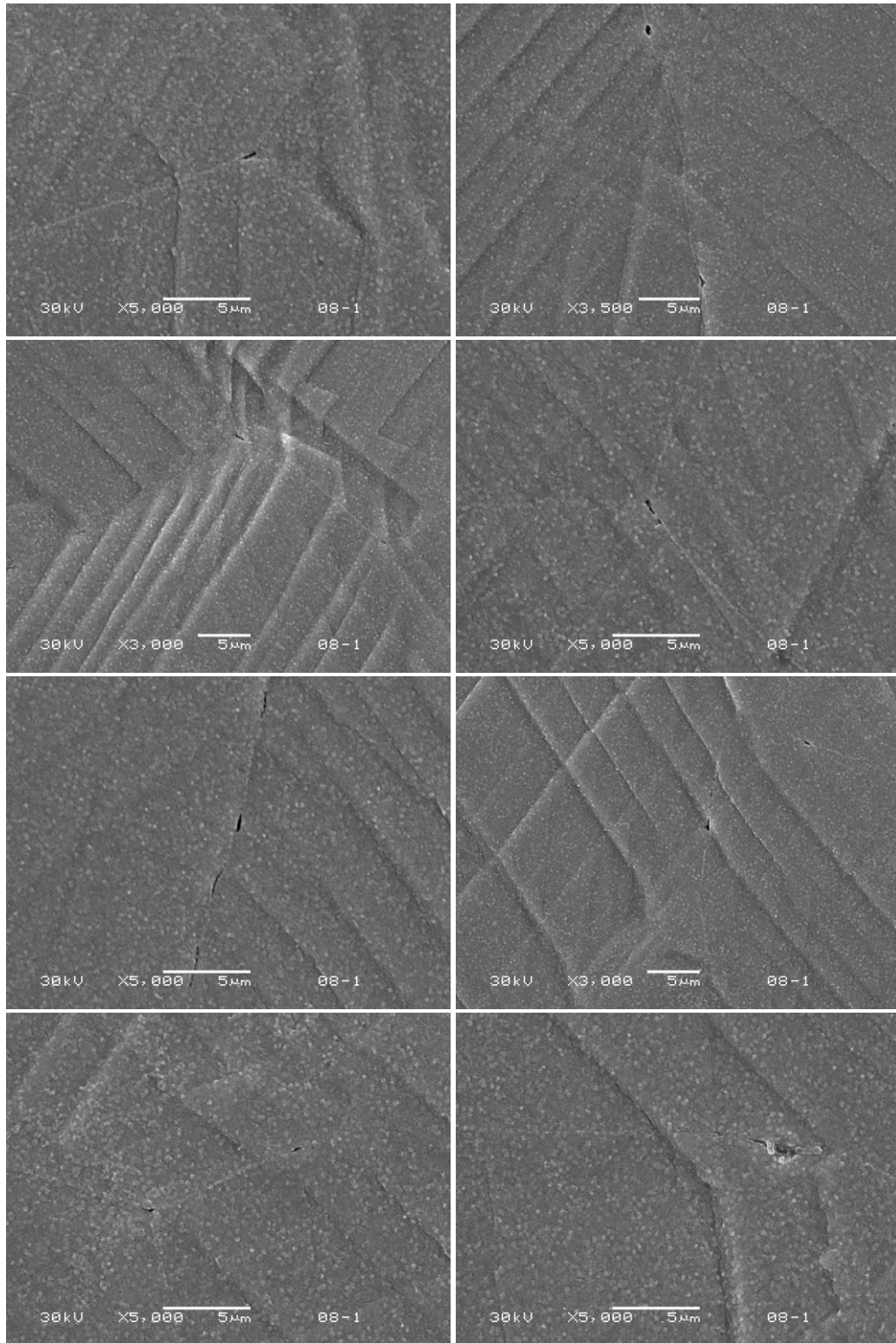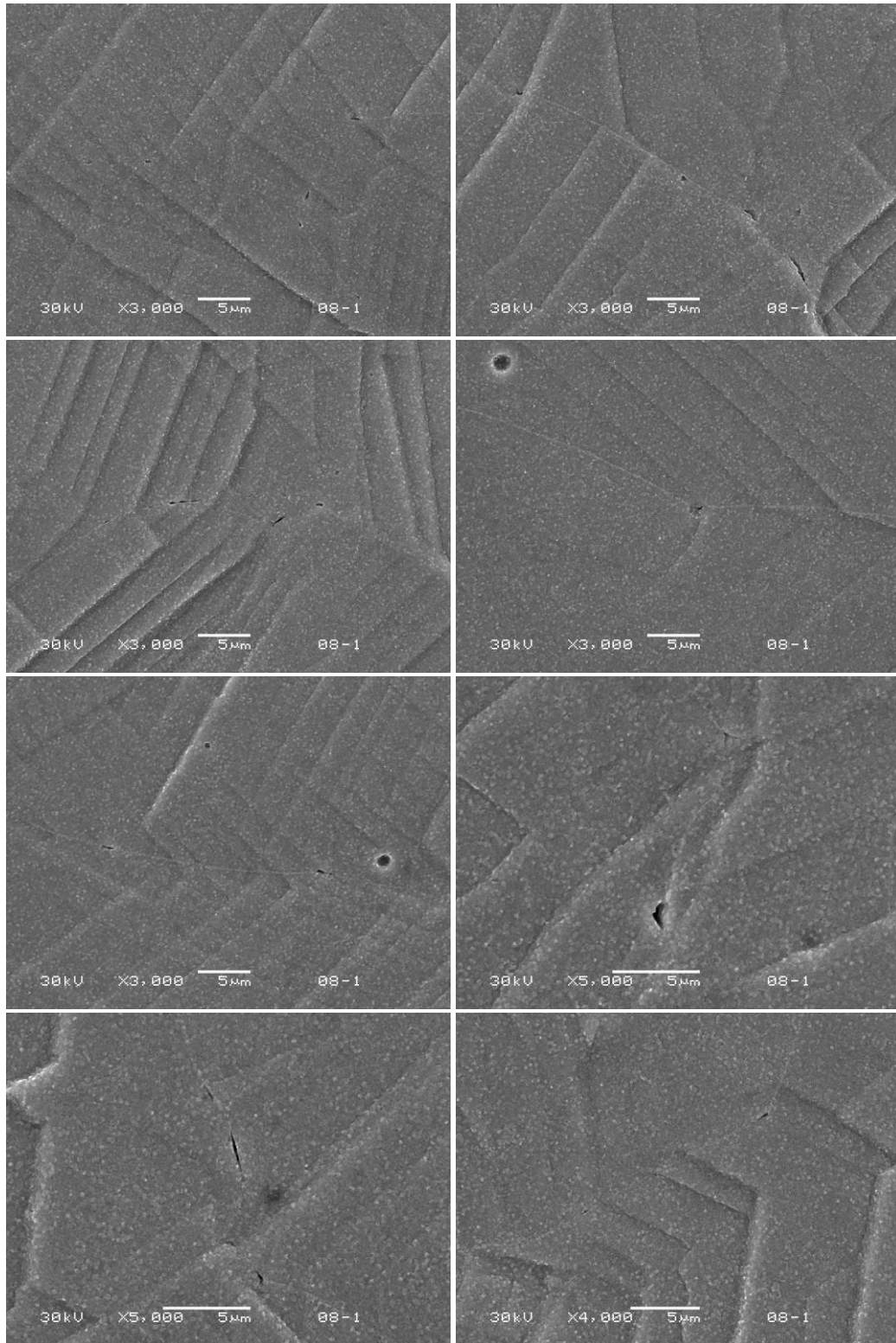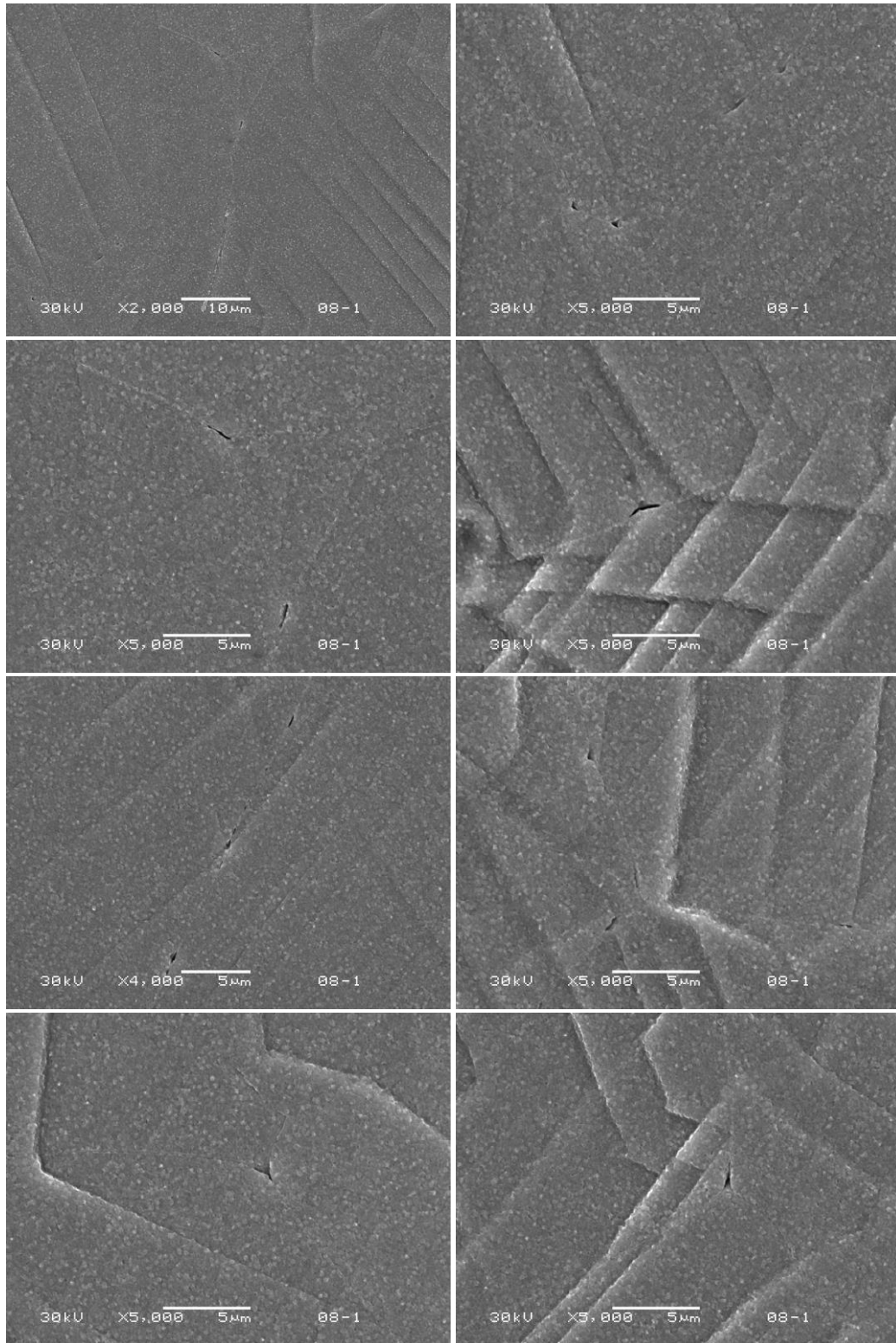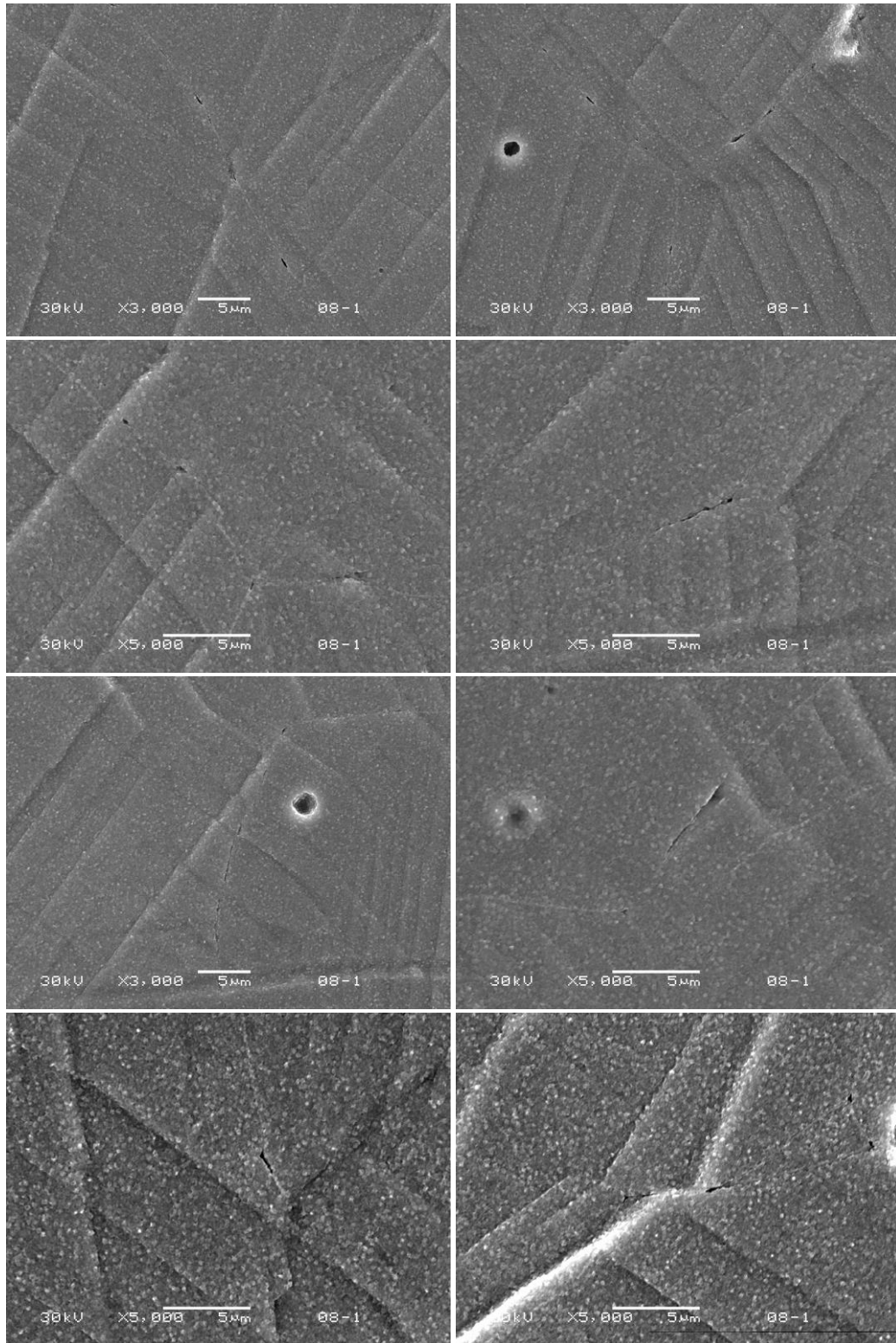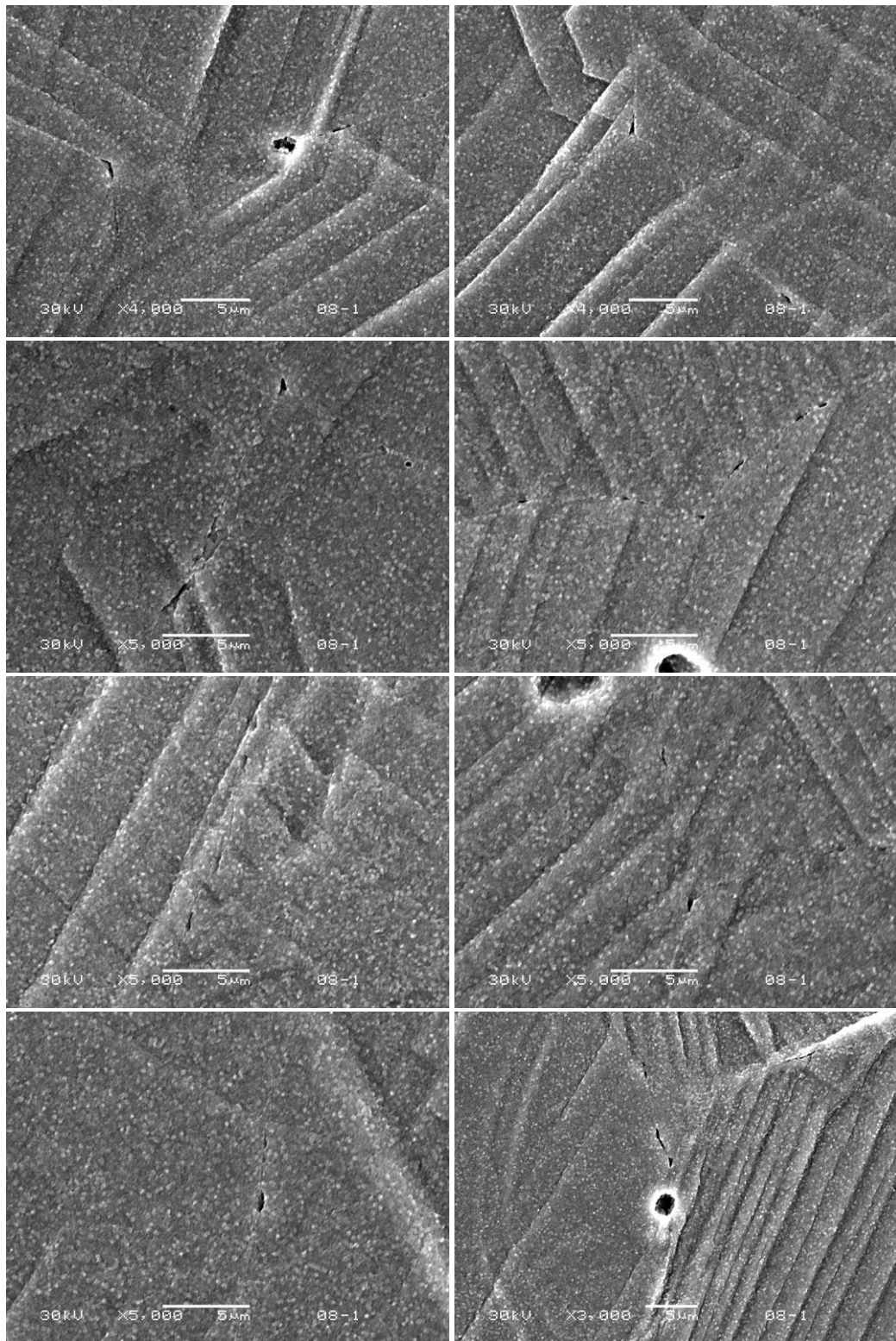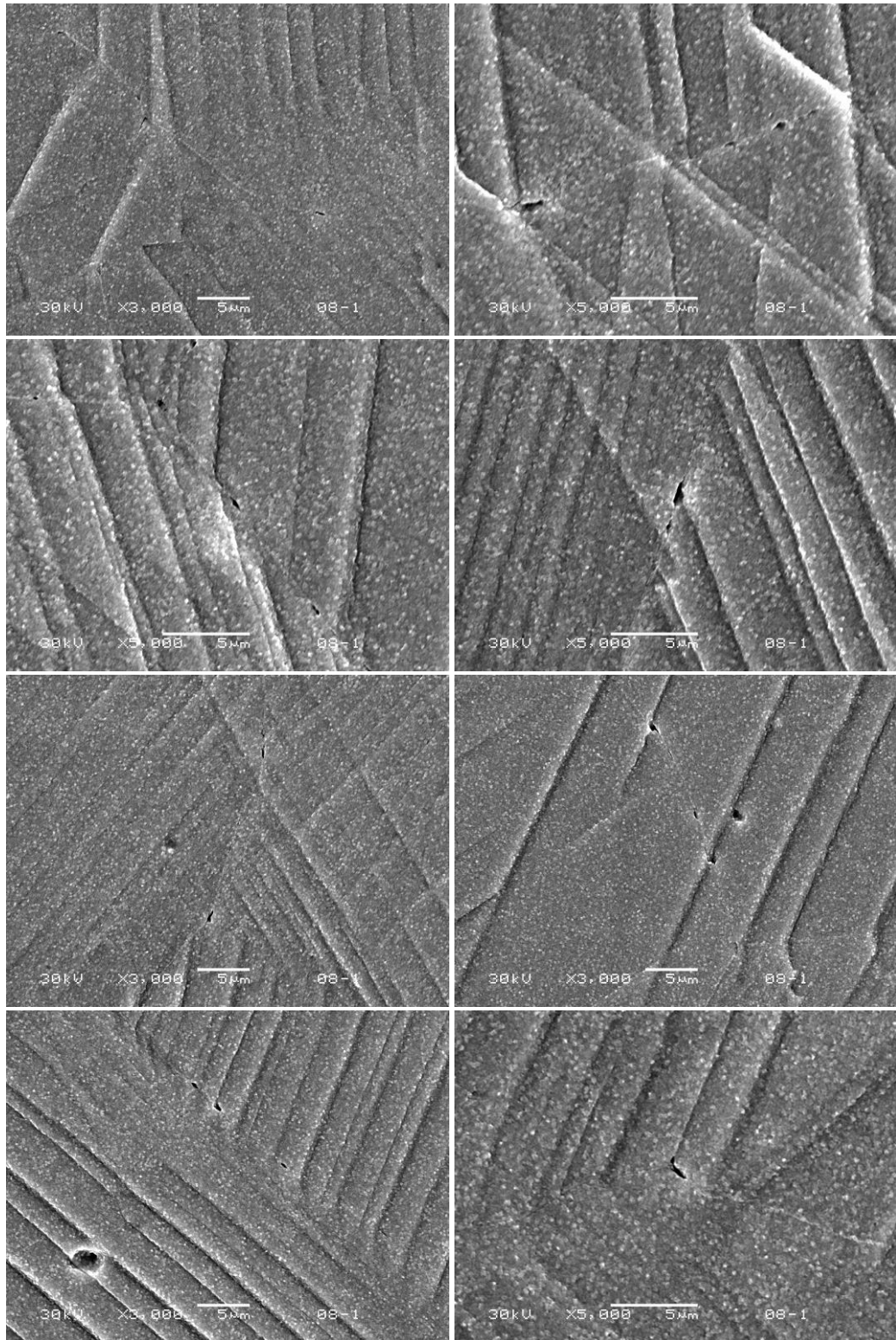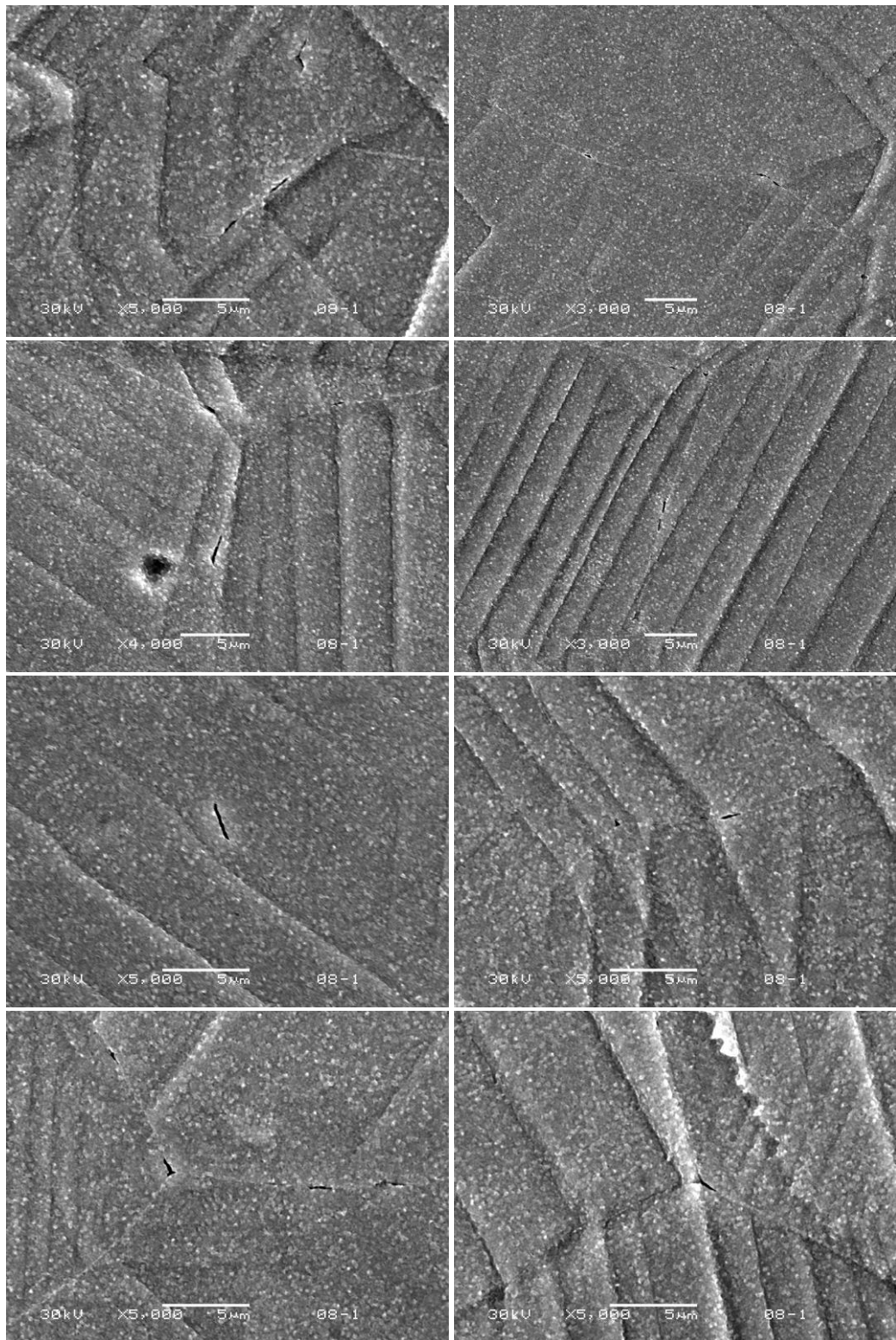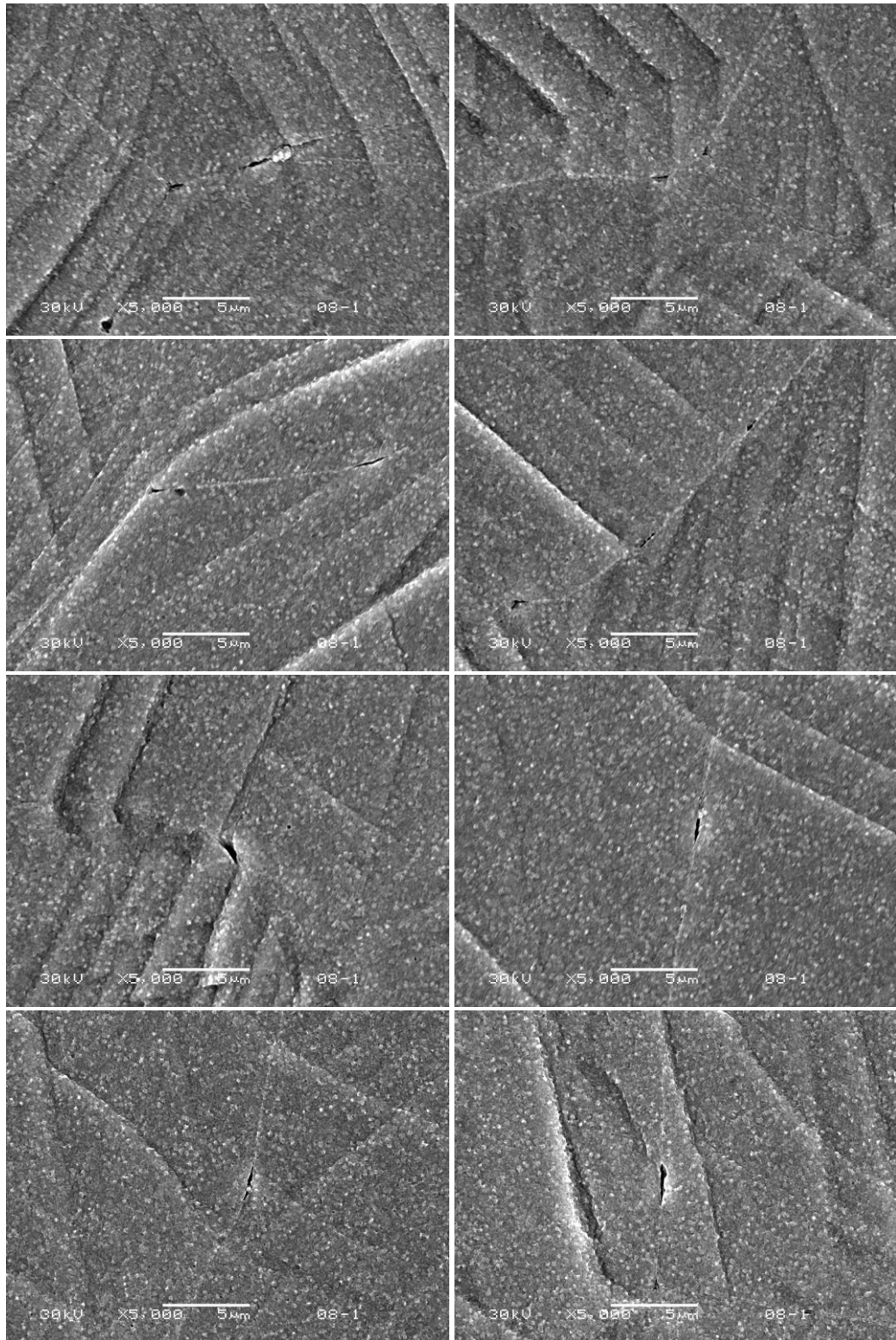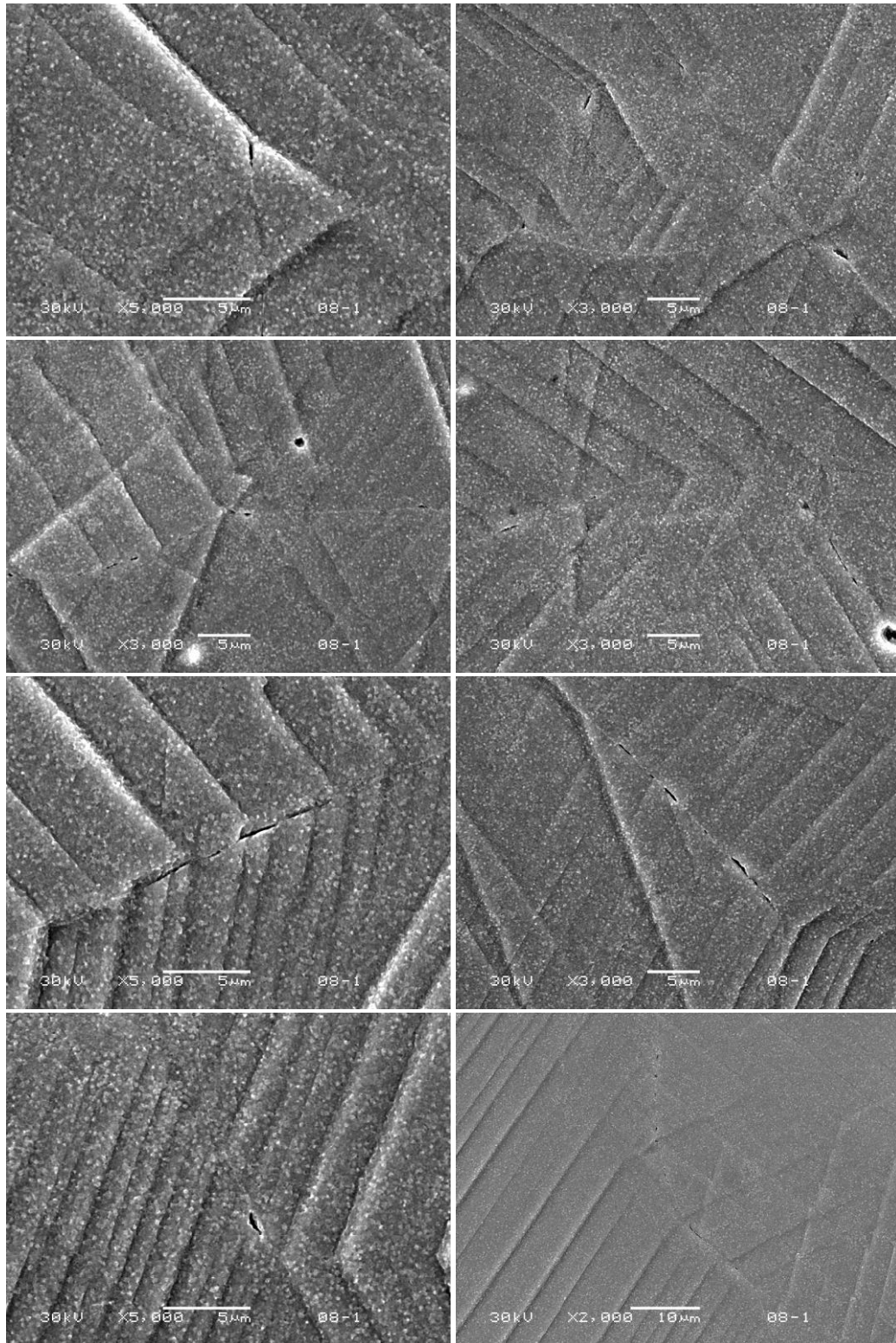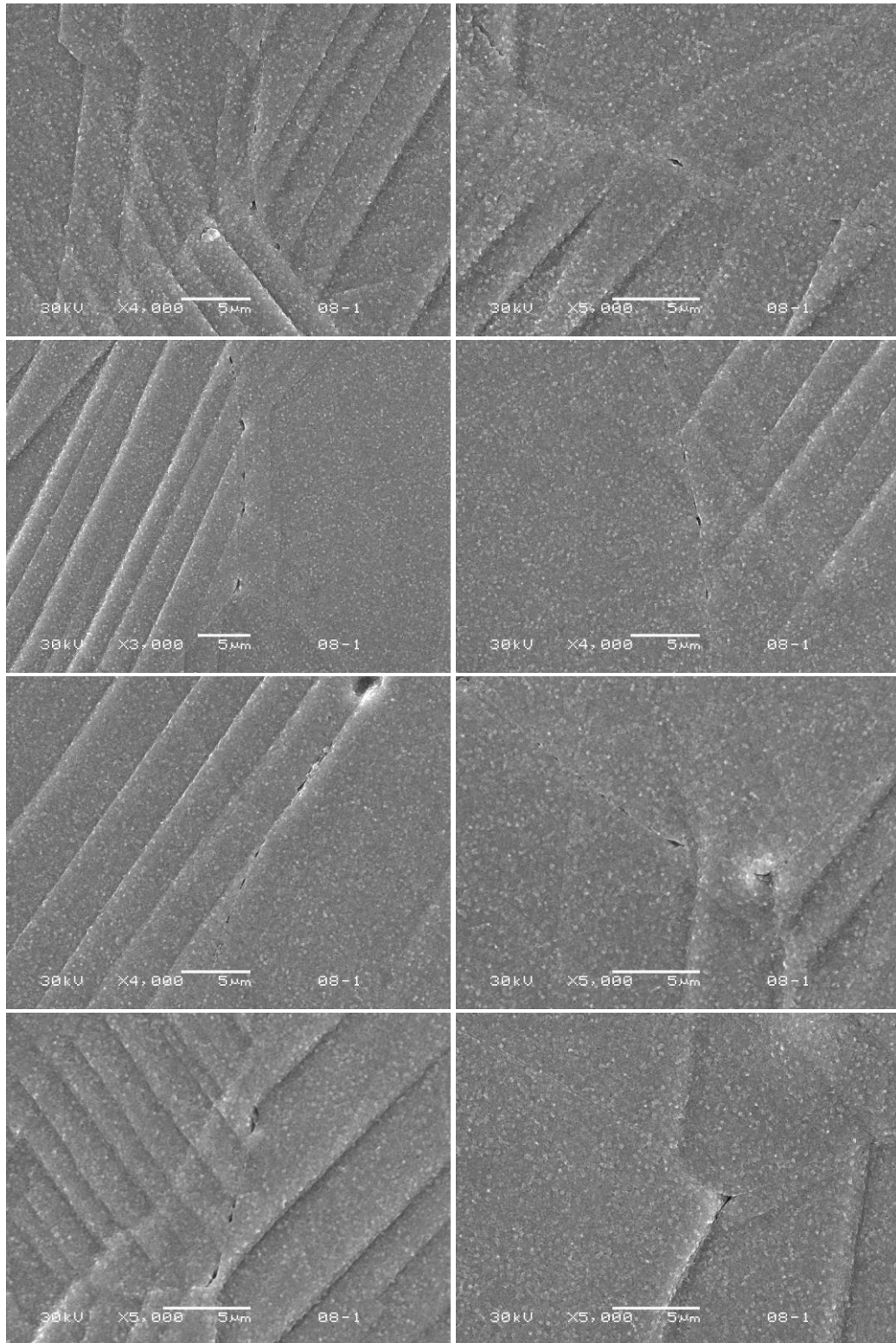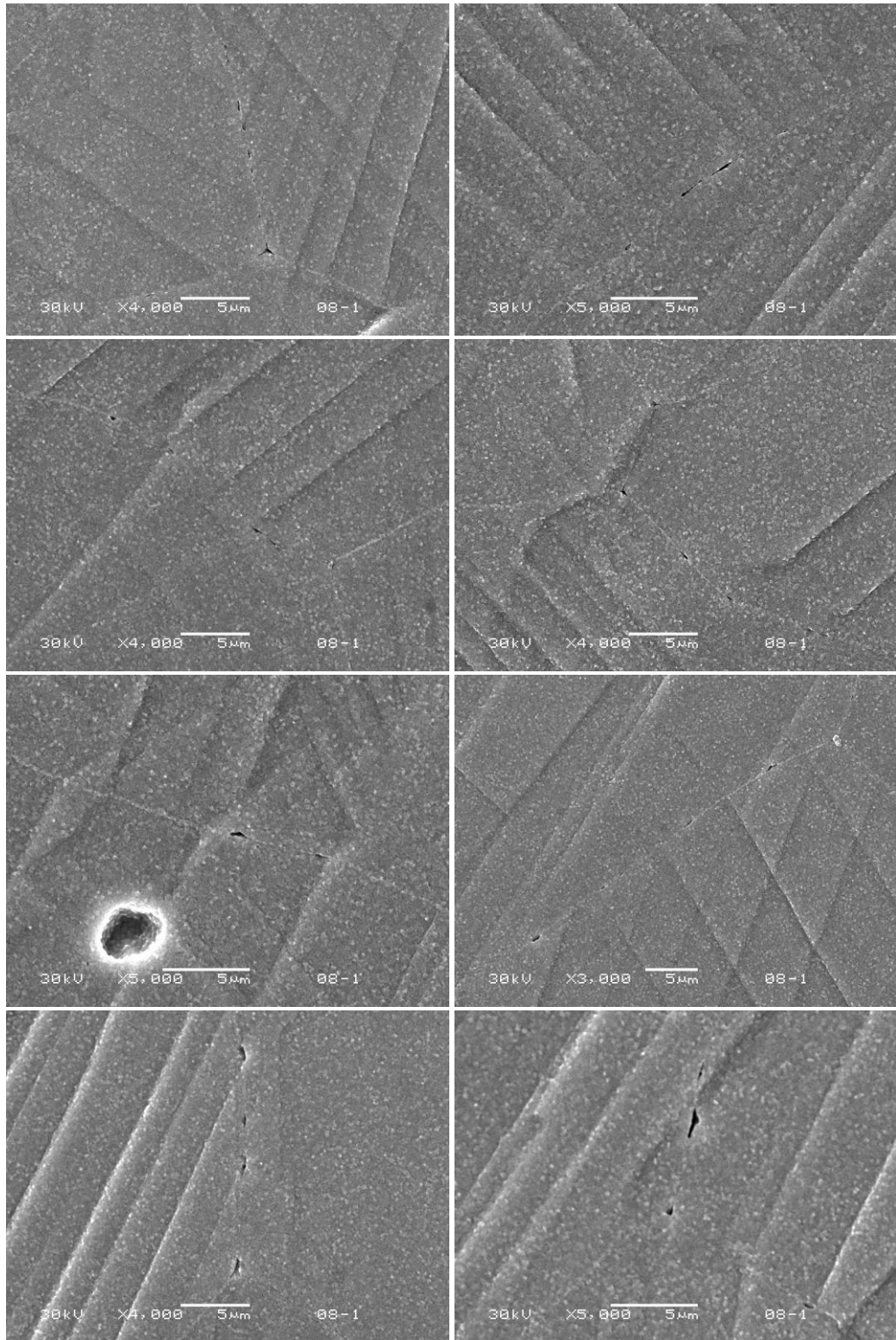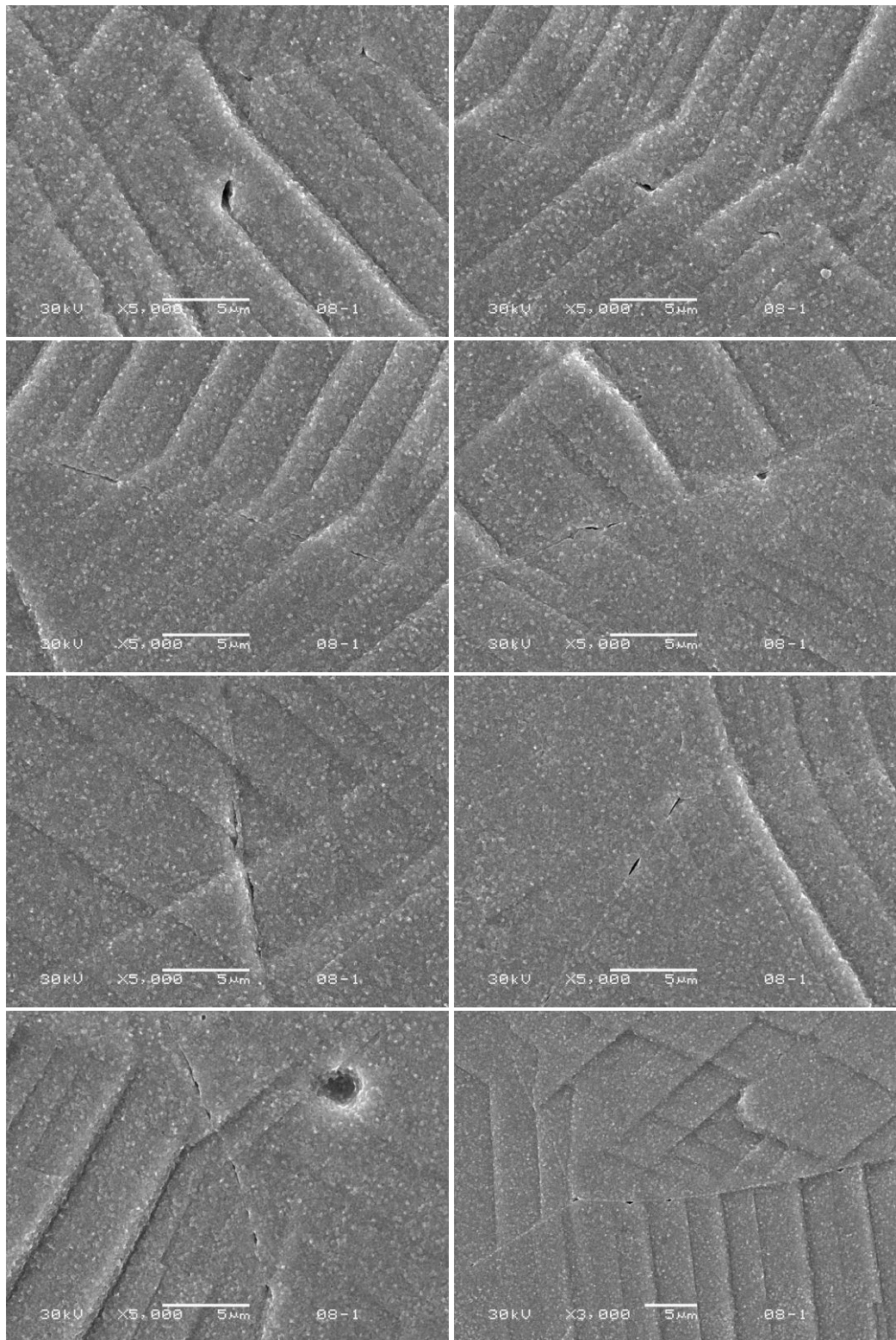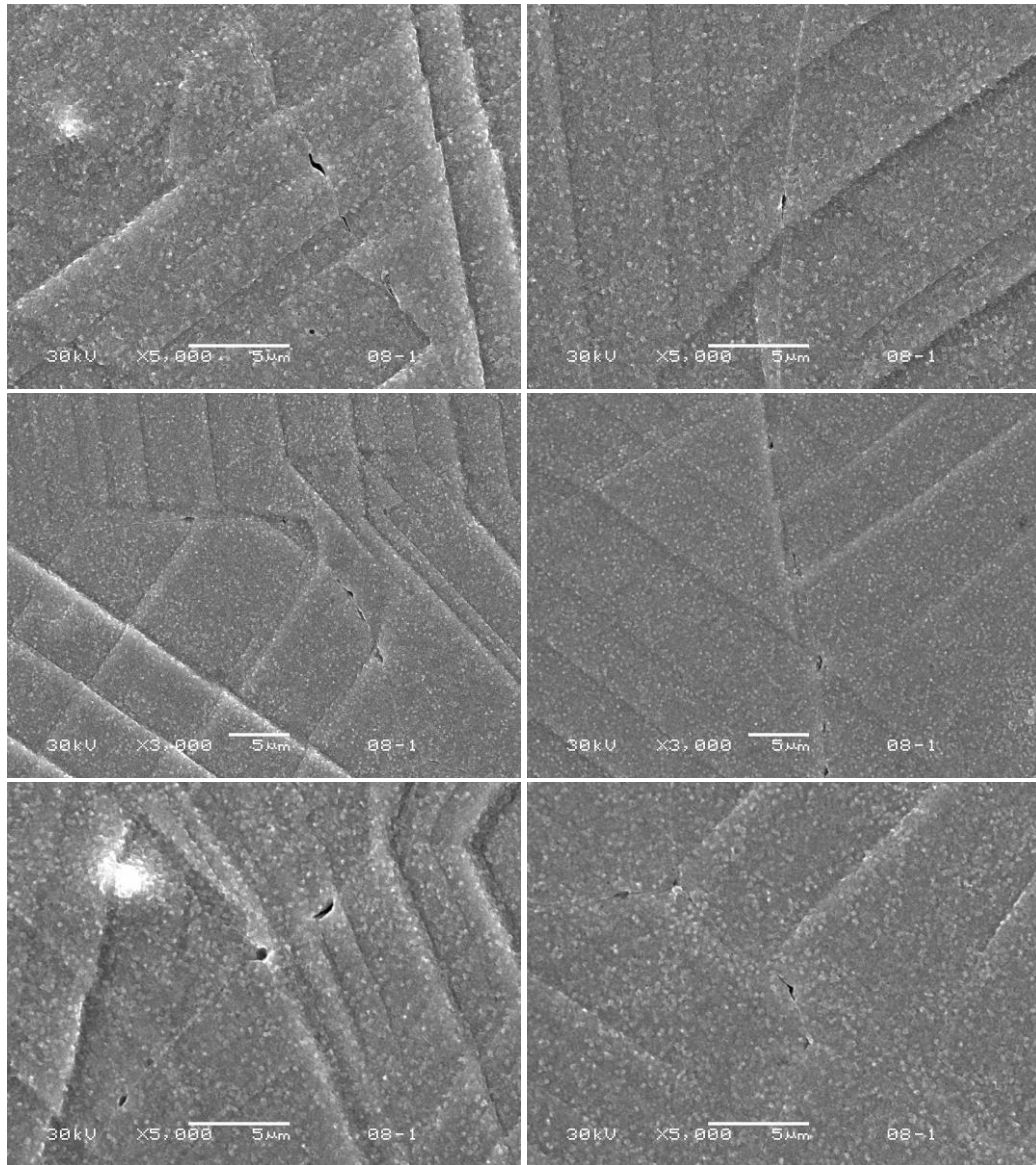
Figure E.1 SEM images of crack initiation sites.

REFERENCES

[1]   NRC, U.S. Nuclear Regulatory Commission 2013-2014 Information Digest, 2013.

[2]   S.J. Zinkle, G.S. Was, Acta Mater. 61 (2013) 735.

[3]   K. Fukuya, J. Nucl. Sci. Technol. 50 (2013) 213.

[4]   P. Scott, J. Nucl. Mater. 211 (1994) 101.

[5]   J.T. Busby, G.S. Was, E.A. Kenik, J. Nucl. Mater. 302 (2002) 20.

[6]   K. Fukuya, M. Nakano, K. Fujii, T. Torimaru, J. Nucl. Sci. Technol. 41 (2004) 594.

[7]   Z. Jiao, G.S. Was, J. Nucl. Mater. 408 (2011) 246.

[8]   T. Onchi, K. Dohi, N. Soneda, M. Navas, M.L. Castano, J. Nucl. Mater. 340 (2005) 219.

[9]   E.A. West, M.D. McMurtrey, Z. Jiao, G.S. Was, Metall. Mater. Trans. A 43 (2011) 136.

[10]   M. Le Millier, O. Calonne, J. Crépin, C. Duhamel, Proc. Fifteenth Int. Conf. Environ. Degrad. Mater. Nucl. Power Syst. Water React. (2011) 1293.

[11]   Z. Jiao, J. Busby, R. Obata, G.S. Was, in:, 12th Int. Conf. Degrad. Mater. Nucl. Power Syst., Warrendale, PA, 2005, p. 379.

[12]   F. Gioacchino, J. Quinta da Fonseca, Exp. Mech. 53 (2012) 743.

[13]   H.S. Khatak, B. Raj, Corrosion of Austenitic Stainless Steels : Mechanism, Mitigation and Monitoring, Narosa Pub. House ; Distribution in North America only by ASM International, New Delhi Materials Park, Ohio, 2002.

[14]   D. Peckner, I.M. Bernstein, Handbook of Stainless Steels, McGraw-Hill, New York, 1977.

[15]   M. Milititsky, N. De Wisplaere, R. Petrov, J.E. Ramos, A. reguly, H. Hanninen, Mater. Sci. Eng. A (2008) 289.

[16]   J.R. Davis, ed., Metals Handbook Desk Edition, 2nd ed., ASM International, 1998.

[17]   D.J. Kotecki, Weld. J. 78 (1999) 180s.

[18]   D.J. Kotecki, Weld. J. (2000) 346s.

[19]   T. Angel, J. Iron Steel Inst. 177 (1954) 165.

[20] A.F. Padilha, P.R. Rios, Isij Int. 42 (2002) 325.

[21] J.L. R. Voicu E. Andrieu, D. Poquillon, J. furtado, Mater. Sci. Eng. A (2008).

[22] ASTM International, ASTM Special Technical Publication No. 264, "Report on Stress-Corrosion Cracking of Austenitic Chromium-Nickel Stainless Steels," Philadelphia, 1960.

[23] ASTM International, ASTM Special Technical Publication 656 "Intergranular Corrosion of Stainless Alloys," Philadelphia, 1978.

[24] B.M. Gordon, Corrosion 69 (2013) 1039.

[25] B. Leffler, Stainless-Stainless Steels and Their Properties, 1998.

[26] BWR Normal Water Chemistry Guidelines: 1986 Revision, Palo Alto, 1986.

[27] R. Asay, J. Blok, J. Holloway, Water Quality in Boiling Water Reactors, Palo Alto, 1980.

[28] BWR Water Chemistry Guidelines - 2000 Revision, Palo Alto, 2000.

[29] G. Sachs, Zeitschrift Des Vereines Dtsch. Ingenieure 72 (1928) 734.

[30] G.I. Taylor, J. Inst. Met. 62 (1938) 307.

[31] J.F.W. Bishop, R. Hill, Philos. Mag. (London, Engl. 1945) 42 (1951) 1298.

[32] W.F. Hosford, Mechanical Behavior of Materials, 2nd ed., Cabridge University Press, Cambridge, 2010.

[33] W.T. Read, W. Shockley, Phys. Rev. 78 (1950) 275.

[34] M.L. Krongber, F.H. Wilson, Trans. Am. Inst. Min. Metall. Eng. 185 (1949) 501.

[35] G.C. Hasson, G. C, Scr. Metall. 5 (1971) 889.

[36] A.P. Sutton, R.W. Balluffi, Acta Metall. 35 (1987) 2177.

[37] A. Seki, O. Hellman, S. Tanaka, Scr. Mater. 34 (1996) 1867.

[38] A. Needleman, J. Mech. Phys. Solids 38 (1990) 289.

[39] V.R. Coffman, J.P. Sethna, Phys. Rev. B 77 (2008).

[40] M. Kamaya, Y. Kawamura, T. Kitamura, Int. J. Solids Struct. 44 (2007) 3267.

[41] J. Wan, Z. Yue, J. Mater. Sci. Technol. 19 (2003) 303.

[42] T. Tsuru, Y. Kaji, T. Tsukada, Y. Shibutani, 2 (2011) 20.

[43] B.T. Britton, A.J. Wilkinson, Acta Mater. 60 (2012) 5773.

[44] J.D. Eshelby, F.C. Frank, F.R.N. Nabarro, Philos. Mag. 42 (1951) 351.

[45] Z. Shen, R. Wagoner, W. Clark, Acta Metall. 36 (1988) 3231.

[46] T.C. Lee, I.M. Robertson, H.K. Birnbaum, Scr. Metall. 23 (1989) 799.

[47] T.C. Lee, I.M. Robertson, H.K. Birnbaum, Acta Metall. Mater. 40 (1992) 2569.

[48] P. Wo, A. Ngan, J. Mater. Res. 19 (2004) 189.

[49] B. Lagow, I. Robertson, M. Jouiad, D. Lassila, T. Lee, H. Birnbaum, Mater. Sci. Eng. A 309-310 (2001) 445.

[50] G.E. Dieter, Mechanical Metallurgy, 3rd ed., McGraw-Hill, New York, 1986.

[51] F.A. Sherrill, M.C. Wittels, T.H. Blewitt, J. Appl. Phys. 28 (1957) 526.

[52] G.T. Gray, Acta Metall. 36 (1988) 1745.

[53] J.W. Christian, S. Mahajant, Prog. Mater. Sci. 39 (1995) 1.

[54] G.Y. Chin, W.F. Hosford, D.R. Mendorf, Proc. R. Soc. London Ser. A-Mathematical Phys. Sci. 309 (1969) 433.

[55] A. Venables, J. Phys. Chem. Solids 25 (1964) 693.

[56]   a. Rohatgi, K.S. Vecchio, G.T. Gray, III, Acta Mater. 49 (2001) 427.

[57] T.H. Blewitt, R.R. Coltman, J.K. Redman, J. Appl. Phys. 28 (1957) 651.

[58]   a. S. Hamada, L.P. Karjalainen, R.D.K. Misra, J. Talonen, Mater. Sci. Eng. A 559 (2013) 336.

[59] Y.L. Gong, C.E. Wen, X.X. Wu, S.Y. Ren, L.P. Cheng, X.K. Zhu, Mater. Sci. Eng. A 583 (2013) 199.

[60] C.S. Barrett, Structure of Metals, McGraw-Hill Book Company, Inc., New York, 1943.

[61]   M. Kamaya, J. Quinta Da Fonseca, L.M. Li, M. Preuss, Appl. Mech. Mater. 7-8 (2007) 173.

[62]   A.F. Bower, Applied Mechanics of Solids, CRC Press, Boca Raton, 2009.

[63]   D.J. Edwards, E.P. Simonen, F.A. Garner, L.R. Greenwood, B.M. Oliver, S.M. Bruemmer, J. Nucl. Mater. 317 (2003) 32.

[64]   T.R. Allen, in:, Radiat. Eff. Solids, 2007, pp. 99–121.

[65]   P.R. Okamoto, H. Wiedersich, J. Nucl. Mater. 53 (1974) 336.

[66]   T.R. Allen, G.S. Was, in:, Radiat. Eff. Solids, 2007, pp. 123–151.

[67]   C. Bailat, A. Almazouzi, N. Baluc, R. Schaublin, F. Groschel, M. Victoria, J. Nucl. Mater. 283 (2000) 446.

[68]   C. Bailat, F. Groschel, M. Victoria, L. Fusion Technol Grp Ctr Rech Phys Plasmas Ecolr Polytechn Fed, C. Mat Res Dept Riso Natl Lab, N. Mat Sci Div Lawrence Livermore, lab, in:, Int. Work. Basic Asp. Differ. Irradiat. Eff. Between FCC, BCC HCP Met. Alloy., Elsevier Science Bv, Cangas De Onis, Spain, 1998, pp. 283–288.

[69]   J. Brimhall, J. Cole, J. Vetrano, S. Bruemmer, MRS Proc. 373 (1994) 177.

[70]   K. Farrell, T.S. Byun, N. Hashimoto, J. Nucl. Mater. 335 (2004) 471.

[71]   T.S. Byun, N. Hashimoto, K. Farrell, M. Mineral, O.F.E.S.U.S.D.O.E.O.R.N. Mat Soc, M. Lab, D. Ceram, in:, Symp. Microstruct. Process. Irradiat. Mater. Held Annu. Meet. Miner., Elsevier Science Bv, San Francisco, CA, 2005, pp. 303–315.

[72]   N. Hashimoto, T.S. Byun, J. Nucl. Mater. 367-370 (2007) 960.

[73]   Y. Matsukawa, Y.N. Osetsky, R.E. Stoller, S.J. Zinkle, Mater. Sci. Eng. A 400-401 (2005) 366.

[74]   Y. Matsukawa, Y.N. Osetsky, R.E. Stoller, S.J. Zinkle, Philos. Mag. 88 (2008) 581.

[75]   H. Nishioka, K. Fukuya, K. Fujii, Y. Kitsunai, J. Nucl. Sci. Technol. 45 (2008) 274.

[76]   D.J. Edwards, B.N. Singh, J. Nucl. Mater. 329-333 (2004) 1072.

[77]   D.J. Edwards, B.N. Singh, J.B. Bilde-Sørensen, J. Nucl. Mater. 342 (2005) 164.

[78]   T.R. Allen, J.I. Cole, J. Gan, G.S. Was, R. Dropek, E. a. Kenik, J. Nucl. Mater. 342 (2005) 90.

[79]   R.P. Tucker, J. Appl. Phys. 40 (1969) 400.

[80]   Z. Jiao, G.S. Was, J. Nucl. Mater. 382 (2008) 203.

[81]   J. Sharp, Acta Metall. 22 (1974) 449.

[82]   M. Sauzay, K. Bavard, W. Karlsen, J. Nucl. Mater. 406 (2010) 152.

[83]   P. Evrard, M. Sauzay, J. Nucl. Mater. 405 (2010) 83.

[84]   G.S. Was, P.L. Andresen, Corrosion 63 (2007) 19.

[85]   O.K. Chopra, a. S. Rao, J. Nucl. Mater. 409 (2011) 235.

[86]   Z. Jiao, J.T. Busby, G.S. Was, J. Nucl. Mater. 361 (2007) 218.

[87]   S.M. Bruemmer, S. Mat Res, in:, C.A. Schuh, M. Kumar, C.B. Carter, V. Randle (Eds.), Symp. Interfacial Eng. Optim. Prop. III Held 2004 MRS Spring Meet., Materials Research Society, San Francisco, CA, 2004, pp. 101–110.

[88]   E.A. West, G.S. Was, J. Nucl. Mater. 441 (2013) 623.

[89]   E.A. West, G.S. Was, J. Nucl. Mater. 408 (2011) 142.

[90]   G.S. Was, B. Alexandreanu, J. Busby, Adv. Fract. Fail. Prev. Pts 1 2 261-263 (2004) 885.

[91]   T.R. Bieler, P. Eisenlohr, F. Roters, D. Kumar, D.E. Mason, M. a. Crimp, D. Raabe, Int. J. Plast. 25 (2009) 1655.

[92]   P.L. Andresen, F.P. Ford, Int. J. Press. Vessel. Pip. 59 (1994) 61.

[93]   M. Kamaya, T. Kitamura, Int. J. Fract. 130 (2004) 787.

[94]   B. Simkin, Scr. Mater. 49 (2003) 149.

[95]   H. Mykura, Grain-Boundary Structure and Kinetics, ASM, Metals Park, Ohio, 1980.

[96]   F.B. Pickering, Proc. Conf. Stainl. Steels 84 (1985) 2.

[97]   R.E. Schramm, R.P. Reed, Metall. Trans. A 6A (1975) 1345.

[98] C.G. Rhodes, A.W. Thompson, Metall. Trans. A 8A (1977) 901.

[99] S.M. Bruemmer, D.J. Edwards, B.W. Arcy, L.A. Charlot, in:, Proc. Ninth Int. Symp. Environ. Degrad. Mater. Nucl. Power Syst. - Water React., 1999, pp. 1079–1087.

[100] E.A. Kenik, T. Inazumi, G.E.C. Bell, J. Nucl. Mater. 183 (1991) 145.

[101] J.W. Simmons, Mater. Sci. Eng. a-Structural Mater. Prop. Microstruct. Process. 207 (1996) 159.

[102] G.Y. Chin, W.L. Mammel, M.T. Dolan, Trans. Metall. Soc. AIME 239 (1967) 1111.

[103] J.P. Hirth, J. Lothe, Theory of Dislocations, Wilen, John & Sons, Inc., 1982.

[104] T.A. Khraishi, H.M. Zbib, T.D. de la Rubia, M. Victoria, Philos. Mag. Lett. 81 (2001) 583.

[105] N. Hashimoto, T.S. Byun, K. Farrell, M. Mineral, O.F.E.S.U.S.D.O.E.O.R.N. Mat Soc, M. Lab, D. Ceram, in:, Symp. Microstruct. Process. Irradiat. Mater. Held Annu. Meet. Miner., Elsevier Science Bv, San Francisco, CA, 2005, pp. 295–302.

[106] R.E. Stoller, M.B. Toloczko, G.S. Was, a. G. Certain, S. Dwaraknath, F. a. Garner, Nucl. Instruments Methods Phys. Res. Sect. B Beam Interact. with Mater. Atoms 310 (2013) 75.

[107] G. Frens, Nat. Phys. Sci. 241 (1973) 20.

[108] K.C. Grabar, R.G. Freeman, M.B. Hommer, M.J. Natan, 67 (1995) 1217.

[109] A.D. Kammers, S. Daly, Exp. Mech. 53 (2013) 1333.

[110] Z. Jiao, G.S. Was, J. Nucl. Mater. 407 (2010) 34.

[111] A.J. Wilkinson, G. Meaden, D.J. Dingley, Ultramicroscopy 106 (2006) 307.

[112] B. Alexandreanu, G.S. Was, Philos. Mag. A 81 (2001) 1951.

[113] X.J. Meng, Q.L. Liu, A.M. Zhu, Q.G. Zhang, J. Memb. Sci. 360 (2010) 276.

[114] K. Stephenson, (n.d.).

[115] J. Kacher, I.M. Robertson, Acta Mater. 60 (2012) 6657.

[116] G.S. Was, B. Alexandreanu, P. Andresen, M. Kumar, S. Mat Res, in:, C.A. Schuh, M. Kumar, C.B. Carter, V. Randle (Eds.), Symp. Interfacial Eng. Optim. Prop. III Held 2004 MRS Spring Meet., Materials Research Society, San Francisco, CA, 2004, pp. 87–100.

[117] M. Shimada, H. Kokawa, Z.J. Wang, Y.S. Sato, I. Karibe, in:, Symp. Comput. Thermodyn. Mater. Des., Pergamon-Elsevier Science Ltd, New Orleans, Louisiana, 2001, pp. 2331–2341.

[118] D. Wolf, Acta Metall. 37 (1989) 1983.

[119] E.A. West, G.S. Was, Nucl. Eng. Radiol. Sci. PhD (2010).

[120] B. Alexandreanu, PhD Thesis, University of Michigan: Ann Arbor, 2002.

[121] E.A. West, G.S. Was, J. Nucl. Mater. 408 (2011) 142.

[122] M. McMurtrey, G. Was, B. Cui, I. Robertson, L. Smith, D. Farkas, Int. J. Plast. 56 (2014) 219.

[123] J. Kacher, G.S. Liu, I.M. Robertson, Micron 43 (2012) 1099.

[124] B. Cui, J. Kacher, M. McMurtrey, G. Was, I. Robertson, Acta Mater. 65 (2014) 150.

[125] S. Plimpton, J. Comput. Phys. 117 (1995) 1.

[126] M. Briceño, J. Fenske, M. Dadfarnia, P. Sofronis, I.M. Robertson, J. Nucl. Mater. 409 (2011) 18.

[127] B. Cui, M.D. McMurtrey, G.S. Was, I. Robertson, Philos. Mag. Submitted (2014).

[128] T.M. Angeliu, P.L. Andresen, F.P. Ford, in:, Corrosion, 1998, p. Paper No. 262.

[129] P.L. Andresen, F.P. Ford, Mater. Sci. Eng. A 103 (1988) 167.

[130] Z. Jiao, (2014) Personal communication.

[131] M. Schutze, Mater. Sci. Technol. 4 (1988) 407.

[132] H.E. Evans, Int. Mater. Rev. 40 (1995) 1.