Solving Atomic Structure using Statistical Mechanical Searches on X-ray Scattering Derived Potential Energy Surfaces

by

Christopher James Wright

Bachelor of Science
Brown University 2014

Submitted in Partial Fulfillment of the Requirements

for the Degree of Masters of Science in

Chemical Engineering

College of Engineering and Computing

University of South Carolina

2016

Accepted by:

Xiao-Dong Zhou, Major Professor

Thomas Vogt, Committee Member

Mark Uline, Committee Member

Jochen Lauterbach, Committee Member

Lacy Ford, Vice Provost and Dean of Graduate Studies

© Copyright by Christopher James Wright, 2016
All Rights Reserved.

DEDICATION

Acknowledgments

ABSTRACT

TABLE OF CONTENTS

DEDICATION iii

ACKNOWLEDGMENTS iv

ABSTRACT v

LIST OF TABLES ix

LIST OF FIGURES x

CHAPTER 1 ATOMIC STRUCTURE: EXTRACTION AND APPLICATION . . . 2

1.1 Atomistic Goals 2

1.2 Atomistic Experiments 3

1.3 Atomistic Simulations 3

CHAPTER 2 STATISTICAL MECHANICAL ENSEMBLES AND POTENTIAL ENERGY SURFACES 4

2.1 Introduction 4

2.2 Potential Energy Surfaces 4

2.3 Ensembles 6

CHAPTER 3 ATOMIC PAIR DISTRIBUTION FUNCTION: THEORY AND COMPUTATION 11

3.1 Theory 11

3.2 Computation 11

3.3 Experiment 14

3.4 Data Processing Workflow 14

CHAPTER 4 BENCHMARKING 22

4.1 PDF 22

4.2 PDF with ADPs 22

CHAPTER 5 ANNEALING AND AGGREGATION OF 2NM AU NANOPARTICLES 23

5.1 Experiments 23

5.2 Data Processing 23

5.3 Data Analysis 23

5.4 Simulation 23

5.5 Structural Analysis 23

5.6 Conclusions 23

CHAPTER 6 PHASE CHANGES AND ANNEALING DYNAMICS OF PR_2NIO_4 AND ITS DERIVATIVES 24

6.1 Experiments 24

6.2 Data Processing 24

6.3 Data Analysis 24

6.4 Simulation 24

6.5 Structural Analysis 24

6.6 Conclusions 24

Chapter 7 Conclusion 25

List of Tables

LIST OF FIGURES

Figure 3.1 Q resolution as a function of Q. 17

Figure 3.2 Generated beamstop masks for a beamstop holder with 10% transmittance. From left to right: the raw image, the masked image, and the missed pixels 19

Figure 3.3 Generated beamstop masks for a beamstop holder with 30% transmittance. From left to right: the raw image, the masked image, and the missed pixels 19

Figure 3.4 Generated beamstop masks for a beamstop holder with 50% transmittance. From left to right: the raw image, the masked image, and the missed pixels 19

Figure 3.5 Generated beamstop masks for a beamstop holder with 90% transmittance. From left to right: the raw image, the masked image, and the missed pixels 20

Figure 3.6 Generated dead/hot pixel masks for a detector. From top left clockwise: 100, 300, 500, and 1000 dead/hot pixels. 21

INTRODUCTION

This is the introduction to the thesis.

Chapter 1

Atomic Structure: Extraction and Application

1.1 Atomistic Goals

The only way to truly understand the fundamental souce of material and chemical properties is through atomic structure. The goal of atomistic engineering is to produce novel structures and combinations of structures to engender new properties and functions. This includes producing stronger materials, more durable catalysts, more energy dense batteries, and many more engineering applicatons. The true power of atomistic engineering has been shown in biochemistry and pharmicutical design. Although the production of drugs and biomeical treatments is usually considered to be rather far from the field of catalyst design and materials science, the atomistic nature of these fields can not be denied. The field of protiene structural analysis stands as an example of structural science, elcuidating the three dimensional coordinates of thousands of atoms. These structures are then used to describe how the molecular machinery of the biological world works, enabling the development of new drugs and treatments for diseases and a deeper understanding of how we evolved. The development of protene inhibiter drugs, which are important to so many treatments, would have not been possible without very detailed atomic structures. The asperation of this work is to create this level of accuracy and utility, generating structures which allow for the understanding of how materials work on a fundamental level.

1.2 Atomistic Experiments

Single Crystal Diffraction

Electron Microscopy

X-ray Total Scattering

1.3 Atomistic Simulations

The goals of atomistic simulations are usually to produce atomic structures from quantum mechanical first principles, as in the case of Density Functional Theory (DFT), or classical approximations to quantum mechanics.

Density Functional Theory

Classical Force Field

Monte Carlo and Statistical Mechanics

Maybe put the ensemble and PES work here, since it is more general than the PDF per say. Also the rational for the gradients and fast computation make much more sense knowing we are going to be very sample happy and follow the gradient of the PES.

CHAPTER 2

STATISTICAL MECHANICAL ENSEMBLES AND POTENTIAL ENERGY SURFACES

2.1 INTRODUCTION

The approach taken in this work for solving the atomic structures of materials is one of optimization. The positional variables of the system are optimized so as to minimize the value of a potential energy surface (PES). The

2.2 POTENTIAL ENERGY SURFACES

A PES simply describes the potential energy of the system as a function of all its relevent coordinates in phase space, essentially providing a mapping $\mathbb{R}^n \to \mathbb{R}$. Usually these coordinates are the positions of the atoms q and their conjugate the momenta p. Note that there could be more variables associated with the system, for instance the magnetic moments of the atoms could play a role in describing the system. In this magnetic system there would be positional variables for the atomwise spin vectors and their "momenta". Application of the term "momenta" might seem odd here, as the magnetic spin does not have a mass or a velocity. However, since the magnetic "position" is defined on the PES we need to describe its conjugate varible to properly formulate Hamitonian dynamics and the kinetic portion of the PES.

Experimentally Derived Potential Energy Surfaces

Generally PESs are obtained from purely computational experiments including: ab-initio DFT, classical approximations via the embedded atom method, or even parameter driven models with experimentally fitted parameters. However, one can dervive a PES from an experiment which describes how well the model reproduces the experimental data. In this case one needs a theoretical and computational framework mapping the atomistic variables of the simulation to the same space of the data obtained from the experiment. This allows the experiment to be compared directly against the predicted data via an experimentally derived PES.

Potentials

For an experiment which produces 1D data, like powder diffraction, EXAFS or XPS, the implemented potentials are:

$$\chi^2 = \sum_{a=a_{\text{min}}}^{a_{\text{max}}} (A_{_\text{obs}} - \alpha A_{_\text{calc}})^2 \tag{2.1}$$

$$Rw = \sqrt{\frac{\sum_{a=a_{\text{min}}}^{a_{\text{max}}} (A_{_\text{obs}} - \alpha A_{_\text{calc}})^2}{\sum_{a=a_{\text{min}}}^{a_{\text{max}}} A_{_\text{obs}}^2}} \tag{2.2}$$

$$\chi^2_{\text{INVERT}} = \frac{1}{N} \sum_j \sum_r [A_{obs}(r) - \alpha A_{j_{\text{calc}}}(r)]^2 \tag{2.3}$$

$$\alpha = \frac{\sum_{a=a_{\text{min}}}^{a_{\text{max}}} A_{_\text{obs}} A_{_\text{calc}}}{\sum_{a=a_{\text{min}}}^{a_{\text{max}}} A_{_\text{calc}}^2} = \frac{\vec{A}_{_\text{obs}} \cdot \vec{A}_{\text{calc}}}{|\vec{A}_{\text{calc}}|^2} \tag{2.4}$$

where A_{calc} and A_{obs} are the calculated and observed 1D experimental data and $A_{calc,j}$ is the calculated data for a single atom interacting with the other atoms of the system. Note that A_{calc} has a dependence on q, the positions of the system.

Forces

$$\vec{\nabla}\chi^2 = -2 \sum_{a=a_{\text{min}}}^{a_{\text{max}}} (\alpha \frac{\partial A_{_\text{calc}}}{\partial \gamma_{i,w}} + A_{_\text{calc}} \frac{\partial \alpha}{\partial \gamma_{i,w}})(A_{_\text{obs}} - \alpha A_{_\text{calc}}) \tag{2.5}$$

$$\vec{\nabla} Rw = \frac{Rw}{\chi^2} \sum_{a=a_{\mathrm{min}}}^{a_{\mathrm{max}}} (\alpha \frac{\partial A_{\mathrm{calc}}}{\partial \gamma_{i,w}} + A_{\mathrm{calc}} \frac{\partial \alpha}{\partial \gamma_{i,w}})(\alpha A_{\mathrm{calc}} - (A_{\mathrm{obs}})) \tag{2.6}$$

$$\frac{\partial \alpha}{\partial \gamma_{i,w}} = \frac{(\sum_{a=a_{\mathrm{min}}}^{a_{\mathrm{max}}} A_{\mathrm{obs}} \frac{\partial A_{\mathrm{calc}}}{\partial \gamma_{i,w}} - 2\alpha \sum_{a=a_{\mathrm{min}}}^{a_{\mathrm{max}}} A_{\mathrm{calc}} \frac{\partial A_{\mathrm{calc}}}{\partial \gamma_{i,w}})}{\sum_{a=a_{\mathrm{min}}}^{a_{\mathrm{max}}} A_{\mathrm{calc}}^2} \tag{2.7}$$

$$\vec{\nabla} \chi^2_{\mathrm{INVERT}} = \frac{-2}{N} \sum_{a=a_{\mathrm{min}}}^{a_{\mathrm{max}}} \sum_{j} (\alpha \frac{\partial A_{j_{\mathrm{calc}}}}{\partial \gamma_{i,w}} + A_{j_{\mathrm{calc}}} \frac{\partial \alpha}{\partial \gamma_{i,w}})(A_{\mathrm{obs}} - \alpha A_{j_{\mathrm{calc}}}) \tag{2.8}$$

where $\gamma_{i,w}$ is the ith arbitrary positional variable in the wth direction. The concept of an "arbitrary positional variable" might seem a bit cumbersome but it allows us to define the forces for any atomic parameter which can be represented as a vector in 3-space. This comes in handy when trying to define the forces acting on variables like anistropic displacement parameters or atomic magnetic spins.

DISCUSS INVERT A BUNCH. ALSO COMPARE RW AND CHI**2, POTENTIALY WITH A FIGURE.

2.3 Ensembles

While PESs describe which atomic configurations are the most desirable and how the atoms would like to get there, the ensemble describes how the atoms move on the PES. The abstraction of the PES from the ensemble is an important one, as it allows for the reuse and exchange of both PESs and ensembles for a wide array of problems. Statistical mechanical ensembles can be described in two ways, analyticly and stochasticly. For long simulation times and fine enough numerical or analytical integration these two descriptions should be identical. In either case one starts by defining the Hamiltonian $\mathcal{H}$ as the total energy of the system. Thus, the Hamiltonian is described as the sum of the potential $U(q)$ and kinetic $K(p)$ energies, where q is the positions of the atoms and p is their momenta

$$\mathcal{H}(q, p) = U(q) + K(p) \tag{2.9}$$

where $K(p) = \frac{1}{2} \sum_i \frac{p_i^2}{m_i}$ and i denotes the ith particle. Analyticly one generally defines a partition function, which describes the sum of probabilities over all potential atomic states.

$$\Xi = \sum_i P_i(q,p)$$

where P_i is the probability of the ith state and is a function of the total energy of that state. This partition function can then be used to obtain the probabilty of any specific state.

Hamiltonian Monte Carlo

In order to model dynamics we need to describe the motion of the particles in our system, thus:

$$\frac{dq_i}{dt} = \frac{\partial \mathcal{H}}{\partial p_i} = p_i \tag{2.10}$$

$$\frac{dp_i}{dt} = -\frac{\partial \mathcal{H}}{\partial q_i} = -\vec{\nabla} U \tag{2.11}$$

Using these equations we can derive the position and momentum vectors at any point in time using the leap-frog algorithm:

$$p_i(t+\delta t/2) = p_i(t) - \frac{\delta t}{2}\frac{\partial}{\partial q_i}U(q(t)) \tag{2.12}$$

$$q_i(t+\delta t) = q_i(t) + \delta t * p_i(t+\delta t/2) \tag{2.13}$$

$$p_i(t+\delta t) = p_i(t+\delta t/2) - \frac{\delta t}{2}\frac{\partial}{\partial q_i}U(q(t+\delta t)) \tag{2.14}$$

Note that $\frac{\partial}{\partial q_i}$ is the gradient with respect to q where i denotes the ith atom being moved. Using this notation the gradient is

$$\vec{\nabla} U = \begin{bmatrix} \frac{\partial U}{\partial q_{0,x}} & \frac{\partial U}{\partial q_{0,y}} & \frac{\partial U}{\partial q_{0,z}} \\ \vdots & \frac{\partial U}{\partial q_{i,w}} & \vdots \\ \frac{\partial U}{\partial q_{n,x}} & \frac{\partial U}{\partial q_{n,y}} & \frac{\partial U}{\partial q_{n,z}} \end{bmatrix} = \begin{bmatrix} \vec{\mathcal{F}}_0 \\ \vdots \\ \vec{\mathcal{F}}_i \\ \vdots \\ \vec{\mathcal{F}}_n \end{bmatrix} \tag{2.15}$$

where $\frac{\partial}{\partial q_{i,w}}$ is the derivative with respect to q where w denotes direction of the derivative (x, y, or z), n is the number of atoms and U is the potential which depends on q, and $\vec{\mathcal{F}}_i$ is the "force" on the ith atom.

No-U-Turn-Sampling

Grand Canonical Ensemble

Ensemble description

In the Grand Canonical Ensemble (GCE) two sets of variables are allowed to change, the atomic positions and the total number of atoms and their associated identities. These two variables are controlled by temperature and chemical potential. The partition function is

$$\Xi = e^{-\beta(\mathcal{H}+\mu)} \tag{2.16}$$

This is translated into a Monte Carlo system, producing Grand Canonical Monte Carlo (GCMC).

Grand Canonical Monte Carlo

While the probabilities for atomic motion are the same as in the Canonical Ensemble, the addition or removal of an atom have their own probabilities. For the addition of an atom the probability is formally:

$$\min[1, \frac{V}{(N+1)\Lambda(T)^3} e^{-\beta\Delta U + \beta\mu}] \tag{2.17}$$

Similarly the removal of an atom has the probability:

$$\min[1, \frac{(N)\Lambda(T)^3}{V} e^{-\beta\Delta U - \beta\mu}] \tag{2.18}$$

However, both of these equations depend of the overall simulation volume and the thermal wavelength, which is undesirable as these are not really properties that we are of interest to these simulations. Thus, we roll them into the definition of the chemical potential, essentially setting the base chemical potential to counteract these effects. This makes certain that our simulation does not change if we change the overall cell volume. A GCMC move consists of creating a new atomic configuration, where an atom has been added or removed, and checking the above criteria. However, previous results have shown that this method is computationally expensive in dense liquids, and exceedingly expensive in solid materials. The long simulation times are due to the random nature of the atomic additions or removals which produce: over-tightly packed atoms, atoms in the middle of nowhere, or unphysical vacancies. These configurations are rejected by the GCMC criteria but their probability of being sampled is much higher than configurations which are lower in energy, since the number of incorrect ways to add/remove atoms is much larger than the correct ways. Thus we have implemented methods for biasing the atomic addition positions and the atomic removals toward configurations which are more likely to be accepted.

GCMC biasing

The first method is to remove some of the excess options from the probability pool. Initially the insertion positions are calculated at random using a random number generator and scaled to the size of the simulation cell. This produces probabilities which have floating point level precision, which is effectively infinite. While this produces a potentially infinite number of ways to create energetically favorable configurations, the infinite ways to produce bad configurations is much larger. Thus we can limit this by moving to voxels. In this case atoms are added to the center of voxels which have a pre-set resolution, limiting our total number of valid addition points. While this could produce some problems with ergodicity, we avoid this by allowing the atoms to translate throughout the system. Each voxel has a probability of being tried:

$$P_{i,j,k} = \frac{xyz}{abc} \tag{2.19}$$

where x, y, z and a, b, c are the resolutions and cell side lengths in the cardinal directions, respectively. While this does help to limit the total probability space it does not tell us which voxels are likely to lead to better configurations, leading to many rejected atomic additions. To combat this issue we can weigh the individual voxels, giving more probability to voxels which show promise and less to those with less likelihood to be accepted.

The approach most likely to yield success would be to measure the change in potential energy associated with the addition of an atom at the center of the voxel where the probability of a voxel to be tried is:

$$P_{i,j,k} = \frac{e^{\beta \Delta U_{i,j,k}}}{\sum_{i,j,k} e^{\beta \Delta U_{i,j,k}}} \tag{2.20}$$

where $\Delta U_{i,j,k}$ is the change in energy. However, calculating $\Delta U_{i,j,k}$ can be particularly expensive, especially when calculating scattering from atomic positions. The computational expense can be mitigated by using a cheaper potential, if only for the evaluation of the voxel energy, as previously shown. Similar to previous work we can use the Lennard Jones potential to approximate the addition potential.

CHAPTER 3

ATOMIC PAIR DISTRIBUTION FUNCTION: THEORY AND COMPUTATION

3.1 THEORY

To properly understand the PDF and its limitations we need to derive its mathematics. The following derivation has been performed numerous times but most recently and completely by Farrow and Billinge, it is reproduced here for clarity and completeness.

Derivation

Consider a wave incident on a volume of variable density...

Analytical Gradients

Many optimization algorithms and simulations methodologies, including HMC, require not only the potential energy of a given configuration but also the forces acting on that configuration. These forces are described by the gradient of potential energy of the system.

3.2 COMPUTATION

Simply deriving the equations for the PDF is not enough. The many body nature of the PDF equation make analytical solution of the structure from the PDF impossible.

Thus, the PDF must be computed from a structural candidates and compared against experimental results to evaluate the relability of the model.

HPC and GPUs

To properly solve the structure of materials the PDF will need to be computed many times and checked against experimental results. This requires computation of the PDF, potentialy over many atoms. Calculating these PDFs requires a fast, highly parallized, computational framework.

GPUs and Parallelization

Computing the PDF is an embarrassingly parallel problem. The basic procedure is to calculate the reduced structure factor $F(Q)$ for each atom pair and momentum transfer vector, sum over all the atom pairs, and Fourier transform the structure to the PDF. The first part of this procedure is perfectly parallizable, as each atom pair is seperate from the others. The summation over all the atomic reduced structure factors can be parallelized via distributed summing. Lastly the FFT can be parallelized using existing parellel FFT algorithms.

GPUs are particularly well suted to the task of computing PDFs. GPU chip architecture is designed to perform many task simultaniously by having potentially thousands of cores.

Map from ij space to k space

The above equations, although formally correct, are very ineffiecent. $F(Q)$ and its gradient are indexed over all the atoms twice, however there are symmetries that allow us to only compute over the atom pairs esentially mapping from an nxn space, ij space, to a $\frac{n(n-1)}{2}$ space, k space. For $F(Q)$ we apply the following mapping where E denotes the atomic coordinates in ij space, E' denotes $F(Q)$ before the summation

$$\begin{tikzcd}
E \arrow[r, "\psi"] \arrow[d, "\phi"'] & E' \arrow[r, "\Sigma"] & Z \\
B \arrow[r, "\psi'"'] & B' \arrow[ur, "\Sigma'"] &
\end{tikzcd}$$

in ij space, B denotes the atomic pairs in k space, B' denotes $F(Q)$ in k space, and Z denotes the final summed $F(Q)$. For the operators, ϕ denotes the mapping from ij space to k space $k = j + i * \frac{i-1}{2}$, ψ and ψ' denote the $F(Q)$ operation in ij and k space, respectivly. Σ denotes the sum over all the atoms.

To properly define Σ' we must establish whether $F(Q)$ is an even function. We can accomplish this by examining each of the portions of $F(Q)$, $\alpha, \beta, \tau, \Omega$. Ω is even, since r_{ij} is the interatomic distance, which is the same despite a flip of indicies, Q does not depend on the atomic indicies, and since Qr_{ij} is even so is $\sin Qr_{ij}$. Thus, Ω is even. Providing similar analysis to τ we can see that while $\vec{u}_{ij}$ is odd, so is the unit displacement vector between the two atoms, thus the two odds cancel out. Intuitivly this makes sense, since the $F(Q)$ equation is fundamentally interested in the interatomic distances which is even. Thus, switching atom indicies does not change $F(Q)$. Due to the even nature of the $F(Q)$ operator the Σ' operator sums over all the atom pairs, and multiplies by two to reflect the double counting of the Σ operator.

For the gradient a similar mapping is used:

$$\begin{tikzcd}
E \arrow[r, "\psi"] \arrow[d, "\phi"'] & E' \arrow[r, "\Sigma"] & Z \\
B \arrow[r, "\psi'"'] & B' \arrow[ur, "\tilde{\phi}\Sigma"] &
\end{tikzcd}$$

In this mapping, however, we use the $\tilde{\phi}\Sigma$ operator. This operator simultaniously performs a reverse mapping from k to ij space, and a summation with the correct symmetry. In this case the ψ and ψ' operators, which denote the $\vec{\nabla}F(Q)$ operator in ij and k space, are antisymmetric. Intuitivly this makes sense as an extension of Newton's Second Law, since each particle's interation is felt oppositely by its partner.

Periodic Boundary Conditions

Periodic boundary conditions can be helpful when simulating extended solids or large nanoparticles. In this case all the non-crystallinity is contained within the simulation box and the box is repeated to create the longer distance peaks observed in the PDF. To perform this we can break up the Debye equation into two main parts, the part that describes the interatomic distances within the simulation box and those between boxes. Neglecting the thermal motion portion:

$$F(Q) = \frac{1}{N\langle f\rangle^2}(\sum_{j\neq i} f_i^*(Q)f_j(Q)\frac{\sin(Qr_{ij})}{r_{ij}} + \sum_{i,j} f_i^*(Q)f_j(Q)\frac{\sin(QR_{ij})}{R_{ij}}) \tag{3.1}$$

where

$$R = |\vec{r} + \vec{u}| \tag{3.2}$$

$$\vec{u} = \gamma_1 * \vec{a} + \gamma_2 * \vec{b} + \gamma_3 * \vec{c} \tag{3.3}$$

3.3 EXPERIMENT

PDF experiments are generally performed at synchrotron light sources, as only these sources can provide the need flux, energy, and high momentum transfer vectors needed to obtain relyable PDFs.

3.4 DATA PROCESSING WORKFLOW

Processing the raw pixel intensities to the PDF is very important as we are extracting most of our interesting information out of very high Q data. This data relies on good statistics and sound background subtraction. Talk about papers from Billinge Group with thin film PDF and dilute NP solutions. Diagram of the overall data processing workflow. Discuss the NSLS-II data stack.

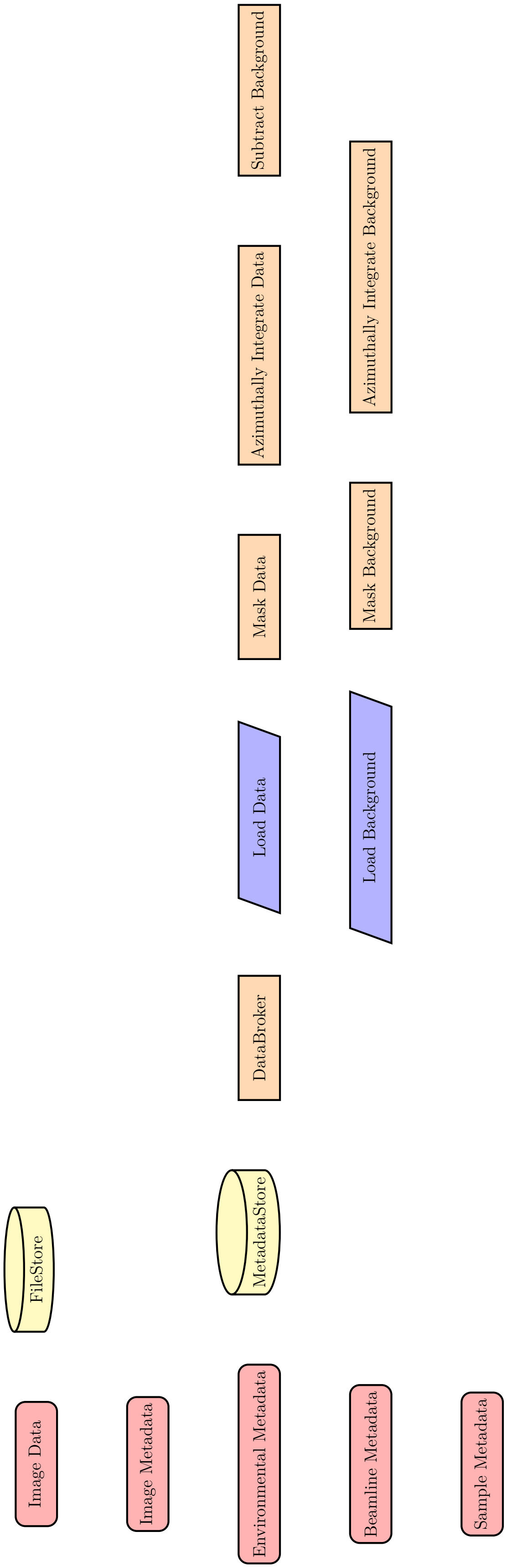
Image Data
Image Metadata
Environmental Metadata
Beamline Metadata
Sample Metadata
FileStore
MetadataStore
DataBroker
Load Data
Load Background
Mask Data
Mask Background
Azimuthally Integrate Data
Azimuthally Integrate Background
Subtract Background

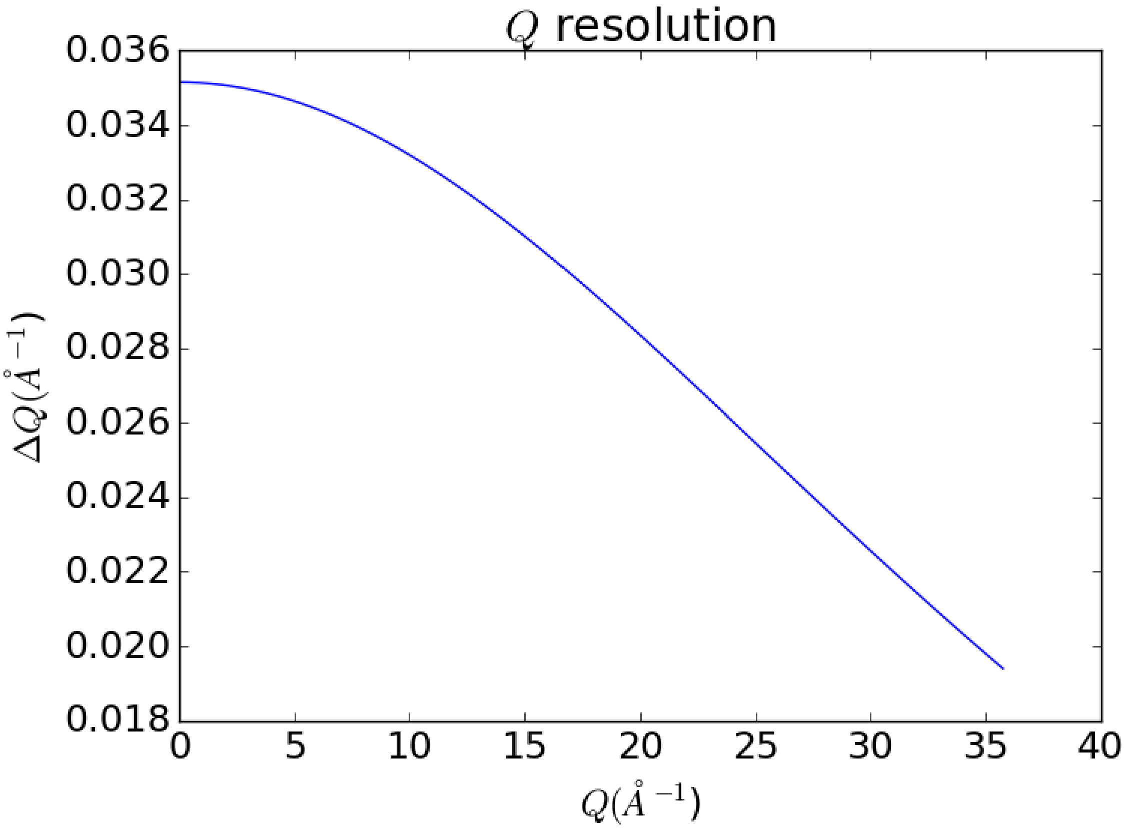

Figure 3.1 Q resolution as a function of Q.

MetadataStore Side Loading

Design of sidewinder-spec for loading the data into metadatastore. Most of the design considerations went into the loaders, which are different for each experiment.

Automated Image Azimuthal Integration

Mux data as needed. Use pyFAI to get the radial distance array. Note that to properly mask and integrate the system we need to compute the bin edges for the pixels. The bin edges change as a function of Q, as the angle subtended by a pixel shrinks essentially giving high Q pixels more resolution.

Mask the image.

Automated Mask Generation

Detector masking is an important part of any x-ray scattering workflow as dead/hot pixels, streak errors, and beamstop associated features can be averaged into the data changing the signal and its statistical significance. While some features, like the beamstop holder, can be easily observed and masked by hand other are much more difficult to observe even on large computer monitors. Additionally, while dead/hot pixels and streaks are usually static the hot pixels associated with textured or single crystal scattering or cosmic rays are not. Thus, coming up with an automated method for finding such erroneous pixels is important, especially as high flux diffraction beamlines can generate data very quickly.

While this problem can be quite complex in the most general case, we can use the annular symmetry of the powder scattering pattern to our advantage, by comparing a pixel against pixels in the same ring. Since non-textured powder scattering should produce the same pixel intensity for a given ring we can mask any pixels which are x standard deviations away from the mean. This method relies on the aforementioned pixel binning algorithm, as using miss sized bins will cause some pixels which should be in separate rings to be put together, and others which should be in the same ring to be separated. In that case the masking algorithm will overestimate the number of pixels to be masked due to the additional statistical variation in the sample.

NEED TO INCLUDE FIGURES OF POORLY BINNED PIXELS MASK

ALSO NEED ROTATED BEAMSTOP MASKS

ALSO ALSO NEED TO SHOW THE EFFECT OF CHANGING ALPHA FOR THE MASKING

ALSO ALSO ALSO SHOW SOME MASKS OF REAL DATA, INCLUDING DATA WITH SINGLE CRYSTAL/TEXTURE

Note that we do miss some pixels when the number of dead pixels grows very large. However, most detectors do not have that many dead pixels. We can run into

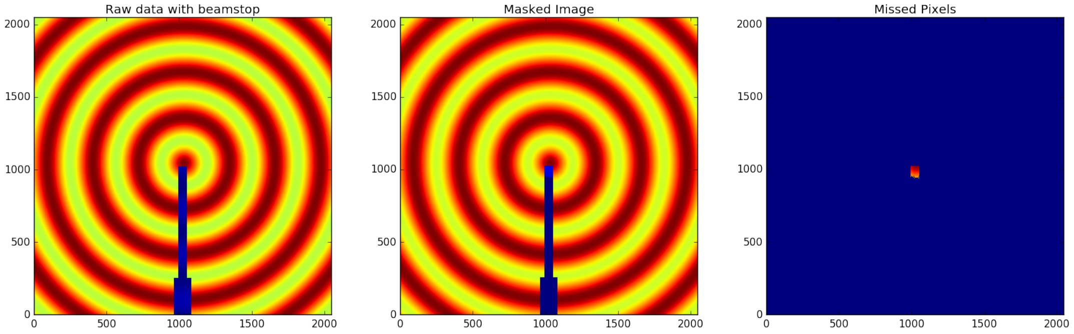

Figure 3.2 Generated beamstop masks for a beamstop holder with 10% transmittance. From left to right: the raw image, the masked image, and the missed pixels

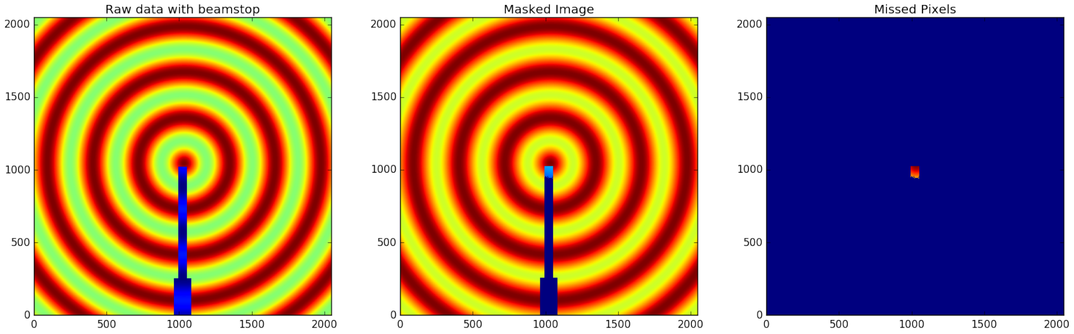

Figure 3.3 Generated beamstop masks for a beamstop holder with 30% transmittance. From left to right: the raw image, the masked image, and the missed pixels

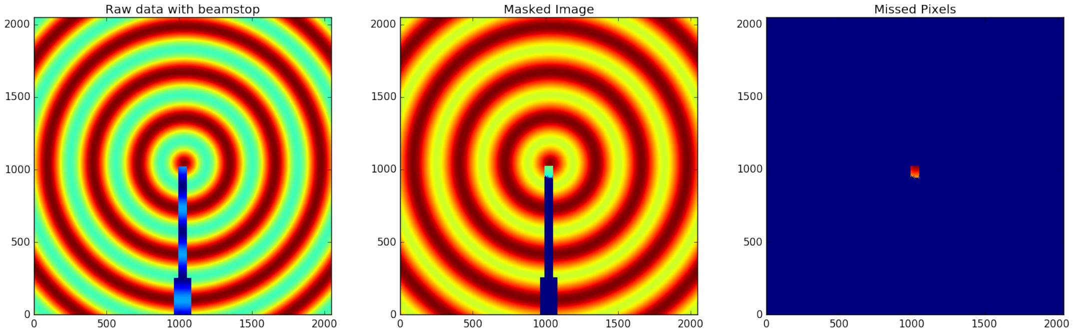

Figure 3.4 Generated beamstop masks for a beamstop holder with 50% transmittance. From left to right: the raw image, the masked image, and the missed pixels

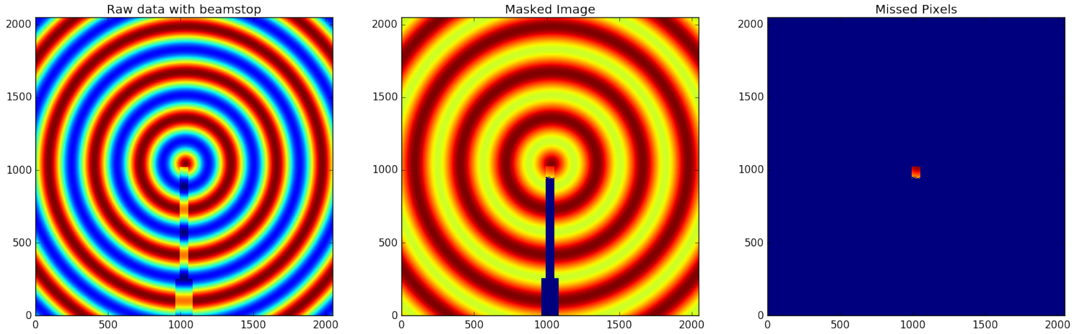

Figure 3.5 Generated beamstop masks for a beamstop holder with 90% transmittance. From left to right: the raw image, the masked image, and the missed pixels

these kinds of situations when running samples with some single crystal or textured components. However, when this is the case the contrast between the affected pixels and the desired signal is very large enabling easier masking.

We then mask data points Additionally the standard deviation threshold can be a function of the pixel distance from the center, allowing the mask generator to be more forgiving at certain points and stricter at others. This is particularly helpful as the small number of pixels near the point of incidence combined with the very sharp peaks causes some pixels to be improperly masked. Similarly it is important to remove dead pixels at the edge of the detector as these have an outsized effect on the integration as the pixel intensity is low to begin with. In practice this results in the removal of almost all dead pixels and potentially the beamstop holder. Removal of the holder depends on its individual properties, since a holder which is more x-ray opaque will cause a larger shift in the pixel intensity distribution. The method was benchmarked on synthetic data, with both hot and cold pixels added. Additional benchmarking was performed with synthetic beamstop holders of various x-ray transmittance. Anomalous corner masking most likely due to the small number of pixels out at the corners.

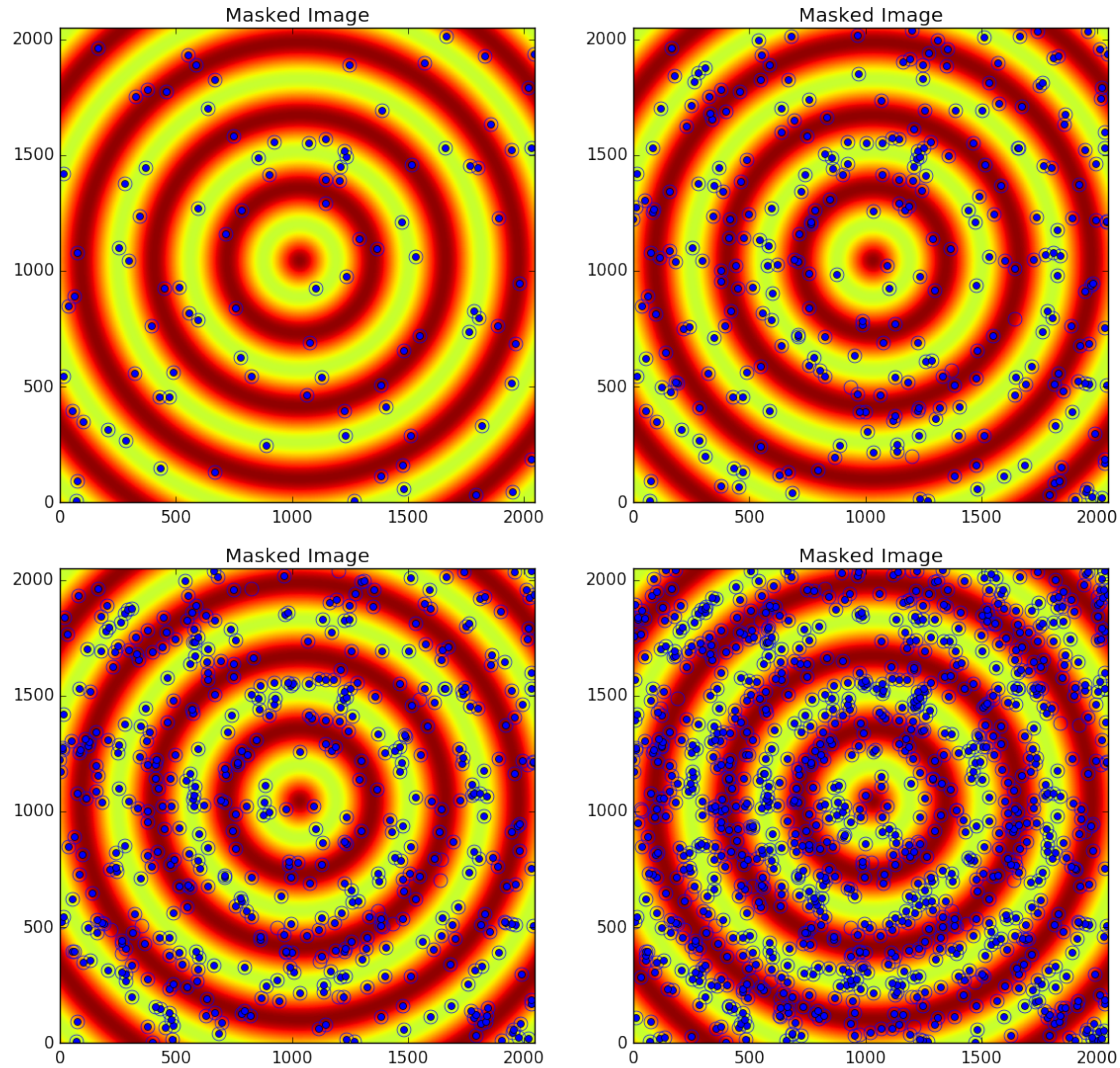

Figure 3.6 Generated dead/hot pixel masks for a detector. From top left clockwise: 100, 300, 500, and 1000 dead/hot pixels.

Chapter 4

Benchmarking

4.1 PDF

Au55: surface relaxed

Au55: surface disordered

Au55: amorphous

Au102: triple phase

Au147

C60

4.2 PDF with ADPs

CHAPTER 5

ANNEALING AND AGGREGATION OF 2NM AU NANOPARTICLES

5.1 EXPERIMENTS

NP Synthesis

X-ray Total Scattering Measurements

5.2 DATA PROCESSING

5.3 DATA ANALYSIS

5.4 SIMULATION

5.5 STRUCTURAL ANALYSIS

5.6 CONCLUSIONS

Chapter 6

Phase Changes and Annealing Dynamics of Pr_2NiO_4 and its derivatives

6.1 Experiments

Pr_2NiO_4 Synthesis

X-ray Total Scattering Measurements

6.2 Data Processing

6.3 Data Analysis

Intra Sample Comparison

Inter Sample Comparison

6.4 Simulation

Small Box

Large Box

6.5 Structural Analysis

6.6 Conclusions

CHAPTER 7

CONCLUSION