Optimization Improving Tomography Compared with HEDM Image Using Data Mining Methods

Rulin Chen

Department of Physics and Machine Learning Department, Carnegie Mellon University rulinc@andrew.cmu.edu

Committee: Barnabás Póczos, Florin B. Manolache

May 02, 2018

0.1 Introduction of Tomography and HEDM (High Energy Diffraction Microscopy

Computed tomography (CT) is widely known technique. It's actually making use of X-ray detecting the interior of objects. The basic idea of CT consists of two steps: Firstly, using some form of physical radiation such as X-rays to light on the sample, record the intensity of transmitted X-rays which is so called projection images and rotate the sample to create hundreds of such projections from different angles. Secondly, using the projection slices to reconstruct the interior of the object.

Besides CT image, there is another technique named HEDM [1, 2, 3], another interior imaging technique containing more information such as orientation of crystal in solids. It is simply a 3D X-ray rotational method to obtain the microscopy of crystal orientation. The Forward Modeling Method simulates the experimental geometry and diffraction from a known crystal structure and attempts to find the unit cell orientation that maximizes overlap of simulated diffraction with the experimental Observations [4]. Further analysis of changes in solids can be made given HEDM reconstruction. [5]. Fragmentation and slip are two typical micrometer scale crystal orientation changing. Many data mining methods can be utilized to analyze the changes in solids at micro meter scale to study and predict material response for industry applications. Usually people combine these two interior imaging techniques.

0.2 Data Process of Tomography and HEDM

I used three different solids data: Cu, Zr and Au here in this project. I performed data collection process with group members at Argonne National Lab. The Cu raw tiff data are reduced, optimized and finally reconstructed by me at CMU using cluster in physics department. The Zr dataset is existing group dataset which was processed by previous group member Jonathan Lind. The Au raw tiff images are provided by previous group member David Mesnache and are reconstructed by me using the optimization methods discussed in this thesis.

I study and implement several optimization algorithms to perform tomography reconstruction for Au with the algebraic reconstruction technique formalism. Then I do simulation to verify correctness before applying to real data. Next, further analysis based on HEDM images are presented in this thesis: I apply Singular Value Decomposition (SVD) method to study real physics slip planes in 3D, apply k-means method to detect fragmentation under increasing external strains.

0.3 Tomography Optimization Study

Given the projection data and the geometrical relations, ART formulates the problem to be linear equations, and iteratively finds the solution with different optimization methods. As shown in Fig. 1 (a), we denote f_j as the intensity value in the *j*-th cell of the region enclosing the object, this square which contain N^2 pixels represent the 2D sample space containing the cut of sample. Then we simulate the X-ray beam to pass through the 2D sample. Assume there are number of M parallel rays going through the 2D sample space. Each p_i , i = 1, 2...Mis the intensity on the projection screen. We define nA as the the number of rotation angles. For example, we will rotate the sample 180° with step size 5° , then the length of rotation angle is $180^{\circ}/5^{\circ}=36^{\circ}$. Since M is the total number of parallel rays at a specific angle, then after going through all the rotation angles, there are nA*M intensity equation in the detector space. In ART, the intensity sum-up equation is:

$$\sum_{j=1}^{N^2} \omega_{ij} * f_j = p_i, \quad i = 1, 2...nA * M,$$
(1)

where $w_{ji} = \frac{\operatorname{areaofABC}}{d^2}$ is shown in Fig. 1 and d is the width of one pixel. The equations can be represented in matrix form: Ax = b, where A is a nA * M by N^2 matrix with $A_{ij} = \omega_{ij}$ vector x represents the sample space's each pixel with $x_i = f_i$. Vector b represents the detector space with $b_i = p_i$. Given a figure, the simulation program constructs A and b. Since A is typically very large, it is generally infeasible to directly solve the linear equations. The reconstruction is thus usually achieved with iterative optimization methods.

There is a widely traditional algorithm Alternative projection method can be used to reconstruct the tomography.

Alternative projection method(Kaczmartz method)

The unique solution $(f_1, f_2, ..., f_{N^2})$ can be considered as a single point in the N^2 -dimensional space and each of the above equation represents a hyperplane in this space. The 2-dimension case is the typical famous alternating projections algorithm, keep projecting back and forth



Figure 1: (a) The basic problem of tomography reconstruction and definition of w_{ji} . (b) The illustration of Kaczmarz method (alternative projection) optimization for the case of two unknowns.

as illustrated in Fig(b). We start with initial guess $\mathbf{f}^{(0)}$ in the N dimensional space, then project the initial guess $\mathbf{f}^{(0)}$ to the hyperplane of the first equation and get $\mathbf{f}^{(1)}$. Denoting $\omega_i = (\omega_{i1}, \omega_{i2}, ..., \omega_{iN^2})$, the ith projection is described by

$$\mathbf{f}^{(i)} = \mathbf{f}^{(i-1)} - \frac{\mathbf{f}^{(i-1)} * \omega_i - p_i}{\omega_i * \omega_i} * \omega_i \qquad i = 1, 2...nA * M$$
(2)

We call a full nA * M times projection as one iteration. The algorithm iterates until converge.

Various Optimization Methods Implemented in Tomography Reconstruction

As the alternative projection method is slow and sometimes might be stuck if there's a bad point on the slice (which is corresponding to one line of equations). Since the objective function of tomography reconstruction is continuous and convex, I utilized various convex optimization methods to solve this tomography reconstruction problem. The formulation is seen below in gradient descent method section.

Gradient descent

The solution of these linear equations can be written as the optimizer of the following optimization problem,

$$\min_{x \in \mathbb{R}^N} F(x) = \min_{x \in \mathbb{R}^N} \frac{1}{2} ||Ax - b||_2^2 = \min_{x \in \mathbb{R}^N} \frac{1}{2} x^T A^T A x - b^T A x + b^2.$$
(3)

Note that from later on, to follow the convention in the optimization field, we use x to denote the f vector.

the gradient descent algorithm is suitable to perform optimization to reconstruct the figure. The update rule is,

$$x^{(k)} = x^{(k-1)} - t * A^T (A x^{(k-1)} - b),$$
(4)

where t is the step size. In practice, we find a fixed step size is not suitable for this problem. Instead, we perform backtracking line search method. In every gradient descent step, we start with a constant t = 1, then we shrink t to be 0.9t until the following condition is met:

$$F(x^{(k-1)} - t * A^T(Ax^{(k-1)} - b)) < F(x^{(k-1)}) - t * ||A^T(Ax^{(k-1)} - b)||_2^2/2.$$
(5)

Next, I explored with **Conjugate gradient descent**, **Newton method**, **Quasi-Newton BFGS methods** after gradient descent. The detailed implementations can be seen in Appendix.

Results of Simulation Data

I first implemented the tomography simulation to get the matrix A and b, then implemented the above five reconstruction (optimization) algorithms. I first generated a figure of 50x50pixels. With suitable number of projections and angles, the simulated A matrix is of dimension 18000×2500 and b is 18000×1 . We set the termination criteria to be the error



 $||Ax - b||_2^2/||b||_2^2 < 10^{-6}$ for all the algorithms. The origin and reconstructed figures are shown below in Fig. 2. With the same stop criteria, the reconstructions are similarly good.

Figure 2: The origin and reconstructed figures: (a) origin (b) Alternative projection (c) Gradient descent (d) Conjugate gradient (e) Newton method (f) Quasi-Newton: BFGS

To compare these optimization algorithms, the errors $||Ax - ||_2^2/||b||_2^2$ vs iterations are shown in Fig. 3.

We see that the Newton method takes fewest iterations and alternative projection takes the most. Note that the iterations of alternative projection is counted as the number of a full iteration, namely 2500 projections in this simulation.

To further compare these method, the time used for these methods are listed in Tab. 1 below, for both 50×50 and 100×100 simulated figures.

Based on computation time, the conjugate gradient method is the fastest. The fact BFGS is slowest for largest cell may be because of its large space complexity and virtual memory is



Figure 3: (a) The error vs iterations in simulation. The error is defined as $||Ax - ||_2^2/||b||_2^2$.

used. Newton's method is reasonably good because we used the fast way to compute $A^{-1}b$ with $A \setminus b$. The advantage of alternative projection is its small space complexity since in one update only one row of A matrix is used. We find that the sparse matrix stored using Matlab is column-major. Thus to improve cache efficiency, we first transpose A and then pick each column of A^T to perform update. The speed is 8 times faster than use A directly. Based on these comparisons, we will use either conjugate gradient or alternative projection as my optimization methods. Moreover, my conjugate gradient solver perform even faster than Matlab quadprog solver as shown in the Table. This is because every iteration of quadprog is expensive although effective.

Method	50x50 time (s)	100x100 time (s)			
Alter. Proj.	67.42	471.34			
Grad. Desc.	7.14	39.69			
Conj. Grad.	0.34	2.43			
Newton	6.45	166.59			
BFGS	55.78	3748			
Matlab solver	1.44	38.2			

Real Experiment Tomography Reconstruction and MPI Implementation

I reconstruct the Au experimental data and the results is shown in Fig. 4. The left panel is our ART reconstruction with conjugate gradient algorithm, middle is our ART reconstruction with alternative projection. In our reconstruction, CG takes 12 iterations, 26 seconds to achieve error of 0.005, where the error is defined as $||Ax - b||_2^2/||b||_2^2$. Our alternative projection takes 50 iterations, 2312 seconds to achieve error of 0.122. The error of alternative projection does not decrease with further iterations. This difference indicates that alternative projection is sensitive to bad (noisy) equations since it projects to every hyperplane one by one. Our methods improves the tomography reconstruction and give more clear results on this particular data.



Figure 4: The reconstruction of a slice of the sample using our CG, alternative projection and Tomopy alternative projection from left to right.

The real X-ray tomography data obtained from experiment at Argon National Lab is very large, which is usually 10 GB. To perform the reconstruction with MPI will improve the reconstruction speed. The steps of MPI implementation of the most straight forward gradient descent algorithm with C++ are described below.

Step1:We distribute the ATA, ATb and x to different processes from rank=0. We use

MPI underline Bcast : Broadcasts a message from the process with rank "root" to all other processes of the communicator

```
//distribute the ATb,ATA to all process, in order to
//parallelize the Gradient Descent Calculation
MPI_Bcast(ATb,ncol*1,MPI_FLOAT,0,MPI_COMM_WORLD);
MPI_Bcast(ATA,ncol*ncol,MPI_FLOAT,0,MPI_COMM_WORLD);
MPI_Bcast(x,ncol*1,MPI_FLOAT,0,MPI_COMM_WORLD);
MPI_Bcast(xnew,ncol*1,MPI_FLOAT,0,MPI_COMM_WORLD);
```

Step2: We partition the work for different processes, a good way is indexing with the rank for i - for - loop:

Partition work by i - for - loop, and do calculation on different process separately

```
istart = (ncol/nproc)*rank;
iend = (ncol/nproc)*(rank+1)-1;
for (int i = 0; i < MaxIter && (diff(x, xnew) > error || i == 0); i++) {
    x = xnew;
    for(int j = istart; j < iend; ++j) {
        xnew[j] += 2*stepSize*ATb[j];
        for (int k = istart; k < iend; ++k) {
            xnew[j] -= 2*stepSize*ATA[j][k]*x[k];
        }
        }
        cout << "Iter: " << i << " diff: " << diff(x, xnew) << endl;
    }
    return 0;
}
```

Figure 4 shows the i - for - loop partition. n=ncol is the row dimension of the matrix ATA, I divided it into nproc processes and the starting index is istart = (ncol/nproc)*rank; and the ending index of row is iend = (ncol/nproc)*(rank + 1) - 1;.



Figure 5: Distribute the ATAx term row block figure

step3: We collect values from each process and send them to rank=0.

```
//Gather computated result
MPI_Gather(x+(ncol/nproc*rank),ncol/nproc,
MPI_FLOAT,x+(ncol/nproc*rank),ncol/nproc,
MPI_FLOAT,0,MPI_COMM_WORLD);
```

In this section, I performed the ART formalism of X-ray tomography reconstruction and implemented various optimization algorithms. Both simulation and real data reconstruction are performed. Successfully MPI GD on local machine which can be further applied to cluster. An comparison of tomography and HEDM reconstruction will be shown in next section with further analysis adopted data mining methods.



Figure 6: (a) Tomography reconstruction of Cu (b) HEDM reconstruction of Cu

0.4 HEDM Analysis Compared With Tomography Reconstruction

I performed tomography reconstruction for Cu dataset as well as HEDM reconstruction. Because HEDM image contains the crystal orientation information, and in the process of plastic deformation, solids tend to have orientation rotations [5]. Fragmentation and slip are such two typical phenomenon. So I develop an algorithm detecting slip and fragmentation in Cu dataset, then using K-means clustering method to detect fragmentation across different strain state in Cu dataset[6]. See in Fig. A.2.

Algorithm Detecting the Face Centered Cubic Slip Systems in Cu

I develop an algorithm and write an automatic script for detecting these crystallographic orientation changes due to FCC slip activity (see algorithms in Appendix). For this Cu dataset, I explored the Inverse Pole Figure space by adopting k-means for fragmentation detection (see algorithm in Appendix) As fragmentation becomes common to grains under higher strain state, an automatic clustering method that can quantify and detect the fragmentation can be more useful than eye bow observation. K-means clustering method can be utilized here to detect the change of IPF shape under dynamic strain states. By labeling the data points to k centroids [7], the converged positions of k centroids would reveal the IPF shape's change of individual grain.

I used two different Cu datasets here. The first Cu data set reaches the highest strain of 3.2% in S7. The IPF's of the 100 largest grains are monitored from initial state, S1, to highest strain state, S7. In this interval, no orientation change or spreading is observed. As an example, the largest grain is shown in Fig. 7.



Figure 7: IPF of Cu S1, S2, S7-the highest strain we got reconstructed

The second Cu data set, from Pokharel's work, reaches 14%. In the high strain states, I observe the spreading of orientations in the IPF plots starting in S8 which has 9.2% strain. Under high strain states, fragmentation happens and the IPF shapes split with the crystallographical orientation changing significantly. Seen in the Fig. 8(a), blue points are the IPF shape under 9.2% strain state and its has a bifurcation shape. If we assign k = 3 centroids to the blue points in Fig. 8 and perform the k means clustering algorithm, the converged centroids will form an obtuse angle between the two vectors connecting the three centroids. The data points belonging to three centroids are plotted with red, green and blue color separately in Fig. 8.

In this section, I present algorithms developed for detecting fcc slip and present the



Figure 8: IPF of Cu Data set2.(a) Red, Green, Blue points are the IPF shape of same grain under strain state: 7.0%, 7.9%, 9.2%. (b) The three black circles are the final converged positions of k = 3 centroids given by k means algorithm.

results in a complete Cu data set with 12 strain states up to 14%[6]. Some inter-grain slip traces are detected. I then presented applying k means clustering to automatically detect fragmentation under increasing strains. Further we could apply this k means more generally and automatically. For detecting many grains' fragmentation in multiple strain states, the centroids positions will also be tracked and can be used as input parameters for a function of strain.

One More Zr Dataset Analysis Using Singular Value Decomposition Fitting The 3D slip evolution in Zr

I have done HEDM reconstruction for another Zr dataset and apply the HCP slip tracking algorithm developed to this dataset, I successfully detected the slip activity. See in Fig. A.4 Next, I study the geometrical features of slip planes under increasing strain in 3D: First I extract the slip trace on each layer and stack these slip traces in 3D sample space to form the prismatic slip plane. Then I use the singular value decomposition (SVD) method with the closeness measured by orthogonal distance regression. The results are presented below.

The singular-value decomposition (SVD) method is a factorization of a real matrix in linear algebra[8, 9]. Assume we have a real matrix M, $M \in \mathbb{R}^{m \times n}$. The SVD decomposition of M is:

$$M = U\Sigma V^T \tag{6}$$

where $U \in \mathbb{R}^{m \times m}$ and $V \in \mathbb{R}^{n \times n}$ are orthogonal matrices. $\Sigma \in \mathbb{R}^{m \times n}$ is a diagonal matrix having the singular values $\sigma_1 \geq \sigma_2 \geq \dots \sigma_l \geq 0$, $l = \min(m, n)$.

Extraction of slip points yield a cluster of N 3-D prismatic slip points $p_1, p_2, ..., p_N$ and we want to find an optimal plane that is closest to these points. We use a least square sum of orthogonal distances to measure the closeness. If the plane's normal unit vector n and a point on the plane are known, then the plane is fixed and found. We assume the center of the N 3-D prismatic slip points, $C = 1/N \sum_{i=1}^{N} p_i$, will be exactly on the plane. The objective function to find the closest plane is

$$\min \sum_{i=1}^{N} ((p_i - C)^T n)^2 = \min \left\| M^T n \right\|_2^2$$
(7)

where $M = [p_1 - C, p_2 - C, ..., p_N - C]$ is a 3 × N matrix[8]. Use the singular decomposition $M = U\Sigma V^T$ and get the objective function as:

$$\min \left\| M^T n \right\|_2^2 = \min \left\| V \Sigma^T U^T n \right\|_2^2 = \min \left\{ (\sigma_1 y_1)^2 + (\sigma_2 y_2)^2 + (\sigma_3 y_3)^2 \right\}.$$
 (8)



Figure 9: In state S3: The 3D prismatic slip plane of GN2 shown in different view angles in (a-c). The colors of the points indicate the magnitude of the rotation angle at each point with scale bar shown in right. The black arrow is the plane normal of the fitted plane. The red arrow is the c-axis in sample space. The angle between c-axis and the fitted plane normal is also almost 90 degree in sample space.

Table 1: Characteristics of planar slip interfaces in grains GN2 and GN3 in state S3

Grain	MSF	SVD Plane nor-	SVD resid-	Angle be-	Angle be-	
Index	of	mal in crystal	uals	tween SVD	tween SVD	
	slip	space		n and c-	n and ten-	
	plane			axis	sile axis	
GN 2	0.44	[0.98, 0.23, 0.03]	10.79 um	87.00	116	
					36	

0.5 Conclusions

In this DAP project, I carried out a complete study of tomography image and HEDM image analysis for three dataset: Au, Cu, Zr. The data collection, cleaning, reduction and reconstruction were performed by me using bloch-cluster at CMU. After obtaining the reconstruction images, I explored with different machine learning methods to further analyze the data.

First, I performed the ART formalism of X-ray tomography reconstruction and imple-

mented several optimization algorithms to solve the equations in ART. Simulated the physical tomography process and performed reconstruction to verify correctness, then apply to real large experimental tomography data. Also MPI GD on local machine which can be further applied to cluster when reconstruct even larger tomography datasets.

Next, I compared the tomography images and HEDM reconstruction image of another dataset: Cu. Then developed algorithm combined with HEDM reconstruction images to carry out further analysis such as slip and fragmentation detection. I used K-means methods and SVD method to study the geometric character of interfaces at micrometer scale in polycrytsal.

In summary, tomography and HEDM data can be complementary when imaging the interior of solids. With the help of various machine learning methods, many physics phenomenon such as slip, fragmentation can be detected, tracked and even predicted quantitatively, which is extremely useful for studying response and selecting robust materials in industry application.

Bibliography

- U Lienert, SF Li, CM Hefferan, J Lind, RM Suter, JV Bernier, NR Barton, MC Brandes, MJ Mills, MP Miller, et al. High-energy diffraction microscopy at the advanced photon source. JOM Journal of the Minerals, Metals and Materials Society, 63(7):70–77, 2011.
- [2] RM Suter, D Hennessy, C Xiao, and U Lienert. Forward modeling method for microstructure reconstruction using x-ray diffraction microscopy: Single-crystal verification. *Review of scientific instruments*, 77(12):123905, 2006.
- [3] SF Li and RM Suter. Adaptive reconstruction method for three-dimensional orientation imaging. *Journal of Applied Crystallography*, 46(2):512–524, 2013.
- [4] Ashley D Spear, Shiu Fai Li, Jonathan F Lind, Robert M Suter, and Anthony R Ingraffea. Three-dimensional characterization of microstructurally small fatigue-crack evolution using quantitative fractography combined with post-mortem x-ray tomography and high-energy x-ray diffraction microscopy. Acta Materialia, 76:413–424, 2014.
- [5] F Roters, Z Zhao, and D Raabe. Development of a grain fragmentation criterion and its validation using crystal plasticity fem simulations. In *Meeting*.
- [6] Reeju Pokharel, Jonathan Lind, Anand K Kanjarla, Ricardo A Lebensohn, Shiu Fai Li, Peter Kenesei, Robert M Suter, and Anthony D Rollett. Polycrystal plasticity: comparison between grain-scale observations of deformation and simulations. Annu. Rev. Condens. Matter Phys., 5(1):317–346, 2014.

- [7] John A Hartigan and Manchek A Wong. Algorithm as 136: A k-means clustering algorithm. Journal of the Royal Statistical Society. Series C (Applied Statistics), 28(1):100–108, 1979.
- [8] Inge Söderkvist. Using svd for some fitting problems. University Lecture, 2009.
- [9] L De Lathauwer, B De Moor, J Vandewalle, and Blind Source Separation by Higher-Order. Singular value decomposition. In Proc. EUSIPCO-94, Edinburgh, Scotland, UK, volume 1, pages 175–178, 1994.
- [10] Jan Tullis, John M Christie, and David T Griggs. Microstructures and preferred orientations of experimentally deformed quartzites. *Geological Society of America Bulletin*, 84(1):297–314, 1973.

Appendix A

Appendix

A.1 Optimization Algorithms Implemented

Gradient descent

The solution of these linear equations can be written as the optimizer of the following optimization problem,

$$\min_{x \in \mathbb{R}^N} F(x) = \min_{x \in \mathbb{R}^N} \frac{1}{2} ||Ax - b||_2^2 = \min_{x \in \mathbb{R}^N} \frac{1}{2} x^T A^T A x - b^T A x + b^2.$$
(A.1)

Note that from later on, to follow the convention in the optimization field, we use x to denote the f vector.

Since the objective function is a smooth convex function, the gradient descent algorithm is suitable to perform optimization to reconstruct the figure. The update rule is,

$$x^{(k)} = x^{(k-1)} - t * A^T (Ax^{(k-1)} - b),$$
(A.2)

where t is the step size. In practice, we find a fixed step size is not suitable for this problem. Instead, we perform backtracking line search method. In every gradient descent step, we start with a constant t = 1, then we shrink t to be 0.9t until the following condition is met:

$$F(x^{(k-1)} - t * A^T(Ax^{(k-1)} - b)) < F(x^{(k-1)}) - t * ||A^T(Ax^{(k-1)} - b)||_2^2/2.$$
(A.3)

Conjugate gradient

Conjugate gradient is usually faster than gradient descent while much less computational costly than Newton's method. For a general quadratic optimization problem, $\min_{x \in \mathbb{R}^n} \frac{1}{2}x^TQx - b^Tx$, the conjugate gradient method first finds conjugate directions d_i that satisfies $d_i^TQd_j = 0$, $\forall i \neq j$. An example of the CG method and its comparison with gradient descent is shown in Fig. A.1. The left panel shows a 2D example of the CG method. The ellipsoid is the contour of the function. In this problem, minimizer is found in two steps. The right panel is adapted from wikipedia. The green curve is gradient descent, where every search direction is perpendicular to the contour. The red is CG. In this case, it takes less steps than gradient descent to find the optimizer.



Figure A.1: The CG method and the comparison between CG and gradient descent directions.

The optimal solution can be written as,

$$x^* = \sum_{i=1}^n \alpha_i d_i, \text{ with } \alpha_i = \frac{\mathbf{d}_i^{\mathrm{T}} \mathbf{b}}{\mathbf{d}_i^{\mathrm{T}} \mathbf{Q} \mathbf{d}_i}.$$
 (A.4)

A systematic way to find the conjugate directions exists. The conjugate gradient algorithm is listed in Algo. 1. In our problem we just need to plug in $Q = A^T A$ and $b = b^T A$.

Algorithm 1: Conjugate gradient for quadratic programming						
Input	Q, b, $x^{(0)}$	The function information and initial value				
Output	x^*	optimal value				
Function	$d_0 = -g_0 = b - Qx^{(0)}$					
	do					
1	$\alpha_k = \frac{d_k^T b}{d_k^T Q d_k}, x^{(k+1)} = x^{(k)} + \alpha_k d_k$					
	$g_{k+1} = Qx^{(k+1)} - b, \qquad \beta_k = \frac{g_{k+1}^T Qd_k}{d_k^T Qd_k}$					
	$d_{k+1} = -g_{k+1} + \beta_k d_k, k = k+1$					
	until the stopping criterion is satisfied					

Newton method

Since the objective function is twice differentiable, we can use Newton method. However, in the quadratic programming case, this is actually cheating because if we can afford to calculate Hessian matrix $(A^T A)^{-1}$, we can find the optimizer directly by, $x^* = (A^T A)^{-1}b$. To make a comparison between first order and second order methods, pretending we do not know the exact solution through Hessian, the update rule for Newton's method is,

$$x^{(k)} = x^{(k-1)} - t * (A^T A)^{-1} * A^T (A x^{(k-1)} - b),$$
(A.5)

where similarly as gradient descent, the suitable step size t can be find by backtracking. Note that $(A^T A)^{-1}$ is usually not sparse even though A is sparse.

Quasi-Newton: BFGS

Since typically the Hessian matrix is hard to calculate or even not positive definite, we need to find approximate Hessian matrix to perform Newton update. One commonly used algorithm is the BFGS method. The intuition is to update the approximate Hessian matrix at each step with symmetric rank one matrices. The algorithm is listed below,

Algorithm	1 2: quasi-Newton	BFGS algorithm
Input	Q, b, $x^{(0)}, H^{(0)}_{BFGS}$	The function and initial value
Output	x^*	optimal value
Function	$d_0 = -g_0 = b - Qx^0$	0)
_	do	
1	$g_k = Qx^{(k)} - b,$	$d_k = -H^{(k)}g_k$
	$\alpha_k = \arg \min_{\alpha > 0}$	$f(x^{(k)} + \alpha d_k), x^{(k+1)} = x^{(k)} + \alpha_k d_k, p_k = \alpha_k d_k g_{k+1} = Qx^{(k+1)} - b_k$
	$q_k = g_{k+1} - g_k,$	$H^{(k+1)} = H^{(k)} + \left(1 + \frac{q_k^T H^{(k)} q_k}{p_k^T q_k}\right) \frac{p_k p_k^T}{p_k^T q_k} - \frac{p_k q_k^T H^{(k)} + H^{(k)} q_k p_k^T}{q_k^T p_k}, k = k+1$
	until the stopping	g criterion is satisfied

A.2 Cu Strain Information

All the strain information of the measurements are shown in table. A.1. Using same axistolerance $\theta_t = 5^o$ for the whole 12 strain states, the results of FCC slip for Cu data is shown in Fig. A.3.

Table A.1: Cu Dataset 2: The nf-HEDM orientation reconstruction is from initial state S1 to highest strain state S14. The true strain ϵ are shown in first row and the true stress τ are shown in second row.

S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12
0.06%	0.4%	1.7%	3.5%	5.3%	7.0%	7.9%	9.2%	10.1%	11.5%	12.5%	14.0%
46.8N	69.8N	104.3N	125.2N	146.4N	168.1N	179.9N	187.1N	197.9N	204.5N	214.9N	221.3N



Figure A.2: Cu Data Set 2: 12 strain state up to 14% engineering strain



Figure A.3: (a)-(i) are FCC slip Detection Results for Cu dataset. From initial state S1, to strain state S12. The strain information are in table A.1

A.3 Algorithms of Cu slip Tracking and K-means For Detecting Fragmentation under increasing Strains

In FCC Cu unit cell, the {111} planes are with the highest planar density. There are three such equivalent planes and each plane has three possible slip directions. The crystal would glide in the slip directions and rotate around the slip rotation axis **r**. Given one slip plane, the three slip rotation axis directions: < 211 >, < 121 > and < 112 >, also including the permutation for each direction, there will be 24 symmetric possible slip rotation axis directions in FCC unit cell.

The HEDM images gives the crystallographic orientation, so we can track the orientation changes and detect slip and fragmentation.

The algorithm for axis selection is shown below.

Algorithm 3: Algorithm Detecting candidate pairs can have rotations related with FCC Slip

1 <u>function:FCCRotationAxisSearch</u> Two neighboring element (A, B);

Input : (A, B):Rotation matrix: g_A, g_B ; Matrix M; Axis-off Tolerance angle α ;

Output: Logic Indicator 0/1, shows whether **g** is also one of the 24 symmetric FCC slip rotation axis;

2 Calculate (A, B)'s s rotation axis **g** in crystal coordinate, given the input g_A, g_B ;

3 if Angle between **g** and $v_i, i = 1, 2, ...24$; axis $\leq \alpha$ then

- 4 Index (A, B) with Logical Indicator 1; ;
- 5 else
- 6 Continue searching through the elements of HEDM orientation reconstruction map;;
- 7 end

The candidate pairs are selected and are marked with logical indicator in Algorithm 1. The candidate pairs with misorientation angles above threshold can be detected by the following Algorithm.

Algorithm 4: Algorithm Calculating the Rotation Angles of FCC slip

1 <u>function:FCCSlipSearch</u> HEDM orientation Map with Logical Indicator; **Input** : (A, B):HEDM orientation Map with Logical Indicator; Rotation Angle Threshold: θ_t ;

Output: Selected pairing element with rotation Angle α above θ_t ;

- 2 Calculate the rotation angle θ around the (A, B)'s s rotation axis $\mathbf{g}(\mathbf{r})$ in crystal coordinate
- 3 if $\theta \ge \theta_t$ then
- 4 Select (A, B) by marking the edge between (A, B) with black line;;
- 5 else
- 6 Continue calculating the rotation angles through the elements of HEDM orientation reconstruction map;;
- 7 end

The standard stereographic triangle of inverse pole figure is utilized to show whether there is any crystallographically orientation change. Grain fragmentation happens when there is such orientation changes [10]. Algorithm 5: Algorithm of k means detecting fragmentation in indivdual grain

- <u>function:K means Detecting Fragmentation</u> IPF data points in individual grain;
 Input : IPF points in individual grain under each strain state i;
- **2** Assign k = 3 centroids to IPF data points under each strain state i;
- 3 Labeling IPF data point to one of the k centroids by measuring the distance to each of the k centroids;
- 4 The centroid of each of the k clusters becomes the new mean, iteratively re-assign new cluster until convergence reached
- 5 if The angle θ between centroid junction vectors changes under increasing strain states; then
- 6 Index Grain with 1 as a logical indicator that there is fragmentation in this grain;;
- 7 else
- 8 Index Grain with 0 indicates no fragmentation;;
- 9 end

A.4 Zr Slip Detection Results



(a)

(b)



(c)



(d)



(f)



Figure A.4: (a)-(i) are the prismatic searching results of HEDM reconstruction of nine layers of Zr specimen %. Black lines are drawn on the shared edges neighboring voxels that qualify as prismatic slip events. The common crystal rotation axis has a 5 degree tolerance away from crystal c-axis, their misorientation angle threshold is > 0.5 degree.