2010-09-02

Defect Detection Microscopy

Stuart Craig Rogers
Brigham Young University - Provo

Defect Detection Microscopy

Stuart C. Rogers

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

David T. Fullwood, Chair
Brent L. Adams
Mike P. Miles

Department of Mechanical Engineering

Brigham Young University

December 2010

ABSTRACT

Defect Detection Microscopy

Stuart C. Rogers

Department of Mechanical Engineering

Master of Science

The automotive industry's search for stronger lighter materials has been hampered in its desire to make greater use of Magnesium alloys by their poor formability below 150°C. One current challenge is to identify the complex structure and deformation mechanisms at work and determine which of these are primary contributors to the nucleation of defects.

Orientation Imaging Microscopy has been the most accessible tool for microstructural analysis over the past 15 years. However, using OIM to analyze defect nucleation sites requires prior knowledge of where the defects will occur because once the defects nucleate the majority of microstructural information is destroyed. This thesis seeks to contribute to the early detection of nucleation sites via three mechanisms: 1. Detection of cracks that have already nucleated, 2. Detection of surface topography changes that may indicate imminent nucleation and 3. Beam control strategies for efficiently finding areas of interest in a scan. Successive in-situ OIM scans of a consistent sample region while strain is increased, while using the three techniques developed in this thesis, will be employed in future work to provide a powerful defect analysis tool.

By analyzing retrieved EBSD patterns we are able to locate defect / crack sites via shadowing on the EBSD patterns. Furthermore, topographical features (and potentially regions of surface roughening) can be detected via changes in intensity metrics and image quality. Topographical gradients are currently only detectable in line with the beam incidence. It is therefore suggested that the tensile specimens to be examined are orientated such that the resulting shear bands occur preferentially to this direction.

The ability to refine the scan around these areas of interest has been demonstrated via an off-line adaptive scan routine that is implemented via the custom scan tool. A first attempt at a defect detection framework has been outlined and coded into MATLAB. These tools offer a first step to accessing the information about defect nucleation that researchers are currently seeking.

Keywords: [magnesium, defects, nucleation, EBSD, custom scan, voids, topography]

ACKNOWLEDGEMENTS

Firstly my thanks go to my advisor Dr. David Fullwood for providing the opportunity and means for me to come to BYU. His advice, support and code optimizing skills were invaluable in my completing this thesis. As members of my graduate committee, Dr. Brent Adams and Dr. Mike Miles also contributed greatly with their insights, support and encouragement through the trying times that culminated in this thesis. My fellow basement dwellers, as co-workers in the MSD research group, have all contributed with insights and code portions that have helped immensely. Ultimately, as always, my thanks go to my wife Lisa for all she has done in our family to enable me to complete this work. She has always been a blessing in our home and never more so than when I am not there.

TABLE OF CONTENTS

LIST OF FIGURES

1 INTRODUCTION

Material failure currently amounts to direct costs of $300 billion in annual losses within the US [1]. These failures are partially attributable to defects within the material causing localized stress concentrations and weak points; resulting in failure before the theoretical homogeneous/isotropic limits engineers currently design to.

Microstructure Sensitive Design (MSD) and other similar methodologies have proven successful in linking properties of a given material to its microstructure [2]. Various tools and approaches have been developed to help manipulate material microstructures to achieve optimum performance for a given application. There are, however, difficulties involved in applying these methods to “extreme value properties”, where toughness, ductility, formability and fatigue are of interest. These properties are linked to heterogeneous deformation, localization and crack/void nucleation and growth. As such, the first step is discovering relations between the local microstructure and defect nucleation. This thesis is focused on that issue.

In recent years the automotive industry’s focus on reduced weight and improved fuel efficiency has led to an increased use of Magnesium alloys, which have a potential weight saving of 60% relative to steel and 40% relative to aluminum [3-6]. However the use of magnesium has been severely limited by its poor ductility at ambient temperatures, limiting its range of industrial processing [7, 8]. This poor ductility is due to very convoluted deformation mechanisms at work

that are yet to be fully understood. This thesis aims to develop the current EBSD methods to identify defect nucleation, allowing for microstructural characterization. The thesis was motivated and funded by an NSF project that focuses on light metals for improved automotive efficiency. Hence the following two sections will discuss some relevant background on magnesium that motivates and provides direction for the later work.

1.1 Deformation Mechanisms in Magnesium

Research has been conducted into magnesium deformation and formability, both theoretical and physical, which points towards potential causes of failure location [9-19]. Some of these failure mechanisms are easier than others to link to specific microstructure features or failure initiation parameters (FIPs). The literature suggests that failure nucleation is linked to the presence of one or more of the following; Grain size, twinning (tension, compression and double), dislocation pile up, grain boundary sliding, triple point, stress concentration, activation of <c+a> slip, basal slip, twin interface (boundary), twin-twin intersect, slip-twin interaction, twin-grain boundary interaction and shear banding [15, 16, 18, 20-37]. These FIPs have yet to be fully characterized and compared for predominance within a strained sample.

Fully modeling the interactions of all of these contributory factors is difficult to the point of impossibility. The problem has always been that experimentally we are only able to see the defects once they have propagated to a given size, by which point much of the contributory microstructural information has been destroyed. In order to find, and eventually predict, the location of defect nucleation we must first be able to find defects to the finest resolution possible, as soon as they occur. Upon finding the nuclei we will then be able to characterize the local microstructure in terms of the proposed FIPs, linking the theoretical models to experimental data.

Cucchiara et al have shown that machine learning techniques, specifically C4.5 learning algorithm, can be used to predict defect occurrence in a sample [38]. The final step will then be to apply these machine learning techniques to predict sites of future defects.

1.2 Shear Bands

Much work has been done to try and identify where cracks occur, and it has been observed that the majority of strain within a deformed magnesium sample is found to be confined in small linear regions called shear bands. When subsequent failure occurs it is located within or near these shear bands, where all the available ductility has been exhausted [39].

As a magnesium sample is strained, the deformation localizes around material inhomogeneities and the adiabatic heating gives rise to a region of lower stiffness which extenuates the localization effect of further deformation. The proposed mechanism for shear band propagation is shown in Figure 1-1. The shear bands nucleate at various points within a sample and propagate by the linear connection of multiple nuclei. These shear bands tend to follow grain boundaries and propagate to the extremes of the material. The adiabatic process makes this phenomenon self perpetuating and it continues to the joining of successive bands through the extent of the material.

When the shear/slip bands meet the free surface of the sample there is a topographical effect, shown in Figure 1-1d. If we can locate these regions of surface roughening then we may be able to apply further strain and watch the subsequent cracks nucleate. It is believed that this roughening of the surface will result in changes to the collected EBSD patterns that can be detected and used to highlight the area as a potential failure sight.

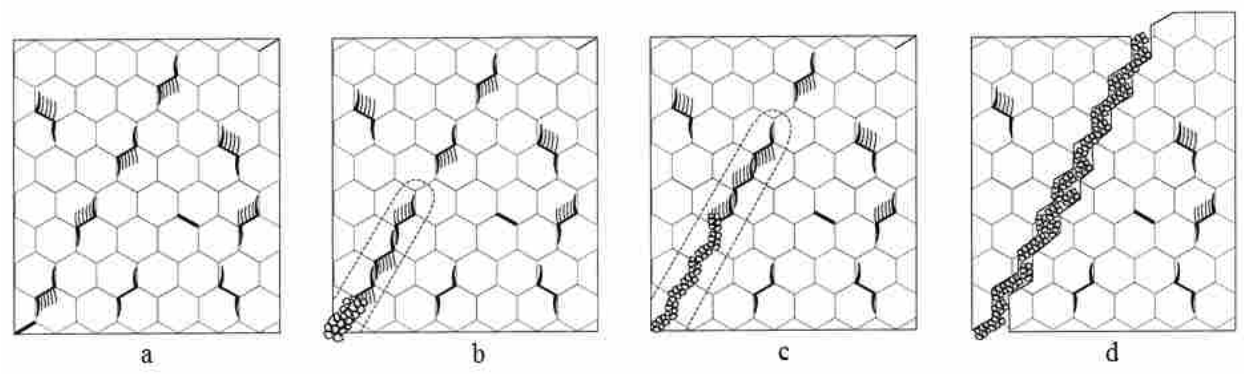

Figure 1-1 : Mechanism of shear bands evolution in a compressed sample [1].

1.3 Scope of Research

In light of this current state of the art we will focus our efforts on two distinct areas: 1) finding cracks (defects that have already occurred) and 2) finding changes in topography (sites where defects are likely to occur next). Our search for defects will be based on the observed phenomenon of shadowing on retrieved EBSD patterns, while our predictions of future failure will be based on the surface topography effect discussed previously. We will also consider search schemes to efficiently find areas of interest in a scan.

2 EBSD METHODS

2.1 EBSD Patterns

When a material sample is placed in a scanning electron microscope (SEM) and hit with an electron beam, some of these beam electrons are diffracted by the charges of the atoms in the interaction volume. These backscattered electrons emerge back out of the sample and can be detected by inclining the sample and introducing a phosphor screen as shown in Figure 2-1 below.

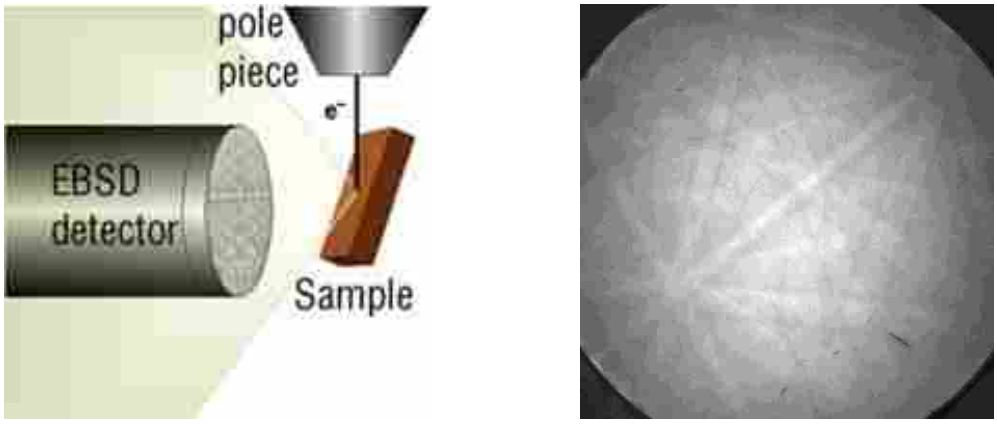

Figure 2-1 : Schematic of EBSD pattern generation (left) and a captured Iron alpha EBSD pattern (right)

The captured electron backscatter diffraction (EBSD) patterns consist of bands of high electron intensity. The position and spacing of these bands results from the crystallographic configuration

and lattice orientation of the atoms they are diffracted by. This characteristic allows for the lattice orientation of the sample to be determined by comparing the collected patterns to those of known crystallographic lattice orientations.

2.2 Orientation Imaging Microscopy

The first automated EBSD indexing System was introduced in 1991 as a collaborative effort between Dr. Brent Adams and TSL Inc [40]. The orientation imaging microscopy (OIM) system applies a Hough transform to the EBSD pattern and identifies the position of the Kikuchi bands. These are then compared to lookup tables to determine the crystallographic lattice orientation. The orientations at each point are resolved to the three Euler angles φ_1, Φ and φ_2 to an accuracy of ~0.5˚. The newly developed High Resolution EBSD (HR-EBSD) compares the capture EBSD patterns to simulated patterns and has been shown capable of increasing the angular resolution to ~0.005˚ and is also capable of measuring elastic strains in the lattice [41-44].

Thanks to increases in the speed of cameras and the associated hardware the current OIM system is capable of indexing up to 450 patterns per second. In reality this is restricted by the level of polish on the sample surface which determines the quality of the EBSD patterns. To increase the accuracy of the indexing we are required to increase the dwell time at each point, giving more backscattered electrons and thus a higher probability of correct indexing.

The current TSL system comes with an analysis package that is able to produce various plots of the scan data. The most commonly used plots are:

1. Inverse Pole Figure (IPF); where the Euler angles are used to determine the lattice orientation at each scan point and represented by distinct colors.

2. Image Quality; where the quality of the EBSD pattern retrieved at each point is shown.

3. Unique Grain Color; where the Euler angles are resolved into individual grains and displayed by distinct colors.

4. Grain Boundaries; where the orientation at each point is compared to its neighbors and if the misorientation is above a threshold value then it is determined to be a different grain.

5. Phase maps; where the EBSD pattern spacing is used to determine the phase in the lattice at each scan point.

These maps each have various applications for materials research and the most applicable to our aim of finding defects is the grain boundary map. The software determines the grain boundaries as the locations within the sample with peek static changes in orientation. The grain boundaries are also known to experience the peek dynamic changes in fields as the microstructure evolves and as such are of great interest to researchers.

2.2.1 Adaptive Scan

In order to increase the scan resolution around the grain boundaries, without a drastic increase in computational time, Yang et al suggested an adaptive scan routine [45]. TLS commercialized this tool and incorporating the adaptive scan into their code. This tool takes a coarse scan of the sample that allows the rough position of the grain boundaries to be determined. An intermediate scan is then taken at points around these positions to refine the scan grid. The final scan then further increases the resolution at the grain boundaries as shown in Figure 2-2 below. Here each

cross represents a single scan point and it is obvious how this tool allows the resolution at the grain boundaries to be increased fourfold with only a marginal increase in computation time.

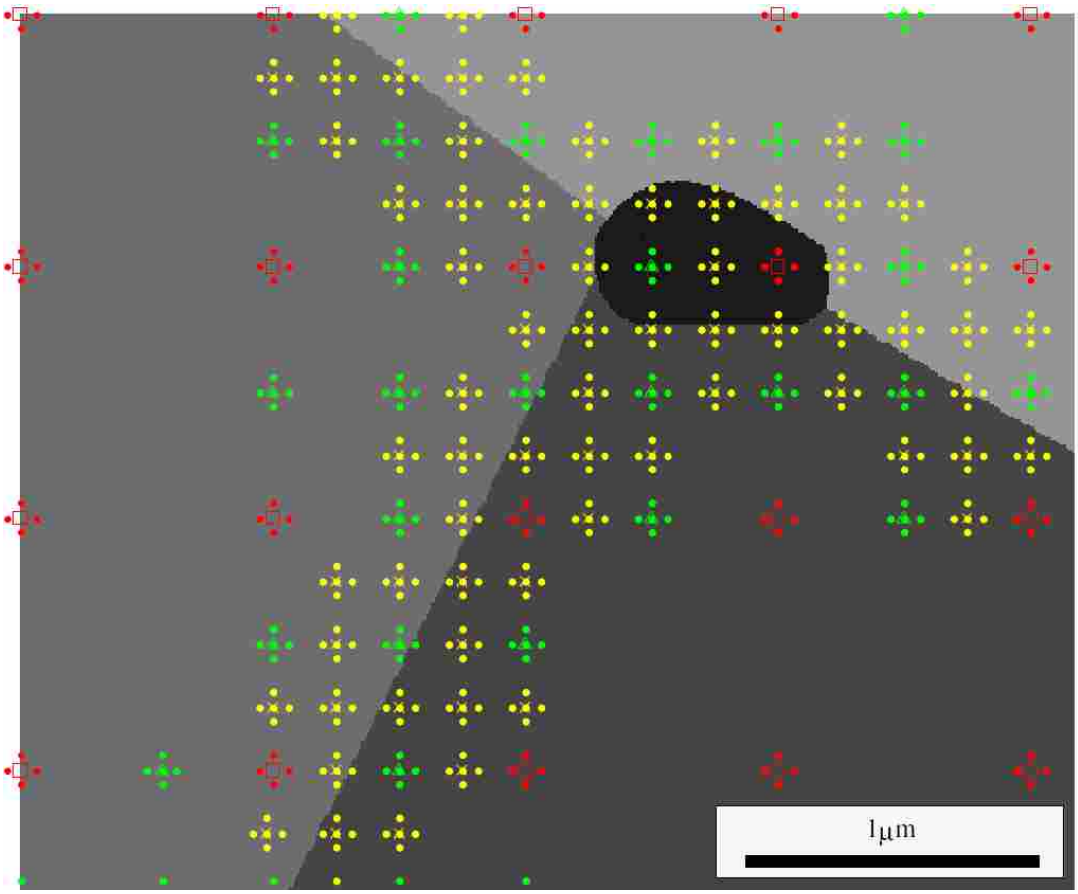

Figure 2-2 : Schematic of the adaptive OIM scan routine. The grey shading shows the individual grains and the black area represents a void. The red crosses with □ centres show the course scan, while the green crosses with ▲ centres and yellow crosses with x centres show the scan refinement.

Despite the time savings the adaptive scan offers, many researchers viewed it as a decrease in resolution at the grain centers and found it hard to accept this potential loss of detail. This reaction, combined with increases in computing power, has resulted in the adaptive scan being deactivated in the most recent versions of the OIM software.

2.2.2 Custom Scan

The OIM software also has the option for running a custom scan. In this instance the coordinates of each scan point must be pre-determined, compiled in a text file and read into the OIM software. This tool has been used for various applications, most recently in conjunction with the HR-EBSD system to allow three sub-points forming an L, or five forming a cross, to be taken at each scan point. The spacing of these sub-points could thus be kept much finer than the overall scan, and by comparing these sub-points the local strain fields and dislocation densities can be found [46].

2.2.3 Proposed Scan Routine

Although the adaptive scan routine is no longer functional, we can use the custom scan to achieve the same result. By taking an initial coarse scan we can determine the points of interest and generate a custom scan file of the coordinates for further investigation. This scan can then be run and the process iterated until the desired resolution is achieved at the points of interest. This methodology also allows us to focus in on more than just grain boundaries i.e. we may search for the sites where defects have already occurred and where surface roughening may be occurring. Potential scan algorithms will be discussed and demonstrated in a later section (see chapter 5).

3 VOID DETECTION

The ability to detect cracks in a sample has been a known feature of OIM analysis since its inception. The cracks are seen in the results as blank regions where the software is unable to resolve the Euler angles from indexing the obtained EBSD patterns, as shown in Figure 3-1 below. It is also known that the ability to index EBSD patterns falls as the material lattice is strained, resulting in a reduction in image quality in these regions [47, 48]. Thus plotting an IQ map shows a crack as a dark region of low image quality, as shown below. These are the two detection methods commonly used but are only accessible after processing the scan data and are indirect measurements that may or may not be the result of the presence of a crack. For our purposes we wish to find a more direct measure that can eventually be applied real time with the OIM scans. We believe this is possible by direct analysis of the obtained EBSD patterns.

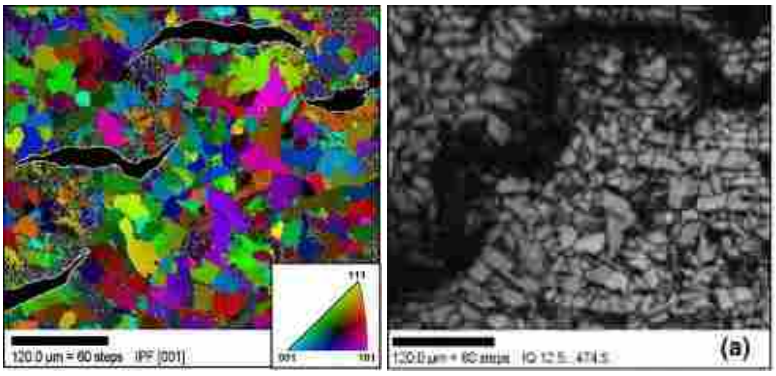

Figure 3-1 : OIM analysis IPF map of a sample with Hydrogen Initiated Cracks (left) and a close up IQ map of a crack region (right) [47].

The occurrence of shadowing on retrieved EBSD patterns is also a well known phenomenon. These shadows hinder the indexing of EBSD patterns and as such have been discussed as part of sample preparation training [49] and work has been performed to demonstrate how to compensate for them [50]. These shadows have always been seen as a nuisance to be avoided but no attempt has ever been made to quantify them. It is our belief that these shadows, when quantified, can give us useful information about cracks and voids in the sample surface that is currently inaccessible.

3.1 Shadowing on EBSD Patterns

The escaping electrons that form EBSD patterns can be blocked by features of the surface, resulting in patterns with varying degrees of shadowing. These features may take the form of artifacts on the sample surface or voids in the surface itself. Voids in the surface result in non-uniform electron scattering and thus the resultant EBSD patterns contain a region with little or no information. This is here defined as shadowing *'in'* the EBSD pattern. On the other hand artifacts on the surface block the scattering electrons in a particular region and thus a crisp shadow is seen on an otherwise pristine EBSD pattern. This is defined as shadowing '*of'* the EBSD pattern and is currently of great concern for three dimensional OIM [51]. This difference can be clearly seen in Figure 3-2 below.

This difference allows for the possibility of distinguishing between the two occurrences. This can be achieved by first defining the region of shadowing and then looking at the contrast across the boundary of that region. This would allow us to further fine tune our search for cracks rather than including surface blemished. However in the current investigation we will take all EBSD

patterns with shadowing as regions of interest and thus use all non-typical patterns as a parameter by which to refine our scan.

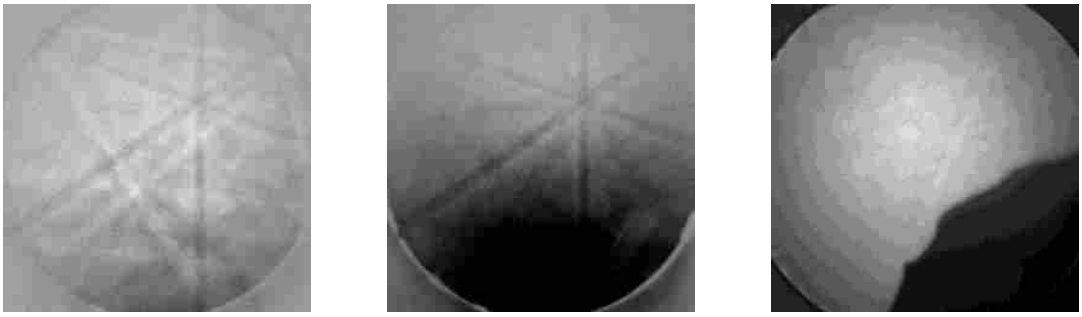

Figure 3-2 : Typical EBSD pattern (left), shadowing '*in*' an EBSD pattern (centre) and shadowing *'of'* an EBSD pattern (right) taken from a scan of the sample shown in Figure 5-4.

3.2 Detection Methodology

The degree of shadowing is defined as the size of the connected region of an EBSD pattern with intensity below a given threshold. This is much easier to see when the shadow is *'of'* as opposed to *'in'* the EBSD pattern. In order to determine the region of shadowing in the later case it is necessary to set a threshold and convert the pattern to black and white. In order to perform this automatically all the EBSD patterns from a scan are read into MATLAB. We are then able to define the threshold and convert them to black and white (0 being black and 1 being white); experience has shown that a threshold of 0.5 is in keeping with our visual deductions. We then calculate the moving average of the result by first defining an element, called herein the defect mask, whose area defines the size over which we perform the averaging. The moving average is efficiently calculated by convolving the defect mask with the black and white image to obtain the resultant shadowing intensity profile using fast Fourier transforms. We define the height of the peak intensity of the convolution as the 'shadow value' and apply a threshold to this to determine if the pattern has a shadow. This process is shown in Figure 3-3 below.

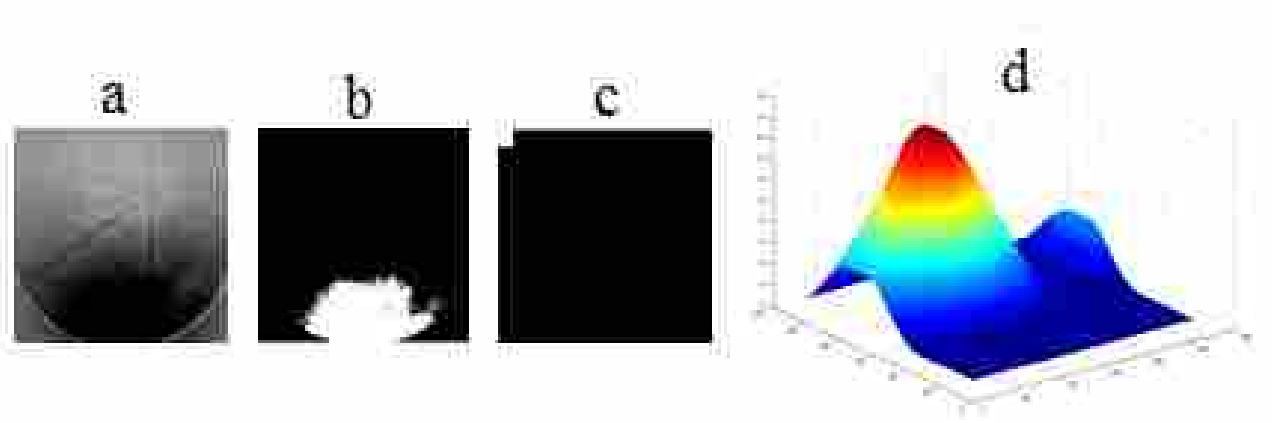

Figure 3-3 : (a) Shadowing *'in'* an EBSD pattern, (b) Pattern converted to black and white, (c) Defect mask that the pattern is convoluted with and (d) Resultant convolution profile.

The three parameters we set allow us to fine tune the process for a given scan condition, material or defect type. Increasing the black and white threshold, the mask size or the peak intensity all serve to filter out the smaller defects. We are also able to disregard large defects in favor of the smaller (and presumably more recent) defects. In our current investigation we will be looking for defects of all sizes and will therefore be setting lower thresholds.

Once we have determined if a pattern contains shadowing we then create a 'Defect Map'. This is just a binary image the size of the initial scan with ones in the places where shadowing, and thus voids or other surface features, were detected. Applying this routine to the scan of a sample with a known defect allows us to show its capabilities. Figure 3-4 below demonstrates the accuracy with which this system can find and represent surface defects. The resolution of this defect map is obviously dependant on the step size of the scan and is limited only by the abilities of the microscope. This defect map will be the basis of our scan refinement regime as it quickly highlights regions of interest.

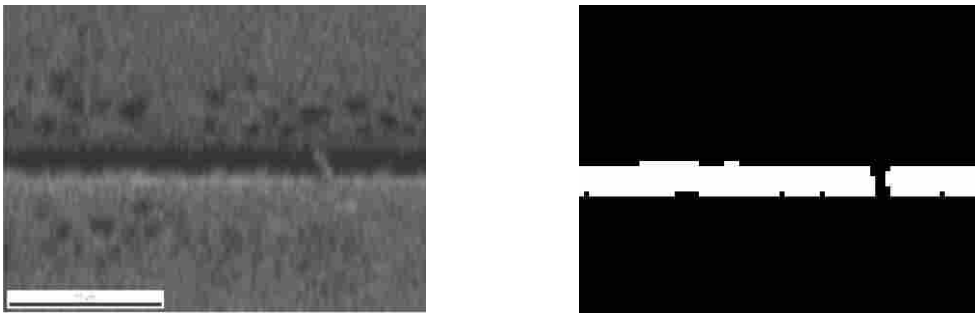

Figure 3-4 : SEM image of a sample with a surface crack (left) and the resolved EBSD shadow based defect map (right).

3.3 Resolution Limits

The resolution of this process is determined by taking a series of line scans across a defect of known dimensions. By varying the step size of the scan we can determine the size of crack that is detectable under a given set of scan parameters or determine the minimum step size required to detect a crack of a given size.

For this analysis we used an ultra-sonically consolidated (UC) nickel sample provided by Brent Stucker (Utah State University). UC samples are manufactured by ultrasonically welding metal foils to form a complex three dimensional structure [52]. Various parameters affect the strength of the bonds formed and when regions do not bond then the resultant crack is approximately perpendicular to the surface. The resultant well-defined geometry of the crack is ideal for the crack detection exercise reported below.

We identified a region where the interface had not properly consolidated and used various portions of this crack to test our process. Figure 3-5 below shows the initial 6.7µm section of the crack we used and our line scan setup. We began our scans roughly 30µm below the crack and then worked our way up the line with various step sizes. Plotting the shadow value against position for each step size we obtain the results shown in Figure 3-6 below. As expected an

EBSD pattern from the centre of a crack is fully shadowed and thus has an intensity of 1. It can also be seen that the scan points from the edge of the crack result in an intensity of 0.62. Thus by setting our threshold as lower we can detect the crack on our defect map at smaller step sizes.

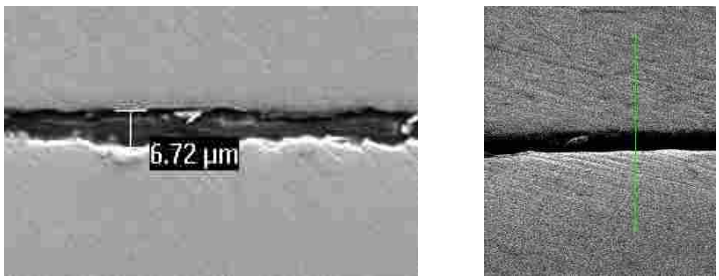

Figure 3-5 : SEM image of an unconsolidated region of Nickel (left) and our line scan setup with the sample inclined to 70° (right).

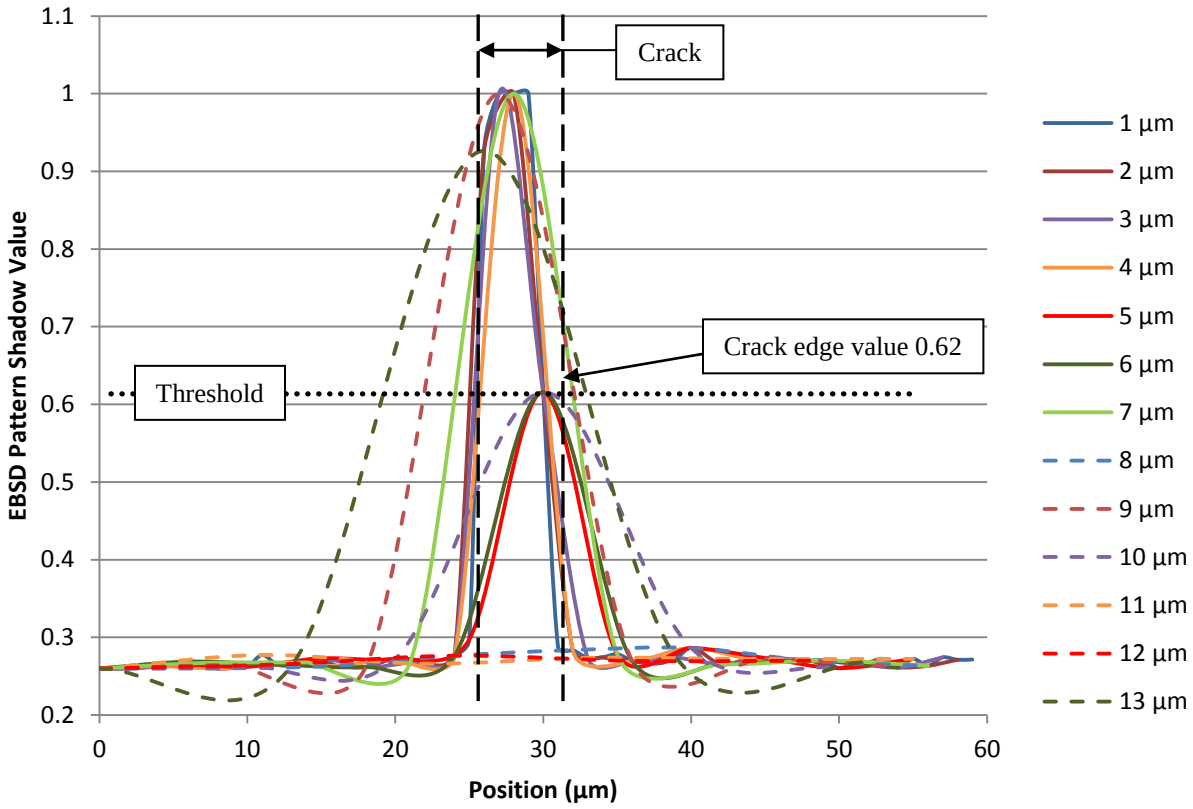

Figure 3-6 : Plot of EBSD shadow value for line scans across a 6.7µm crack at various stepsizes.

The results show that we can detect the 6.7µm crack for all step sizes up to 7µm. For step sizes above this the detection then becomes a matter of luck as to whether the beam lands on the crack or not. It is also believed that the success of the 7µm scan was also a lucky result and that in reality we can only be certain of detecting cracks of equal or greater width than our step size.

To confirm our conclusions we repeated this process for regions of the sample with crack widths of 3.6µm and 1µm and the results are given in Figure 3-7 and Figure 3-8 below. The results from the 3.6µm crack support the majority of our initial conclusions. We can consistently detect the crack for every step size smaller than the crack width and randomly with step sizes above that value. However for the smaller crack size we see that the shadow value for points at the edge of the crack is now 0.32 as opposed to the 0.62 we observed previously. This is likely just the effect of the interaction volume for the scan point incorporating slightly less of the crack in the case of the 0.32 value, but reinforces our belief that for the greatest resolution of our method we need to make the threshold as close as possible to the background level. The resolution for detection of the crack edge (at least for fairly sharp edges as used in this exercise) will then be of the order of the size of the interaction volume.

The key to distinguishing the bounds of the crack will be our ability to identify the background noise and the first points that are outside of this range. We will therefore need to set our threshold as close to the background level as possible or risk false negatives on our defect map (i.e. we may miss some cracks). Due to the nature of the scan refinement we plan to utilize (see chapter 5) we would much prefer false positives as these errors would be corrected in subsequent scans. In our refinement scheme described later we pad our defect map around the edges of crack, this increasing our resolution of the surrounding area (which may of itself be useful) and

reduce the possibility of false negatives that do get through, thus optimizing our resolution of crack boundaries.

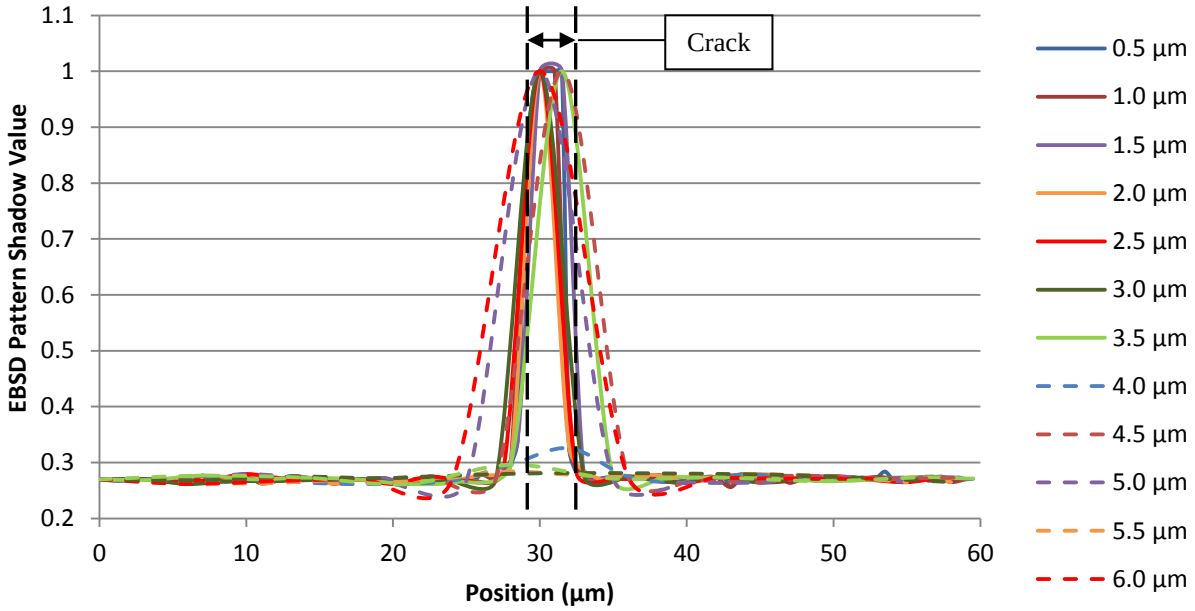

Figure 3-7 : Plot of EBSD shadow value for line scans across a 3.6µm crack at various stepsizes.

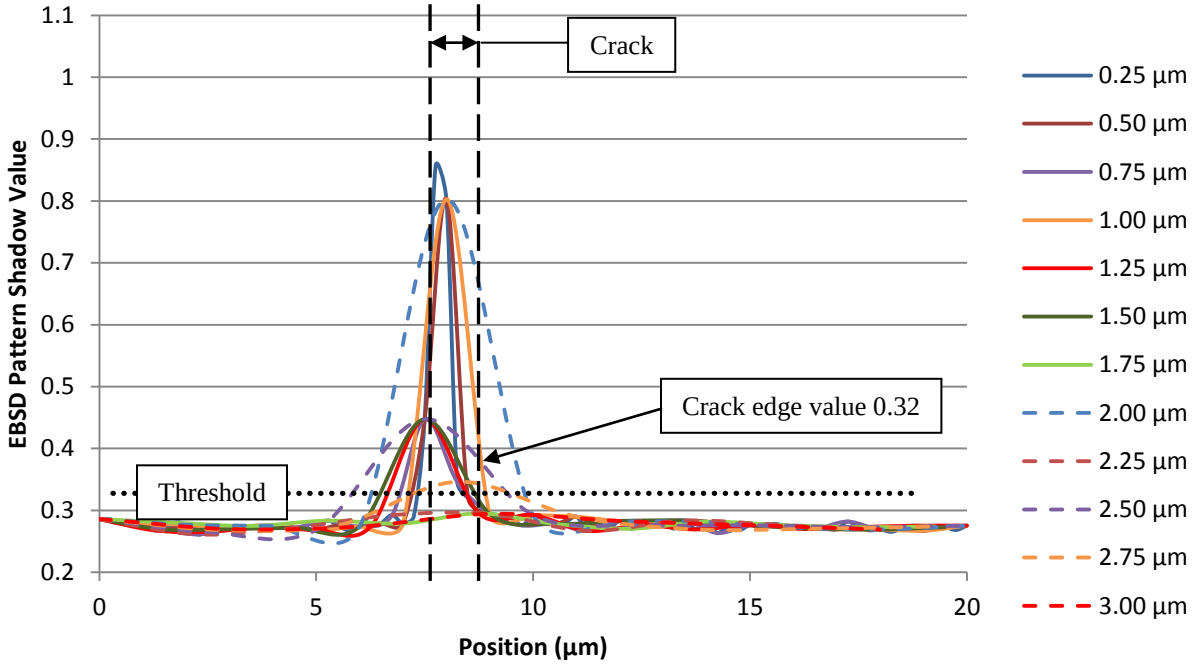

Figure 3-8 : Plot of EBSD shadow value for line scans across a 1µm crack at various stepsizes.

It should also be noted that even the typical EBSD patterns from non-cracked regions result in shadow values of up to 0.29. This phenomenon is attributable to the fact that for this analysis we are analyzing square images of circular patterns, thus we have black corners to our EBSD patterns (as seen in the right hand image of Figure 3-2) that are read as shadows. This background level may be significantly reduced had we used a background subtract routine on our images, where each image is divided by the average value of a number of images in the scan, which is commonly used to remove the effect of defects in our detector and imaging system (as seen in the left hand and centre images of Figure 3-2). In our raw patterns it represents the background level of 'blackness' in our EBSD patterns above which we determine the presence of a shadow from a crack. Thus if we were to process a scan taken utilizing a background subtract then this threshold will need to be reduced below the 0.3 value suggested in Figure 3-8 or we will lose resolution of our crack detection. A test scan may be advisable to determine the exact threshold to apply for each set of microscope / scan parameters.

Returning to the discussion of the 6.7µm crack (Figure 3-6), although we can detect the crack with a 6µm step size we only obtain values above the background level for a single scan point at position 30µm. We know that at the adjacent scan points (24µm and 36µm) the values are back to the background level but we have no way of knowing what exists within these two 6µm boundary regions. Our best conclusion is that the crack is no bigger than 12µm wide. Looking at the results from the 1µm step size scan we obtain values above the background level for positions 26µm - 30µm. Thus we can conclude that the crack is 5-7µm wide and further scan refinement would increase this accuracy. Remembering that the crack is 6.7 µm wide; as a general rule when we detect a crack we are able to also predict it's width to within plus or minus one step size. The ideal strategy to increase the crack detection resolution is to refine the scan in

the regions of the crack edges. The accuracy will be increased with each iteration of our scan refinement method, discussed in chapter 5.

The results from the 1µm crack (Figure 3-8) show shadow values for between the edges of the crack below the typical 1.0 seen from the larger cracks. This is attributable to the fact that the backscattered electrons are emitted from an interaction volume around the point where the electron beam hits the sample. For a non-cracked sample inclined to 70°, the amount of material between the interaction volume and free surface is small and thus electrons are backscattered over the entire phosphor, as shown in Figure 3-9a below.

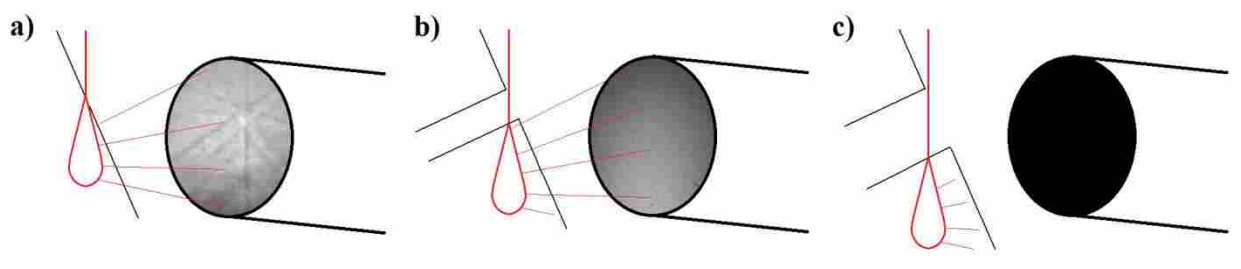

Figure 3-9 : The effect of interaction volume and crack geometry on retrieved EBSD patterns

When a large crack is present, larger than the interaction volume, then the beam hits deep in the sample (relative to the free surface facing the phosphor) and the backscattered electrons are fully blocked, giving a shadow value of 1 as in Figure 3-9c. However, when a small crack is present then the interaction volume encapsulates a portion of either the top edge or lower edge of the outer sample surface and thus we obtain an intermediate shadow level, as shown in Figure 3-9b.

The shadow value we obtain is thus dependent not only upon the crack geometry but also the size of the interaction volume. This is determined by the energy of the electrons in the beam,

controlled through the acceleration voltage and the spot size of the scan, and the atomic number of the sample material [53].

Investigating this relationship further we took scans of 1.5µm and 0.5µm wide cracks and plotted shadow value for each crack size in Figure 3-10 below. The results are normalized to show the difference between the peak shadow value and the background level. A crack of zero width obviously presents as a standard EBSD image and thus only the background level of shadowing exists. As the crack width is reduced then the difference between the shadow value and the background falls off quickly. Clearly a crack cannot be detected that has a shadow value that is within the noise range of a standard EDSB pattern from a non-cracked region of the sample. This noise value will clearly be material and sample dependant. For the sample used in this exercise; the background noise has a value of 0.1; this defines the limit of the crack width we can detect to be about 0.15µm, as shown below.

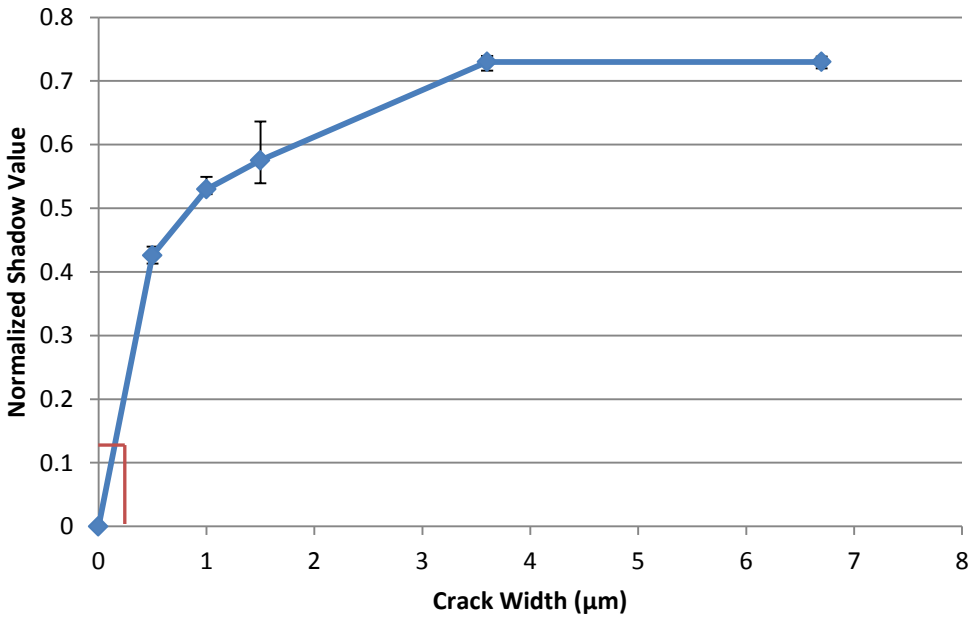

Figure 3-10 : Shadow value for different crack widths taken with a 0.5µm step size.

3.4 Crack Orientation

We subsequently took a line scan with a 4.8µm crack orientated 45° to the beam incidence to see if the resulting shadows presented any differently to when we scan normal to the crack. The results are shown in Figure 3-11 below and they are in line with the results we obtained for scans normal to the crack. The difference seen is that the 45° crack angle allows the effective crack width to increase to about 6.8µm (4.8/cos 45°) and thus the crack is consistently detectable with a step size up to 7µm.

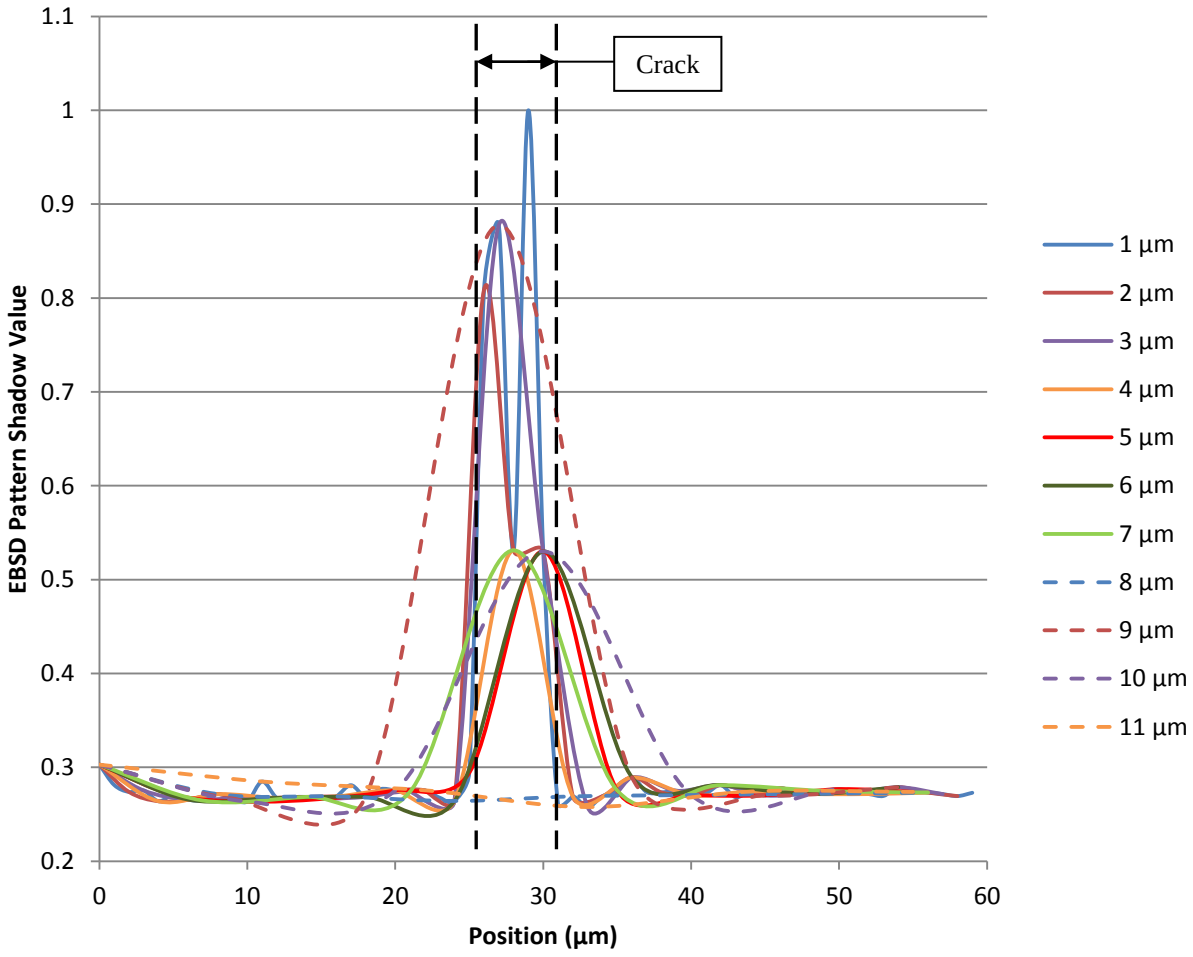

Figure 3-11 : Plot of EBSD shadow value for line scans taken at 45° to a 4.8µm crack at various stepsizes

The results also display a noisy peak value for the smallest step size, which is explained by the rough nature of the crack edge. When we approached normal to the crack then the beam falls into the crack quite suddenly whereas the 45° approach allows the beam to be more affected by the rough edges of the crack and thus we obtain noisy results. The values are still well above our background threshold and so we will still be able to detect it as a crack regardless of the noise.

There is nothing in these shadow value graphs that would enable us to distinguish the angle of the crack from the shadow value of the scan. In hopes of achieving this we took a closer look at the individual EBSD patterns. Comparing EBSD patterns from a crack normal to the beam and at 45° to it reveals the strong possibility for determining not only the presence of a crack but it's direction also. Figure 3-12 below shows a marked difference in the location of the shadowing that seems to correlate with the portion of the interaction volume that is meeting the crack. If we imagine a line from the center of our connected shadow region to the centre of the phosphor, the pattern from the 45° crack would result in a line almost exactly 45° to the vertical. This method would allow us to make our search algorithm much more efficient. This may also allow for the use of an increased step size as we could identify and connect points along a given crack without needing to find every point along the crack. Obviously this method will require a detailed investigation before it can be fully implemented, which is outside of the scope of this thesis.

3.5 Additional Factors

The creation of a defect map lends itself quite naturally to the incorporation of additional defect indicators. For example, we discussed previously how the lattice strain around voids results in the image quality of the obtained EBSD patterns being low [47, 48]. Although no shadowing occurs on the pattern, it's clarity and the ability to index it falls. Thus incorporation of low image

quality into the detection routine can broaden the range of defects that are observed. This and other areas of future interest can thus easily be incorporated into our adaptive scan routine as and when they arise.

Figure 3-12 : Shadowing from a crack normal to the scan direction (left) and at 45° to it (right).

4 TOPOGRAPHY DETECTION

The most complete system for rough surface EBSD analysis was introduced by Semprimoschnig et al, whose new tool incorporated an Automatic Surface Reconstruction System (ASRS) with EBSD analysis.[54]. In this system the fracture surface topography is resolved by analysis of stereo pairs imaging. Subsequent crystallographic analysis on the very rough fracture surfaces requires various rotations of the sample in order to retrieve useable EBSD patterns from each surface plane. This methodology cannot be employed in an automated EBSD system like OIM but the level of topography we are currently concerned with (to detect shear bands, for example) is much less than that of a fracture surface analyzed in the previous reference. Thus it is unlikely that stage re-orientations will be necessary to detect such surface roughening.

The effect of changing the grazing angle (measured between the electron beam and the sample surface) on the intensity profile of the retrieved EBSD pattern was rigorously investigated by Alam et al [55]. Their work with single axis variations showed a strong correlation between the grazing angle and the position of the peak intensity in the EBSD pattern. The effect of two axis variations is imagined to be observable in exactly the same way, but whether the variation of the metric used to determine surface ‘rotation’ change (terminology meant to describe surface slopes in the direction lateral to the beam incidence) is sufficiently different from variations measured for surface ‘inclination’ change (meaning slopes in line with the beam) to distinguish between

the two cases is yet to be determined. They also showed that as the grazing angle increased then the intensity profile broadens. This will give us a second metric to identify surface topography but again the effect in two dimensional variations is yet to be seen.

Field suggests that fracture surface analysis can be conducted by using a reference surface as a background (B) and employing an A/B background subtract routine on each subsequent EBSD image (A) [56]. The results of such a method suggest that 1° changes in surface normal can be detected visually and the effect of a 5° variance may render the resultant EBSD pattern useless. This phenomenon is routinely manifested in microscopic analysis as changes in image quality (I.Q). Ideally we would vary the image used in the background subtraction to include images captured from various plane orientations; we could then identify the regions of the sample surface that match each plane by high local I.Q. However I.Q is affected by various other factors that need to be separated out before this can be used as a reliable metric.

4.1 Detection Methodology

Noting Fields observations, our scans were all taken without using the built-in OIM background subtract, this allows us not only allows us to apply Fields method to the dataset offline if desired but also avoids any skewing of the raw EBSD patterns data that may hide the results we are looking for. Starting with Alam's observations and looking at the intensity profile of the EBSD patterns, we are able to use MATLAB to display and locate the position of the peak intensity on the phosphor screen. The EBSD patterns are typically noisy and can be smoothed with a simple convolution with a small moving region, as shown below in Figure 4-1. The peak intensity position and broadness of the peak (for a given contour) for the smoothed values are easily identifiable.

4.1.1 Comparison of EBSD Patterns

Taking successive scans of the same sample region at varying angles of inclination allows us to compare EBSD patterns and identify features resulting directly from the change in grazing angle. The differences resulting from a 5° change in grazing angle can be seen in Figure 4-2 below. As the grazing angle increases the position of the peak moves and the contour broadens in line with Alam's observations. Even after the convolution there is still noise in the peak value and so from here out we identify the peak position as the centre of mass all the points that lie within the contour chosen for the broadness metric, described below.

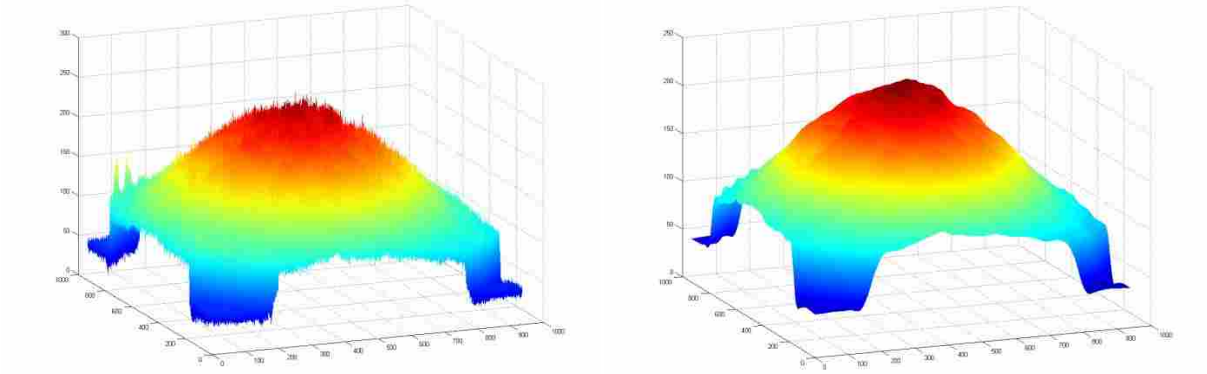

Figure 4-1 : Plot of EBSD pattern intensity across the phosphor (left) and the convolved values (right).

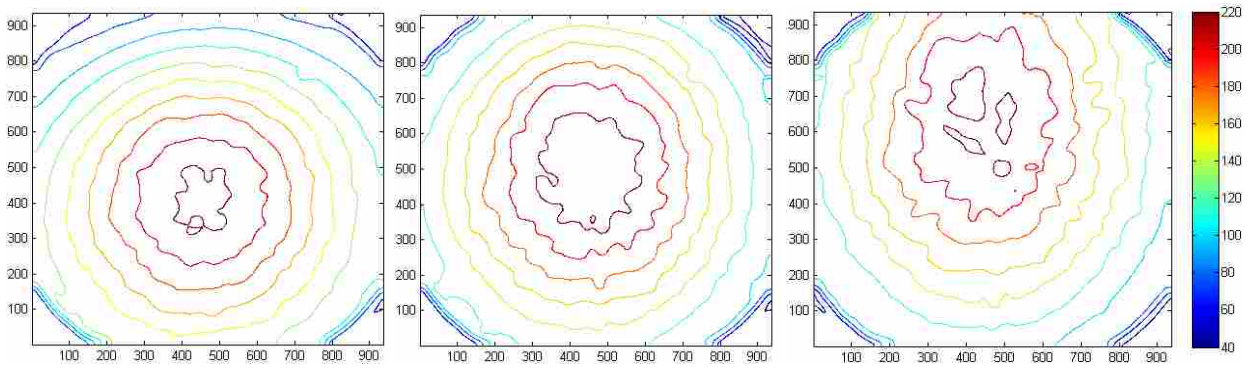

Figure 4-2 : Contour maps of the EBSD intensity profiles from scans of single crystal Germanium inclined to 65° tilt (left), 70° tilt (centre) and 75° tilt (right).

Taking the 70° pattern as a reference, we can find a relationship between the peak position and broadness of the profile for both a positive and negative change in inclination. The broadness is determined as the area of the profile with intensity above the mean intensity value plus 1.2 standard deviations. These relationships are shown in Figure 4-3and Figure 4-4 below.

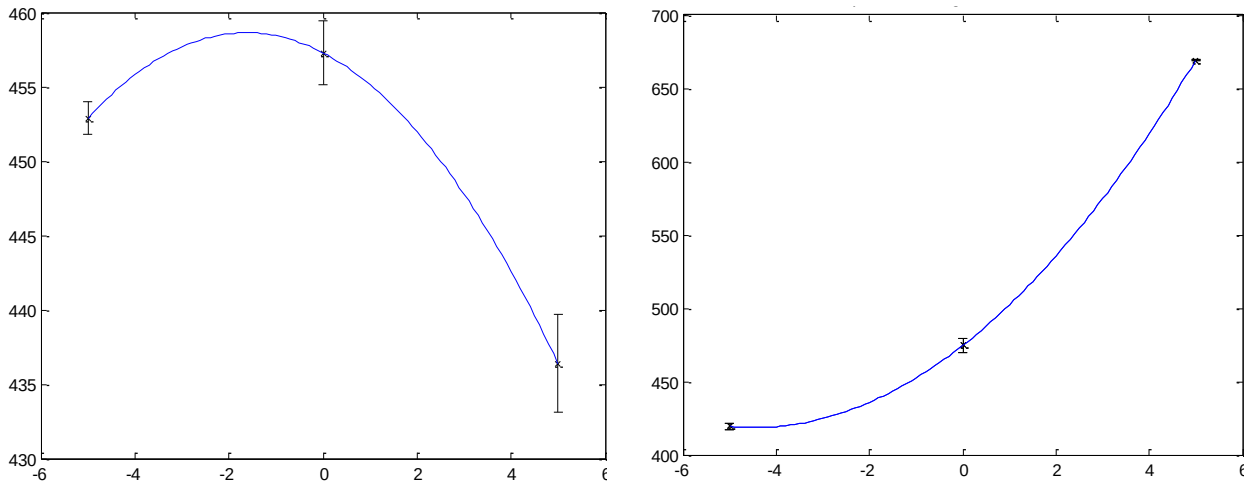

Figure 4-3 : Change in intensity peak position (measured in pixels on the phosphor) for X (left) and Y (right) for changes in inclination (measured in degrees); a quadratic is fitted to the three data points for visualization and calibration purposes. The error bars indicate plus and minus one standard deviation across the 121 data points.

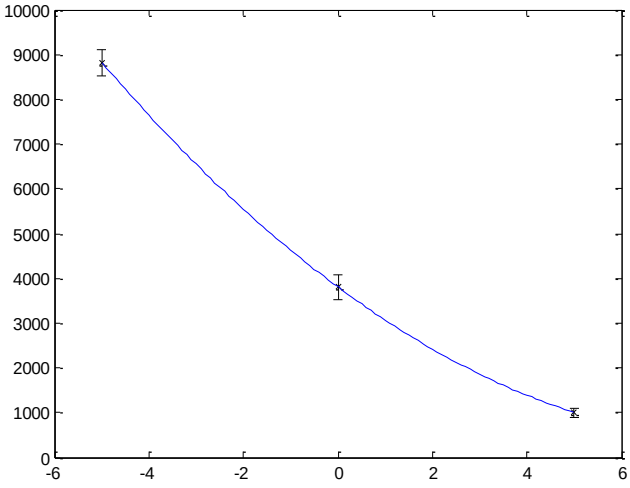

Figure 4-4 : Change in profile area (square pixels) for changes in inclination (degrees). The error bars indicate plus and minus one standard deviation across the 121 data points,

As can be seen, the results for changes in X position are less useful to us as the value is reduced for both a negative and a positive change in inclination, thus predicting topography change from this value becomes more difficult. However, the results for Y position and area both vary smoothly with inclination angle, and are extremely consistent. Both metrics present the possibility for inclination detection based on these obtained values.

4.1.2 Topography Characterization

Field's method of utilizing a background subtract to identify topographical features relies on our ability to gain suitable reference patterns for each surface plane in the scan. This method allows us to identify areas of the scan with a large misorientation from the current background being subtracted but gives us no indication as to a better orientated background to use. Thus this method becomes a matter of trial and error or it requires a vast array of different backgrounds to be successively implemented.

By comparing the location and broadness of the intensity profile of adjacent EBSD patterns we are able to detect changes in topography on a point to point basis. Furthermore, the Germanium test case indicates that a smooth calibration curve can be derived for a given sample, to enable a direct correlation between these metrics and surface inclination. We can then combine this information about the individual changes in topography to reconstruct the sample surface.

Note that the calibration curve is highly likely to be sample dependant – relating to material type, the quality of polishing / the surface roughness, and probably other factors.

4.1.3 Surface Recovery

EBSD patterns were retrieved from the leading quadrant of the surface of an 8mm diameter cylindrical Nickel sample orientated perpendicular to the electron beam, as shown in Figure 4-5 below. This dataset encompasses a range of surface inclinations and allows us to test the validity of our proposed surface recovery process.

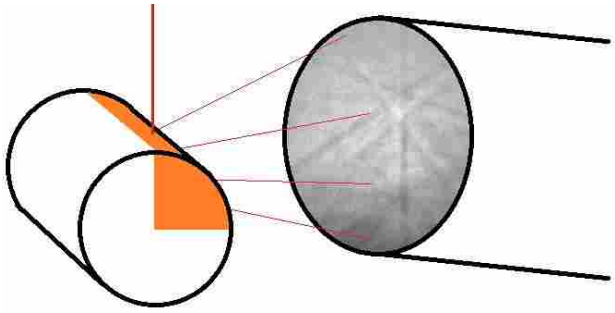

Figure 4-5 : Orientation of scanned Nickel cylinder with scanned quadrant shown in orange.

The results from comparing the EBSD patterns of successive points in the scan are shown in Figure 4-6 and Figure 4-7 below. As can be seen, the results for changes in X position continue to be less useful and contribute little to our knowledge of the absolute change in position. If our sample had been perfectly polished then we wouldn't expect to see any changes in X position, the results we obtained show small undulations of the surface from imperfections in our sample preparation. The results for changes in Y position give us a very smooth result which seems perfectly in keeping with the profile of the sample. The results for changes in the area also give us a smooth result and also look to be starting to suggest the cylindrical profile of the sample (although there is an interesting fluctuation along one edge of the scan).

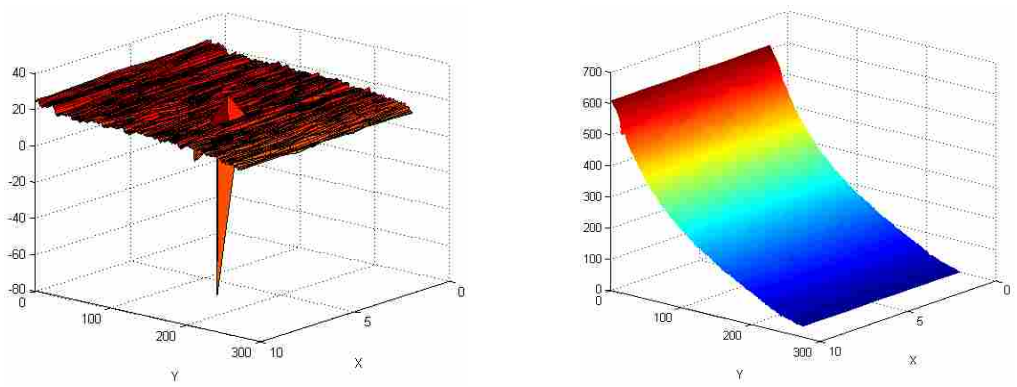

Figure 4-6 : Plots of incremental change in X position (left) and Y position (right) for our cylindrical Nickel sample(the x and y on the axis represent scan position, and are in microns, the X and Y position of the peak intensity are given by the height of the surface, in pixels).

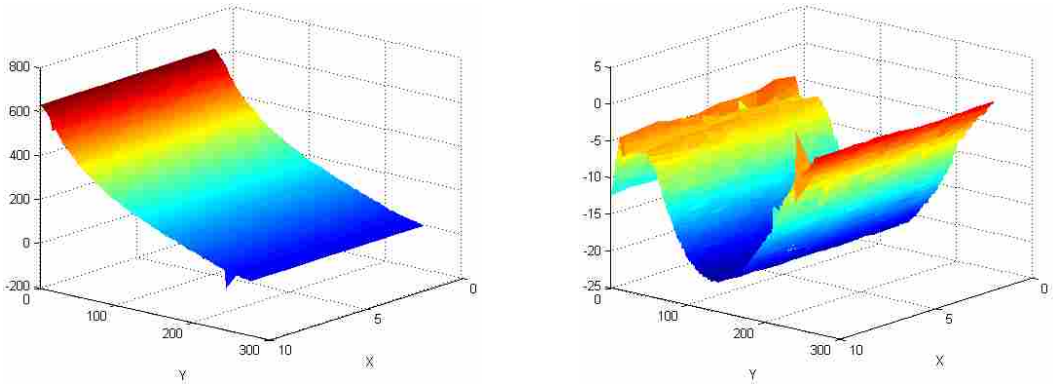

Figure 4-7 : Plots of incremental change in absolute position (microns – using the calibration from the Germainium results) (left) and Area (pixels) (right) for our cylindrical Nickel sample.

These results support our initial conclusion that the Y position and area of the EBSD pattern intensity peak can be used to identify changes in sample topography. In order to fully reconstruct the surface then we need to calibrate the combination of these values (an approximate calibration is given in Figure 4-7, using the germanium results). To do this with our current sample we would need to know several things about the surface. The OIM software assumes a flat sample inclined to 70°. When we specify the step size we want, then the OIM calculates the linear beam translation required in order to get the desired step size on the sample surface. With our curved sample then we need to know the exact profile of the sample in order to determine the actual step

size. This will vary for each data point and so we would also need to know the exact position of each data point along the curve.

This information is unavailable to us with our current sample but the creation of a more accurate sample is currently under investigation. This calibration for surface reconstruction will be performed on the new sample as part of a future work but for now we have sufficiently demonstrated that it is possible to use EBSD patterns to identify changes in sample topography.

Without this calibration we are unable to completely quantify the changes in topography but we are able to qualitatively identify areas of the sample that have higher levels of topography variation. This information can be used in our scan refinement algorithm (where we are simply searching for areas that are different from the norm) without the need for reconstructing the surface.

4.2 2-Dimensional Topography

Our investigations and results so far have been based purely on changes in inclination, as were Alam's [55]. It is assumed that changes in topography perpendicular to the sample inclination would result in changes in the X position of the peak intensity, similar to those seen previously in Y position. In order to fully investigate this, modifications to the current OIM system are required.

The Phillips XL30 SEM we are using does not have the capability to rotate the sample in this desired direction and the only non-flat sample holders available are inclined to 45°, which is far too much for the level of detection we wish to investigate. As a result we cannot take and compare scans from the single crystal Germanium as we did to identify changes due to

inclination. To overcome this issue the author has designed a new sample holder (see Figure 4-8) that is currently being manufactured for this purpose.

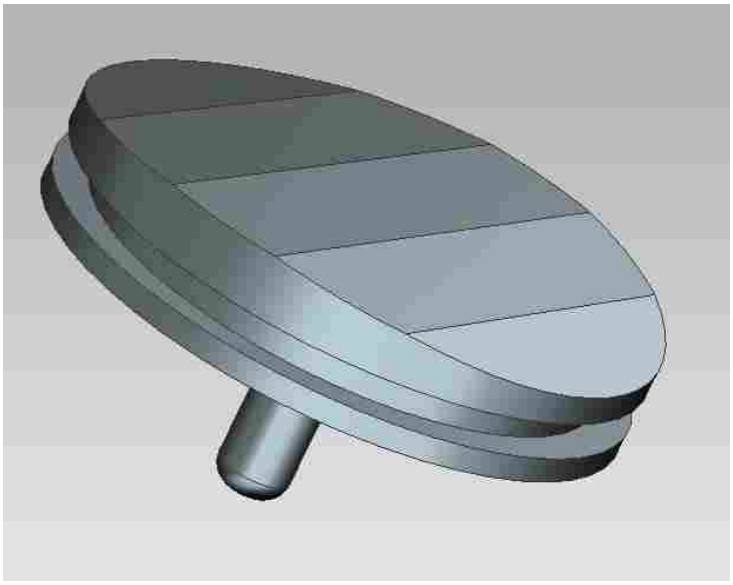

Figure 4-8 : New stub for OIM analysis of inclined samples.

This new stub is 1 inch in diameter and has five flats of equal width milled on the surface. The central region is coplanar to the stage (as a normal stub does) and it then also has flats inclined to 5° and 10° in each direction. This stub allows for five individual samples of single crystal Germanium to be mounted at five different angles. When this stub is manufactured it will allow us to conduct the same investigation into topography changes perpendicular to the inclination direction. When combined with our current results we will then be able to fully qualify the topography of a sample from the EBSD patterns obtained.

5 SCAN REFINEMENT

5.1 Triple Junctions

The presence of triple junctions is one of the most obvious place to look for local failure events, as all the amplification effects seen at grain boundaries are further multiplied. The locations of triple junctions is not reported by the TSL scan software, they can be found using the TSL analysis software but this tool cannot be easily integrated with our current algorithms, thus the author wrote MATLAB code to locate them. Following an OIM scan, the TSL scan code outputs a '.ang file', which is just a text list of scan position and retrieved EBSD pattern data (most importantly the local orientation and confidence index) for each scan point. By reading this file into MATLAB we are able to post process it as desired. By applying the symmetry operators of the crystal rotations to the Euler angles, we can identify the individual grains and grain boundaries the same way the TSL analysis software does. It is then just a step further to identify the triple junctions, as shown in Figure 5-1 below [see Appendix for MATLAB code]. As can be seen, a complicated image can be quickly resolved to find and highlight the points of interest. This map can then be combined with the other microstructural properties discussed in the literature to identify FIP sites.

This search algorithm is a significant first step at creating specific search routines for microstructure properties. It contains many of the basic ingredients that we will need to be used

in our adaptive scan approach to any of the FIPs, specifically crack detection and topography analysis.

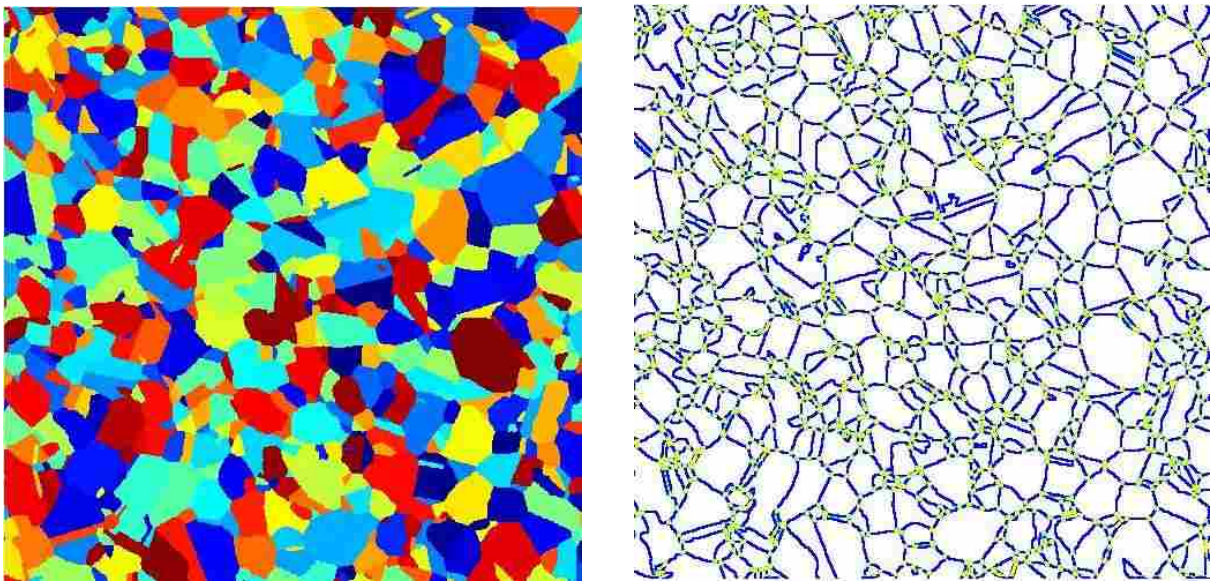

Figure 5-1 : Grain map generated straight from the EBSD images (left) and the resultant blue grain boundaries and yellow triple junctions (right).

In order to efficiently investigate such regions of the scan as triple points, we propose to initially scan the sample using a coarse scan, then subsequently refine the scan in particular regions of interest (in the same spirit as the TSL adaptive scan methodology). The TSL custom scan routine provides a useful platform for facilitating such an approach. The custom scan software requires the input of the desired scan coordinates. In order to determine the additional scan points for a refined scan we simply need to know the coordinates of the points of interest in the existing scan and the level of refinement we wish to employ. The shadowing and topography search algorithms generate a scan map, with highlighted points of interest where either algorithm returns a positive result. A simple example with three points of the initial scan identified for further analysis is shown in Figure 5-2 below. A scaling factor of two has been applied to

identify the new scan points, leading to a two fold increase in resolution in the areas of interest, with the calculated scan time only increasing by a factor of 1.6. This saving is likely to be much higher in reality as in this example 8.3% of the total points were identified as points of interest, which is considerably higher than we'd expect in the majority of samples. Even if we were to pad our areas of interest (as discussed previously in chapter 3.3) we would still obtain a considerable time saving for the resolution.

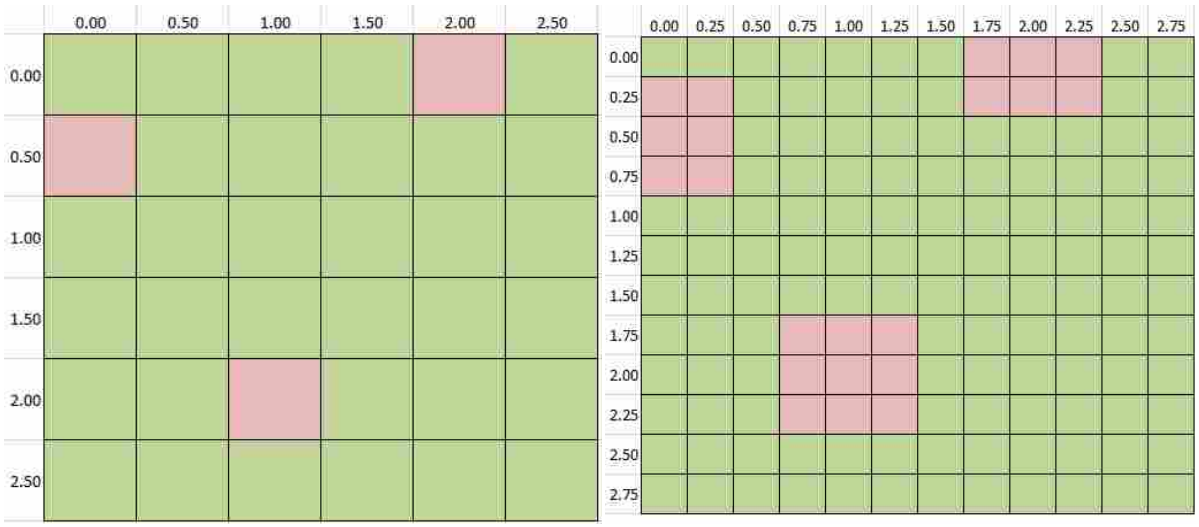

Figure 5-2 : Initial scan map with points of interest highlighted (left) and the grid of refined scan points (right).

5.2 Integration with Coarse Scan

One method of organizing the data for the refined scan (the method used in this exercise) is to form a larger matrix capable of holding a full scan at the finer resolution, and then populate the refined points with multiple instances of the initial data. Thus the four cells in the top left corner of the refined grid are filled with the data from the top left cell from the course grid and so on. The data from the new scan points can then be easily written in over the top at each point that is being analyzed. This process can then be iterated until we achieved the desired resolution.

As all the information is of a square grid nature, it would then be a simple task to write it to a new .ang file, as is output by the TSL scan code for a regular scan, which would allow the subsequent use of TSL OIM analysis software. More usefully the dataset has been augmented to include the location of each scan point's EBSD image file, thus allowing for HR-EBSD to be performed on the final scan.

5.3 Refined Defect Scan

The overall methodology developed in this thesis can be seen in Figure 5-3 and is outlined below.

1. We start with a normal square grid scan (shown as run through the custom scan feature but can equally be run normally) and the resulting .ang file is read into MATLAB to obtain the normal grain maps.
2. We then analyze the individual EBSD patterns and create a defect map of points of interest. This can be performed for multiple indicating factors and then combined.
3. This information is turned into a 'defect scan', which is a custom scan text file that increases the resolution around these areas by a user specified factor.
4. The scan is run and the resulting .ang file and EBSD patterns are analyzed. The new data is spliced with the old giving us a newly refined defect map.
5. Steps 3 and 4 are repeated until the desired resolution is achieved.

In order to demonstrate this, a scan refinement routine was run on an intentionally scratched Nickel sample. The defect maps in this example were created purely based on EBSD shadowing. The SEM image of the scan region is shown in Figure 5-4 below and the successive defect maps can be seen in Figure 5-5.

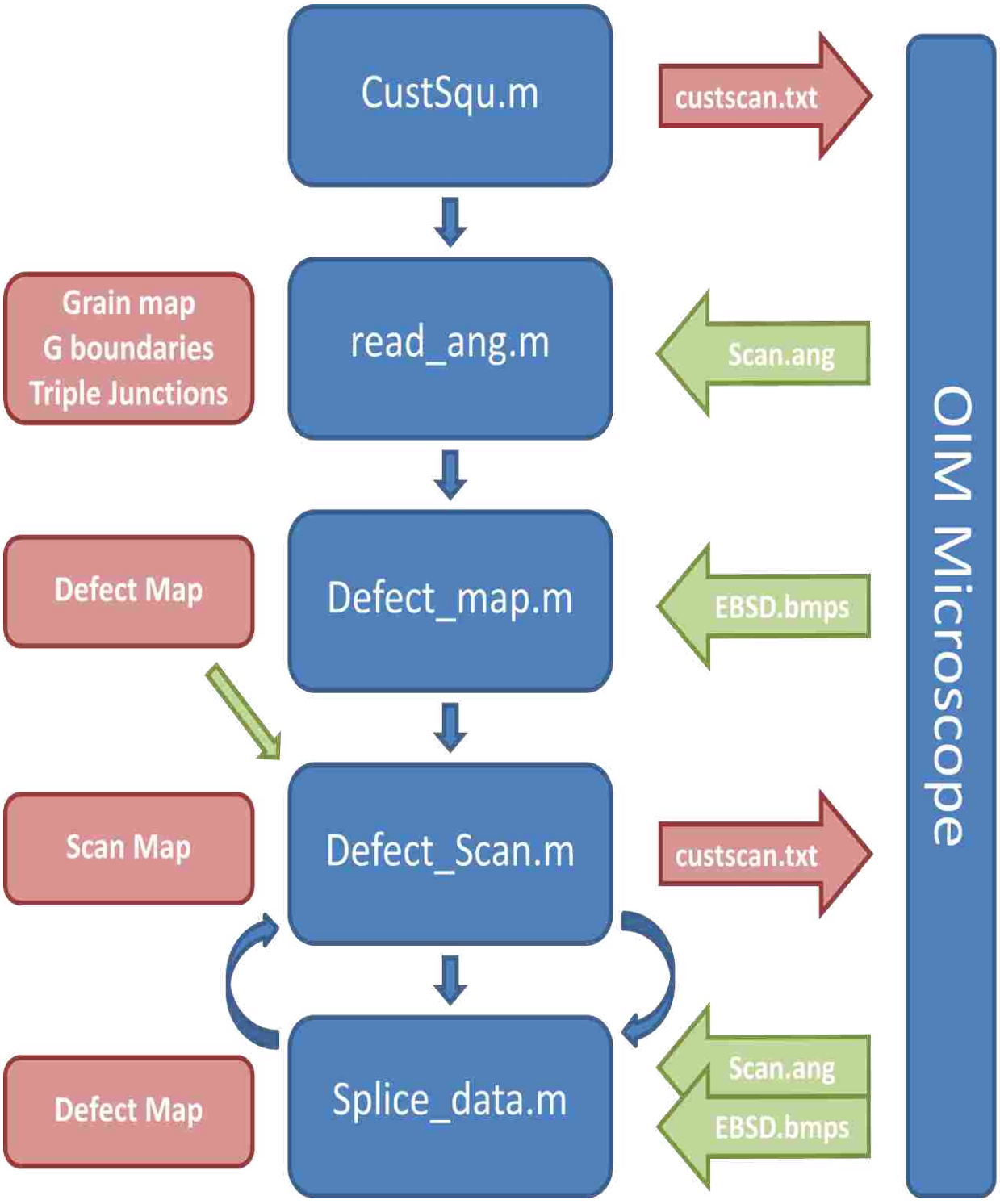

Figure 5-3 : Flow diagram of the various steps (and code portions) used to obtain the detection and scan refinement strategy presented in this thesis (see Appendix for MATLAB code).

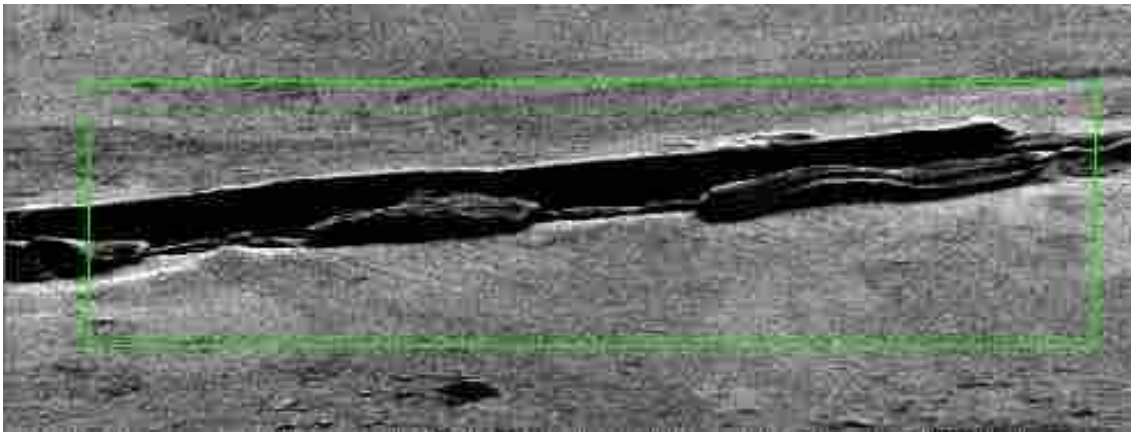

Figure 5-4 : SEM image of a scratched Nickel sample with the scan region outlined in green.

Figure 5-5 : Initial defect map (left) and after scan refinement (right).

Due to the way MATLAB images data the defect maps are inverted but the orientation can be clearly seen. The lower edge of the SEM image correlates to the top edge of the sample. Along this edge of the scratch, even EBSD patterns generated in the non scratched region have a portion of the backscattered electrons blocked by the protrusion of the defect and thus we see bulges in the shadowing observed. As discussed in section 3.1, this effect could be removed by using contrast gradients to distinguish EBSD patterns with shadowing *'in'* them from those presenting with shadowing '*of'* them.

It should be remembered that the scan was taken of a 200µm x 150µm area but appears narrower in the SEM due to the 70° inclination. The scan refinement could be continued as many times as necessary to achieve the desired resolution. The limiting factor to the resolution of this scan routine is what can be detected by the initial scan. Any points missed within the scan grid are not re-evaluated by the current routine. This may be another area where the application of machine learning may prove beneficial.

6 CONCLUSIONS

The aim of this thesis is to provide tools that can be used via EBSD analysis to detect the presence and future occurrence of defects. These tools fall into three categories; 1) Crack detection, 2) Topography detection and 3) Scan refinement.

1. Shadowing is a known issue in EBSD analysis and effort is expended in trying to avoid or compensate for it. We have demonstrated that shadow characteristics can provide quantified measures of crack/void attributes.

   The scan step size must be smaller than the crack width in order for it to be detected and refinement techniques in the region of the crack will increase resolution of crack width prediction. Cracks down to 0.15μm can be detected by the demonstrated methodology.

   The orientation of such cracks can be identified; this is an area of ongoing work.

2. Image quality has been used to give some information about surface topography. Similarly image intensity characteristics have been used to assess topography.

   We have used a new systematic approach to link intensity parameters of position and broadness to determine local inclination. The potential for the approach was demonstrated on a nickel cylinder. Initial results, not reported in this thesis indicate that a similar approach may be possible for topography changes perpendicular to the inclination. A suitable test bed has been suggested.

3. TSL's adaptive scan routine was state of the art but is no longer available. We have demonstrated the current custom scan tool and suggested a new refinement routine that can be implemented through it.

   This off-line adaptive scan can be used with various indicators to significantly increase resolution around areas of interest with a minimal increase in scan time. The future application of machine learning techniques is currently under investigation to increase this further.

   The system has been designed in such a way as to allow for the use of HR-EBSD and offers a new powerful tool for the analysis and prediction of defect nucleation.

The methodology developed in this thesis has been demonstrated in an initial attempt at defect detection. The results are consistent with the conclusions drawn in each individual section of this thesis. This new tool is now working and ready to help in the current areas of interest in materials research.

REFERENCES

1. *Corrosion cost and preventative strategies in the United States*. 2001, CC technologies.

2. Fullwood, D.T., et al., "Microstructure sensitive design for performance optimization". *Progress in Materials Science*, 2010. **55**(6): p. 477-562.

3. Chen, F.-K. and T.-B. Huang, "Formability of stamping magnesium-alloy AZ31 sheets". *Journal of Materials Processing Technology*, 2003. **142**(3): p. 643-647.

4. Das, S., "Primary Magnesium Production Costs for Automotive Applications". *Journal of Metals*, 2008: p. 60-63.

5. Lee, S., Y.-H. Chen, and J.-Y. Wang, "Isothermal sheet formability of magnesium alloy AZ31 and AZ61". *Journal of Materials Processing Technology*, 2002. **124**(1-2): p. 19-24.

6. Li, H., et al., "Determination of Active Slip/Twinning Modes in AZ31 Mg Alloy Near Room Temperature". *Journal of Materials Engineering and Performance*, 2007. **16**(3): p. 321-326.

7. Froes, F., D. Eliezer, and E. Aghion, "The science, technology, and applications of magnesium". *JOM Journal of the Minerals, Metals and Materials Society*, 1998. **50**(9): p. 30-34.

8. Mordike, B.L. and T. Ebert, "Magnesium: Properties -- applications -- potential". *Materials Science and Engineering A*, 2001. **302**(1): p. 37-45.

9. *Forming of Magnesium Alloys*, in *ASM Handbook: 14B*. p. 625.

10. Agnew, S.R. and Ö. Duygulu, "Plastic anisotropy and the role of non-basal slip in magnesium alloy AZ31B". *International Journal of Plasticity*, 2005. **21**(6): p. 1161-1193.

11. Basinski, Z.S., et al., "Transformation of slip dislocations during twinning of copper-aluminum alloy crystals". *Revue de Metallurgie Cahiers D'Informations Techniques*, 1997. **94**(9): p. 1037-1044.

12. Doege, E. and G. Kurz, "Development of a formulation to describe the work softening behaviour of magnesium sheets for heated deep drawing processes". *CIRP Annals - Manufacturing Technology*, 2001. **50**(1): p. 177-180.

13. Knezevic, M., et al., "Anisotropic Stress-Strain Responce and Microstructure Evolution in AZ31.". *Acta Materialia*, in preparation.

14. Kobayashi, T. and J. Koike, "Grain Size Dependence of Active Slip Systems in an AZ31 Magnesium Alloy". *Journal of the Japan Institute of Metals*, 2003. **67**: p. 149-152.

15. Koike, J., "Enhanced deformation mechanisms by anisotropic plasticity in polycrystalline Mg alloys at room temperature". *Metallurgical and Materials Transactions A*, 2005. **36**(7): p. 1689-1696.

16. Koike, J., et al., "The activity of non-basal slip systems and dynamic recovery at room temperature in fine-grained AZ31B magnesium alloys". *Acta Materialia*, 2003. **51**(7): p. 2055-2065.

17. Salem, A.A., S.R. Kalidindi, and S.L. Semiatin, "Strain hardening due to deformation twinning in [alpha]-titanium: Constitutive relations and crystal-plasticity modeling". *Acta Materialia*, 2005. **53**(12): p. 3495-3502.

18. Yoo, M., "Slip, twinning, and fracture in hexagonal close-packed metals". *Metallurgical and Materials Transactions A*, 1981. **12**(3): p. 409-418.

19. Yoo, M.H., "dislocation reactions in H.C.P. metals". *Scripta Metallurgica*, 1968. **2**(10): p. 537-540.

20. Barnett, M.R., et al., "Non-Schmid behaviour during secondary twinning in a polycrystalline magnesium alloy". *Acta Materialia*, 2008. **56**(1): p. 5-15.

21. Chino, Y., et al., "Mechanical anisotropy due to twinning in an extruded AZ31 Mg alloy". *Materials Science and Engineering: A*, 2008. **485**(1-2): p. 311-317.

22. Chino, Y., K. Kimura, and M. Mabuchi, "Twinning behavior and deformation mechanisms of extruded AZ31 Mg alloy". *Materials Science and Engineering: A*, 2008. **486**(1-2): p. 481-488.

23. Cizek, P. and M.R. Barnett, "Characteristics of the contraction twins formed close to the fracture surface in Mg-3Al-1Zn alloy deformed in tension". *Scripta Materialia*, 2008. **59**(9): p. 959-962.

24. Jiang, L., et al., "Influence of {10-12} extension twinning on the flow behavior of AZ31 Mg alloy". *Materials Science and Engineering: A*, 2007. **445-446**: p. 302-309.

25. Kim, J.H., et al., "Effect of calcium content on the microstructural evolution and mechanical properties of wrought Mg-3Al-1Zn alloy". *Materials Science and Engineering: A*, 2009. **525**(1-2): p. 18-29.

26. Koike, J. and R. Ohyama, "Geometrical criterion for the activation of prismatic slip in AZ61 Mg alloy sheets deformed at room temperature". *Acta Materialia*, 2005. **53**(7): p. 1963-1972.

27. Li, X., et al., "Orientational analysis of static recrystallization at compression twins in a magnesium alloy AZ31". *Materials Science and Engineering: A*, 2009. **517**(1-2): p. 160-169.

28. Mehrotra, P., T.M. Lillo, and S.R. Agnew, "Ductility enhancement of a heat-treatable magnesium alloy". *Scripta Materialia*, 2006. **55**(10): p. 855-858.

29. Mishra, R.K., et al., "Influence of cerium on the texture and ductility of magnesium extrusions". *Scripta Materialia*, 2008. **59**(5): p. 562-565.

30. Myshlyaev, M.M., et al., "Twinning, dynamic recovery and recrystallization in hot worked Mg-Al-Zn alloy". *Materials Science and Engineering A*, 2002. **337**(1-2): p. 121-133.

31. Tian, S., et al., "Microstructure evolution and deformation features of AZ31 Mg-alloy during creep". *Materials Science and Engineering: A*, 2006. **415**(1-2): p. 309-316.

32. Tokaji, K., et al., "Fatigue behaviour and fracture mechanism of a rolled AZ31 magnesium alloy". *International Journal of Fatigue*, 2004. **26**(11): p. 1217-1224.

33. Yang, F., et al., "Crack initiation mechanism of extruded AZ31 magnesium alloy in the very high cycle fatigue regime". *Materials Science and Engineering: A*, 2008. **491**(1-2): p. 131-136.

34. Yi, S.B., S. Zaefferer, and H.G. Brokmeier, "Mechanical behaviour and microstructural evolution of magnesium alloy AZ31 in tension at different temperatures". *Materials Science and Engineering: A*, 2006. **424**(1-2): p. 275-281.

35. Yoo, M., et al., "Nonbasal deformation modes of HCP metals and alloys: Role of dislocation source and mobility". *Metallurgical and Materials Transactions A*, 2002. **33**(13): p. 813-822.

36. Zeng, R., et al., "Influence of microstructure on tensile properties and fatigue crack growth in extruded magnesium alloy AM60". *International Journal of Fatigue*, 2010. **32**(2): p. 411-419.

37. Zeng, X., et al., "Effect of strontium on the microstructure, mechanical properties, and fracture behavior of AZ31 magnesium alloy". *Metallurgical and Materials Transactions A*, 2006. **37**(4): p. 1333-1341.

38. Cucchiara, R., et al., "An application of machine learning and statistics to defect detection". *Intelligent Data Analysis*, 2001. **5**(2): p. 151-164.

39. Bach, F., et al., "Macroscopic damage by the formation of shear bands during the rolling and deep drawing of magnesium sheets". *JOM Journal of the Minerals, Metals and Materials Society*, 2005. **57**(5): p. 57-61.

40. Adams, B., S. Wright, and K. Kunze, "Orientation imaging: The emergence of a new microscopy". *Metallurgical and Materials Transactions A*, 1993. **24**(4): p. 819-831.

41. Landon, C.D., B.L. Adams, and J. Kacher, "High-Resolution Methods for Characterizing Mesoscale Dislocation Structures". *Journal of Engineering Materials and Technology*, 2008. **130**(2): p. 021004-5.

42. Troost, K.Z., P. van der Sluis, and D.J. Gravesteijn, "Microscale elastic-strain determination by backscatter Kikuchi diffraction in the scanning electron microscope". *Applied Physics Letters*, 1993. **62**(10): p. 1110-1112.

43. Wilkinson, A.J., G. Meaden, and D.J. Dingley, "High resolution mapping of strains and rotations using electron backscatter diffraction". *Materials Science and Technology*, 2006. **22**: p. 1271-1278.

44. Wilkinson, A.J., G. Meaden, and D.J. Dingley, "High-resolution elastic strain measurement from electron backscatter diffraction patterns: New levels of sensitivity". *Ultramicroscopy*, 2006. **106**(4-5): p. 307-313.

45. Yang, W., B.L. Adams, and M. DeGraef, "Adaptive orientation imaging microscopy". *Microscopy and microanalysis*, 1999. **5**(suppl. 2): p. 246.

46. Gardner, C.J., et al., "EBSD-based continuum dislocation microscopy". *International Journal of Plasticity*, 2010. **26**(8): p. 1234-1247.

47. Venegas, V., et al., "EBSD study of hydrogen-induced cracking in API-5 L-X46 pipeline steel". *Scripta Materialia*, 2005. **52**(2): p. 147-152.

48. Feaver, M.J., "Comparative performance of BSI and CERL image quality indicators in detecting natural cracks in steel". *NDT International*, 1977. **10**(1): p. 13-20.

49. Gholinia, A. *Sample Preparation*. 6th Framework Program Marie Curie funded Summer Schools: Knowledge Based Materials 2005 [cited 2010 8/27]; Powerpoint presentation on sample preparation techniques]. Available from: http://materialsknowledge.org/index.php?option=com_wrapper&view=wrapper&Itemid=2.

50. Wilkinson, A.J. and D. Randman, "Determination of elastic strain fields and geometrically necessary dislocation distributions near nanoindents using electron back scatter diffraction". *Philosophical Magazine*, 2010. **90**(9): p. 1159 - 1177.

51. Zaefferer, S., S.I. Wright, and D. Raabe, "Three-Dimensional Orientation Microscopy in a Focused Ion Beam–Scanning Electron Microscope: A New Dimension of Microstructure Characterization". *Metallurgical and Materials Transactions A*, 2008. **39**(2): p. 374-389.

52. Yang, Y., G.D. Janaki Ram, and B.E. Stucker, "Bond formation and fiber embedment during ultrasonic consolidation". *Journal of Materials Processing Technology*, 2009. **209**(10): p. 4915-4924.

53. Král, V. and Z. Pelzbauer, "Penetration of an electron beam into material and energy dissipation". *International Journal of Radiation Applications and Instrumentation. Part C. Radiation Physics and Chemistry*, 1986. **28**(5-6): p. 511-514.

54. Semprimoschnig, C.O.A., et al., "A New Powerful Tool for Surveying Cleavage Fracture Surfaces". *Fatigue & Fracture of Engineering Materials & Structures*, 1997. **20**(11): p. 1541-1550.

55. Alam, M.N., M. Blackman, and D.W. Pashley, "High-Angle Kikuchi Patterns". *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 1954. **221**(1145): p. 224-242.

56. Field, D.P., "Recent advances in the application of orientation imaging". *Ultramicroscopy*, 1997. **67**(1-4): p. 1-9.

APPENDIX

Defect Detection Code

Read_ang.m

```
% Code to read in .ang file, clean up grain IDs and determine grain
% boundaries and triple junctions
% June 2010
% Stuart Rogers (MS Thesis)
close all; clear all;

%Select OIM .ang file
[OIMFile OIMPath]=uigetfile({'*.ang';'*.txt';'*.*'},'Select OIM .ang file.');
%Read in the scan file and store its info
fin =fopen([OIMPath OIMFile],'r');
curline=fgetl(fin);

% choose scan method
Scanchoose = {'Normal Scan','Custom Scan'};
[ss,vv] = listdlg('PromptString','what type of scan are you
doing?','SelectionMode','single','ListString',Scanchoose);

%Select CustomScan file
if ss ~= 1
    [TextFile TextPath] = uigetfile('*.txt','Select CustomScan text file');
    fid = fopen([TextPath TextFile],'r');
    fin2 = fopen([TextPath TextFile],'r');
    curline2 = fgetl(fin2);
end

%Select the smallest grain size
Small=inputdlg('What minimum grainsize do you want to apply?');%20;  %clean
up any grains smaller than this
small= str2num(Small{1});
mistol=15*pi/180;    %tolerance on misorientation for points to be in the
same grain

switch ss

    case 1 % Normal scan
        %read .ang file
        %skip through header information

        XStart=inputdlg('What was the X start position? ');
Xstart=str2num(XStart{1});
        YStart=inputdlg('What was the Y start position? ');
Ystart=str2num(YStart{1});
```

```
        while numel(curline)==0
            curline=fgetl(fin);
        end

        while curline(1)=='#'
            curline=fgetl(fin);
            while numel(curline)==0
                curline=fgetl(fin);
            end
        end

        %read in the orienation at point 1

        phi1rn = sscanf(curline,'%g',1);
        PHIrn=sscanf(curline,'%*s %g',1);
        phi2rn=sscanf(curline,'%*s %*s %g',1);
        xpos = sscanf(curline,'%*s %*s %*s %g',1);
        ypos = sscanf(curline,'%*s %*s %*s %*s %g',1);
        starty=sscanf(curline,'%*s %*s %*s %*s %g',1);
        IQ = sscanf(curline,'%*s %*s %*s %*s %*s %g',1);
        CI = sscanf(curline,'%*s %*s %*s %*s %*s %*s %g',1);

        %Create a data structure that will hold the orientation of every
point
        data.phi1rn(1)=phi1rn;
        data.PHIrn(1)=PHIrn;
        data.phi2rn(1)=phi2rn;
        data.IQ(1) = IQ;
        data.CI(1)=CI;
        data.xpos(1) = xpos+Xstart;
        data.ypos(1) = ypos+Ystart;

        %Read all remaining information and tore it as well
        numdatacols=0;
        counter=1;
        while curline~=-1
            phi1rn=sscanf(curline,'%g',1);
            PHIrn=sscanf(curline,'%*s %g',1);
            phi2rn=sscanf(curline,'%*s %*s %g',1);
            xpos = sscanf(curline,'%*s %*s %*s %g',1);
            ypos = sscanf(curline,'%*s %*s %*s %*s %g',1);
            cury=sscanf(curline,'%*s %*s %*s %*s %g',1);
            IQ = sscanf(curline,'%*s %*s %*s %*s %*s %g',1);
            CI = sscanf(curline,'%*s %*s %*s %*s %*s %*s %g',1);

            if numdatacols==0
                if cury~=starty
                    numdatacols=counter-1;
                end
            end

            if isempty(phi2rn)
                counter=counter+1;
                break
            end
```

```
            data.phi1rn(counter)=phi1rn;
            data.PHIrn(counter)=PHIrn;
            data.phi2rn(counter)=phi2rn;
            data.IQ(counter) = IQ;
            data.CI(counter)=CI;
            data.xpos(counter) = xpos+Xstart;
            data.ypos(counter) = ypos+Ystart;
            counter=counter+1;
            curline=fgetl(fin);
        end
        counter = counter-1;

        %close scan file
        fclose(fin);

    case 2 % Custom scan %

        %read .ang file
        %skip through header information

        while numel(curline)==0
            curline=fgetl(fin);
        end

        while curline(1)=='#'
            curline=fgetl(fin);
            while numel(curline)==0
                curline=fgetl(fin);
            end
        end

        %read in the orienation at point 1
        xpos = sscanf(curline2,'%*s %g',1);
        ypos = sscanf(curline2,'%*s %*s %*s %g',1);
        phi1rn = sscanf(curline,'%g',1);
        PHIrn=sscanf(curline,'%*s %g',1);
        phi2rn=sscanf(curline,'%*s %*s %g',1);
        starty=sscanf(curline,'%*s %*s %*s %*s %g',1);
        IQ = sscanf(curline,'%*s %*s %*s %*s %*s %g',1);
        CI = sscanf(curline,'%*s %*s %*s %*s %*s %*s %g',1);

        %Create a data structure that will hold the orientation of every
point
        data.phi1rn(1)=phi1rn;
        data.PHIrn(1)=PHIrn;
        data.phi2rn(1)=phi2rn;
        data.IQ(1,1) = IQ;
        data.CI(1)=CI;
        data.xpos(1) = xpos;
        data.ypos(1) = ypos;

        %Read all remaining orientations and store them as well
        % Store coordinates for each point

        counter=1;
```

```
        while curline2~=-1

            xpos = sscanf(curline2,'%*s %g',1);
            ypos = sscanf(curline2,'%*s %*s %*s %g',1);

            data.xpos(counter) = xpos;
            data.ypos(counter) = ypos;
            counter=counter+1;
            curline2=fgetl(fin2);
        end

        counter=2;
        numdatacols=0;
        curline=fgetl(fin);
        while curline~=-1
            phi1rn=sscanf(curline,'%g',1);
            PHIrn=sscanf(curline,'%*s %g',1);
            phi2rn=sscanf(curline,'%*s %*s %g',1);
            cury=sscanf(curline,'%*s %*s %*s %*s %g',1);
            IQ = sscanf(curline,'%*s %*s %*s %*s %*s %g',1);
            CI = sscanf(curline,'%*s %*s %*s %*s %*s %*s %g',1);

            if numdatacols==0
                if cury~=starty
                    numdatacols=counter-1;
                end
            end

            if isempty(phi2rn)
                counter=counter+1;
                break
            end

            data.phi1rn(counter)=phi1rn;
            data.PHIrn(counter)=PHIrn;
            data.phi2rn(counter)=phi2rn;
            data.IQ(counter) = IQ;
            data.CI(counter)=CI;
            counter=counter+1;
            curline=fgetl(fin);
        end
        counter = counter-1;

        %close scan file
        fclose(fin);
        fclose(fin2);
end

maxx=max(data.xpos);
maxy=max(data.ypos);

if data.xpos(length(data.xpos))<maxx;
    maxy=maxy-1;
end
```

```
stepx=data.xpos(3)-data.xpos(2);
nx=(floor((maxx-data.xpos(1))/stepx))+1;
ny=(floor((maxy-data.ypos(1))/stepx))+1;%floor(length(data.ypos)/nx);
X=zeros(nx,ny);   %variable to hold the scan size
angles(:,1) = data.phi1rn(1:(nx*ny));
angles(:,2) = data.PHIrn(1:(nx*ny));
angles(:,3) = data.phi2rn(1:(nx*ny));

N=length(angles(:,1));  % number of orientations
angles = reshape(angles,[size(X) 3]);
anglesR=circshift(angles,[0 -1 0]);   % shift matrix left
anglesD=circshift(angles,[-1 0 0]);   % shift matrix up
angles = reshape(angles,[N 3]);
anglesR = reshape(anglesR,[N 3]);
anglesD = reshape(anglesD,[N 3]);

gmat = @(phi1,Phi,phi2) [cos(phi1).*cos(phi2)-sin(phi1).*cos(Phi).*sin(phi2),
-cos(phi1).*sin(phi2)-sin(phi1).*cos(Phi).*cos(phi2), sin(phi1).*sin(Phi),
...
    sin(phi1).*cos(phi2)+cos(phi1).*cos(Phi).*sin(phi2), -
sin(phi1).*sin(phi2)+cos(phi1).*cos(Phi).*cos(phi2), -cos(phi1).*sin(Phi) ,
...
    sin(Phi).*sin(phi2), sin(Phi).*cos(phi2), cos(Phi)];

ind = reshape(1:9,3,3);
indT=ind';
Q = gmat(angles(:,1),angles(:,2),angles(:,3));
QRtemp= gmat(anglesR(:,1),anglesR(:,2),anglesR(:,3));
QDtemp= gmat(anglesD(:,1),anglesD(:,2),anglesD(:,3));
count=0;
for jj = 1:3
    for ii =1:3
        count=count+1;
        QR(:,count) = sum(QRtemp(:,ind(ii,:)).*Q(:,indT(:,jj)),2);
        QD(:,count) = sum(QDtemp(:,ind(ii,:)).*Q(:,indT(:,jj)),2);
    end
end

load CrystalRotations.mat

MisOrR = 10000*ones(size(Q,1),1);    % initiate misorientation matrices
MisOrD = 10000*ones(size(Q,1),1);    % initiate misorientation matrices

for kk = 1:24
    count = 0 ;

    R = repmat(reshape(CubicTriclinicrot(:,:,kk),1,9),size(Q,1),1);

    for jj = 1:3
        for ii =1:3
            count  =count +1;
            Q_primeR(:,count) = sum(R(:,ind(ii,:)).*QR(:,ind(:,jj)),2);
            Q_primeD(:,count) = sum(R(:,ind(ii,:)).*QD(:,ind(:,jj)),2);
        end
    end
```

```
    MisOr_tempR = acos((sum(Q_primeR(:,diag(ind)),2)-1)/2);
    smaller = find(MisOr_tempR < MisOrR );
    MisOrR(smaller) = MisOr_tempR(smaller);
    MisOr_tempD = acos((sum(Q_primeD(:,diag(ind)),2)-1)/2);
    smaller = find(MisOr_tempD < MisOrD );
    MisOrD(smaller) = MisOr_tempD(smaller);

end

MisOrR=real(reshape(MisOrR,size(X)));
MisOrD=real(reshape(MisOrD,size(X)));
[nrows ncols]=size(X);
grains=zeros(nrows+2,ncols+2);
grains(:,1)=1e20;
grains(:,ncols+2)=1e20;
grains(1,:)=1e20;
grains(nrows+2,:)=1e20;
% make final col of MisOrR large to remove possibility of grains wrapping
around
MisOrR(:,ncols)=1e20;
MisOrD(nrows,:)=1e20;
MisOrR=[ones(1,ncols)*1e20;MisOrR;ones(1,ncols)*1e20];
MisOrR=[ones(nrows+2,1)*1e20 MisOrR ones(nrows+2,1)*1e20];
MisOrD=[ones(1,ncols)*1e20;MisOrD;ones(1,ncols)*1e20];
MisOrD=[ones(nrows+2,1)*1e20 MisOrD ones(nrows+2,1)*1e20];
grains(2,2)=1;
flag=0;
ngrain=1;   %used to enumerate the grains

tic
while flag==0
    flag=1;
    nadds=0;
    for i=2:nrows+1
        for j=2:ncols+1
            if ((j>1)&&(MisOrR(i,j-1)<mistol)) && ((i>1)&&(MisOrD(i-
1,j)<mistol)) && (grains(i,j)~=min(grains(i,j-1),grains(i-1,j)));
                grains(i,j)=min(grains(i,j-1),grains(i-1,j));
                flag=0;
                nadds=nadds+1;
            elseif (i>1)&&(MisOrD(i-1,j)<mistol) && (grains(i,j)~=grains(i-
1,j))
                grains(i,j)=grains(i-1,j);
                flag=0;
                nadds=nadds+1;
            elseif (j>1)&&(MisOrR(i,j-1)<mistol) && (grains(i,j)~=grains(i,j-
1))
                grains(i,j)=grains(i,j-1);
                flag=0;
                nadds=nadds+1;
            elseif grains(i,j)==0
                ngrain=ngrain+1;
                grains(i,j)=ngrain;
            end
        end
```

```
    end
    for i=nrows+1:-1:2
        for j=ncols+1:-1:2
            if ((j<ncols+2)&& (MisOrR(i,j)<mistol)) && ((i<nrows+2)&&
(MisOrD(i,j)<mistol)) && (grains(i,j)~=min(grains(i,j+1),grains(i+1,j)))
                grains(i,j)=min(grains(i,j+1),grains(i+1,j));
                flag=0;
                nadds=nadds+1;
            elseif (i<nrows+2)&& (MisOrD(i,j)<mistol) &&
(grains(i,j)~=grains(i+1,j))
                grains(i,j)=grains(i+1,j);
                flag=0;
                nadds=nadds+1;
            elseif (j<ncols+2)&& (MisOrR(i,j)<mistol) &&
(grains(i,j)~=grains(i,j+1))
                grains(i,j)=grains(i,j+1);
                flag=0;
                nadds=nadds+1;
            end
        end
    end
    %    nadds
    if nadds<1; flag=1;end

end
toc
grains(:,1)=0;
grains(:,ncols+2)=0;
grains(1,:)=0;
grains(nrows+2,:)=0;

% now work out grain sizes at each point
grains=grains(2:nrows+1,2:ncols+1);
grains=reshape(grains,(ncols)*(nrows),1);
grainsize=zeros(1,max(grains));
for i=1:length(grains)
    grainsize(grains(i))=grainsize(grains(i))+1;
end

grains=reshape(grains,(nrows),(ncols));
sizes=grains;
sizes(:,:)=grainsize(grains(:,:));

RGB = label2rgb(grains,'jet','c','shuffle');
% figure('Name','Grains','NumberTitle','off'); imshow(RGB);

% Remove small grains %
numsmall=sum((grainsize<small).*(grainsize>0))
goodgrains=grains;
goodgrainsize=grainsize;
while numsmall>0

    [goodgrains,goodgrainsize] = cleanup(goodgrains,goodgrainsize,small);
    numsmall=sum((goodgrainsize<small).*(goodgrainsize>0));
end
goodsizes=goodgrains;
```

```
goodsizes(:,:)=goodgrainsize(goodgrains(:,:));

RGB1 = label2rgb(goodgrains,'jet','c','shuffle');
% figure('Name','GoodGrains','NumberTitle','off'); imshow(RGB1);

rc = size(goodgrains);

% Find all grain boundaries %
Search = 1;
BOUND = zeros(nx,ny);

for i=Search+1:nx-Search-1;
    for j=Search+1:ny-Search-1;
        ker = goodgrains(i-Search:i+Search,j-Search:j+Search);
        Ker = unique(ker);
        if length(Ker)~=1;
            BOUND(i,j)=1;
        end
    end
end

RGB2=label2rgb(BOUND);
% figure('Name','Grain Boundaries','NumberTitle','off'); imshow(RGB2);

% Find Triple Junctions %
search = 1;
TRIP=zeros(nx,ny);

for j=search+1:nx-search-1;
    for i=search+1:ny-search-1;
        Kernel = goodgrains(j-search:j+search,i-search:i+search);
        uniquelist = unique(Kernel);
        if length(uniquelist) > 2
            TRIP(j,i)=1;
        end
    end
end

RGB3=label2rgb(TRIP);
% figure('Name','Triple Junctions','NumberTitle','off'); imshow(RGB3);

RGB4=label2rgb(TRIP+BOUND);
% figure('Name','Hot Spots','NumberTitle','off'); imshow(RGB4);

subplot(2,2,1); imshow(RGB); title('Grains');
subplot(2,2,2); imshow(RGB1); title('Good Grains');
subplot(2,2,3); imshow(RGB2); title('Grain Boundaries');
subplot(2,2,4); imshow(RGB4); title('Triple Junctions');


save GoodStuff1 goodgrains RGB RGB1 BOUND RGB2 TRIP RGB3 RGB4 counter...
    data numdatacols nx ny maxx maxy;
```

Defect_map.m

```
% Shadow detection after running read_ang.m
% June 2010
% Stuart Rogers (MS Thesis)

clear all; close all;

load GoodStuff1;

%Prompt for the first image in the series
%it must have 1 before the file extension
[ImageName,ImagesPath] = uigetfile({'*.bmp';'*.jpeg';'*.*'},['Select the
First Image ']);
if ImageName==0
    delete(gcf)
    error('Exited by user')
end

%Determine the number of characters in the file extension
dotpos=find(ImageName=='.');
if length(dotpos) > 1
    dotpos = dotpos(length(dotpos));
end
charsinext=length(ImageName)-dotpos;
if charsinext~=3 && charsinext~=4
    error('I had trouble reading in the file names')
end

%Determine the number of characters in the file extension
dotpos=find(ImageName=='.');
if length(dotpos) > 1
    dotpos = dotpos(length(dotpos));
end
charsinext=length(ImageName)-dotpos;
if charsinext~=3 && charsinext~=4
    error('I had trouble reading in the file names')
end

ImageNames = cell([counter/numdatacols,numdatacols]);
for r = 1:counter/numdatacols
    for c = 1:numdatacols
        ImageNames{r,c} = [ImageName(1:end-charsinext-4) num2str(r-1) 'c'
num2str(c-1) ImageName(end-charsinext:end)];
    end
end


Mask = 200; %Can change this value to check for a bigger shape
DefectMap = zeros(size(ImageNames,1), size(ImageNames,2));
PatternSize = imread ([ImagesPath ImageNames{1,1}]); %read in pattern size
from first EBSD pattern
PatternSize = length(PatternSize);
template=zeros(PatternSize,PatternSize); %template for fft shadow testing
```

```
template(1:Mask,1:Mask)=1;
templateFFT = conj(fftn(template));

%Reading in and testing each pattern, and building a defect map
counter=1;
for kk = 1:size(ImageNames,1) %Microstructure map loop
    for ll = 1;%:size(ImageNames,2)
       ImageNames{kk,ll}
       pattern = imread([ImagesPath ImageNames{kk,ll}]);
       % pattern = sum(pattern,3)/3;
       NewPattern=im2bw(pattern,.3);
       NewPattern = 1 - NewPattern;

       convol=ifftn(templateFFT.*fftn(NewPattern))/(Mask^2);
       shadow(counter)=max(max(convol));
       if max(max(convol))>=1; %>Mask.^2-1
           DefectMap(kk,ll)=1;
       else
           DefectMap(kk,ll)=0;
       end
  counter=counter+1;
       clear pattern NewPattern MaxShadow
    end
end
toc
imshow(DefectMap);

count=1;
for i=1:kk;
    for j=1:ll;
        data.DM(count) = DefectMap(i,j);
        data.IN(count) = ImageNames(i,j);
        data.IP{count} = ImagesPath;
        count=count+1;
    end
end


save GoodStuff1 goodgrains RGB RGB1 BOUND RGB2 TRIP RGB3 RGB4 counter...
     data numdatacols nx ny maxx maxy kk ll PatternSize;
```

Defect_scan.m

```
% Code to create a custom scan file that refines the scan grid around
% possible defects after running Defect_map.m
% June 2010
% Stuart Rogers (MS Thesis)

clear all; close all

% Read in coarse defect map %
load GoodStuff1;

% Specify scan refinement factor %
Scale=inputdlg('What factor do you want to refine the mesh by?');
scale= str2num(Scale{1});

% Determine grain refinement %
xstep = data.xpos(2)-data.xpos(1);
ystep = xstep; %data.ypos(nx+2)-data.ypos(1);

Xstep = xstep/scale;
Ystep = ystep/scale;

Xpos = [data.xpos(1):Xstep:maxx];
Ypos = [data.ypos(1):Ystep:maxy];

% Create scan matrix %
scan = zeros(length(Xpos),length(Ypos),3);

for i = 1:length(Xpos);
    scan(i,:,2) = Xpos(i);
end

for j = 1:length(Ypos);
    scan(:,j,3) = Ypos(j);
end

% Reshape defect map %
count=1;
for i=1:kk;
    for j=1:ll;
        Defectmap(i,j)=data.DM(count);
        count=count+1;
    end
end

% Scale edge points %
% Horizontals


for i = 2:kk-1;
    for j = 1;
        if Defectmap(i,j)==1;
```

```
            scan((scale*i)-2:scale*i, (scale*j)-1:scale*j,1)=1;
        end
    end
end

for i = 2:kk-1;
    for j = ll;
        if Defectmap(i,j)==1;
            scan((scale*i)-2:scale*i, (scale*j)-2:(scale*j)-1,1)=1;
        end
    end
end

% Verticals
for i = 1;
    for j = 2:ll-1;
        if Defectmap(i,j)==1;
            scan((scale*i)-1:scale*i, (scale*j)-2:scale*j,1)=1;
        end
    end
end

for i = kk;
    for j = 2:ll-1;
        if Defectmap(i,j)==1;
            scan((scale*i)-2:(scale*i)-1, (scale*j)-2:scale*j,1)=1;
        end
    end
end

% Main body

for i = 2:kk-1;
    for j = 2:ll-1;
        if Defectmap(i,j)==1;
            scan((scale*i)-2:scale*i, (scale*j)-2:scale*j, 1)=1;
        end
    end
end

numpoints = sum(sum(scan(:,:,1)));

% create vector of points
% bx = zeros(1,numpoints);
% by = zeros(1,numpoints);

count=1;
for i = 1:(nx*scale)-1;
    for j = 1:(ny*scale)-1;
        if scan(i,j,1)==1;
            bx(1,count) = scan(i,j,2);
            by(1,count) = scan(i,j,3);
            count = count+1;
        end
    end
```

```
end

% bx = bx/1000;
% by = by/1000;

row = 1;
col = 1;

fid = fopen('custscan.txt','w');

%write first line of file
fprintf(fid,'%s\t %1.3f\t %s\t %1.3f\t %s\t %1.3f\t %s\t %1.3f\t %s\t
%05i%s%i\n',...
    'bx',bx(1),'by',by(1),'rx',bx(1),'ry',by(1),'pn',row,'c',col);

col = col+1;

%write the rest of the file
for i = 1:length(bx)-2
    fprintf(fid,'%s\t %1.3f\t %s\t %1.3f\t %s\t %1.3f\t %s\t %1.3f\t %s\t
%05i%s%i\n',...

'bx',bx(i+1),'by',by(i+1),'rx',bx(i+1),'ry',by(i+1),'pn',row,'c',col);
    if bx(i+2)~=bx(i+1);
        row = row+1;
        col = 1;
    else
        col = col+1;
    end
end

% write last line twice because of issue with OIM custom scan stuff
for j=i:i+1;

    fprintf(fid,'%s\t %1.3f\t %s\t %1.3f\t %s\t %1.3f\t %s\t %1.3f\t %s\t
%05i%s%i\n',...

'bx',bx(i+2),'by',by(i+2),'rx',bx(i+2),'ry',by(i+2),'pn',row,'c',col);
end

fclose(fid);

plot(bx,by,'.');
axis('equal'); xlabel('X');

save GoodStuff1 goodgrains data RGB RGB1 BOUND RGB2 TRIP RGB3 RGB4...
     counter numdatacols nx ny maxx maxy scale scan;
```

Splice_data.m

```
% Code to integrate refined scan into original scan data
% Runs between itterations of Defect_Scan.m
% June 2010
% Stuart Rogers (MS Thesis)

close all; clear all;

% Read in coarse defect map %
load GoodStuff1;

% rearrange existing data %
count = 1;
for j=1:ny;
    for i=1:nx;
        X(i,j)=data.xpos(count);
        Y(i,j)=data.ypos(count);
        phi1(i,j)=data.phi1rn(count);
        PHI(i,j)=data.PHIrn(count);
        phi2(i,j)=data.phi2rn(count);
        iq(i,j)=data.IQ(count);
        DM(i,j)=data.DM(count);
        IN(i,j)=data.IN(count);
        IP(i,j)=data.IP(count);
        count=count+1;
    end
end

nx = (nx*scale)-1;
ny = (ny*scale)-1;

xstep=(data.xpos(2)-data.xpos(1))/scale;
ystep=xstep;

Xpos = [data.xpos(1):xstep:maxx];
Ypos = [data.ypos(1):ystep:maxy];

% create new data structure %
for j=1:ny;
    for i=1:nx;
        X2(i,j)=Xpos(i);
        Y2(i,j)=Ypos(j);
        phi12(i,j)=phi1(ceil(i/scale),ceil(j/scale));
        PHI2(i,j)=PHI(ceil(i/scale),ceil(j/scale));
        phi22(i,j)=phi2(ceil(i/scale),ceil(j/scale));
        iq2(i,j)=iq(ceil(i/scale),ceil(j/scale));
        DM2(i,j)=DM(ceil(i/scale),ceil(j/scale));
        IN2(i,j)=IN(ceil(i/scale),ceil(j/scale));
        IP2(i,j)=IP(ceil(i/scale),ceil(j/scale));
    end
end

% Vectorize Result %
```

```
MeshedX = X2(:);
MeshedY = Y2(:);
Meshedphi1 = phi12(:);
MeshedPHI = PHI2(:);
Meshedphi2 = phi22(:);
MeshedIQ = iq2(:);
MeshedDM = DM2(:);
MeshedIN = IN2(:);
MeshedIP = IP2(:);

% read in new scan data %

% Select OIM .ang file
[OIMFile OIMPath]=uigetfile({'*.ang';'*.txt';'*.*'},'Select refined OIM .ang
file.');
%Read in the scan file and store its info
fin =fopen([OIMPath OIMFile],'r');

%Select CustomScan file
[TextFile TextPath] = uigetfile('*.txt','Select CustomScan text file');
fid = fopen([TextPath TextFile],'r');

%Prompt for the first image in the series
%it must have 1 before the file extension
[ImageName,ImagesPath] = uigetfile({'*.bmp';'*.jpeg';'*.*'},['Select the
First Image ']);
if ImageName==0
    delete(gcf)
    error('Exited by user')
end

preamble = sscanf(ImageName, '%c',6);

%read .ang file
%skip through header information
curline=fgetl(fin);
while numel(curline)==0
    curline=fgetl(fin);
end

while curline(1)=='#'
    curline=fgetl(fin);
    while numel(curline)==0
        curline=fgetl(fin);
    end
end

% Read in Custom Scan info
fin2=fopen([TextPath TextFile],'r');
curline2=fgetl(fin2);

%read in the orienation at point 1
nxpos = sscanf(curline2,'%*s %g',1);
nypos = sscanf(curline2,'%*s %*s %*s %g',1);
nIN = sscanf(curline2,'%*s %*s %*s %*s %*s %*s %*s %*s %*s %s',1);
```

```
nphi1rn = sscanf(curline,'%g',1);
nPHIrn=sscanf(curline,'%*s %g',1);
nphi2rn=sscanf(curline,'%*s %*s %g',1);
nstarty=sscanf(curline,'%*s %*s %*s %*s %g',1);
nIQ = sscanf(curline,'%*s %*s %*s %*s %*s %g',1);

newdata.phi1rn(1)=nphi1rn;
newdata.PHIrn(1)=nPHIrn;
newdata.phi2rn(1)=nphi2rn;
newdata.IQ(1,1) = nIQ;
newdata.IN{1} = [preamble num2str(nIN) num2str('.bmp')];
newdata.xpos(1) = nxpos;
newdata.ypos(1) = nypos;

%Read all remaining orientations and store them as well
curline2=fgetl(fin2);
counter=2;
while curline2~=-1
    nxpos = sscanf(curline2,'%*s %g',1);
    nypos = sscanf(curline2,'%*s %*s %*s %g',1);
    nIN = sscanf(curline2,'%*s %*s %*s %*s %*s %*s %*s %*s %*s %*s %s',1);

    newdata.IN{counter} = [preamble num2str(nIN) num2str('.bmp')];
    newdata.xpos(counter) = nxpos;
    newdata.ypos(counter) = nypos;
    counter=counter+1;
    curline2=fgetl(fin2);
end

%close scan file
fclose(fin2);

counter=2;
numdatacols=0;
curline=fgetl(fin);
while curline~=-1
    nphi1rn=sscanf(curline,'%g',1);
    nPHIrn=sscanf(curline,'%*s %g',1);
    nphi2rn=sscanf(curline,'%*s %*s %g',1);
    ncury=sscanf(curline,'%*s %*s %*s %*s %g',1);
    nIQ = sscanf(curline,'%*s %*s %*s %*s %*s %g',1);

    if numdatacols==0
        if ncury~=nstarty
            numdatacols=counter-1;
        end
    end

    if isempty(nphi2rn)
        counter=counter+1;
        break
    end

    newdata.phi1rn(counter)=nphi1rn;
    newdata.PHIrn(counter)=nPHIrn;
```

```
    newdata.phi2rn(counter)=nphi2rn;
    newdata.IQ(counter) = nIQ;
    counter=counter+1;
    curline=fgetl(fin);
end

% Analyse new images

%Determine the number of characters in the file extension
% dotpos=find(ImageName=='.');
% if length(dotpos) > 1
%     dotpos = dotpos(length(dotpos));
% end
% charsinext=length(ImageName)-dotpos;
% if charsinext~=3 && charsinext~=4
%     error('I had trouble reading in the file names')
% end
%
% %Determine the number of characters in the file extension
% dotpos=find(ImageName=='.');
% if length(dotpos) > 1
%     dotpos = dotpos(length(dotpos));
% end
% charsinext=length(ImageName)-dotpos;
% if charsinext~=3 && charsinext~=4
%     error('I had trouble reading in the file names')
% end
%
% ImageNames = cell([counter/numdatacols,numdatacols]);
% for r = 1:counter/numdatacols
%     for c = 1:numdatacols
%         ImageNames{r,c} = [ImageName(1:end-charsinext-4) num2str(r-1) 'c'
num2str(c-1) ImageName(end-charsinext:end)];
%     end
% end

ImagesNames=newdata.IN;

Mask = 200; %Can change this value to check for a bigger shape
DefectMap = zeros(length(ImagesNames));
PatternSize = imread([ImagesPath ImagesNames{1,1}]); %read in pattern size
from first EBSD pattern
PatternSize = length(PatternSize);
template=zeros(PatternSize,PatternSize); %template for fft shadow testing
template(1:Mask,1:Mask)=1;
templateFFT = conj(fftn(template));

%Reading in and testing each pattern, and building a defect map
for Q = 1:length(newdata.IN) %Microstructure map loop
       newdata.IN(Q)
       pattern = imread([ImagesPath newdata.IN{Q}]);
       pattern = double(pattern) + 1;%sum(pattern,3)/3;
       NewPattern=im2bw(pattern,.3);
       NewPattern = 1 - NewPattern;

       convol=ifftn(templateFFT.*fftn(NewPattern))/(Mask^2);
```

```
        if max(max(convol))>=1; %>Mask.^2-1
            DefectMap(Q)=1;
        else
            DefectMap(Q)=0;
        end

        clear pattern NewPattern MaxShadow
end
toc

NDM = DefectMap(:);

% vectorize new Images %

NIN = ImagesNames(:);
NIP = ImagesPath;

% Determine overlap points %

XYBig = [ X2(:) Y2(:)];
XYSmall = [newdata.xpos(:) newdata.ypos(:)] ;

% [c, iBig, iSmall] = intersect(XYBig,XYSmall);
%
% % Overwrite data structure with new data %
%
% for hh =1 : length(Bigi)
%     MeshedX(Bigi) = newdata.xpos(Smalli);
%     MeshedY(Bigi) = newdata.ypos(Smalli);
%     Meshedphi1(Bigi) = newdata.phi1rn(Smalli);
%     MeshedPHI(Bigi) = newdata.PHIrn(Smalli);
%     Meshedphi2(Bigi) = newdata.phi2rn(Smalli);
%     MeshedIQ(Bigi) = newdata.IQ(Smalli);
%     MeshedDM(Bigi) = NDM(Smalli);
%     MeshedIN(Bigi) = NIN(Smalli);
%     MeshedIP(Bigi) = NIP;
% end
%
for l=1:length(XYBig);
    for m = 1:length(XYSmall);
        if XYBig(l,1) == XYSmall(m,1);
            if XYBig(l,2) == XYSmall(m,2);
                MeshedX(l) = newdata.xpos(m);
                MeshedY(l) = newdata.ypos(m);
                Meshedphi1(l) = newdata.phi1rn(m);
                MeshedPHI(l) = newdata.PHIrn(m);
                Meshedphi2(l) = newdata.phi2rn(m);
                MeshedIQ(l) = newdata.IQ(m);
                MeshedDM(l) = NDM(m);
                MeshedIN(l) = NIN(m);
                MeshedIP{l} = NIP;
                count=count+1;
            end
        end
    end
```

```
end


% Reshape coloumn to grid %

count=1;
for j=1:ny;
    for i=1:nx;
        DefectMap(i,j) = MeshedDM(count);
        MX(i,j) = MeshedX(count);
        MY(i,j) = MeshedY(count);
        Mphi1(i,j) =  Meshedphi1(count);
        MPHI(i,j) = MeshedPHI(count);
        Mphi2(i,j) = Meshedphi2(count);
        MIQ(i,j) = MeshedIQ(count);
        MDM(i,j) = MeshedDM(count);
        MIN(i,j) = MeshedIN(count);
        MIP(i,j) = MeshedIP(count);
        count = count+1;
    end
end

subplot(1,2,1); imshow(DM); title('Defect Map1');
subplot(1,2,2); imshow(MDM); title('Refined Defect Map');

% Reshape grid to row %

count=1;
for j=1:ny;
    for i=1:nx;
        data.xpos(count) = MX(i,j);
        data.ypos(count) = MY(i,j);
        data.phi1rn(count) = Mphi1(i,j);
        data.PHIrn(count) = MPHI(i,j);
        data.phi2rn(count)= Mphi2(i,j);
        data.IQ(count) = MIQ(i,j);
        data.DM(count) = MDM(i,j);
        data.IN(count) = MIN(i,j);
        data.IP(count) = MIP(i,j);
        count = count+1;
    end
end

save GoodStuff1 goodgrains RGB RGB1 BOUND RGB2 TRIP RGB3 RGB4 counter...
     data numdatacols nx ny maxx maxy scale;
```