University of California
Santa Barbara

Minimum Angle Transformation Loss for Superresolution of Electron Microscope Orientation Data

A Thesis submitted in partial satisfaction
of the requirements for the degree
Master of Science
in
Electrical and Computer Engineering
by
Joaquin Matias Giorgi

Committee in charge:

Professor B.S. Manjunath, Chair
Professor S. Chandrasekaran
Professor Kenneth Rose

August 2024

The Thesis of Joaquin Matias Giorgi is approved.

Professor S. Chandrasekaran

Professor Kenneth Rose

Professor B.S. Manjunath, Committee Chair

August 2024

Minimum Angle Transformation Loss for Superresolution of Electron Microscope Orientation Data

by

Joaquin Matias Giorgi

I would like to dedicate this Thesis to my parents, for their encouragement, inspiration and prioritizing my educational fulfillment.

Acknowledgements

First and foremost, I would like to thank Professor B.S. Manjunath for believing in my abilities to contribute to a technically difficult project, in a novel and interesting domain; also for taking care of the day-to-day operations of the lab and ensuring students have all the infrastructure required to collaborate and work comfortably. Secondly, Dr. Devendra Jangid for his mentorship in machine learning fundamentals, as well as it's application to novel scientific datasets. And of course a special thank you to Dr. Neal Brodnick, for patiently explaining many fundamental materials science concepts, which were necessary to form a clear understanding of this domain, as well as constantly proposing and troubleshooting ideas in the intersection of materials science and machine learning. Thank you as well to Bowen, Raphael and Satish, for helping debug my project in the initial phases, as well as Connor Levenson for the software engineering tips and assistance. Thank you as well to the other lab members for making the time in the lab a memorable experience.

Abstract

Minimum Angle Transformation Loss for Superresolution of Electron Microscope Orientation Data

by

Joaquin Matias Giorgi

Electron backscatter diffraction (EBSD) is a scanning electron microscopy technique used for collecting orientation properties of a material sample over space at the micrometer scale. Because collecting this data is known for being costly and time-consuming, various methods have been proposed to upsample collected data, or generate new microstructures from a latent space. We propose a novel interpolation algorithm for quaternions that is imprevious to symmetry switching, named Minimum Angle Transformation Spherical Linear Interpolation (Slerp-MAT). We also propose a new Physics-based loss function based on on this algorithm, to obtain state-of-the-art results, in terms of the angular difference of the superresolved data and the ground truth. The result is a 882% reduction in mean angular distance of Superresolved versus Ground Truth data for the collected Nickel dataset with respect to the previous state-of-the-art loss function, and a 321% reduction for the collected Titanium dataset.

Contents

Chapter 1

Background

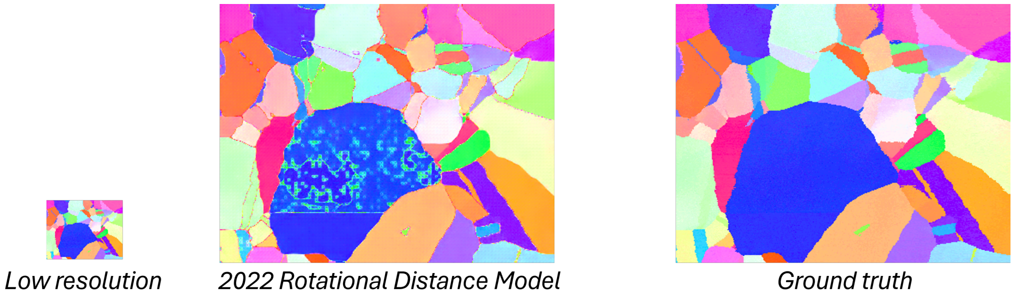
1.1 Computational Materials Science

Computational materials science sits at the intersection of optimization, materials science, and engineering. The performance of materials studied by scientists and engineers can have key societal impacts, leading to performance improvements in areas such as battery degradation, engine degradation, solar panel efficiency, and mechanical joint sustainability. These materials, such as various classes of Titanium and Nickel, have atoms arranged into repeating structures, or unit cells. These patterns of unit cells are called crystal structures, which are visible when observed with an optical instrument at the micrometer scale. The data captured by these instruments are commonly referred to as microstructures.

Due to the importance of new and improved materials, the exploration of how engineering ideas can enter this optimization process is an area under investigation in academia as well as industry. For the case of this thesis, we focus on how interpolation and machine learning models can be used to superresolve electron backscatter diffraction data.

1.2 Research Context

The research for this thesis focuses on addressing incorrect treatment of symmetry observed in color maps when using both interpolation and machine learning based super-resolution architectures. Despite seemingly accurate results in the color mapping for Titanium data, according to common superresolution metrics (e.g. Peak Signal-to-Noise Ratio), minor incorrect color mapping was still observed at the edges of some EBSD colorized images for Titanium. Moreover, when training and testing the same model on Nickel data, very apparent incorrect coloring was found during evaluation, as seen in Figure 1.1. The primary objective was to identify the cause of these mapping errors, and adapt the models to work effectively across various materials datasets. This investigation revealed several underlying issues in interpolation and deep learning approaches, which we present here. The findings provide valuable insights into implementing and testing computer vision and machine learning algorithms for Electron Backscatter Diffraction (EBSD) data.


Low resolution

2022 Rotational Distance Model

Ground truth

Figure 1.1: Left-to-right: decimated EBSD Image (LR), Upsampled image of LR using state-of-the-art physics-based rotational distance loss in [1] (SR), ground truth undecimated image (HR). Rotational distance loss SR EBSD Images have areas of aliasing, such as the green patterns in the otherwise blue center region.

As further explained in section 1.3.4, the color mapping for the discretized orientations of crystal structures indicates the vector in the local frame that is parallel to a chosen basis vector of the sample orientation. In Figure 1.1, we observe a superresolved version (SR) of the low-resolution (LR) EBSD Image - obtained by decimating a high-resolution (HR) ground truth image from the microscope. When comparing the SR to the HR, we observe a green pattern in the blue center region of the SR, which is not present in the HR, due to incorrect treatment of symmetrically equivalent orientations when calculating angular distances.

1.3 Microstructures

1.3.1 Data Collection

Crystal structure data is obtained by placing a metal sample (a Nickel or Titanium variant, for instance) under what is known as an electron backscatter diffraction microscope (abbreviated as EBSD), as in Figure 1.2. Once the sample is placed, the microscope sends an electron beam towards the metal sample, producing electron orientation scattering patterns known as Kikuchi patterns [2]. A commercial software (in this case, Dream3D) [3], then translates the analog electron orientation patterns into discrete digital orientation patterns at each array element.

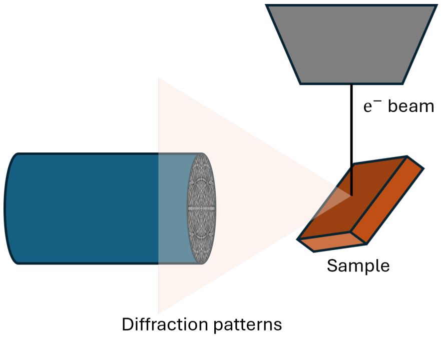


Figure 1.2: EBSD Microscope data collection. A focused electrom beam is incident to a metal sample, resulting in Kikuchi diffraction patterns. Orientation data is inferred from these patters, which is digitized for further processing.

The orientation at each micro-meter scale element is obtained from a nano-meter scale diffraction spot, and it is assumed that all atoms captured within the spatial size represented by each output array element are packed into repeating unit cell structures with identical orientation. Typically, array elements representing closely oriented unit cells (angular difference of less than $\theta = 3°$) are adjacent to each other, and the group made up by such pixels is known as a grain.

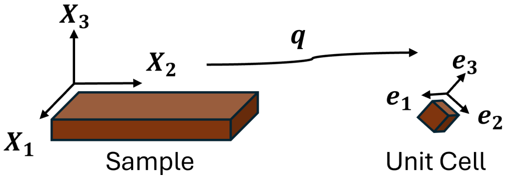


Figure 1.3: Orientation of the metal sample versus Orientation of the microscopic unit cell in a micro-meter scaled section of the sample. The EBSD microscope outputs a quaternion array, where each quaternion q is the transformation from the sample Orientation to the Orientation of the unit cell represented by that array element.

In its raw form, the collected and digitized data is a three-dimensional matrix, with a quaternion stored at each array element. These quaternions represent the orientation transformation from the macroscopic sample orientation, to the orientation of the microscopic unit cell of that element, as shown in Figure 1.3. The three-dimensional orientation data is typically viewed graphically by materials scientists using a series of color orientation maps, which can provide a visual intuition to different phenomena occurring within grains, and at the boundary between them.

1.3.2 Crystal Poles

Each element in a microstructure data array contains the quaternion determining the orientation of the unit cells in the spatial area represented by that element, with respect to the orientation of the sample. In order to make this data more easily digestible in a visual format, unit cell orientations are approximated by color orientation maps, which

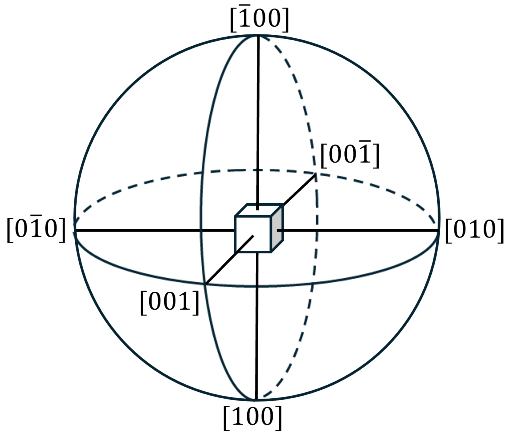


Figure 1.4: Poles of the faces of a cube: [001], [010], [100], and the negative versions of these, denoted with an overscore. For each pole, there are four symmetrically equivalent orientations of the cube by rotating about the pole by a factor of 90°.

allow material scientists to visually infer precise insights about the material - which can't be directly extracted through computation on the data.

Industry standard color orientation maps do not directly translate euler angles or quaternion components to color, but instead denote the point of intersection of a crystal direction vector to a unit radius sphere centered at its origin (the pole), which is parallel to the reference sample direction for the mapping (one of the sample's basis vectors: X_1, X_2, or X_3). The convention for a crystal direction vector of a unit cell is simply

$$[lmn] = l\mathbf{e_1} + m\mathbf{e_2} + n\mathbf{e_3} \tag{1.1}$$

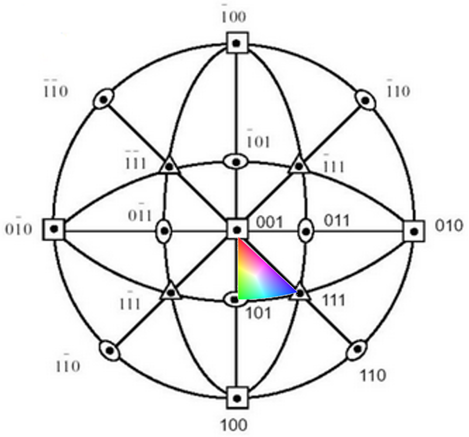
where $\mathbf{e_1}$, $\mathbf{e_2}$, and $\mathbf{e_3}$ are the unit cell basis vectors as shown in Figure 1.3. This type of color orientation mapping is known as an inverse pole figure (IPF), since we color based on the unit cell crystal direction parallel to the sample reference direction while, conversely, the EBSD output quaternion is a transformation from the sample orientation to that of the unit cell.

In Figure 1.4, we show the poles for the crystal directions normal to the six faces of a cube. So, for instance, if we are plotting the inverse pole figure for the macroscopic sample orientation X_3 from Figure 1.3, an array element representing an orientation with the [001] crystal direction pole as shown in Figure 1.4, would have it's crystal direction vector parallel to X_1. As such, the element would be assigned the color mapped to the [001] pole in an IPF-X_1 mapping.

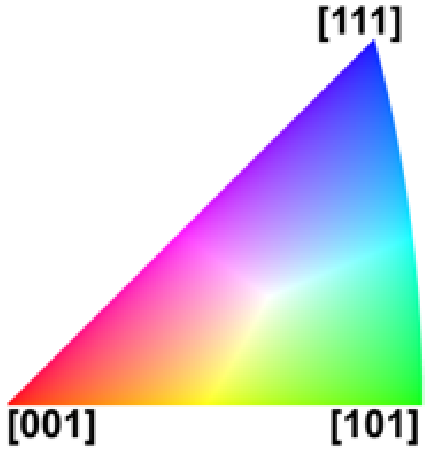
1.3.3 Fundamental Zone

To avoid different colors representing symmetrically equivalent orientations, color is mapped to an area formed by a continuous set of poles that are limited by axes of symmetry for the crystal symmetry group in question. We can observe this in Figure 1.5a, where the poles of the crystal direction vectors which act as a symmetry axis have been stereographically projected to a circle. Because the area delineated by each of these sectors is only made up of unique orientations (it's symmetrically equivalent orientations are located in other zones), they are defined as a fundamental zone.

The symbols at the intersection of lines in the stereographic projection represent the number of symmetry operations at that boundary: two for an oval, three for a triangle, and four for a square. For instance, a cubic unit cell can be rotated about it's [001] crystal direction vector, to obtain four symmetrically equivalent orientations, by rotating 0°, $90°, 180°$, or 270°. Hence, it is marked with a square, as are the other stereographically projected poles of the remaining cube faces. In total, for cubic crystal systems, any orientation can be described by twenty four mathematically distinct transformations, that map to equivalent orientations for a shape with unmarked faces. For this symmetry group, the area bounded by the crystal direction poles [001], [101], and [111] is referred to as the standard stereographic triangle. Color is continuously mapped to the this area,



(a) Stereographic projection of symmetry axes for cubic crystal system. The stereographic triangle for color orientation mapping is shown within the area bounded by crystal vector poles 001, 101, and 111. This figure is reproduced, with permission from [4].



(b) Color assigned to the area bounded by stereographic triangle, for cubic symmetry.

as shown in Figure 1.5b. Finally, each element in the quaternion array can be mapped to a color, according to the pole in the stereographic triangle which represents the inverse transformation for the chosen sample direction. This produces an RGB EBSD image, which is also referred to as an inverse pole figure.

1.3.4 Color Orientation Map

In summary, we are able to visualize EBSD data as color images with the following convention. First, a reference sample direction is chosen, based on one of the axes of the sample (X_1, X_2, or X_3, as labelled in Figure 1.3). Then the raw quaternions, which represent the orientation of the unit cells at an array element with respect to the sample orientation, are reduced to the minimum angle equivalent symmetric transformation. It is assumed that the orientations lie in a single fundamental zone, that of the standard stereographic triangle in 1.5a. Finally, to form an EBSD Image, the pixels representing each quaternions are assigned a color according to the inverse pole figure coloring mapping shown in Figure 1.5b.

The (inverse) pole figure based mapping is not a perfect proxy for the orientation of the unit cells. This is because a single inverse pole figure provides no information about the rotation of unit cells about the crystal pole defining the color of the pixel. Hence a color mapped EBSD Image is not invertible to the orientation quaternion array it was derived from.

1.4 Quaternions

EBSD microscopes return an x-y-z spatial array of orientations represented as quaternions - a hyper-complex number used to store three-dimensional rotations. We describe the definition, properties and intuition behind them in this section, to build up to the improved interpolation and deep learning loss functions in subsequent chapters.

A quaternion $q \in \mathbb{H}$ is a four dimensional number, with the imaginary components $\mathbf{i}$, $\mathbf{j}$, and $\mathbf{k}$.

$$q = a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k} \tag{1.2}$$

The multiplication rules for the components $\mathbf{i}, \mathbf{j}, \mathbf{k}$ are:

$$\mathbf{i^2} = \mathbf{j^2} = \mathbf{k^2} = \mathbf{ijk} = -1 \tag{1.3}$$

$$\mathbf{ij} = \mathbf{k},\ \mathbf{ji} = -\mathbf{k} \tag{1.4}$$

$$\mathbf{jk} = \mathbf{i},\ \mathbf{kj} = -\mathbf{i} \tag{1.5}$$

$$\mathbf{ki} = \mathbf{j},\ \mathbf{ik} = -\mathbf{j} \tag{1.6}$$

Quaternions are the natural extension of complex numbers, and are typically used in engineering applications such as EBSD to represent orientations and rotations in 3D, similar to how complex numbers are used for 2D. Because the axis of rotation is not explicitly defined for three dimensional rotations as it is for two dimensions (where it's by default the out of plane z-axis), a quaternion object encodes not just an angle, but also the axis of rotation.

The full transformation encoded by a quaternion, is apparent when expressed in polar form. To obtain the polar form, we first define

$$\mathbf{v} = b\mathbf{i} + c\mathbf{j} + d\mathbf{k} \tag{1.7}$$

$$\|\mathbf{v}\| = \sqrt{b^2 + c^2 + d^2} \tag{1.8}$$

$$\hat{\mathbf{v}} = \frac{1}{\|\mathbf{v}\|}\mathbf{v} \tag{1.9}$$

The component multiplication rules were designed such that $\hat{\mathbf{v}}^2 = -1$. This means that Euler's formula can be applied. If q is a unit quaternion, we have

$$q = a + \|\mathbf{v}\|\hat{\mathbf{v}} \tag{1.10}$$

$$q = e^{\hat{\mathbf{v}}\frac{\theta}{2}} = \cos(\frac{\theta}{2}) + \hat{\mathbf{v}}\sin(\frac{\theta}{2}) \tag{1.11}$$

More generally, for any (non-unit) quaternion:

$$q = \|q\|e^{\hat{\mathbf{v}}\frac{\theta}{2}} = \|q\|(\cos(\frac{\theta}{2}) + \hat{\mathbf{v}}\sin(\frac{\theta}{2})) \tag{1.12}$$

The transformation encoded by this quaternion q is thus a rotation by $\frac{\theta}{2}$ radians about the axis $\hat{\mathbf{v}}$, as well as a scaling factor $\|q\|$. The angle in the exponential form of quaternions is $\frac{\theta}{2}$, or in other words half of the angle of the represented orientation transformation, because of quaternions' double mapping of orientation, which is further discussed in Chapter 2.

It is also possible to string two quaternion transformations together. This is handled by the Hamilton product, which defines how quaternions are multiplied and is non-commutative:

$$q_1 = a_1 + \mathbf{v_1} \tag{1.13}$$

$$q_2 = a_2 + \mathbf{v_2} \tag{1.14}$$

$$q_1 q_2 = a_1 a_2 - \mathbf{v_1} \cdot \mathbf{v_2} + a_1 \mathbf{v_2} + a_2 \mathbf{v_1} + v_1 \times v_2 \tag{1.15}$$

where · and × are the dot and cross product respectively (in fact, the definition of these products stems from the Hamilton product).

As with complex numbers, the conjugate of a quaternion cancels the imaginary components, and leaves us with the square magnitude.

$$q = a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k} = \|q\| e^{\hat{\mathbf{v}}\frac{\theta}{2}} \tag{1.16}$$

$$q* = a - b\mathbf{i} - c\mathbf{j} - d\mathbf{k} = \|q\| e^{\hat{\mathbf{v}}\frac{-\theta}{2}} \tag{1.17}$$

$$qq* = \|q\|^2, \tag{1.18}$$

For unit quaternions, $\|q\|^2 = 1$, so the conjugate $q*$ is equivalent to the multiplicative inverse.

$$qq^* = \|q\|^2 = 1 \tag{1.19}$$

$$q^{-1} = q^* \tag{1.20}$$

The last property we introduce is that of a quaternion to a scalar power, which is used for orientation interpolation; this operation affects both the scaling of the transformation, if it's not a unit quaternion, and also parametrizes the angle we rotate about the axis: from $t = 0$ for no rotation, to $t = 1$ for the full rotation angle.

$$q = \|q\| e^{\hat{\mathbf{v}}\frac{\theta}{2}} \tag{1.21}$$

$$q^t = \|q\|^t e^{\hat{\mathbf{v}}\frac{\theta t}{2}} \tag{1.22}$$

Finally, in order to find the orientation of one quaternion with respect to another, we must find the transformation T between them.

$$T = q_1^{-1} q_2 \tag{1.23}$$

In this case, T is the transformation from the orientation represented by q_1 to the orientation represented by q_2.

Chapter 2

Orientation and Symmetry

In this chapter, we lay the groundwork for the symmetry-reduced interpolation technique and the neural network methods discussed in the subsequent chapter. We begin with an overview of orientation space, explaining why interpolation is feasible within this space. We then get into the treatment of symmetry minimization for both two-dimensional (2D) and three-dimensional (3D) orientations. Given the intuitive nature of visualizing 2D orientations, we start with a detailed example of 2D symmetry. Following this, we explore the nuances of 3D symmetry, highlighting the key differences and additional complexities involved in its treatment.

2.1 Two Dimensional Orientations on $\mathbb{C}$ Unit Circle

A complex number, $z \in \mathbb{C}$ on the Unit Circle represents two dimensional orientation. z encodes a positive or negative angle about the out-of-plane z-axis, which can be superimposed with other complex numbers through multiplication. For instance, as shown in the polar form below and in Figure 2.1:

$$z = e^{j\theta} \tag{2.1}$$

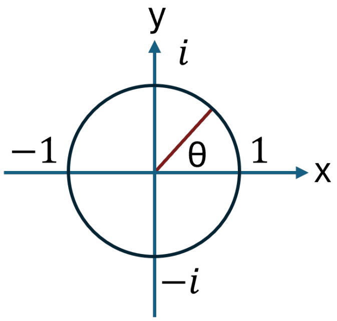


Figure 2.1: Two-dimensional orientation encoded by the angle θ of a complex number on the Unit Circle. As per convention, a positive angle corresponds to counter-clockwise rotation, with respect to the positive x-axis.

If complex numbers are used to represent the orientations, two squares are identically oriented if the rotational distance between them is a multiple of 90°.

By convention, we make the orientation angles lie in the range $\theta \in [0, 2\pi]$, and the symmetry operator angles from $[-2\pi, 0]$. The permutation symmetry operator for a square, O_{symm}, without reflections is as follows:

$$O_{symm} = \{1, e^{-90^\circ j}, e^{-180^\circ j}, e^{-270^\circ j}\} \tag{2.2}$$

For instance if we consider a scenario where our microscope is working with two-dimensional orientation data, and one of the recorded data points is $z = e^{300^\circ k}$, after multiplying with our square symmetry operator, O_{symm}, we obtain

$$O_{symm} z = O_{symm} e^{300^\circ j} = \{e^{300^\circ j}, e^{210^\circ j}, e^{120^\circ j}, e^{30^\circ j}\} \tag{2.3}$$

All the complex numbers in the set above represent symmetrically equivalent squares, where orientation angle is counter-clockwise with respect to the positive x-axis, and the rotation is about the out of plane z-axis. In this case, of the symmetrically equivalent options within the set, we choose the one with the smallest magnitude angle, which in this case is $e^{30^\circ j}$. We can state this as:

$$Min_{|\theta|}\{O_{symm}e^{300^\circ j}\} = e^{30^\circ j} \tag{2.4}$$

2.2 Orientation Space

An Orientation Space is a continuous set of all possible orientations, typically in reference to a 2D or 3D object. Orientations in this space are represented by a complex number z or quaternion q, each defined relative to a physical orientation reference. The orientation of the reference for that mathematical object is implicitly assigned to $z = 1$ or $q = 1$. Complex numbers and hypercomplex numbers do not encode complete information about the relationship between orientations. Instead, they describe a single transformation with respect to a reference, which can be combined with other transformations through the established rules for complex multiplication for $z \in \mathbb{C}$, or the Hamilton product for $q \in \mathbb{H}$.

To obtain the transformation between two orientations represented by unit quaternions, we can use the Hamilton product to multiply the inverse of a quaternion q_1 by another quaternion q_2, where q_1 and q_2 have the same physical orientation reference for $q = 1$. For the resulting quaterion, $T = q_1^{-1}q_2$, the reference orientation $q = 1$ is now the orientation of the object represented by q_1.

For EBSD data collection, the reference orientation, $q = 1$ is assigned to the orientation of the sample. The quaternions in the array elements store the transformation from the orientation of the sample to the orientation of the unit cell in the spatial area covered by that array element. Since the EBSD unit cells are symmetrical in nature (no markings are assigned to each face), there is more than one way to transform between the orientations of these two (or any) unit cells. In this field the equivalent transformation with the smallest angle is called the misorientation, which as will be discussed further in

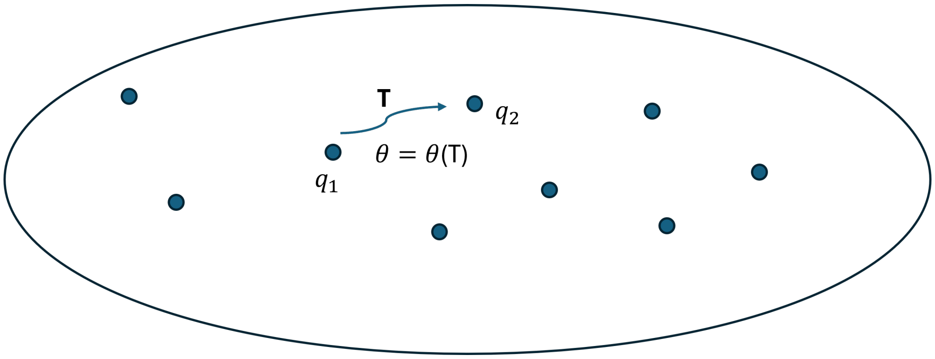


Figure 2.2: An Orientation Space, illustrating a quaternion T which transforms between two orientations described by quaternions with respect to common reference q_1 and q_2.

this chapter, is obtained by looping through all the equivalent symmetry permutations and selecting the minimum angle equivalent transformation.

When represented by quaternions, the orientation space for 3D objects is simply connected, which means interpolation between orientations is always possible. The simply connected property is linked to the double cover of quaternions (q and $-q$ are equivalent transformations). This results in the angles stored by a quaternion transformation being double that of the angles of the equivalent transformation in $\mathbb{R}^3$ that they represent.

2.3 Angle Minimization over Shape Group Symmetry

For the two dimensional case, the transformation between two unit complex numbers z_1 and z_2 in an orientation space is $T = z_1^{-1} z_2$.

Now we would like to frame the minimum angle transformation between z_1 and z_2. Because all transformations for two-dimensional orientations are positive or negative rotations about the out-of-plane z-axis, it is equivalent to find the minimum angle symmetry equivalent of both orientations separately, or find the minimum angle equivalent of the transformation itself.

$$\theta(T_{min}) = Min_{|\theta|}\{O_{symm}z_1^{-1}\}Min_{|\theta|}\{O_{symm}z_2\} \equiv Min_{\theta}\{O_{symm}z_1^{-1}z_2\} \tag{2.5}$$

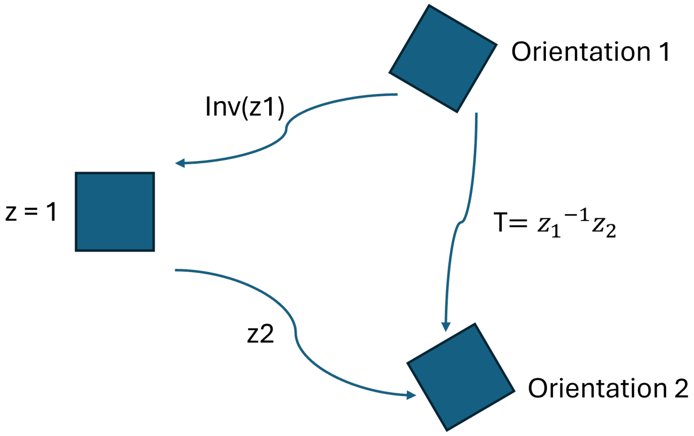


Figure 2.3: Transformations between orientations in 2D, represented by complex numbers on the Unit Circle. z_1^{-1} transforms from orientation 1, to the reference $z = 1$, and z_2 transforms from this reference to orientation 2. $T = z_1^{-1}z_2$ transforms from orientation 1 to orientation 2.

2.4 Angle Minimization over Crystal Group Symmetry

For the three dimensional case, the transformation between two unit quaternions q_1 and q_2 in an orientation space is $T = q_1^{-1} q_2$.

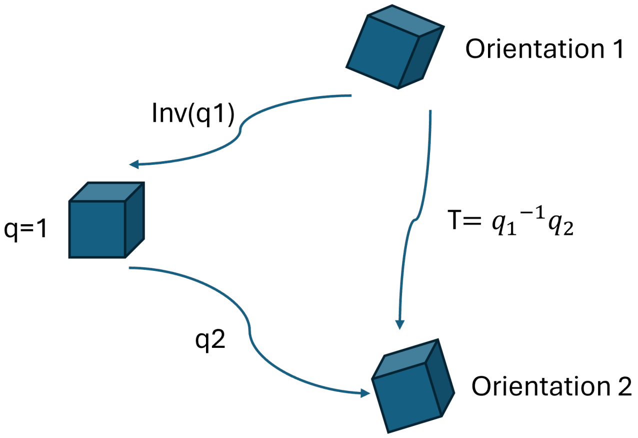


Figure 2.4: Symmetry Minimization of Two-Dimensional Orientation

In this case, separately minimizing the quaternions which map transformations from the reference is not equivalent to minimizing the transformation between them, i.e.,

$$\theta(T_{min}) = Min_{|\theta|}\{O_{symm} q_1^{-1} q_2\} \not\equiv Min_{|\theta|}\{O_{symm} q_1^{-1}\} Min_{|\theta|}\{O_{symm} q_2\} \tag{2.6}$$

This is because for three-dimensional orientations, the full transformation between two orientations cannot be encoded in a single quaternion. Instead, it involves the conjunction of two quaternions and the definition of the Hamilton product between one and the inverse of the other. This means that symmetry reduction must be considered within this specific context.

To further understand this, consider the concept of symmetry switching. The minimum angle transformation from a sample orientation to Orientation 1 might correspond to a different symmetrically equivalent orientation for Orientation 1 compared to the one required for the minimum angle transformation between Orientation 1 and Orientation 2. In this scenario, the symmetry of the quaternion transformation changes, or ”switches.”

To illustrate this, imagine each face of a cube is marked with a different color, making them distinguishable. If the faces are distinguishable, symmetry switching is not possible because you cannot rotate to a different visually identical orientation. In other words, the transformation is unique, and the concept of symmetry switching does not apply.

In summary, for 3D orientations, the minimum angle transformation relation does not hold consistently due to the necessity of involving multiple quaternions and the potential for symmetry switching between different equivalent orientations.

So, the minimization over crystal group symmetry for q_1^{-1} and q_2 separately, is not equivalent to the minimization over crystal group symmetry for the transformation between Orientation 1 and Orientation 2, as stated earlier in Equation 2.6:

In summary, these two minimization problems are different for 3D and will not necessarily result in the same $\theta_{min} = \theta(T_{min})$. Understanding this 2D vs 3D distinction clearly is essential for symmetry-reduced interpolation and to optimize the loss function for the neural network based approaches.

Symmetry reduction	**Cumulative Symmetry Equivalent Orientations**
Shape Symmetry	24 (fcc), 12 (hcp)
Switching Symmetry	48 (fcc), 24 (hcp)
+Q, -Q Equivalence	96 (fcc), 48 (hcp)

Table 2.1: O_{symm} number of permutations for face-centered-cubic (fcc) and hexagonal closed pack (hcp) symmetries

In Table 2.1, we provide a cumulative tally of the number of quaternions representing symmetrically equivalent orientations for Face-Centered Cubic (FCC), and Hexogonal Closed Pack (HCP) crystal groups. Shape symmetry is dependant on the crystal group, and hence the number of equivalents varies, as indicated in Table 2.1. Switching symmetry accounts for both possible directions for a transformation between orientations, resulting in twice as many symmetry equivalents, as does the $+q, -q$ equivalence.

Chapter 3

EBSD Image Superresolution

Superresolution is the process of algorithmically increasing the resolution of a lower resolution (LR) image by a specified scaling factor and, in doing so, recovering higher frequency information present in an equivalent high resolution (HR) image. The super-resolved image (SR) will have the dimensions of the desired HR image, and should be as close as possible to it's respective HR in terms of pixel by pixel comparison. The classical method for superresolution is to fill in the missing pixels introduced by the scaling factor with interpolation between pixels in the LR image. Some of the most common methods for this approach are bilinear interpolation, cubic interpolation, and nearest neighbor. More recently, deep neural networks have been used to learn complex non-linear mapping between LR and HR pixels, and recapture spatial frequencies at higher than the Nyquist threshold.

For Superresolution of EBSD data specifically, interpolation and neural network implementations are conducted using the raw experimental quaternion data, before converting the superresolved quaternion array into an SR Image. This is preferred over using the color data resulting from Color Orientation Mapping, since the orientation data inherently captures more information, and because the color maps are not directly invertible to orientation for the purpose of calculating misorientation accuracy.

Since we are interpolating between orientations, spherical linear interpolation is a more accurate method than the bilinear, cubic, and nearest neighbor interpolations. However, when applying Slerp to superresolve EBSD data, the SR Images tend to exhbit very similar aliasing in its color orientation map to the state-of-the-art EBSD superresolution model. To remedy this issue, we propose the use of a minimum angle transformation, which replaces the standard transformation between two quaternions in the Slerp algorithm, and thus avoids the issue of symmetry switching. This is the first known attempt to introduce Slerp, accounting for symmetry switching, for EBSD Superresolution.

Next, we develop a Superresolution framework, based on [1], with an updated loss function, which computes the rotational distance for the minimum angle transformation between HR and SR pixels during the training phase.

3.1 Spherical Linear Interpolation (Slerp) Interpolation for EBSD Data

Slerp [5] is the standard method for interpolating between two orientations expressed as quaternions q_1 and q_2, as it interpolates along the minimum great circle path connecting the orientations in that space.

3.1.1 Slerp Algorithm

The Slerp algorithm exponentiates a quaternion to a scalar value (the interpolation parameter t), which is computed as follows:

$$q^t = \|q\|^t e^{\hat{\mathbf{v}}\frac{\theta}{2}t} \tag{3.1}$$

Suppose we have two three-dimensional orientations encoded by $q_1 \in \mathbb{H}$ and $q_2 \in \mathbb{H}$, with respect to the same physical reference. Slerp allows us to interpolate through the path by toggling the interpolation parameter t from 0 to 1. In our area of application, q_1 represents the orientation of a unit cell 1 with respect to the orientation of the sample, while q_2 represents the orientation of a unit cell 2, with respect to the orientation of the sample.

$$T = q_1^{-1} q_2 \tag{3.2}$$

$$Slerp(q_1, q_2, t) = q_1 (q_1^{-1} q_2)^t \tag{3.3}$$

$$Slerp(q_1, q_2, t) = q_1 T^t \tag{3.4}$$

When $t = 0.5$, Slerp reaches the point halfway along the minimum great circle connecting the quaternions. We can also trivially note that if $t = 0$, the expression reduces to q_1, and if $t = 1$, q_1 and q_1^{-1} cancel, leaving us with q_2.

The direction of transformations, and relationship between them is illustrated in Figure 3.1

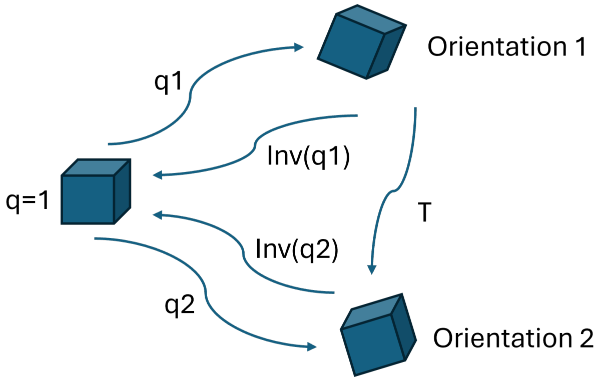


Figure 3.1: All transformations between the orientations $q = 1$ (reference), Orientation 1, and Orientation 2. Slerp interpolates between Orientation 1 and Orientation 2 by stringing together transformation q_1 with a parametrized transformation T, exponentiated to the interpolation factor t.

3.1.2 Minimum Angle Transformation Spherical Linear Interpolation (Slerp-MAT)

The Slerp algorithm (3.4) does not account for objects with group symmetry, so we propose a novel interpolation algorithm that adds symmetry minimization to the transformation between orientations represented by quaternions q_1 and q_2, $T = q_1^{-1}q_2$. In order to reduce computations, since in our EBSD application both objects have the same symmetry, we apply the symmetry operator to the transformation quaternion, and minimize for θ. We obtain a quaternion T_{min} (3.6), whose versor $\mathbf{\hat{v}_{min}}$ is the axis of minimum rotation between the two orientations, for two shapes with the same group symmetry.

$$T = q_1^{-1}q_2 \tag{3.5}$$

$$T_{min} = Min_\theta\{O_{symm}T\} = a + \mathbf{v_{min}} = a + \|\mathbf{v_{min}}\|\mathbf{\hat{v}_{min}} = e^{\mathbf{\hat{v}_{min}}\frac{\theta}{2}} \tag{3.6}$$

$$T_{min}^t = e^{\mathbf{\hat{v}_{min}}\frac{\theta}{2}t} \tag{3.7}$$

Plugging the minimum angle transformation into the Slerp formula, we obtain:

$$Slerp - MAT(q_1, q_2, t) = q_1 T_{min}^t \tag{3.8}$$

Slerp-MAT is a distinct algorithm with respect to traditional SLERP, as there is no way to pre-process or post-process the inputs our outputs to obtain the same result (hence, why Slerp had not been applicable in this field). The use cases of Slerp-MAT could go beyond the scope of the superresolution problem formulation, as it unlocks the potential to create a probability distribution function, in order to noise synthetic microstructures. Further than the scope of EBSD, it can also be used to interpolate between orientations of objects with symmetry, which can have use cases in the more general fields of graphics, computer vision, and virtual reality.

3.1.3 Parallelized Interpolation Method

In order to efficiently superresolve a large amount of EBSD data utilizing Slerp-MAT, we propose a parallel computational framework. If the collected two dimensional plane is of resolution $M \times N$, and we wish to superresolve by a scale S, we will end up with a superresolved image of dimensions $MS \times NS$. In the example below we consider a trivial example, where the low resolution image which we wish to superresolve is of size 2×2 pixels (storing four quaternions in total), and a scale factor $S = 4$, resulting in an output of dimensions 8×8.

We start by placing the low resolution values S positions apart, so we may then first interpolate column-wise with parameter t, up to column $(N - 1)S + 1$.

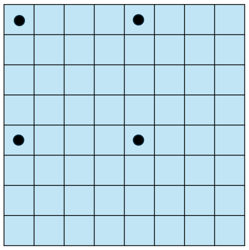


Figure 3.2: Dark circles denote elements from original array placed with equal spacing in a new array with the desired output dimensions, as per the defined scaling factor.

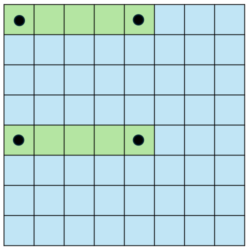


Figure 3.3: Column-wise interpolation using the known array elements. Green highlighted array elements denote areas that have been superresolved with EBSD-Slerp–MAT.

We may then compute in parallel all the interpolations row-wise, down to the row $(M-1)S+1$.

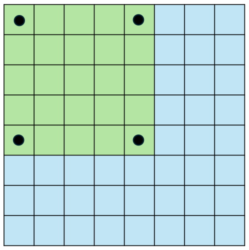


Figure 3.4: Row-wise interpolation using the known array elements. Green highlighted array elements denote areas that have been superresolved with EBSD-Slerp-MAT.

For the remaining pixels, we simply select a nearest neighbor, as we don't have two distinct points to interpolate between. This allows us to maintain the desired dimensions for the superresolved output, $MS \times NS$.

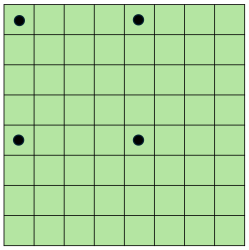
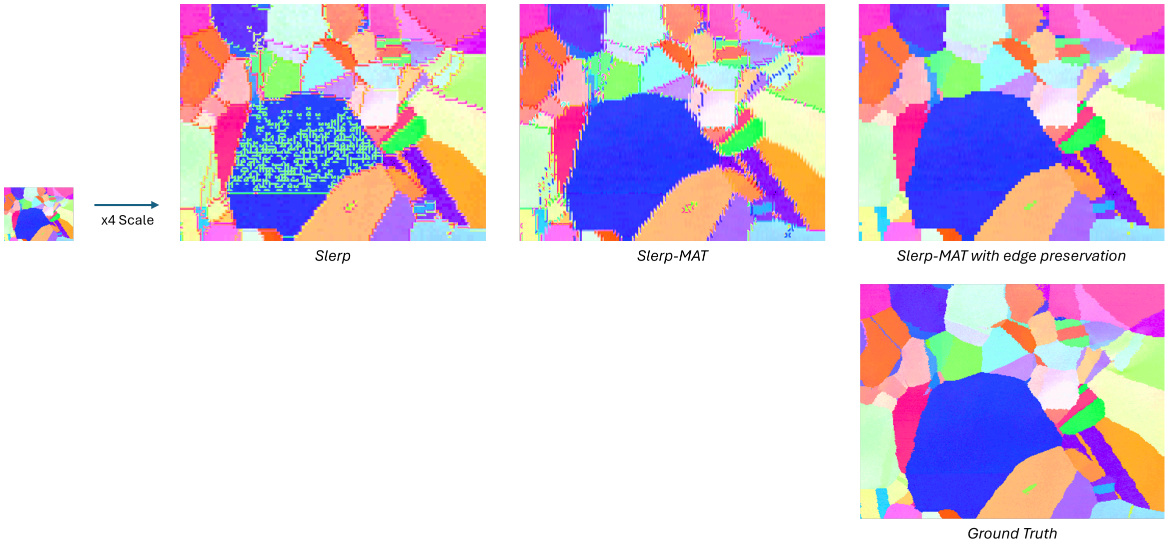


Figure 3.5: We utilize nearest neighbor interpolation for the last $M-1$ rows and $N-1$ columns. Green highlighted array elements denote that the parallelized Slerp-based superresolution algorithm is complete.

This parallelized approach to Slerp Superresolution allows us to interpolate a large amount of EBSD data in an order of minutes rather than hours. These can be further processed by a pre-trained Deep Learning Network, without the need for the resolution increase in the deep learning approach to rely on Pixel Shuffle [6], which struggles to produce high frequency spatial information on globally simple but locally textured data, such as EBSD. It is by parallelizing the interpolation process that we are able to reduce the computation to minutes as opposed to hours, by running the algorithm on GPU hardware - in this case, on an NVIDIA A100 GPU.

3.1.4 Superresolution Results with Slerp versus Slerp-MAT

Slerp interpolation of an EBSD quaternion array results in symmetry switching aliasing, as shown in Figure 3.6. Replacing Slerp with Slerp-MAT resolves the symmetry switching. We can see incorrect color orientation mapping is observed at the grain boundaries, and this is because they have large angular differences and are hence not fit for a smooth interpolation function. To remedy this, we interpolate the array with Slerp-MAT, but with an edge preservation threshold. To introduce edge preservation, we default to nearest neighbor interpolation, if the angle between the interpolating quaternions is $\theta > 3°$. This occurs because the angle minimization across symmetry operators gets computed separately for q_1 and q_2 instead of for the transformation between them. In the case of Nickel, which has cubic symmetry, the results are more apparent, since its symmetry group is twice as large.



Slerp

Slerp-MAT

Slerp-MAT with edge preservation

Ground Truth

Figure 3.6: Results of Superresolving to original dimensions a Low Resolution (LR) 4x Decimated EBSD quaternion array, illustrated with color orientation mapping. Left–to-right: Slerp interpolation function, Slerp-MAT interpolation function, Slerp-MAT interpolation function with nearest neighbor (NN) if $\theta > 3°$. Slerp-MAT is impervious to symmetry switching, and Slerp-MAT with edge preservation replaces Slerp-MAT with a discontinous interpolating function (NN) for grain boundaries.

3.2 Deep Learning Superresolution for EBSD Data

Slerp-MAT with edge preservation provides more accurate results on Nickel data than the previous state-of-the-art superresolution networks, as it avoids incorrect color mapping. However, a deterministic interpolation method is not able to reconstruct grain shapes that were sufficiently deconstructed by decimation, which happens at a 4x Decimation scale for materials with smaller grain sizes. This is the case for the Titanium dataset, that the state-of-the-art network architecture [1] was trained and evaluated on. We can improve the representational power of superresolution, in comparison to a deterministic interpolation like Slerp-MAT, by creating a more complex, non-linear map, between LR and HR images using a deep learning feature extraction approach.

In Figure 3.7 we adapt the loss function of the state-of-the-art Physics-based Loss presented in [1] to avoid symmetry switching, by calculating the rotational distance with the Minimum Angle Transformation employed in Slerp-MAT.

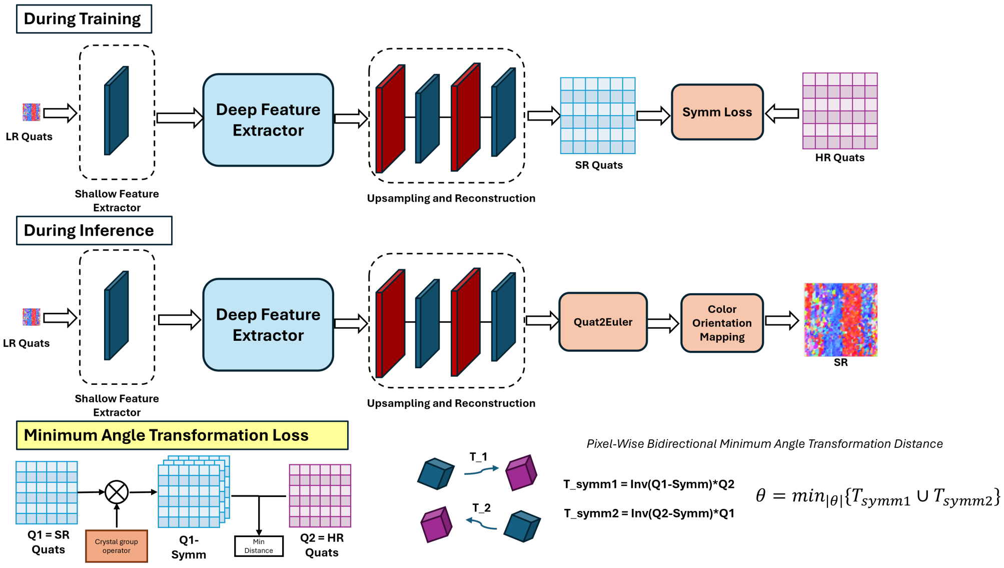

During Training
LR Quats
Shallow Feature Extractor
Deep Feature Extractor
Upsampling and Reconstruction
SR Quats
Symm Loss
HR Quats
During Inference
LR Quats
Shallow Feature Extractor
Deep Feature Extractor
Upsampling and Reconstruction
Quat2Euler
Color Orientation Mapping
SR
Minimum Angle Transformation Loss
Q1 = SR Quats
Crystal group operator
Q1-Symm
Min Distance
Q2 = HR Quats
Pixel-Wise Bidirectional Minimum Angle Transformation Distance
T_1
T_2
T_symm1 = Inv(Q1-Symm)*Q2
T_symm2 = Inv(Q2-Symm)*Q1
$\theta = min_{|\theta|}\{T_{symm1} \cup T_{symm2}\}$

Figure 3.7: Network Architecture and Loss Function

3.2.1 Network Architecture

The network architecture proposed in [1] comprises of two feature extractors - a shallow feature extractor, and a deep feature extractor. The shallow feature extractor employs a single convolution layer to extract shallow features, which are passed on the deep feature extractor.

$$F_0 = H_{SF}(I_{LR}) \tag{3.9}$$

where F_0 are the shallow features generated by the trained mapping H_{SF} applied to a low resolution EBSD quaternion array, I_{LR}. The shallow features are then hierarchically passed on to the deep feature extractor

$$F_{DF} = H_{DF}(F_0) \tag{3.10}$$

Finally, the deep features extracted are passed to an Upscale and Reconstruction Module. This module upscales the features using a Pixel Shuffle operation [6], and then maps the feature onto an SR EBSD Image, with the use of convolution layers.

$$F_{\uparrow} = H_{\uparrow}(F_{DF}) \tag{3.11}$$

$$I_{SR} = H_R(F_{\uparrow}) \tag{3.12}$$

The subsequent deep feature extractor comprises of a state-of-the-art single image superresolution architecture: Enhanced Deep Residual Network (EDSR), or a Holistic Attention Network (HAN).

$$F_{DF} = H_{DF}(F_0) \tag{3.13}$$

In this section, we describe how the Minimum Angle Transformation was used to train industry standard superresolution networks used for images, based on those used in [1]

3.2.2 Loss Function

For the state-of-the-art EBSD Superresolution approach proposed in [1], the loss function computes a rotational distance between the superresolved and respective high-resolution quaternions.

$$\begin{aligned}\theta &= 2\arccos(\mathbb{R}\{q_1^{-1}q_2\}) &(3.14)\\ &= 2\arccos(1-\frac{1}{2}\|q_1-q_2\|_2^2) &(3.15)\\ &= 4\arcsin(\frac{1}{2}\|q_1-q_2\|_2) &(3.16)\\ &= 4\arcsin(\frac{d_{euclid}}{2}) &(3.17)\end{aligned}$$

A linear approximation is calculated at $d_{euclid} = 1.9$, and used for points where $d_{euclid} > 1.9$, in order to avoid gradients tending to ∞.

However, q_1 and q_2 are the symmetry reduced quaterion transformations between the sample orientation, and that of the HR and SR orientations respectively. In other words, the symmetry choice has been minimized in the frame of reference of the sample orientation, instead of that of the respective HR and SR element orientations, leaving this approach vulnerable to symmetry switching, as explained in chapter 2.

$$q_1 = q_{SR-fz} = Min_\theta\{q_{SR}\} \tag{3.18}$$

$$q_2 = q_{HR-fz} = Min_\theta\{q_{HR}\} \tag{3.19}$$

We adapt the original loss function to account for the minimum angle transformation between respective SR and HR orientations, by making q_1 the bidirectional MAT between them, and making q_2 the zero rotation unit quaternion.

$$q_1 = T_{min} = Min_{|\theta|}\{O_{symm}(q_{SR}^{-1}q_{HR}) \cup O_{symm}(q_{HR}^{-1}q_{SR})\} \tag{3.20}$$

$$q_2 = 1 \tag{3.21}$$

The minimization accounts for both transformation directions, by minimizing over the union of symmetry equivalent transformations from SR to HR orientations, and HR to SR orientations.

3.2.3 Results for Titanium Data

A Misorientation probability density function, is generated through random sampling of 10,000 elements out of the 1,689,660 elements in the entire dataset. The area under the probability density functions in Figure 3.8 indicate the probability of a randomly sampled pair of array elements in the same respective position of the HR and SR arrays having a misorientation angle falling in that range. The output array based on minimum angle transformation loss has most of its area surrounding a notable peak at 0°, while the array based on rotational distance has its area much more spread out on the horizontal axis, from 0° to 120° (maximum possible rotation angle for a minimum angle transformation between two hexagonal closed pack unit cells [7]). It does have a peak close to 0°, but also another approximately equal area peak close to 60°. The mean misorientation for the MAT loss versus HR is 11.16°, a 321% improvement compared to 47.05° for the previous state-of-the-art. The results for the color orientation mapping of a single 2D plane of the 3D output array is shown in Figure 3.9.

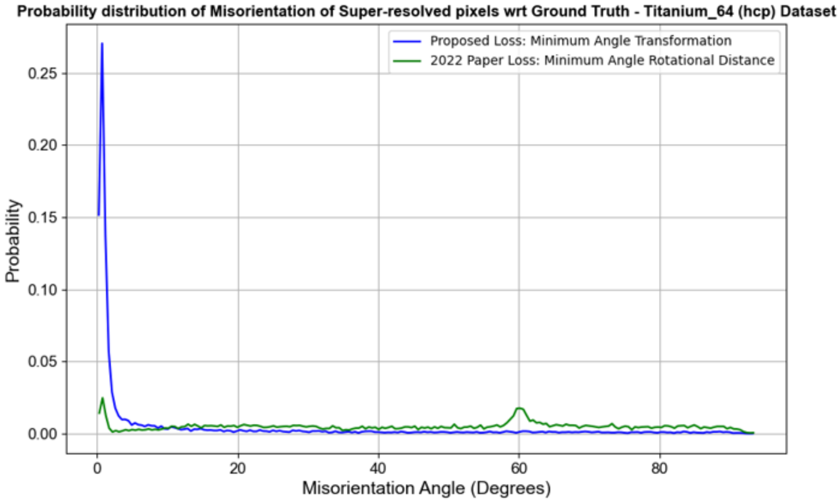


Figure 3.8: Probability density function of Titanium element-wise minimum angular distance between output quaternion array, superresolved from a 4x decimated Ground Truth quaternion array, with EDSR as the deep feature extractor.

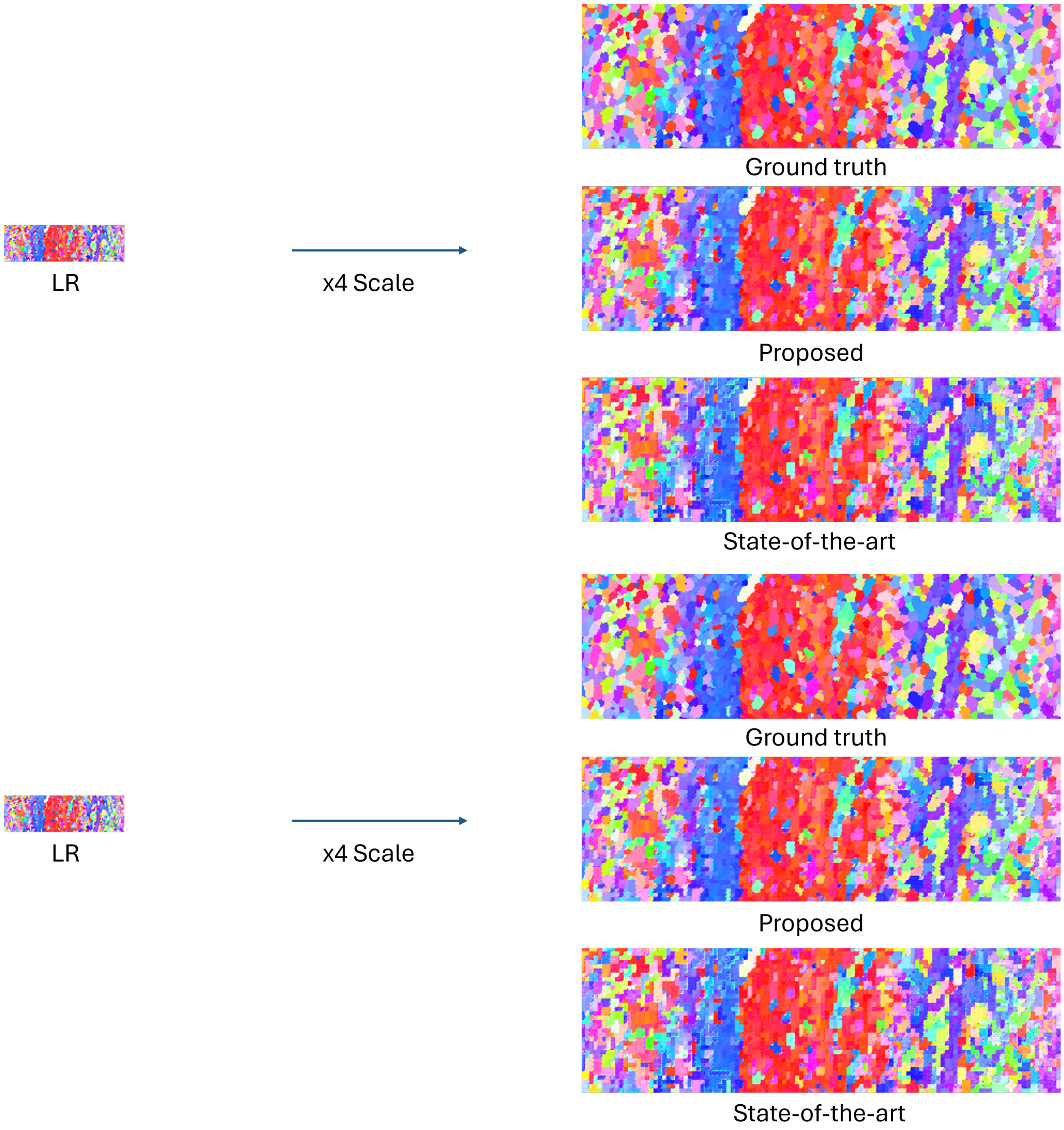


Figure 3.9: Color orientation mapping for two arbitrary 2D planar slices of the Titanium quaternion arrays. Bottom to top: Ground truth experimental array. Superresolved quaternion array of the 4x decimated array using minimum angle transformation based loss during neural network training, with EDSR as the deep feature extractor. State-of-the-art network rotational distance loss, with EDSR as the deep feature extractor.

3.2.4 Results for Nickel Data

The probability density functions for the Nickel dataset in Figure 3.10 are generated through random sampling of 10,000 elements out of the total 22,421,490 elements in the dataset. The probablity density function based on the minimum angle transformation loss exhibits a peak close to 0°, with most of the area distributed in that region of the horizontal axis. On the other hand, the state-of-the-art loss has a much higher peak at 30° than at 0°, and its area is more spread out on the horizontal axis. The mean misorientation for the MAT loss versus HR is 3.52°, a 882% improvement compared to 34.60° for the previous state-of-the-art. Two arbitrary results for the color orientation mapping of single 2D planes of the 3D output array are shown in Figure 3.2.4 and Figure Figure 3.11.

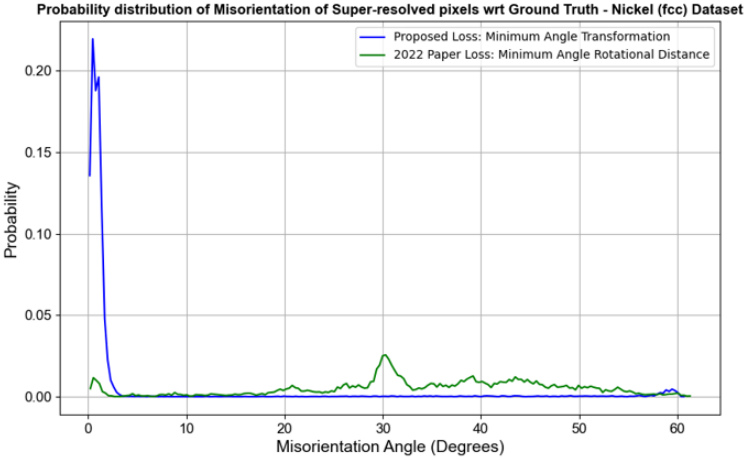


Figure 3.10: Probability density function of element-wise minimum angular distance between Nickel output quaternion array, superresolved from a 4x decimated Ground Truth quaternion array, with EDSR as the deep feature extractor.

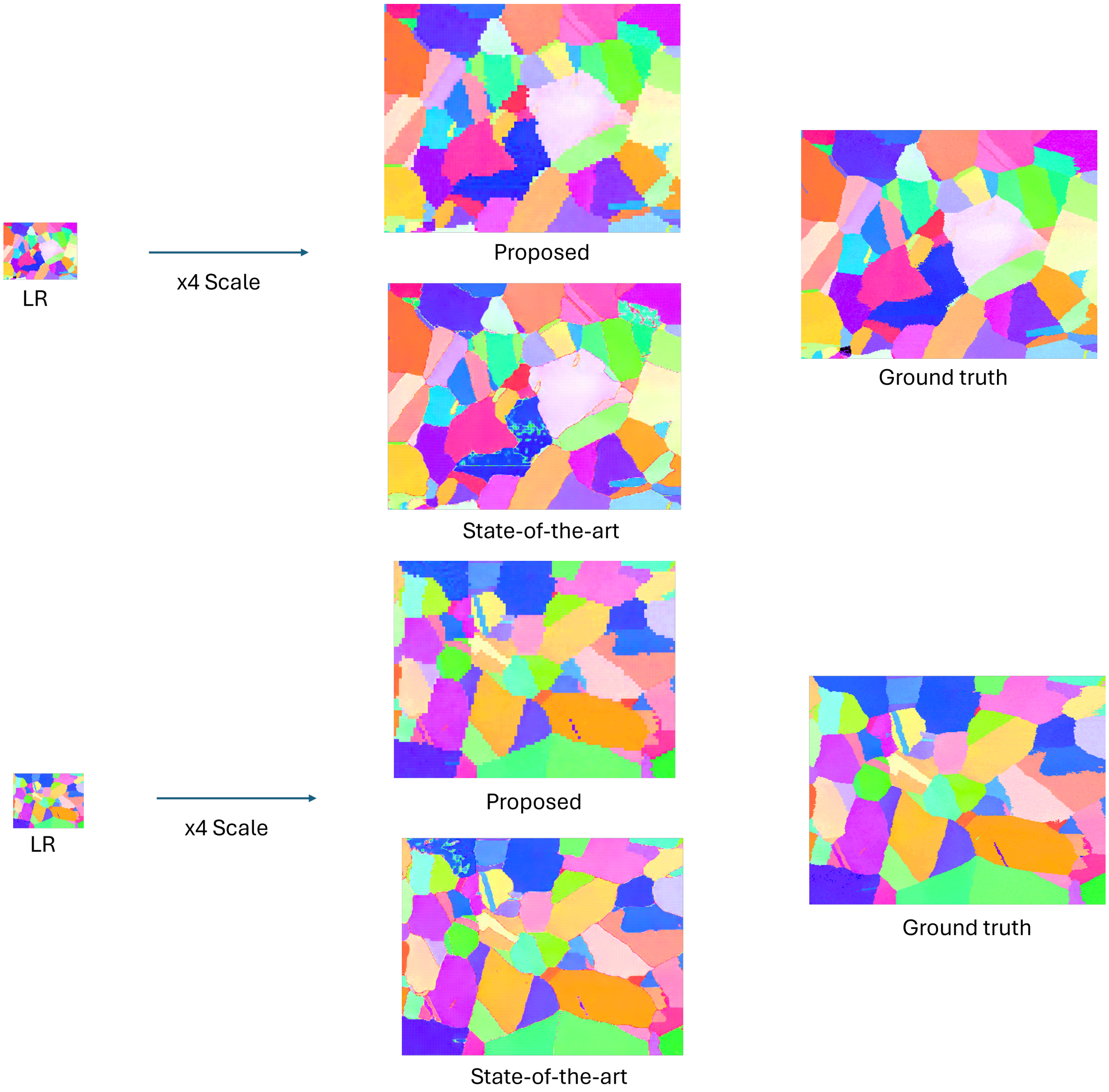


Figure 3.11: Color orientation mapping for two arbitrary 2D planar slices of the Nickel quaternion arrays. Bottom to top: Ground truth experimental array. Superresolved quaternion array of the 4x decimated array using minimum angle transformation based loss during neural network training, with EDSR as the deep feature extractor. State-of-the-art network rotational distance loss, with EDSR as the deep feature extractor.

Chapter 4

Discussion

In this thesis we have demonstrated how to correctly apply crystal group symmetry - with a minimum angle transformation - to interpolate through the shortest great circle path connecting two orientations represented by quaternions in an S^3 sphere orientation space. We also explain how this transformation can be used as a basis for a neural network loss function, which is tested on a deep residual superresolution network (EDSR), and an attention network (HAN). Both of these results have applications outside of just EBSD, to any computer vision or graphics application that involves an interpolation between the orientations of three-dimensional objects that have no individual markings on each face (hence the existence of symmetry equivalents).

In terms of short-term further research, the area for improvement we see in the ML output with minimum angle transformation is that the super-resolved color orientation maps seems to have course spatial features, which looks quite close to that of Slerp-MAT-based output. Because we're using an attention network for Superresolution, we expect model to synethsize higher frequency spatial information, which seems to be missing. In fact, there seems to be a trade-off between creating smoother boundaries, using for instance a simple l_1 loss, with producing accurate misorientation superresolved output.

Adding transformers to the network architecture might be useful for coupling accurate angular distance with higher frequency spatial information, which is what will actually improve metrics such as PSNR that are tracked for resolution increases for RGB images.

For the longer term and bigger picture, the key areas to explore are experimenting with generative adversarial and diffusion networks on the orientation data, to generate textured synthetic microstructures directly in a three-dimensional volume.

Bibliography

[1] D. K. Jangid, *Adaptable physics-based super-resolution for electron backscatter diffraction maps*, *npj Computational Materials* **8** (2022) 255.

[2] A. Winkelmann, *Ebsd orientation analysis based on experimental kikuchi reference patterns*, *Acta Materialia* **188** (2020) 376–385.

[3] "Dream3dnx." `https://www.dream3d.io/`. Accessed: 2024-07-30.

[4] B. D. Cullity and C. D. Graham, *Introduction to magnetic materials*. IEEE/Wiley, 2015.

[5] K. Shoemake, *Animating rotation with quaternion curves*, in *Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '85, (New York, NY, USA), p. 245–254, Association for Computing Machinery, 1985.

[6] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, *Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network*, 2016.

[7] R. Krakow, R. J. Bennett, D. N. Johnstone, Z. Vukmanovic, W. Solano-Alvarez, S. J. Lainé, J. F. Einsle, P. A. Midgley, C. M. Rae, and R. Hielscher, *On three-dimensional misorientation spaces*, *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* **473** (Oct, 2017) 20170274.

[8] B. Cullity, "Elements of x-ray diffraction, 3rd edition."

[9] M. Bedzyk, "Standard projections in a cubic system."

[10] T. Rollett, *Intro to x–ray pole figures*, Jan, 2016.

[11] K. Shoemake, *Animating rotation with quaternion curves*, *SIGGRAPH Comput. Graph.* **19** (jul, 1985) 245–254.

[12] S. Wright, *Orientation Texture*. Academic Press, 2005.

[13] M. Bedzyk, “Crystallography and diffraction.” `https://msecore.northwestern.edu/361/361text.xhtml`. Accessed: 2024-07-16.

[14] B. Niu, W. Wen, W. Ren, X. Zhang, L. Yang, S. Wang, K. Zhang, X. Cao, and H. Shen, *Single image super-resolution via a holistic attention network*, 2020.

[15] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, *Enhanced deep residual networks for single image super-resolution*, 2017.

[16] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, *Image super-resolution using very deep residual channel attention networks*, 2018.

[17] J. Guo, X. Zou, Y. Chen, Y. Liu, J. Hao, J. Liu, and Y. Yan, *Asconvsr: Fast and lightweight super-resolution network with assembled convolutions*, 2023.

[18] J. Ho, A. Jain, and P. Abbeel, *Denoising diffusion probabilistic models*, 2020.

[19] D. Bank, N. Koenigstein, and R. Giryes, *Autoencoders*, 2021.

[20] T. Parcollet, M. Ravanelli, M. Morchid, G. Linarès, C. Trabelsi, R. D. Mori, and Y. Bengio, *Quaternion recurrent neural networks*, 2019.