2008

Chemical analysis of polymer blends via synchrotron X-ray tomography

Heath Alan Barnett
Louisiana State University and Agricultural and Mechanical College, hbarne4@lsu.edu

CHEMICAL ANALYSIS OF POLYMER BLENDS VIA SYNCHROTRON X-RAY TOMOGRAPHY

A Dissertation

Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

In

The Department of Chemistry

by
Heath A. Barnett
B.S., Louisiana Tech University, 2003
December, 2008

Acknowledgments

My parents deserve many thanks for their continual support over the years as I have pursued my higher education. Without them none of this would have occurred.

I would also like to thank my committee, without whose advice and guidance, I would have not have been able to realize my full potential.

Table of Contents

ACKNOWLEDGEMENTS ..... **ii**

LIST OF TABLES ..... **v**

LIST OF FIGURES ..... **vi**

ABSTRACT ..... **xii**

CHAPTER 1 INTRODUCTION **1**
1.1 Bibliography 3

CHAPTER 2 EXPERIMENTAL AND COMPUTATIONAL METHODS **5**
2.1 Introduction to Tomography 5
2.1.1 The Radon Transform 6
2.1.2 Tomographic Imaging 8
2.2 The Tomography Experiment 11
2.2.1 Reconstructing the Data 13
2.3 Converting Absorptions to Chemical Concentrations 16
2.4 Synchrotrons 19
2.4.1 A Brief History of Synchrotrons 20
2.4.2 The Physics of Synchrotron Radiation 26
2.4.3 How a Synchrotron Is Built 28
2.4.4 Making Synchrotron Radiation Brighter 31
2.4.5 Materials Science With Synchrotron Radiation 32
2.5 Diffusion Theory 40
2.6 3D Image Processing 42
2.6.1 Thresholding 44
2.6.2 Neighborhoods 47
2.6.3 Edge Detection 48
2.6.4 Connected Components 50
2.6.5 Collision Detection 51
2.6.6 Domain Extraction and Neighbor Finding 52
2.7 Bibliography 58

CHAPTER 3 3D CHEMICAL DISTRIBUTION OF A FLAME RETARDANT AND SYNERGIST IN A POLYMER BLEND **67**
3.1 Introduction 67
3.2 Experimental Section 69
3.2.1 Sample Preparation 69
3.2.2 Synchrotron X-ray Tomography 70
3.2.3 3D Composition Calculation 71

3.2.4 Scanning Electron Microscopy . . . 72
3.3 Results . . . 73
3.3.1 X-ray Absorbance and Tomographic Reconstructions . . . 73
3.3.2 3D Binary Mask of SiO_2-Rich Voxels . . . 73
3.3.3 Voxel Absorbance Values . . . 75
3.3.4 3D Chemical Composition . . . 77
3.3.5 Radial Concentration about Fiber Bundles . . . 78
3.4 Conclusion . . . 81
3.5 Bibliography . . . 81

CHAPTER 4 SYNCHROTRON X-RAY TOMOGRAPHY FOR 3D CHEMICAL DIFFUSION MEASUREMENT OF A FLAME RETARDANT IN POLYSTYRENE . . . 84
4.1 Introduction . . . 84
4.2 Experimental Section . . . 84
4.3 Results and Discussion . . . 85
4.4 Bibliography . . . 86

CHAPTER 5 MEASUREMENT OF DIFFUSION CONSTANTS WITH SYNCHROTRON X-RAY TOMOGRAPHY . . . 87
5.1 Introduction . . . 87
5.2 Experimental and Computational Section . . . 88
5.2.1 Sample Preparation . . . 88
5.2.2 Tomography Reconstruction . . . 88
5.2.3 3D Image Analysis . . . 90
5.2.4 Diffusion Measurements . . . 91
5.3 Results and Discussion . . . 91
5.4 Conclusions . . . 93
5.5 Bibliography . . . 93

CHAPTER 6 CONCLUSIONS AND EXPECTATIONS . . . 95
6.1 Fourth Generation Light Sources . . . 95
6.2 Tomography . . . 96
6.2.1 Nanoscale Tomography . . . 96
6.2.2 Atom Probe Tomography . . . 96
6.2.3 Electron Tomography . . . 97
6.2.4 Focused Ion Beam Tomography . . . 97
6.2.5 High Throughput X-Ray Tomography . . . 100
6.2.6 Neutron Tomography . . . 101
6.3 New Mathematics and Computational Tools . . . 102
6.3.1 New Mathematics and Algorithms for Reconstruction and 3D Image Analysis 103
6.3.2 High Level Hierarchal Data Formats . . . 104
6.3.3 Visulization Tools . . . 104
6.3.4 Data Storage . . . 105
6.4 Remarks . . . 105
6.5 Bibliography . . . 106

APPENDIX A PERMISSION FORM . . . 110

APPENDIX B ALGORITHMS . . . 111

APPENDIX C CHAPTER 3 SUPPLEMENTAL INFORMATION . . . 136

VITA . . . 141

List of Tables

2.1 Various Tomography Experiments 6

2.2 Requirements for an X-ray Source 19

2.3 Notable/Major Synchrotrons 25

2.4 The Many Advantages of Synchrotron Radiation 33

2.5 Experimental Beamlines Found at Most Synchrotrons 34

2.6 3D Algorithm Classifications 44

3.1 Calculated Voxel Linear Attenuation Coefficients of Pure Components for 3.26 μm Voxels. 71

5.1 Results of Domain Extraction Algorithm 91

5.2 Fit Results with Error Reports 93

6.1 Visualization Software 104

List of Figures

1.1 The fiberglass component of this complex polymer blend presented unique and interesting challenges to the X-ray tomography technique. 2

1.2 Chapter 3 explores a imperfectly blended polymer and the effects that temperature has on its spatial structure. 3

1.3 Fitting a diffusion model to experimental tomography data was performed in Chapter 4. 3

2.1 Citations reported every year from 1994 to mid-2008 that include tomography and materials science in the keywords.[11] 6

2.2 General theory of the Radon transform with $p_\theta(x')$ representing the projection data generated from the line integrals for that θ. 7

2.3 Representation of the Radon transform defined parametrically rather than by the traditional cartesian system. 8

2.4 The basic experimental setup using in tomography experiments. This particular system represents a parallel beam source that passes through a sample directly to a detector where the projection data is generated. 8

2.5 The Shepp-Logan phantom depicted here is used as a standard for tomographic reconstruction algorithms. 11

2.6 The filtered backprojection algorithm with increasing amounts of projections with a)1 b) 30 c) 60 d) 90 e) 120 and f) 150 projections. 12

2.7 The results of a filtered backprojection algorithm on the Shepp-Logan phantom with 180 projections but with different filters being applied a)no filter b) the standard and widely used ramp filter c) Shepp-Logan filter 12

2.8 Hardware setup showing CCD, mirrors, scintillator, lens, sample and all motors used to control the tomography experiment. The sample stage deserves extra attention however due to its critical importance to the tomography experiment. It is shown in a blown up view. In the experiment the sample must be rotated in a precise and controlled manner. Issues such as centering, pitch and yaw must all be controlled to the highest degree possible to avoid introducing artifacts into the reconstruction. . . 13

2.9 A typical dark field image with a colorbar underneath. 13

2.10 A typical white field image with a colorbar underneath. 14

2.11 A typical raw field image with triple line plot underneath. The line plots show the counts of the CCD detector along each line as they pass through the bottom, middle and top horizontal parts of the sample. 14

2.12 Absorption field image for one angle of acquired projection data. It is calculated from Equation 2.17 and raw data that would be similar to Figure 2.11. A a line plot of the column sums of the absorption image is underneath. 15

2.13 The conversion of an absorption image to a sinogram. Every angle produces a raw projection image that is converted to an absorption image. The column sum of an absorption image for one angle gets mapped onto one row of a sinogram. Thus, each row of a sinogram corresponds to the unique angle associated with the raw projection data that helped generate the absorption data that was used for the column sum. . . 15

2.14 The traces show the calculated voxel linear attenuation coefficients for bromine, antimony and nylon. Sharp changes in the absorption of bromine and antimony are indicated at 13.474 and 30.491 keV respectively. The voxel linear attenuation coefficients were calculated by using mass attenuation coefficients from XCOM[25], their respective density and a voxel dimension length of 1 cm. 18

2.15 Donald Krest with his invention the first ever betatron. The betatron was developed at the University of Illinois and was a 20 MeV machine.[39] 21

2.16 Synchrotron light from the 70-MeV electron synchrotron at GE. The arrow points to the actual synchrotron radiation coming of the machine while it is running.[28] . . . 22

2.17 Tantalus I which was built as a test unit for advanced accelerator concepts. Research on the Tantalus ring was discontinued at the beginning of 1987 and in late 1993, it was accepted by the Smithsonian National Museum of American History for display in their exhibit on Electricity and Modern Physics Collections.[28] 24

2.18 Number of published articles each year from 1961 to 1980 that included synchrotron radiation in the title. Journal's polled to generate this data were limited to the applied journals, as to rule out any hits relating to fundamental theory research.[11] 24

2.19 Simulated power decay for one electron in a storage ring with CAMD's properties. This electron never receives a boost from RF cavities as it actually would in real life storage ring. 28

2.20 A general representation of a typical synchrotron device. (1) and (2) Electron gun and linear accelerator, typically they are quite close together, (3) Booster ring, this ring is found in newer third generation light sources and retrofitted older generations. It allows the synchrotron to operate in a "top up" mode. (4) Storage ring, (5) Various beamlines for different experiments.[71] 29

2.21 A typical day at CAMD (a second generation light source). Three injections periods occur throughout the day with the beam having an average lifetime of 12.3 hours. Current level at time of snapshot was 114.74 mA. It should be noted that even though this figure appears to be similar to Figure 2.19 , the decay in this figure is due to a imperfect vacuum leading to electron collision and loss. 30

2.22 This figure shows the effect on the magnetic flux of compounds when they are superimposed into what is called a Halbach array. The magnetic flux due to the alternating magnetization will cancel itself out below the plane and double its strength (for the ideal case) above the plane. 32

2.23 a) Undulator schematic with (1) magnets, (2) electron beam, (3) synchrotron radiation b)Wiggler schematic showing the Halback array (1), electron beam path (2), synchrotron radiation (3), parallel mirrors used to create a a resonant cavity to trap the produced photons. 33

2.24 A) A single crystal is need for the experiment. Growth of single crystals can be the hardest part. [80] B) X-ray diffraction pattern formed due to Bragg's law. [81] C) Diffraction pattern is used to calculate 3D electron density maps. D) From the electron density maps a model of the molecule can be derived. Many times at this step the experimentalist will continually refine his model by going back to the diffraction data and itierativly improve his/her model. 34

2.25 a) Block diagram of the generalized SAXS experiment. b) Simulated two dimensional image of SAXS scattering experiment of a perfect sphere. c) Simulated spectra of a typical SAXS experiment, y-axis is in arbitrary units while the x-axis is in reciprocal wavelengths. 36

2.26 Absorption vs photon energy plot for the element Cobalt. Absorption spectra vary across all the elements elements, some have multiple edges and others have none. XAS a prized technique at probing the local electronic structure of materials.[88, 89, 90] 37

2.27 A block diagram of a Deep X-ray Lithography instrument. The table/base is actually one of the most important components since the mask can not touch the resist layer alignment of the two must be highly precise and thus the base that holds the sample must have high stability.[100] The substrate is typically some for of silicon wafer, silicone wafers as a substrate is well studied and understood making it an ideal choice. There are many types of resist layers regardless of the one used its purpose is to ensure that when X-rays come in contact with it there will be a favorable reaction.[98, 99] The mask is composed of two parts, one part is transparent to X-rays and surrounds the other component. It's main function is to maintain the shape of the mask by preventing deformations.[100] The other component is the material in the mask that is the X-ray shield for the experiment 38

2.28 The base is not shown, the substrate is colored black,resist layer yellow, mask shown slightly above resist layer as blue, the transparent portion, and black, the X-ray blocking material. a) Exposure stage, X-rays are not shown but in the experiment they would hit perpendicular to the surface. b) After exposure and mask is removed from view c) Electrodeposition stage voids created in previous steps filled with desired atoms/compounds. d) Striping stage where the resist layer still left is removed this can be done chemically or via physical methods. e) The mold is now ready to be used to mass produce the device.[101] 39

2.29 Graphical representation of the diffusion process in two dimensions. The diffusional process depicted is called a point source diffusion model due to uniform distribution of particles surrounding a central point. 41

2.30 Various graphical depictions of the diffusion model for a spherical point source (Equation 2.35)over time. a) Each line represents a new time step in the model. Important features include the decreasing amplitude and increasing diffusion length parameters (a and b respectively) of the model as time evolves. b) 3D plot of the model with axes for r radius,C concentration and Dt the diffusion length. c) Graphical 3D representation of the model. 43

2.31 A digital image and its grey codes.[128] 45

2.32 a) Grey squares represent the foreground pixels and white squares the background pixels. b) Another representation of foreground/background pixels with black circles being foreground and white circles background pixels. This representation is more abstract than the first but is useful when discussing pixel connectedness and neighborhoods which is to be discussed later. 46

2.33 A binary image constructed by applying a threshold algorithm on the greyscale image from Figure 2.31.[128] 46

2.34 a) 4 nearest neighbor representation with the coordinates of the neighbors being set along the discrete horizontal and vertical axis. 47

2.35 Visual representation of the central voxel as the dark sphere in the center of the 3×3×3 grid of light gray voxels. 48

2.36 a) The 6 nearest neighbors of a central voxel. b) The 18 nearest neighbors. c) The 26 nearest neighbors of a voxels. Selection of which neighborhood to use can be of great importance in a image segmentation study. 49

2.37 Flow diagram of the algorithm and the corresponding pseudocode. The numbers indicate which portions of code are executed in the flow diagram. The numbers are also directly related to List 10. In the flow diagram portion, rectangles represent code that can only produce new data, diamonds represent code that has the power to eliminate data, trapezoids do not represent code but rather the user supplying input to the algorithm about what it's next step should be. The HDF5 table simply represents saving the data in that format. There are two other miscellaneous objects, start and delete, they do not represent code and are self-explanatory. 56

2.38 a) Randomly chosen sample domains with various radii, these are meant to represent the domains of high hexabromobenzene concentration in the sample. b) Lines are generated for every possible pair of domain connections. This is step 7 of List 10. c) The lines are then culled until the only connections that remain are the ones that represent the nearest neighbor domain pairs. Step 8 of List 10. d) The domain's radii are then set to half the distance to their respective nearest neighbor. Step 9 of List 10. 57

3.1 Chemical structure of brominated polystyrene (x = 3) used as a flame retardant (**FR**). 69

3.2 Small glass fibers will assemble into crude bundles in the extruded polymer sample. In tomography, the concentration of **FR** is measured in cubic volume elements (voxels). The volume elements can be partially occupied by SiO_2, which should cause **FR** concentration to decrease within the bundles. 69

3.3 SEM image of freeze-fractured of the sample showing the embedded fiberglass in a nylon matrix. Each glass fiber is ≈10μm in diameter [36] and ≈100μm in length. . . 73

3.4 A comparison of 2D absorbance images, (a) 13.4 keV and (b) 25 keV, with XZ-slices from 3D reconstructions, (c) 13.4 keV and (d) 25 keV, respectively. The colorbars for the 2D images (a,b) are total absorbance in the sample while the colorbars for the slices (c,d) describe absorbance in the 3.26 μm voxels. The white rectangle (c,d) outlines the cuboid subvolume chosen for analysis. The light striations (a,c) are the fiberglass bundles which have, at 13.4 keV, more X-ray absorbance than the FR/nylon blend. . 74

3.5 A $200 \times 200 \times 500$ cuboid binary mask for fiberglass based on the reconstructed volume for SiO_2. This mask is conservative with 12.58% of the cuboid voxels selected, compared to the 23.56 vol% of fiberglass in the sample based on formulation. 75

3.6 The traces show the calculated voxel linear attenuation coefficients for the pure components (see Table 3.1). The average voxel absorbance values for the two sets of voxels identified by the binary fiberglass mask: the polymer-rich voxels are represented by filled (●, ◄ , ►) symbols and the average values for the SiO_2-rich voxels by open (○, ◁, ▷) symbols, which at some X-ray energies overlap the filled symbols. 75

3.7 Histogram of air-filled voxels. The standard deviation is determined by photon counting statistics and noise introduced by the back projection reconstruction. Dashed lines show apparent absorbance prior to scale and offset corrections, Eq.3.1 76

3.8 Voxel absorbances in the sample-filled $200 \times 200 \times 500$ cuboid for the polymer-rich voxels, selected based on the binary fiberglass mask, Figure 3.5. Overall, the sample becomes more transparent at higher X-ray energies, but with the Br and Sb K-edge absorbances evident at 17 and 30.53 keV, respectively. 76

3.9 Concentrations of (a) **FR** and (b) Sb_2O_3 in the 3.26 μm voxels in the cuboid. The mean values agree with sample formulation and the relatively small standard deviations indicate good blending with nylon. 78

3.10 3D renderings of the concentrations for (a) **FR** and (b) Sb_2O_3 for the $200 \times 200 \times 500$ cuboid subvolume with 3.26 μm voxels. The colorbars indicate concentrations in vol%. 78

3.11 The radial concentration of (a) SiO_2 as a function of distance from the fiber bundle axes, sampled over 10 fiber bundles. Collision detection reduces the number of sampled voxels (b) at large r. The radial concentrations for (c) **FR** and (d) Sb_2O_3, and comparisons with a space-filling model based on the radial concentration of SiO_2. . 80

4.1 XY slice of sample composition (Z = 230) of the non-annealed sample in volume percents of BT-93. Bright spots correspond to high concentrations of the BT-93 and indicate an imperfect blending process. 86

4.2 A plot of the BT-93 concentrations a function of distance about the center of high concentration domains dispersed throughout the sample and the exponential fits applied to the data. 86

5.1 A 461×461 XY slice from the reconstructed volume for the 50 ^{o}C sample. Areas that contain voxels that have high hexabromobenzene concentration are labeled. 89

5.2 Domains of high concentration hexabromobenzene segmented out of a 461×461×161 reconstructed volume for the 50 ^{o}C sample. Segmentation was done with a simple threshold algorithm as described in Chapter 2. Various ranges of sizes and shapes are shown indicating a need to "clean up" the data before fitting. 89

5.3 a) Hexabromobenzene concentrations as a function of distance about the center of mass for the sample baked at 50 ^{o}C b) 53 ^{o}C . c) 56 ^{o}C d) 59 ^{o}C 92

5.4 All four exponential fits plotted on the same axis. Decreases in the a fit parameter follow expected trends with the exception of the 53 ^{o}C sample but the difference is minor. Diffusion coefficients get progressively larger with increasing temperature indicating that diffusion is occurring. 92

6.1 Three-dimensional atom probe tomography chemical map of a sample containing vanadium, carbon, phosphorus, magnesium, silicone, and cooper. [19]) 97

6.2 Electron tomography of virus a-d) Tomographic data slices of hydrated-virus. Some proteins have been labeled with arrowheads in image (a). e) Surface of averaged densities calculated from the volume. Certain biologically active sites are labeled and a superimposed X-ray crystallography structure is also shown. f) Surface colored regions of image (e). [23]) 98

6.3 Principles of a FIB tomography experiment. For this figure a Cu-Al metallic mulitlayer is used for the sample. a) Sample preparation with the platinum protection layer. b) Destructive milling of sample to generate 3D volume data. This is the setup used with a double beam FIB experiment c) Titling over a range of angles is required for single beam FIB experiments. b) Visualization of a system of cement particles/grains imaged with FIB tomography. [23]) 99

6.4 a) Whole, hydrated yeast cell (diameter $\approx$ 5 μm). Notable cellular structures such as the nucleus, vacuole, and vesicles are labeled. b) Tomographic slice from reconstructed volume data for two $Escherichiacoli$ bacteria, whose width is $\approx$ 0.5 μm. Both experiments acquired data at 517 eV with 45 projected images covering angles up to 180^o in 4^o intervals with exposure times of 1-3 seconds. Resolution of both experiments was $\approx$ 60 nm [28]) 100

6.5 a) Series of selected time frames from a dynamic imaging experiment of piston positions. b) A test sample used to illustrate the wavelength selectivity in neutron imaging. c) Phase contrast experiment of a spark plug at two different neutron wavelengths with the third image indicating the success of the phase contrast technique to show unseen new features in the sample. 102

Abstract

Material properties of industrial polymer blends are of great importance. X-ray tomography has been used to obtain spatial chemical information about various polymer blends. The spatial images are acquired with synchrotron X-ray tomography because of its rapidity, good spatial resolution, large field-of-view, and elemental sensitivity. The spatial absorption data acquired from X-ray tomography experiments is converted to spatial chemical information via a linear least squares fit of multi-spectral X-ray absorption data.

A fiberglass-reinforced polymer blend with a new-generation flame retardant is studied with multi-energy synchrotron X-ray tomography to assess the blend homogeneity. Relative to other composite materials, this sample is difficult to image due to low x-ray contrast between the fiberglass reinforcement and the polymer blend. To investigate chemical composition surrounding the glass fibers, new procedures were developed to find and mark the fiberglass, then assess the flame retardant distribution near the fiber.

Another polymer blending experiment using three-dimensional chemical analysis techniques to look at a polymer additive problem called blooming was done. To investigate the chemical process of blooming, new procedures are developed to assess the flame retardant distribution as a function of annealing time in the sample. With the spatial chemical distribution we fit the concentrations to a diffusion equation to each time step in the annealing process.

Finally the diffusion properties of a polymer blend composed of hexabromobenzene and o-terphenyl was studied. The diffusion properties were compared with computer simulations of the blend.

Chapter 1

Introduction

Recent advances in the field of tomography[1, 2, 3] and image analysis algorithms[4, 5, 6] have opened up an exciting new area of research. In the past, tomography was regulated to the medical fields and image analysis algorithms were simple.[7] Imaging the basic shapes of hearts, lungs, brains, etc.[8] was easily done in a clinical environment and techniques provided medical doctors with valuable information. As the field progressed it became apparent that the technique could be used to look at a wide variety of objects.

At first glance the use of tomography in chemistry seems limited, investigating the three dimensional shape of objects can be useful in many ways but in and of itself it isn't really chemistry. Work pioneered by Butler and others [9, 10, 11, 12] showed that you can take tomography data and along with some knowledge of your sample's physical properties you can generate a concentration map in three dimensions of certain compounds. Not all compounds can be imaged in this manner but a wide range of them can, this opens up a unique area of chemical research with many applications.

The primary objective of the research derived herein is to demonstrate that X-ray tomography can be applied to chemical problems, particularly issues with polymer blends and their material properties. During the time of this work, new procedures and algorithms have been developed for extracting relevant chemical information form X-ray tomography data. This work sets the foundation of for future X-ray tomography work as it applies to the fields of chemistry, materials science and polymers.

In Chapter 2, a background and introduction are given for relevant topics pertaining to the research presented in this dissertation. Topics include tomography both in theory and in experiment, conversion of tomographic data to chemical concentration, diffusion theory, synchrotrons,

and also an overview of the algorithms used for 3D image analysis.

In Chapter 3, a fiberglass-reinforced polymer blend with a new-generation flame retardant is studied with multi-energy synchrotron X-ray tomography to assess the blend homogeneity. Relative to other composite materials, this sample is difficult to image due to low x-ray contrast between the fiberglass reinforcement and the polymer blend. To investigate chemical composition surrounding the glass fibers, new procedures were developed to find and mark the fiberglass, then assess the flame retardant distribution near the fiber.

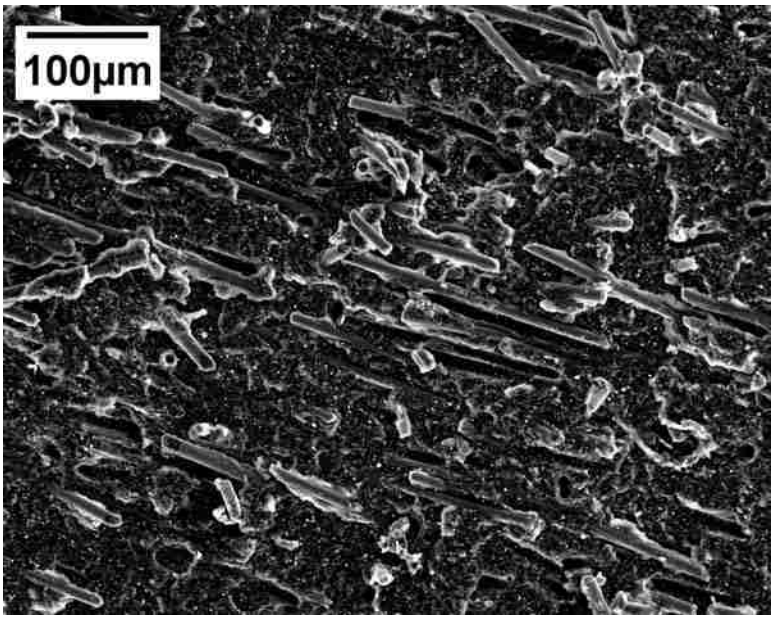


Figure 1.1: The fiberglass component of this complex polymer blend presented unique and interesting challenges to the X-ray tomography technique.

In Chapter 4, another polymer blending experiment using three-dimensional chemical analysis techniques was done to look at a polymer additive problem called blooming. To investigate the chemical process of blooming, new procedures are developed to assess the flame retardant distribution as a function of annealing time in the sample. With the spatial chemical distribution, we fit the concentrations to a diffusion equation to each time step in the annealing process. To calculate diffusion measurements around the pockets of high concentration in the sample, new methods had to be developed.

In Chapter 5, the diffusion properties of a polymer blend composed of hexabromobenzene and o-terphenyl was studied. The diffusion properties were compared with computer simulations of of the blend. The computer simulations will help us interpret the images and guide the improvement of techniques for image analysis of diffusion processes. The glass simulations make

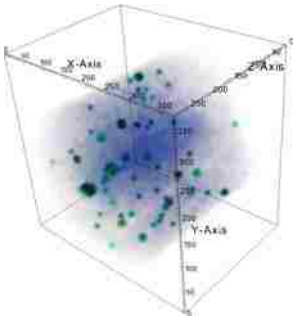


Figure 1.2: Chapter 3 explores a imperfectly blended polymer and the effects that temperature has on its spatial structure.

high demands on supercomputer resources, as do some of the 3D analyses.

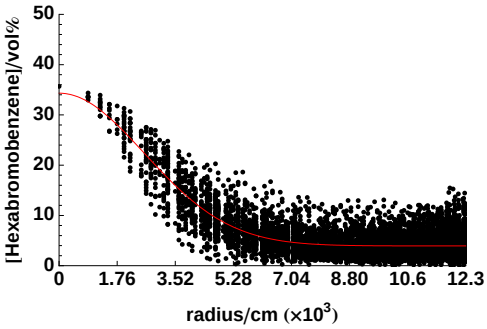


Figure 1.3: Fitting a diffusion model to experimental tomography data was performed in Chapter 4.

Chapter 6 concludes with an overall summary of my research along with comments on the current state of tomography research and where it can go.

1.1 Bibliography

[1] David N. Seidman. Three-Dimensional Atom-Probe Tomography: Advances and Applications. *Annual Review of Materials Research*, 37:127–158, 2007. Annual Review of Materials Research.

[2] F. Natterer and E.L. Ritman. Past and Future Directions in X-Ray Computed Tomography (CT). *International Journal of Imaging Systems and Technology*, 12(4):175–187, 2002.

[3] B.D. Smith. Cone-Beam Tomography - Recent Advances and a Tutorial Review. *Optical Engineering*, 29:524–534, 1990.

[4] Jundi Ding, Runing Ma, and Songcan Chen. A Scale-Based Connected Coherence Tree Algorithm for Image Segmentation. *IEEE Transactions on Image Processing*, 17:204–216, 2008.

[5] P. Stelldinger and U. Kothe. Shape Preserving Digitization of Binary Images after Blurring. *Discrete Geometry for Computer Imagery, Proceedings*, 3429:383–391, 2005.

[6] A. Ammouche, J. Riss, D. Breysse, and J Marchand. Image Analysis for the Automated Study of Microcracks in Concrete. *Cement & Concrete Composites*, 23:267–278, 2001.

[7] M. Lavaljeantet and H. Fischgold. History of Tomography. *Journal De Radiologie D Elctrologie Et De Medecine*, 56(8-9):547–549, 1975.

[8] PittsburghSupercomputer Center. The Visible Human Project at the Pittsburgh Supercomputing Center., 2003.

[9] K. Ham, H. Jin, R. I. Al-raoush, X. G. Xie, C. S. Willson, G. R. Byerly, Larry S. Simeral, M. L. Rivers, R. L. Kurtz, and L.G. Butler. Three-Dimensional Chemical Analysis with Synchrotron Tomography at Multiple X-Ray Energies: Brominated Aromatic Flame Retardant and Antimony Oxide in Polystyrene. *Chemistry of Materials*, 16(4032-42), 2004.

[10] K. L. D'Amico, H. W. Deckman, J. H. Dunsmuir, B. P. Flannery, and W.G. Roberge. X-Ray Microtomography with Monochromatic Synchrotron Radiation. *Rev. Sci. Instrum.*, 60(7, Pt. 2A):1524–6, 1989.

[11] B. Golosio, A. Simionovici, A. Somogyi, L. Lemelle, M. Chukalina, and A. Brunetti. Internal Elemental Microanalysis Combining X-Ray Fluorescence, Compton and Transmission Tomography. *Journal of Applied Physics*, 94(1):145–156, 2003.

[12] M. Chukalina, A. Simionovici, L. Lemelle, C. Rau, L. Vincze, and P. Gillet. X-Ray Fluorescence Tomography for Non-Destructive Semi-Quantitative Study of Microobjects. *Journal De Physique IV*, 104:627–630, 2003.

Chapter 2

Experimental and Computational Methods

In this chapter I will provide a background/introduction for topics that are relevant to this research. Section 2.1 is a short introduction to tomography and its current state in today's research. Section 2.2 covers the tomography experiment that shows the basic steps that a typical tomography experiment will go through. In Section 2.3 concepts behind conversion of absorption data to chemical concentrations is discussed in the third section. Section 2.5 covers a small introduction to diffusion theory and the model of diffusion chosen for this research. Synchrotrons are covered in Section 2.4; it has a brief history and a technical description of synchrotrons and its uses are discussed. In Section 2.6 the theory behind the algorithms that were developed and used in processing the multi-dimensional chemical data is covered.

2.1 Introduction to Tomography

There was not much work done with the Radon transform until Hounsfield's invention of the X-ray computed tomographic scanner. This invention earned Hounsfield (along with Allan Cormack) a Nobel prize in 1972 [1]. Today, tomography is most widely used in medical fields with computed tomography (CT), positron emission tomography (PET), and magnetic resonance imaging (MRI) being the most widely known.[2, 3, 4, 5] Tomography is also being used in archaeology, biology, geology, materials science, and many other sciences[1]. Using tomography scientists are looking at biological specimens[4], polymers[6], geological materials[7], cement[8], foams[9], medical research into artificial joints[10], and many others. A table of the most common types of tomography experiments is given in Table 2.1. Tomography research has rapidly increased over the past decade and it is only expected to increase as new uses and techniques become apparent.

A survey of the number of publications that list tomography and materials science as keywords is shown in Figure 2.1.

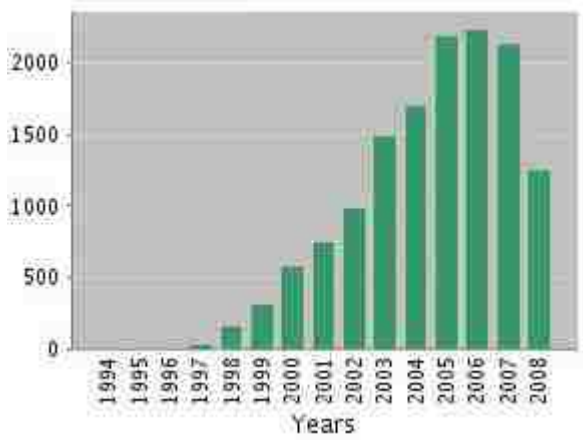


Figure 2.1: Citations reported every year from 1994 to mid-2008 that include tomography and materials science in the keywords.[11]

Depending on various factors the tomography experiment can be fast, on the order of 1 hour data collection and 1 hour data processing, to taking longer periods of time to obtain higher resolutions. Tomography dataset sizes can range from big to huge with resolutions getting down to 2 nanometers in size (electron tomography). A major advantage that tomography has over traditional techniques is that the results are readily accessible by the public. Almost everyone can look at a 3D dataset and take something of value away from it; most other techniques are limited to specialists.

Table 2.1: Various Tomography Experiments

Physical Probe	Technique
x-rays	Computed Tomography
neutrons	Neutron Tomography
knife/slicer	Microtome
Ion Beam	Focused Ion Beam Tomography
electron	Electron Tomography
nuclear magnetic resonance	Magnetic Resonance Imaging
ions	Atom Probe Tomography

2.1.1 The Radon Transform

The mathematical basis for tomographic imaging was done by Johann Radon in 1917[12, 13, 1]. Radon stated, from a purely mathematical standpoint, the solution of how to reconstruct a function from its projections (a projection at a given angle is the line integral of the image in

the direction specified by that angle). This concept is shown in Figure 2.2 and can be defined mathematically as:

$$R(m,b) \quad = \quad \int_{-\infty}^{\infty} f(x, mx+b)\, dx \tag{2.1}$$

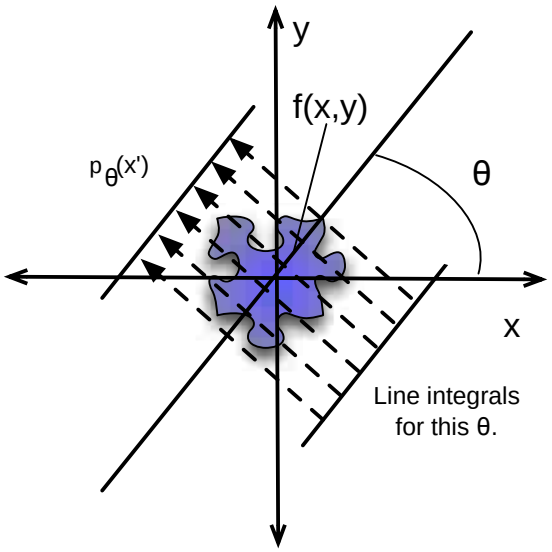


Figure 2.2: General theory of the Radon transform with $p_\theta(x')$ representing the projection data generated from the line integrals for that θ.

A more practical definition using the dirac delta function for the radon transform[14] is shown as:

$$R(m,b) \quad = \quad \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\delta(y-(mx+b))\, dxdy \tag{2.2}$$

If one considers defining a new Radon function, R(r,θ), where r is the perpendicular distance of a line to the origin and θ is the angle formed between the line and x-axis, as shown in Figure 2.3.

Obviously, all points along L can be expressed parametrically by:

$$r = xcos\theta + ysin\theta \tag{2.3}$$

With this coordinate system in mind, the Radon transform can be defined as:

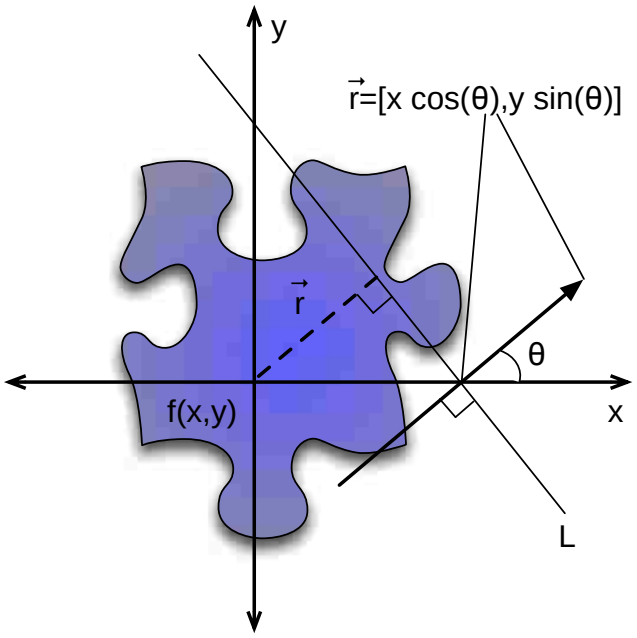


Figure 2.3: Representation of the Radon transform defined parametrically rather than by the traditional cartesian system.

$$R(r,\theta) \quad = \quad \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\delta(r-(xcos\theta+ysin\theta))\,dxdy \tag{2.4}$$

This is an important form of the Radon transform and its significance will be discussed at the end of the next section.

2.1.2 Tomographic Imaging

Consider the basic experimental layout of a parallel X-ray beam passing through and attenuating with the sample to a detector as shown in Figure 2.4.

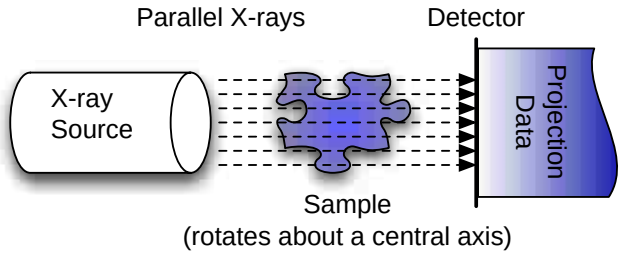


Figure 2.4: The basic experimental setup using in tomography experiments. This particular system represents a parallel beam source that passes through a sample directly to a detector where the projection data is generated.

Each parallel X-ray beam that passes through the sample can be considered a line integral since attenuation is governed by the Beer-Lambert law[15, 13]:

$$I = I_0 e^{-\mu x} \tag{2.5}$$

Which is the general form of the law, where μ is the attenuation coefficient, I_0,I is the initial and final intensities respectively, and x is equal to the path length. Equation 2.5 can rewritten with the μx being represented by a line integral:

$$\begin{aligned} I &= I_0 e^{-\int_L \mu(x,y)\, dL} && (2.6) \\ ln\left(\frac{I}{I_0}\right) &= -\int_L \mu(x,y)\, dL && (2.7) \end{aligned}$$

L represents all the line integrals for that projection at θ (in the case of our experiment in Figure 2.4 there is only one θ). We can now state that the projection data, defined as $p(r,\theta)$, to be equal to Equation 2.7:

$$\begin{aligned} p(r,\theta) &= ln\left(\frac{I}{I_0}\right) && (2.8) \\ &= -\int_L \mu(x,y)\, dL && (2.9) \\ &= -\int_{-\infty}^{\infty} \mu(x, mx+b)\, dx && (2.10) \end{aligned}$$

By using the dirac delta's sifting property we can rewrite Equation 2.9 to:

$$p(r,\theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \mu(x,y)\delta(y-(mx+b))\, dydx \tag{2.11}$$

Which is the same form as the Radon Transform of Equation 2.2. Then shifting to the coordinate system that defines all line integrals parametrically (Equation 2.3) we have:

$$p(r,\theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \mu(x,y)\delta(r-(xcos\theta + ysin\theta))\, dydx \tag{2.12}$$

Which is the same form as Equation 2.4, the Radon transform. Let's assume for the moment that there exists an inverse Radon transform ($R^{\dagger}$) that satisfies the conditions:

$$R^{\dagger}R(r,\theta)[f(x,y)] = f(x,y) \tag{2.13}$$

$$R^{\dagger}[p(r,\theta)] = f(x,y) \tag{2.14}$$

This means that by taking a infinite series of two dimensional X-ray absorption images of a sample between a range of angles the inverse Radon transform could reconstruct the full three dimensional function of the sample. Derivation of the inverse radon transform is well cited in mathematical literature.[16, 13, 17] Everything concerning the radon transform in this dissertation is concerned only with the case of parallel beams. This is due the experimental conditions of synchrotron tomography which uses only parallel beam X-rays. Many other experimental setups exist such as cone and fan beam sources.[18, 19, 20] All of these setups require different formulations of the radon transform but most are well studied.

The exact inverse Radon transform can be derived but it has limited practical applications.[13, 12] The inverse Radon transform does not produce reliable results when dealing with noisy discrete data and formulation of the inverse Radon transform that would be computationally efficient would serve much better. [14] The exact inverse Radon transform does not seem to be used anywhere but mathematical literature and textbooks.

The author is not inclined to derive the inverse Radon transform for a parallel beam system when he has not actually ever used it. Instead I will discuss a discretized inverse Radon transform that is called the filtered back-projection algorithm, which is widely used and studied.[14, 16, 12, 21] Filtered back-projection is the algorithm used to reconstruct all the tomographic data presented in this dissertation. The filtered back-projection formula can be described mathematically as:

$$f(x,y) = \mathcal{F} * \int_0^{\pi} p(xcos\theta + ysin\theta, \theta)\, d\theta \tag{2.15}$$

$$f(x_i,y_j) \approx \mathcal{F} * \Delta\theta \sum_{a=1}^{A-1} p(x_i cos\theta_a + y_j sin\theta_a, \theta_a) \tag{2.16}$$

Where $\mathcal{F}$ is the filter in spatial coordinates. Notice that since the filter is a convolution it is computationaly more efficient to transform to the frequency domain via a Fouler Transfrom where the filter convolution becomes a multiplication. Equations 2.15 and 2.16 are the general and discrete form of the filtered back-projection algorithm. The derivation of Equations 2.15 and 2.16 are

related to the Fourier slice theorem [14] or, as it is known in medical fields, the central slice theorem [13]. The central slice theorem states that a 1D Fourier transform of our projection data is the equivalent of a 2D Fourier transform on the slice image.[13, 14] With this knowledge we can apply the 2D inverse Fourier transform to our 1D Fourier transformed projection data and get a slice image. Obviously, due to the discrete nature of our projection data, interpolation in Fourier space is required but the smaller the $\Delta\theta$ of Equation 2.16 the more accurate the reconstruction will be. The exact number of required projections and the improvement that each projections adds to the reconstructed function is well studied [14]. As a general rule the closer you get to an infinite number of projections between the angles of 0 and π the more accurate your reconstruction will be. When talking about the accuracy of the reconstruction algorithms, such as the filtered back-projection, a standard image called Shepp-Logan phantom, shown in Figure 2.5, is used for testing purposes.

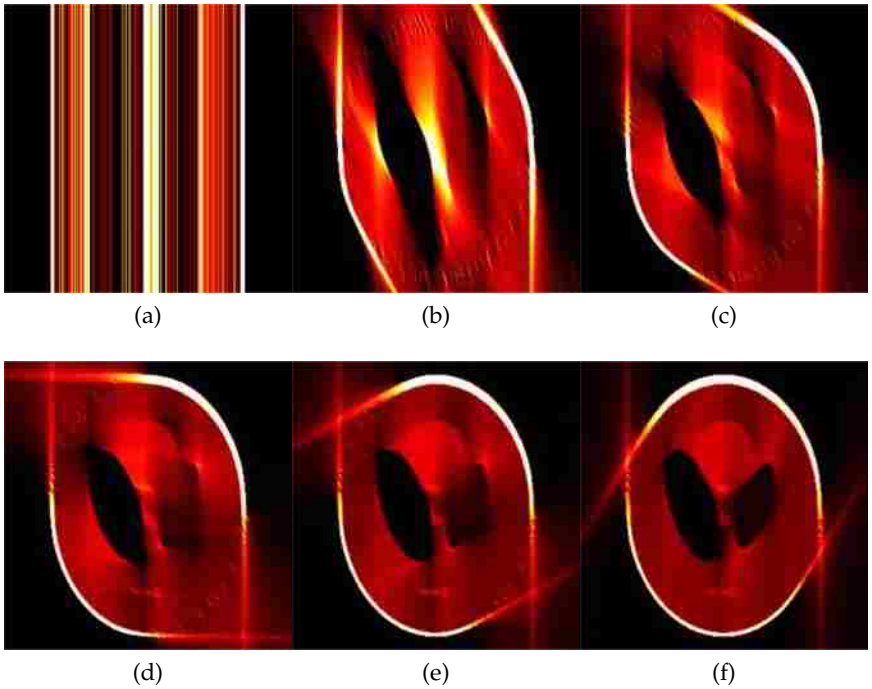
Figure 2.5: The Shepp-Logan phantom depicted here is used as a standard for tomographic reconstruction algorithms.

Figure 2.6 shows the results of a filtered back-projection algorithm applied to a Shepp-Logan phantom that has increasing numbers of projections at various θ's. Figure 2.7 shows the effect of different filters being applied to the phantom in the frequency domain.

2.2 The Tomography Experiment

The design and operation of a synchrotron tomography beam-line is covered quite well in the literature[22, 23] with many beam-lines throughout the world. Tomography presented in this dissertation was mostly done at the Center for Advanced Microstructures and Devices (CAMD) at the tomography beamline. In this section a typical tomography experiment at CAMD is described at each step and effort is taken to relate the steps to previously established theory.

Transmitted X-rays come from a source that passes through a monochromator that selects the energy range of the X-rays that will reach the sample. The X-rays are then converted to light



(a) (b) (c)

(d) (e) (f)

Figure 2.6: The filtered backprojection algorithm with increasing amounts of projections with a)1 b) 30 c) 60 d) 90 e) 120 and f) 150 projections.

(a) (b) (c)

Figure 2.7: The results of a filtered backprojection algorithm on the Shepp-Logan phantom with 180 projections but with different filters being applied a)no filter b) the standard and widely used ramp filter c) Shepp-Logan filter

with a scintillator and then imaged with a CCD camera. Figure 2.8 illustrates this setup:

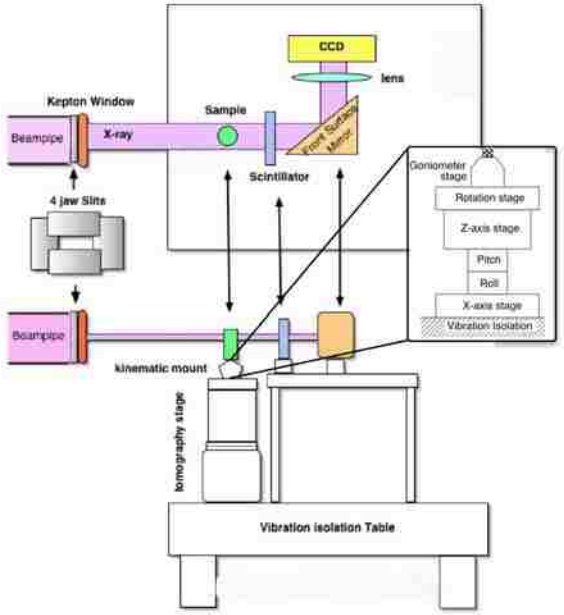


Figure 2.8: Hardware setup showing CCD, mirrors, scintillator, lens, sample and all motors used to control the tomography experiment. The sample stage deserves extra attention however due to its critical importance to the tomography experiment. It is shown in a blown up view. In the experiment the sample must be rotated in a precise and controlled manner. Issues such as centering, pitch and yaw must all be controlled to the highest degree possible to avoid introducing artifacts into the reconstruction.

2.2.1 Reconstructing the Data

Every tomography experiment begins with the acquisition of three field images which are dark, white and raw. Dark images are CCD captures when the hutch has no X-rays or any other light interfering with the scintillator, it normally sufficient to only acquire five of these images at the start of the experiment. Figure 2.9 shows a typical dark field image.

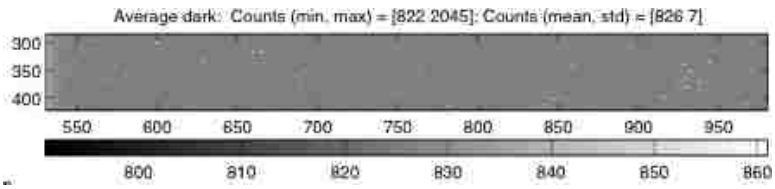


Figure 2.9: A typical dark field image with a colorbar underneath.

White field images are CCD captures of just the X-rays interacting with the scintillator with no sample in the way. Due to the nature of the X-rays being generated at CAMD (frequent beam dumps and scheduled injections) it is necessary to have at least three white field images taken for every thirty raw images in order to insure that the absorption image is scaled correctly. The frequency of white field images acquired differs from beamline to beamline.

Raw images are the CCD captures of the X-rays that have passed through the sample

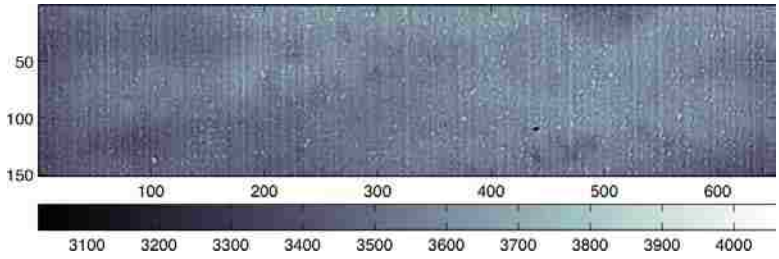


Figure 2.10: A typical white field image with a colorbar underneath.

and made it to the scintillator. Raw images will make up the bulk of images acquired during an experiment. The quality of the final data you reconstruct is directly related to the number of raw images you take at different θ's of the sample rotation[14, 13, 16] (as explained in Section 2.1.2).

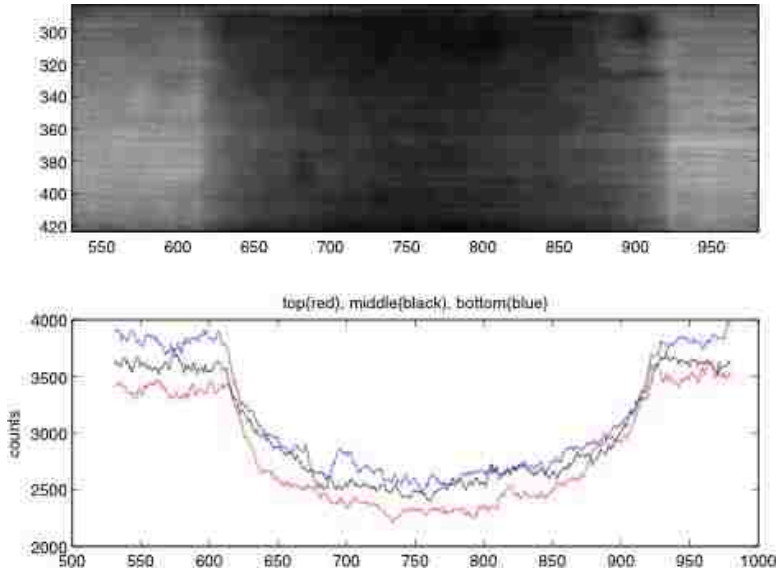


Figure 2.11: A typical raw field image with triple line plot underneath. The line plots show the counts of the CCD detector along each line as they pass through the bottom, middle and top horizontal parts of the sample.

Each raw image is converted to a absorption image with white field and average dark count correction via equation 2.17.

$$Abs = ln\left[\frac{I_{white} - I_{dark}}{I_{raw} - I_{dark}}\right] \tag{2.17}$$

Equation 2.17 is modified form of the Beer-Lambert Law (Equation 2.5) that correctly scales the raw data with white and dark field values. Figure 2.12 shows the results of applying Eq. 2.17

to Figures 2.9,2.10 and 2.11.

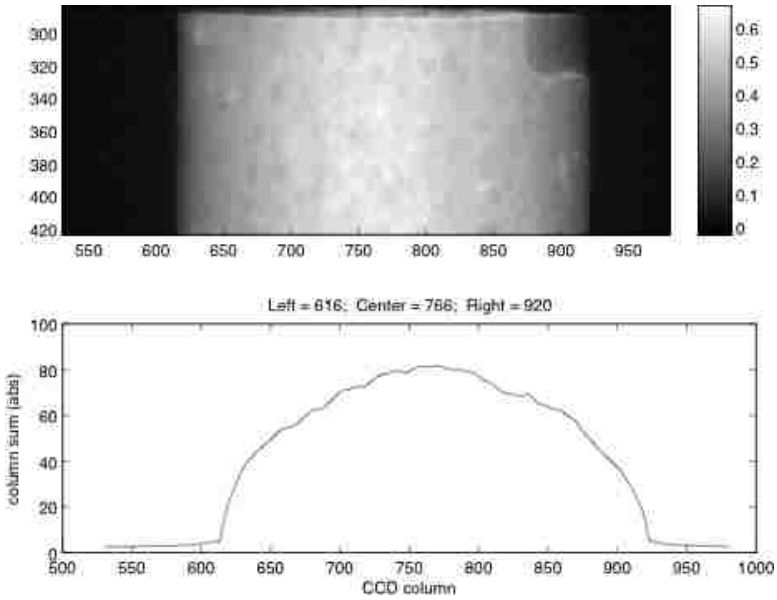


Figure 2.12: Absorption field image for one angle of acquired projection data. It is calculated from Equation 2.17 and raw data that would be similar to Figure 2.11. A a line plot of the column sums of the absorption image is underneath.

From the absorption data sinograms are constructed, efforts are made to reduce zingers and ring artifacts, and the transmission values are converted to line integrals of the absorption. [24, 6] Sinograms are constructed by taking each raw data field image and performing a column sum, this column sum becomes one row in the sinogram. This process is shown visually in Figure 2.13.

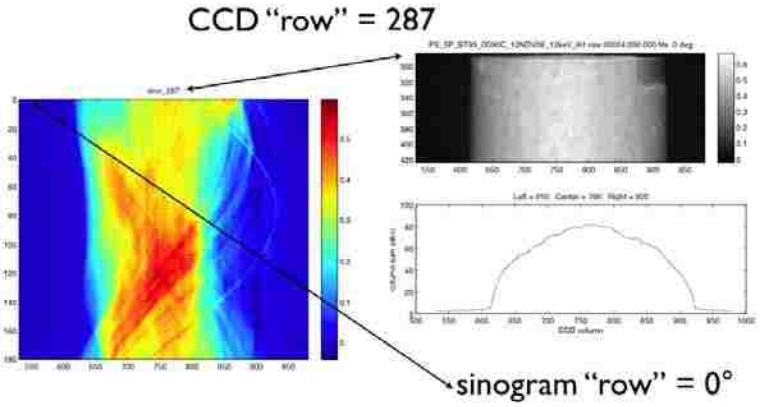


Figure 2.13: The conversion of an absorption image to a sinogram. Every angle produces a raw projection image that is converted to an absorption image. The column sum of an absorption image for one angle gets mapped onto one row of a sinogram. Thus, each row of a sinogram corresponds to the unique angle associated with the raw projection data that helped generate the absorption data that was used for the column sum.

The column sum is the projection data function $p_\theta(x')$ shown in Figure 2.2 and the sinogram which collects all the θ's is $p(r, \theta)$. If we look back to Equation 2.8, we see the mathematical

expression relating our absorption image and projection data. We know from Section 2.1.2 that the filtered back-projection algorithm can take our sinogram and reconstruct one slice image of our sample.

For the reconstructions at CAMD we use the function iradon from the Matlab Image Processing Toolbox. It is a straightforward implementation of the filtered back-projection algorithm discussed at the end of Section 2.1.2. The function can change the interpolation method method used in the frequency domain and has access to several of the most commonly used filters such as ramp, Shepp-Logan, Hamming, Cosine, etc.. With Matlab the ability to define your own interpolation method if desired and the Signal Processing Toolbox can allow the user to create novel new filters to be used with the iradon function. However, the core part of the iradon function, namely the part that implements Equation 2.16, can not be modified which, depending on the user, may be undesirable.

The slices were assembled into a single reconstructed volume of voxel elements, each voxel contains the value of the voxel linear attenuation coefficient. Voxel linear attenuation coefficients are slightly different from the normal linear attenuation coefficients, where the linear attenuation coefficient has units of cm^{-1} voxel linear attenuation coefficients are unit-less. The next section will go into the explanation of voxel linear attenuation coefficients in more detail.

2.3 Converting Absorptions to Chemical Concentrations

Determining the 3D absorption data of a sample as shown in previous sections is typically the end result for most tomographers from the experimental standpoint. For X-ray tomography to work as chemical analysis tool we need to be able to convert the 3D absorption values that we generate into chemical concentrations. Details about the chemical concentration of samples in spatial cartesian coordinates could be used to problems that are unsolvable or difficult to study with other standard techniques.

Conversion of absorption data to chemical concentrations is done by using the Beer-Lambert Law and multi-spectral X-ray tomography. This section will describe the overall algorithm and the theory behind it, that is used to transform absorption data to chemical concentrations.

Simplifying the Beer-Lambert Law (Equation 2.5)[15] we get:

$$
\begin{aligned}
I &= I_0 e^{-\mu x \rho} \\
ln\left(\frac{I}{I_0}\right) &= -\mu x \rho \\
Abs &= -ln\left(\frac{I}{I_0}\right) \\
Abs &= \mu x \rho
\end{aligned} \tag{2.18}
$$

Where Abs is the absorption, μ is the linear attenuation coefficient, ρ is the density of the sample and x the path length the radiation travels. We now see that absorption is defined the product of the linear attenuation coefficient with the path length of the sample, which yields η a unit-less quantity. Most fields of research call this quantity absorption but it is useful in tomography to call it the more descriptive name of voxel linear attenuation coefficient. The formal definition of a voxel linear attenuation coefficient (η) is:

$$
\eta = \left(\frac{\mu}{\rho}\right) \times \rho \times \chi \tag{2.19}
$$

Where $\frac{\mu}{\rho}$ is the mass attenuation coefficient $\left(\frac{cm^2}{g}\right)$, ρ $\left(\frac{g}{cm^3}\right)$ and χ (cm) is the voxel dimension length. The explicit inclusion of the ρ's in the equation may seem pointless since it obviously cancels out of the equation but in practice mass attenuation coefficients are defined in the manner presented.

For the rest of this section consider that a simple, solid mixture of bromine, antimony, and nylon has been made. This sample will be used to help illustrate the techniques used that will produce the chemical concentration data from the absorption data.

For a voxel that is completely filled with the sample, (i.e., no cracks, voids, or sample surface voxels) the absorption of the sample at that voxel must be equal to:

$$
Abs^{sample} = \left([Br] \times Abs_E^{Br}\right) + \left([Sb] \times Abs_E^{Sb}\right) + \left([nylon] \times Abs_E^{nylon}\right) \tag{2.20}
$$

Where [X] are volume percent values for that particular sample. We know that the absorption of a voxel is equal to the voxel linear attenuation coefficient. Voxel linear attention values for pure elements and compounds can be calculated with Equation 2.19 using mass attenuation coef-

ficients, sample density's and voxel dimension lengths. Density and voxel dimension lengths are easy to acquire and mass attenuation coefficients of elements and compounds are readily available from internet databases such as XCOM from NIST[25]. Figure 2.14 shows the calculated voxel linear attenuation coefficients vs photon energy for all three components of our sample.

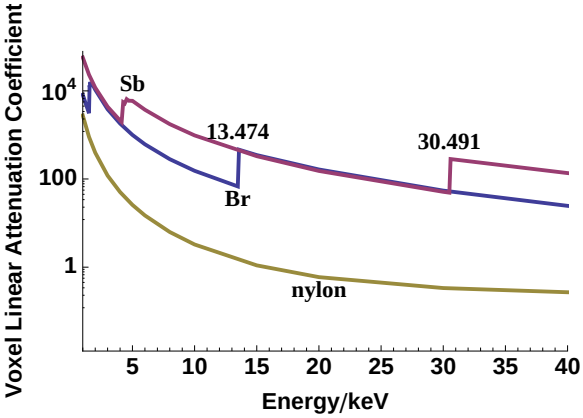


Figure 2.14: The traces show the calculated voxel linear attenuation coefficients for bromine, antimony and nylon. Sharp changes in the absorption of bromine and antimony are indicated at 13.474 and 30.491 keV respectively. The voxel linear attenuation coefficients were calculated by using mass attenuation coefficients from XCOM[25], their respective density and a voxel dimension length of 1 cm.

With the newly calculated absorptions for the pure components we will have an equation with four known and three unknown variables for our 3 component sample. Clearly, this is not enough solve for the concentration data that we desire. However, heavy elements (typically Z$\geq$ 35) produce sharp absorption changes at certain energies. Both bromine and antimony have an edges that visible in Figure 2.14. By taking advantage of the edges present in heavy elements we can image a sample with X-rays at energies just above and below the edge for the element in the sample. This will give us two equations for every component that has a edge in our sample. This will enable us to calculate the concentration of that component. In our three component sample nylon does not have an edge and so its concentration can not be directly calculated. We can indirectly calculate it via a simple subtraction:

$$Abs^{sample} = \left([Br] \times Abs_E^{Br}\right) + \left([Sb] \times Abs_E^{Sb}\right) + \left((1 - [Br] + [Sb]) \times Abs_E^{nylon}\right) \quad (2.21)$$

By selecting the X-ray energy values above and below the edges we can calculate the volume percents for each component by expanding Eq. 2.21 across the X-ray energies. This yields a over-determined set of linear equations. To solve of this system of linear equations a constrained linear least-squares fit of the data to the concentration variables will be required. This is easily implemented with functions from the Optimization toolbox for Matlab. The linear least-squares fit solves the above system of linear equation with a minimization of difference between the experimental X-ray voxel absorptions and the calculated absorptions.

$$\underset{Bromine,\,Antimony}{\overset{minimization}{}} \left\| \mathbf{Abs}_{xyz}^{sample} - \mathbf{Abs}_{xyz}^{calc} \right\|^2 \tag{2.22}$$

Expanding Equation 2.22:

$$\underset{Bromine,\,Antimony}{\overset{minimization}{}} \left\| \begin{bmatrix} A_{13.4}^{Br} - A_{13.4}^{nylon} & A_{13.4}^{Sb} - A_{13.4}^{nylon} \\ A_{13.5}^{Br} - A_{13.5}^{nylon} & A_{13.5}^{Sb} - A_{13.5}^{nylon} \\ A_{30.43}^{Br} - A_{30.43}^{nylon} & A_{30.43}^{Sb} - A_{30.43}^{nylon} \\ A_{30.53}^{Br} - A_{30.53}^{nylon} & A_{30.53}^{Sb} - A_{30.53}^{nylon} \end{bmatrix} \begin{bmatrix} [Br]_{xyz} \\ [Sb]_{xyz} \end{bmatrix} - \begin{bmatrix} A_{xyz,13.4}^{sample} - A_{13.4}^{nylon} \\ A_{xyz,13.5}^{sample} - A_{13.5}^{nylon} \\ A_{xyz,30.43}^{sample} - A_{30.43}^{nylon} \\ A_{xyz,30.53}^{sample} - A_{30.53}^{nylon} \end{bmatrix} \right\|^2 \tag{2.23}$$

In conclusion by using multi-energy imaging and Beer-Lambert Law, absorption values are converted to volume percents for each voxel. This is done via a constrained linear least square fit; constraints are initially chosen to limit calculated volume percents to physically allowed values.

2.4 Synchrotrons

In the previous sections I have laid out the some of the theory and experimental concepts in the field X-ray tomography. What I did not discuss were the requirements of the physical instrumentation that will provide the X-rays to the experimental system. Now that the tomography experiment has been explained the major requirements can be understood fully. Table 2.2 lists the major requirements and the reasons behind the requirements.

Table 2.2: Requirements for an X-ray Source

Requirement	Reason
High X-ray brightness/flux	Allows for thicker samples and faster experiments
Highly Collimated X-rays	Poor collimation violates the parallel beam model for reconstruction
Wide energy range	Wide energy ranges allow us to analyze at more elements
Selectable X-ray energy	Can't calculate chemical concentrations with polychromatic X-rays

These requirements are pretty steep by conventional laboratory X-ray source standards. However, synchrotron light sources meet all of these requirements and more. All of the tomography data presented in this research was generated at a synchrotron facility. In the following sections I will present an overview of the history behind synchrotrons, the synchrotron device

itself, how it constructed and works, and the applications of its primary product synchrotron radiation/light. This overview of synchrotrons will show how it uniquely suited to the purposes of chemical analysis with X-ray tomography.

2.4.1 A Brief History of Synchrotrons

The earliest research into synchrotron radiation dates back to 1898, when Linard derived an expression from classical electrodynamics for the instantaneous total power radiated by an relativistically accelerated charged particle in circular orbit the results were first published in English by Heaviside in 1903 in Nature. [26] Linard's calculation showed that power generated from a relativistic electron in circular orbit was proportional to:

$$P = \frac{\left(\frac{E}{m_0 c^2}\right)^2}{r^2} \tag{2.24}$$

where E is particle energy, m_0 is the rest mass, and r is the radius of the trajectory. By 1907 the power of a relativistic electron in circular orbit in a magnetic field had been solved by Schott.[27] Schott believed that a intense study of the radiation produced would explain the discrete nature of atomic spectra.[28] He eventually produced a 362 page exhaustive book on the subject in 1912 [29] but his use of classical physics, in particularly a planetary orbit model[1], doomed his attempts to failure. Despite the research and progress not much research was done with synchrotron radiation for some thirty years. In 1949 Schwinger had worked out the theoretical framework for a classical theory of radiation of relativistic electrons in magnetic fields that predicted many of the effects we see in modern day synchrotron radiation. [30, 31, 32] Sokolov and Tersov later showed that Schwinger work was indeed confirmed by a quantum mechanical treatment of his theory. [33, 34, 35]

During the hiatus on research involving radiation energy losses from relativistic atoms betatrons received much attention.[28] Betatrons are also an electron particle accelerator that have two coils, the primary coil produces a AC current and the secondary coil is a torus-shaped vacuum tube. When the electrons leave the primary coil they are accelerated at the center of the secondary coil where a a vacuum tube is changing the magnetic field and producing an electrical field. By 1945 three separate betatrons had been built. The first was built by Donald Kerst at the Univeristy of Illinois[2] and was a 20 MeV machine [36]. Figure 2.15 shows Donald Krest with the first con-

[1] Planetary orbit model is where electrons orbit the nucleus of an atom in circles similar to planets orbiting the sun.

[2] Krest decided the name of his invention based on a departmental contest. A variety of names was suggested for it and Krest eventually settled on betatron but names suggested by colleges were: rheotron, inductron, and Ausseror-

structed betatron. The second and third were both developed by GE with 20 and 100 MeV max energy levels. [37] The betatron however was ultimately not meant to be, energy loss from the radiating electrons would put an upper limit on the energy obtainable by betatrons to be $\approx$ 500 MeV. [38]

Figure 2.15: Donald Krest with his invention the first ever betatron. The betatron was developed at the University of Illinois and was a 20 MeV machine.[39]

Belwett, a researcher at GE, believed that that the radiation losses predicted by Schwinger would be able to be seen in these new accelerators.[40] Schwinger's calculations stated that the radiation spectrum for the 100-MeV betatron should show up in the infared and visible spectrum. Blewett ran an exhaustive set of experiments in search of what he called "Schwinger Radiation". Unfortunately, Belwett either miscalculated or made a mistake and always looked in the radio/microwave region of the spectrum. He also used opaque tubes which limited the visibility of the chamber which would prove hindering. Nevertheless, the experimental data collected by Belwett was roughly in accordance with expected energy losses predicted by Schwinger but there was never a confirmation of radiation being produced since he was looking in the wrong spectrum range.[40] Ultimately, Belwett believed that Schwinger's theory was the correct but had no experimental data to prove it.

During the same time a interesting effect had been observed by experimentalists concerning relativistic particles. The observable mass of particles increase as their speeds approach the relativistic range, this was well known and explained by relativistic mechanics. The interesting part was that due to the heavier mass the particles had acquired they were getting to the RF electrodes at non peak times and thus were not getting the maximum amount of acceleration possible. This ultimately limits the maximum possible particle energy and is undesirable. The problem was that the RF frequency was set at constant value, both McMillan and Veksler suggested, independently, modulating the frequency of the RF voltage to match the changing particles parameters.[41, 42]

dentlichhochgeschwindigkeitelektronenentwickelndenschwerarbeitsbeigollitron (I'm really not making that up! Check the citation!)[28]

This would allow the experimentalist to keep his particle stream together and at the same acceleration. This is commonly called the phase-stability principle and is still in wide use at all synchrotrons today today. In 1947 a researcher at GE, Pollock, wanted to test McMillian and Vekslers proposal and GE allowed him to construct a 70 MeV accelerator for this purpose.[43, 44, 45].

Proof of synchrotron radiation was acquired by poor safety standards and dumb luck. At GE Pollock was using a partially shielded 70 MeV accelerator with a transparent torus electron tube, the shielding had been removed temporarily to perform a calibration.[28, 46] A lab technician, Floyd Haber, was looking for sparking in the electron tube, he did not see sparking, instead Floyd saw a bright arc/beam of synchrotron light that was visible in daylight. [28] This discovery generated a lot of discussion within the lab over the source of this light. Langmuir and Elder claimed that this was clearly "Blewett Radiation" [47, 48] but most researchers believed that a gas discharge or Cerenkov effect was responsible. Over the next few years GE characterized and studied the properties of the synchrotron radiation.[46] Figure 2.16 shows the original 70 MeV betatron in which synchrotron light was first discovered. Blewett was ultimately correct but never received the credit he probably entitled to for his earlier work[3].

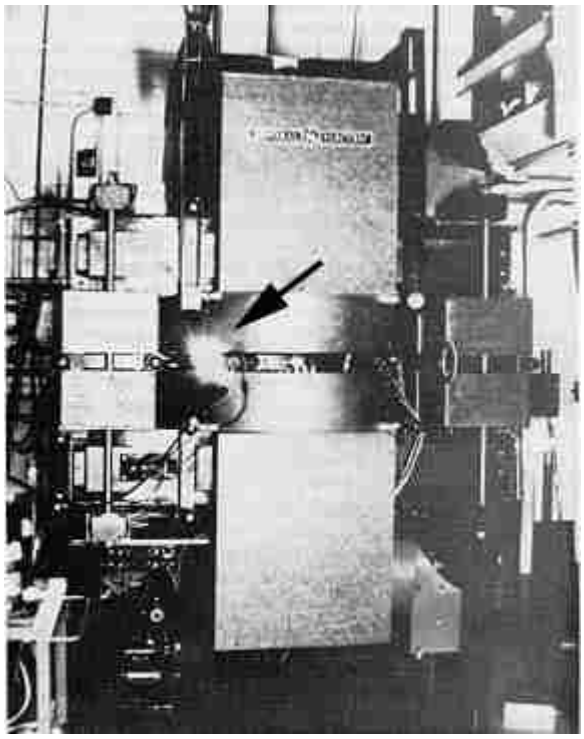


Figure 2.16: Synchrotron light from the 70-MeV electron synchrotron at GE. The arrow points to the actual synchrotron radiation coming of the machine while it is running.[28]

By 1960 electron synchrotrons were available and experiments on them were being done.[52, 53, 54] Despite the advantages of synchrotron radiation that were detailed by the scientists [55, 56,

[3] Belwett has since contributed greatly to synchrotron research [49, 50, 51, 50] and written several papers concerning the history of synchrotron radiation.[37]

57] little progress towards using the radiation generated in electron synchrotrons for experiments was achieved. Largely this was still due to a pervasive attitude among the subatomic physics research community that thought of the energy loss by synchrotrons as a annoyance one had to put up with to have the advantages that the synchrotrons offered them, namely high energy electrons.[58] In 1961 the National Institute of Standards and Technology (at the time it was called the National Bureau of Standards) added beamline to their 180-MeV electron synchrotron and allowed research to be performed on that beamline. [28] This would usher in the synchrotron as a tool for a wide variety of researchers not just high energy particle physicists. The new instrument was called the Synchrotron Ultraviolet Radiation Facility (or SURF).

As more synchrotron facilities added beamlines to access the energy loss a new name for these ”first” generation synchrotron radiation sources was born. The new synchrotron radiation beamlines were called parasitic facilities, these synchrotrons often ran at low beam energies and parasitic operations on synchrotrons that ran at low beam energies meant a severely limited output of synchrotron radiation.[28] This lead to a desire for synchrotrons built for the sole purpose of generating radiation for experiments but the user community was still small and couldn't justify the cost of such a venture. Most often older synchrotrons that had outlived their usefulness to the physicists that had built them would be eagerly taken over by synchrotron radiation experimentalists.[52] Shortly after SURF was up and running synchrotrons being used for their radiation began to appear in America, Europe and Asia.[37]

Storage rings were the first big upgrade that synchrotrons underwent. Storage rings were vastly superior to the previous parasitic facilities that were often limited by the high energy particle physicists requirements. The beam when in a storage ring continuously orbits at a fixed energy for many hours, this leads to much less downtime, higher beam currents, higher flux, better beam stability and other improvements. The incorporation of storage rings into the synchrotron facilities marks a clear distinction between them and first generation/parsitic sources and are called second generation light sources to reflect this difference. The Midwest Universities Research Association (MURA) funded a project that lead to the development of the first storage ring.[59] This project, ultimately, lead to a template for synchrotron radiation facilities that is still in use today. MURA was actually dissolved before the storage ring was actually constructed but USAF Office of Scientific Research and the University of Wisconsin stepped in and completed the storage ring, the facility was known as Tantalus I.[28] In 1968 the first experiment at Tantalus I was performed using X-rays acquired from a storage ring in a synchrotron. Eventually Tantalus I had installed ten beamlines with monochromators and supported multi-users and experiments. [59] Figure 2.17 shows a picture of the original machine.

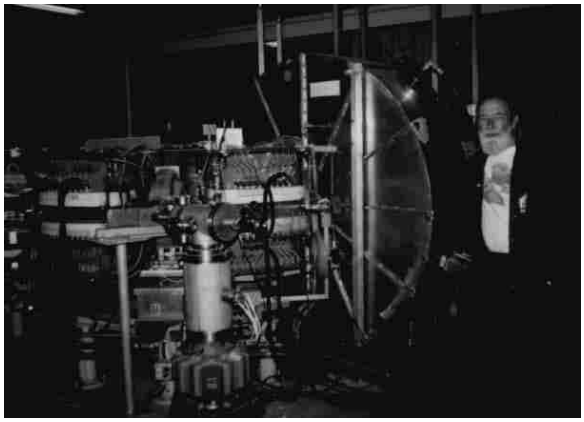


Figure 2.17: Tantalus I which was built as a test unit for advanced accelerator concepts. Research on the Tantalus ring was discontinued at the beginning of 1987 and in late 1993, it was accepted by the Smithsonian National Museum of American History for display in their exhibit on Electricity and Modern Physics Collections.[28]

With the success of Tantalus I, the following years saw an explosion of synchrotrons being built for the sole purpose of generating X-rays for users. By 1980 use of synchrotron radiation as a applied research tool began to grow and publications including synchrotron radiation in the title rose dramatically, Figure 2.18.

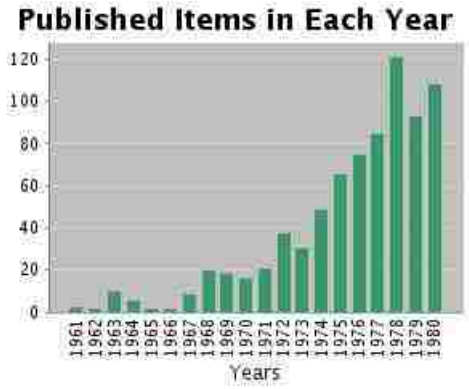


Figure 2.18: Number of published articles each year from 1961 to 1980 that included synchrotron radiation in the title. Journal's polled to generate this data were limited to the applied journals, as to rule out any hits relating to fundamental theory research.[11]

Today many third generation synchrotron radiation facilities are being/have been constructed. The term third generation typically indicates a synchrotron radiation source that provides significant lower beam emittance and an increase in the brightness of the radiation produced.[60, 61, 62] This is accomplished with the use of insertion devices such as wigglers and undulators.[63] Undulators and wigglers are discussed in more detail in a later section.

Currently there are too many synchrotrons to list all of them in a readable format; with that in mind Table 2.3 attempts to list the major synchrotrons currently in operation with their location and electron beam energy. The data for Table 2.3 is complied from a variety of sources.[64, 65, 66]

Table 2.3: Notable/Major Synchrotrons

Name	Location	Energy (GeV)
Australian Synchrotron	Melbourne, Australia	3
LNLS	Campinas, Brazil	1.35
Soleil	Paris, France	3
ESRF	Grenoble, France	6
ELETTRA	Trieste, Italy	2-2.4
DLS	Oxfordshire, England	3
BESSY	Desy, Germany	4.5
SPring-8	Riken, Japan	8
CLS	Candian Light Source	2.9
Tevatron	Fermi National Accelerator Laboratory, USA	1,000
ALS	Lawrence Berkeley Lab., USA	1.5-1.9
NSLS I	Brookhaven National Lab., USA	0.8
NSLS II	Brookhaven National Lab., USA	2.5-2.8
SPEAR (SLAC/SSRL)	Stanford Linear Accelerator Center, USA	3
SURF III	Gaithersburg, USA	0.386
DFELL	Durham, USA	1-1.3
CAMD	Baton Rouge, USA	1.2-1.5
APS	Argonne National Lab.,USA	7
Aladdin	Stoughton, USA	0.8-1.0
CHESS	Cornell High Energy Synchrotron Center	5.5

2.4.2 The Physics of Synchrotron Radiation

This section is meant to give an overall introduction to the physics that govern synchrotron radiation. It is not meant to be an exhaustive treatment on the subject but enough to give the reader a understanding of the process. With this basic knowledge a understanding of why this radiation is so useful to researchers can begin to form and why synchrotrons are built the way they are.

Synchrotron radiation is generated whenever a electron at relativistic speeds is forced to change direction. Obviously, there is energy loss whenever the electron is forced to change direction and Equation 2.25[67] gives the power radiating from an electron in the synchrotron accelerator.

$$P = \frac{2\kappa e^2 \gamma^2 v^4}{3c^3 r^2} \tag{2.25}$$

Where γ is the Lorentz factor:

$$\gamma = \frac{1}{\sqrt{1 - \frac{v^2}{c^2}}} \tag{2.26}$$

κ is Coulomb's constant , e is the elementary charge constant, v is the velocity, c the speed of light constant, and r is the radius of the accelerator ring. It should be noted here that the angular distribution of synchrotron radiation is distorted due to the doppler effect at the relativistic speeds of the electrons. The distortion causes the radiation to have a pronounced "gun" effect: it is directed ahead in parallel to the motion of an electron and is in the shape of a slender cone.[28] This means that synchrotron radiation will have a high collimation at creation.

Take for example the Center for Advanced Microstructures and Devices has a energy range of 1.2-1.5 GeV and a radius of 2.928 meters. Thus at these parameters (we will use the high energy range of 1.5 GeV) the power for a single electron is:

$$\begin{aligned} P &\approx \frac{2\kappa e^2 \gamma^4 c}{3r^2} \\ &\approx \frac{2(8.987 \times 10^9 \frac{Nm^2}{C^2})(1.6 \times 10^{-19} C)^2 (2935.43)^4 (3 \times 10^8 \frac{m}{s})}{3(2.928m)^2} \\ P &\approx 3.99335 \times 10^{-7} \frac{watts}{electron} \end{aligned} \tag{2.27}$$

This value may not seem to be significant but remember this is for one electron, this is a huge number considering the number of electrons that will be typically injected into a an accelerator. A minor modification of Equation 2.25 can allow us to calculate the energy loss per orbit in a

synchrotron ring as shown below:

$$\Delta E = P\left(\frac{2\pi r}{v}\right) = \frac{4\pi\kappa e^2\gamma^4\beta^3}{3r} \tag{2.28}$$

Where :

$$\beta = \frac{v}{c} \tag{2.29}$$

If it were not for the RF cavities placed along the storage ring the energy loss from just the electrons orbiting in the synchrotron would be a major problem.[68] Notice that the radiation loss is inversely proportional to the radius of the synchrotron. Since the radius is set in stone once the ring is constructed the selection of the size of the ring needs to be carefully considered. If energy loss will be a large issue then building the ring with the largest radius possible will help minimize the inherent energy loss found synchrotrons.

Taking the CAMD beamline (1.5 GeV) as our example once again and setting the β variable to 1 (the assumption that $v \approx c$ is valid):

$$\begin{aligned} P &\approx \frac{4\pi\kappa e^2\gamma^4\beta^3}{3r} \\ &\approx \frac{2(8.987\times10^9\frac{Nm^2}{C^2})(1.6\times10^{-19}C)^2(2935.43)^4\beta^3}{3(2.928m)} \\ P &\approx 1.529\times10^{-4}\frac{GeV}{electron\ orbit} \end{aligned} \tag{2.30}$$

Thus, beam energy will eventually reach a point where it can no longer maintain the beam path in a orbit. A simulated decay of one electron in CAMD's synchrotron is shown in Figure 2.19.

We now know two major things about synchrotron radiation: it's highly collimated and the electrons that produce it lose energy rapidly. The high collimation of the X-rays is actually a benefit but the energy loss per orbit must be addressed. The energy loss is compensated by placing RF cavities along the storage ring and they will be discussed more in the next section. The following section discusses the mechanical aspects of what composes synchrotrons.

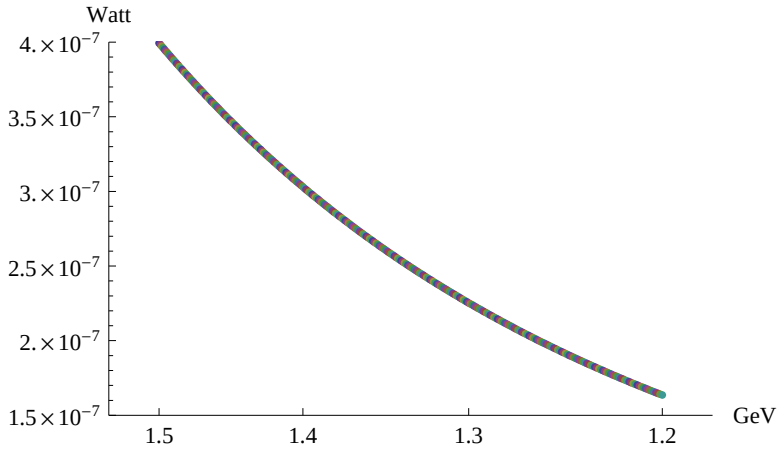


Figure 2.19: Simulated power decay for one electron in a storage ring with CAMD's properties. This electron never receives a boost from RF cavities as it actually would in real life storage ring.

2.4.3 How a Synchrotron Is Built

A synchrotron is a cyclic particle accelerator, it is a variant of a cyclotron but whereas the cyclotron keeps a constant magnetic and electric field the synchrotron does not.[68] The magnetic field and electric field are used in conjunction to produce high intensity light/radiation, the magnetic field ensures that the particle stream moves in the correct manner/direction and the electric field accelerates the particle stream of electrons to relativistic speeds. The first idea for the design of a electron based synchrotron ever built can is sometimes attributed to Luis Alvarez[4] but it was actually Ed McMillan.[37] For clarity, all discussion in the rest of this chapter is concerned with electron based synchrotrons.

When synchrotrons are described in theory the electron particle stream occupies a vacuum container that is shaped like a large thin torus.[69] However, in practice, it is easier to use straight vacuum containers that are joined with bending magnets on small curved sections. [70] This ultimately leads to the inner ring having the shape of a polygon that approximates a torus. Modern third generation synchrotrons can be described as having five major parts:

1. Electron Gun
2. Linear Accelerator
3. Booster Ring

[4] Alvarez has written that he had been trying to think up of a linear electron accelerator and McMillan came up with better version.[37] The false attribution of Alvarez as the father of electron based synchrotrons is most likely due to the fame surrounding Alvarez and his work on the first proton based synchrotron

4. Storage Ring

5. Beamline

A generalized schematic of a modern synchrotron is shown in Figure 2.20.

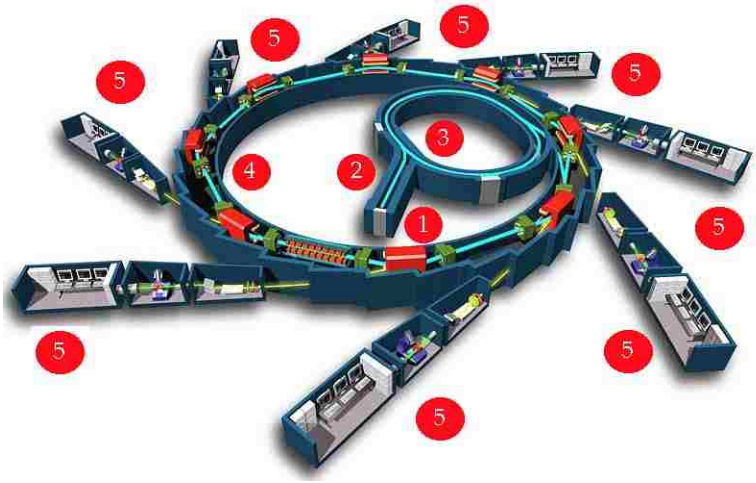


Figure 2.20: A general representation of a typical synchrotron device. (1) and (2) Electron gun and linear accelerator, typically they are quite close together, (3) Booster ring, this ring is found in newer third generation light sources and retrofitted older generations. It allows the synchrotron to operate in a "top up" mode. (4) Storage ring, (5) Various beamlines for different experiments.[71]

The electron gun provides the electrons that are to be accelerated in the ring. There are many type of electron guns that can be used for synchrotrons thermionic emission, photocathode, RF sources and plasma sources are just a few of them.[15] Thermionic emission is popular and widely used, a thermionic emission electron gun works on the same principles as your television's cathode ray tubes. The cathode is typically a tungsten-oxide disk that gets electrically heated to the point of thermionic emission; this occurs at 1000^{o}C. A anode, sometimes called a screen, located near the focusing point of cathode gives short, very strong positive charge pulses which pull the electrons away from the disk and into the linear accelerator.

There are three basic types of linear accelerators: electrostatic, induction, and radio frequency (RF) models. The RF uses microwave radio frequency fields to accelerate the electrons to relativistic speeds and exists in different types.[72] Induction methods use a pulsed voltage around on magnetic cores to produce a electric field that propels the electrons to relativistic speeds. Finally, electrostatic methods accelerate the electrons by using a electric field between two separate potentials.[72] The chamber of all linear accelerators, no matter what type, is under a vacuum of 10^{-11} torr or lower. This is to prevent the electron particle stream from colliding with anything

that would slow them down.

The booster ring is the major advancement of third generation light sources over previous generation light sources. The booster ring provides energy from microwaves generated in a radio frequency cavity.[70] The booster ring will typically start an injection of electrons and ramp up their energy to the desired value by using RF cavities. RF cavities are devices placed along the storage and booster ring that will provide ”replacement” energy for the energy that has been radiated away due to Equation 2.28. Eventually though due to imperfect vacuums the ring will require a re-injection in order to bring the electron particle stream back to the desired electron density.[73] Once the desired beam energy has been attained the electron beam with transfered to the storage ring. [74, 75, 76] By using the separate smaller booster ring inside the storage ring the storage ring never has to go down for injections. Figure 2.21 shows a snapshot of beam schedule for 24 hours at CAMD, it undergoes periodic re-injections. If a booster ring is available then periodic ”boosting” from the booster ring keeps the storage ring in the so called ”top-up” mode where the current of the storage remains approximately constant. [69]

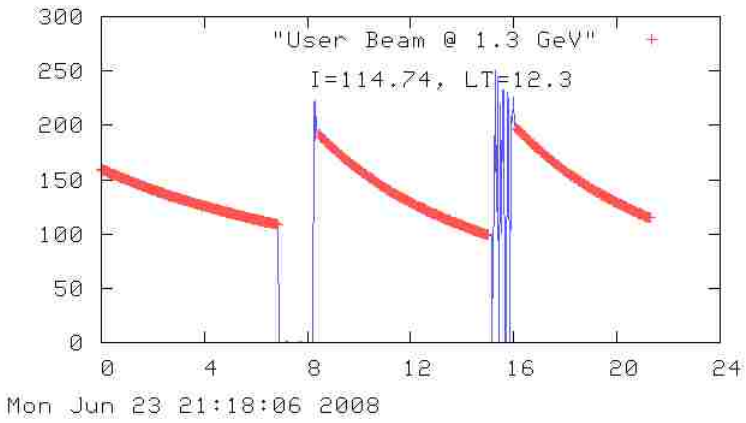


Figure 2.21: A typical day at CAMD (a second generation light source). Three injections periods occur throughout the day with the beam having an average lifetime of 12.3 hours. Current level at time of snapshot was 114.74 mA. It should be noted that even though this figure appears to be similar to Figure 2.19 , the decay in this figure is due to a imperfect vacuum leading to electron collision and loss.

Once the electrons are in the storage ring they will travel in orbit around the ring generating photons whenever magnets change their direction,[69] this is true for all synchrotrons. It should noted that second generation light sources do not have a booster ring so their storage rings must contain the RF cavities to replace the lost energy, although in practice all synchrotrons use this technique regardless of having a booster ring or not. Eventually the electron particle beam will be depleted of electrons, no vacuum is perfect and collisions will occur.[70] Third generation

light sources have the booster ring to continually provide the storage ring with fresh electrons but second generation light sources have shut down to prepare for re-injection of a new electron particle stream.

Beamlines are the final part of any synchrotron and arguably the most important. Beamlines are where the X-rays that are being generated get used by researchers. The beamlines themselves are basically pipes that go to the source of the X-rays in the storage ring and allow the newly generated X-rays access to the experimentalist's workstation. Very often a monochromator will inserted in a beamline before the X-rays can reach the hutch. The monochromator allows the users to select which wavelength of X-rays make it into the experimentalists station. There is a wide variety of uses that the light produced by synchrotrons can applied to. In the next section I will give an overview of many of the most popular uses of synchrotron radiation.

Synchrotrons are similar to cyclotrons but synchrotrons offer significant benefit over cyclotrons which require strict disc-shaped containers. [72] When bending magnets placed inside the accelerator alter the course of the electrons at these high speeds brilliant highly focused radiation is given off. This synchrotron radiation is very useful for scientists[5] and is the desired effect for many synchrotron users.

2.4.4 Making Synchrotron Radiation Brighter

Just about every experimentalist uses X-rays can list a several reason why increased brilliance would be good for their experiment. Take spectroscopy, for example, they reach their highest spectral resolution when they have the smallest contact size possible but having the maximum amount of flux possible. Thus having a more brilliant beam means they can reduce the size of their vertical slits and achieve a better resolution. Crystallography experiments, have high requirements on brightness, the closer they get to matching the incident beam to the crystal size the better their diffraction patterns are which lead to better structure characterization.[77] In tomography experiments greater brilliance can allow the experimentalist to reduce over exposure time on samples thus reducing the overall time a experiment takes, if the sample is sensitive to X-rays this is a boon.

There is no optical method for improving brilliance, you cannot focus the beam or create a aperture that increase it. A electron particle stream with low emittance will have a high

[5] Synchrotrons were originally built by physicists for their research in subatomic particles and synchrotron radiation was originally an annoyance to the physicists because it meant their electron beams lost energy every time they went through a bending magnet. However, the many uses of this light were soon recognized, and researchers began to come up with ways to use it.

brightness.[78] Since emittance is defined as the product of the beam size and its divergence the best place you can improve brilliance is in the storage ring itself.[60, 76] There are specialized devices known as insertion devices which are carefully crafted magnets. The two most popular insertion devices are undulators and wigglers.[63]

Undulators contain a regular array of alternating dipole magnets that creates a static magnetic field that forces electrons passing through it to oscillate and radiate which in turn produces more radiation than a straight path in the synchrotron would have.[68, 73] Undulator's, on average, result in a large increase in brilliance of the beam. A wiggler is very similar to a undulator but its magnets are typically arranged in Halbach Array.[73] The Halbach array is the of magnet that you find on your refrigerator, meaning that one side is magnetic but opposite has had its magnetic cancelled out by careful ordering of the magnetic components[69], as shown in Figure 2.4.4. The wiggler is basically two Halbach array sheets that force the electrons to oscillate. Due to its similarity with that of a laser[79] it has more continuous spectrum with a higher flux and shorter wavelengths than what a undulator produces. Figure 2.23 shows idealized schematics for a wiggler and undulator.

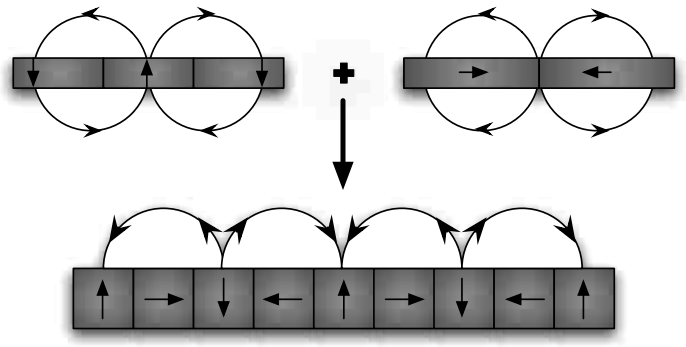
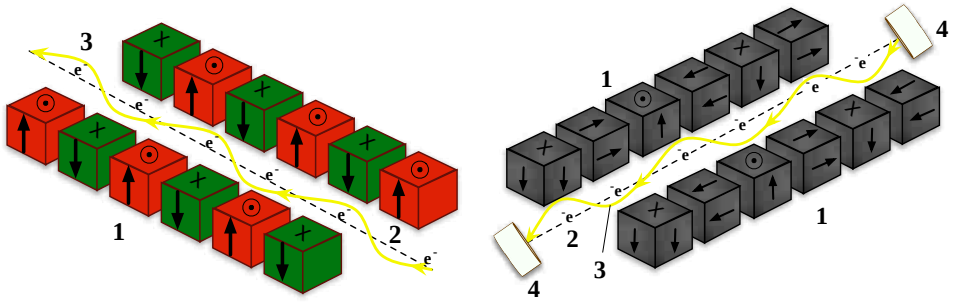


Figure 2.22: This figure shows the effect on the magnetic flux of compounds when they are superimposed into what is called a Halbach array. The magnetic flux due to the alternating magnetization will cancel itself out below the plane and double its strength (for the ideal case) above the plane.

2.4.5 Materials Science With Synchrotron Radiation

The popularity of using synchrotron radiation for materials research has been well documented as having a wide variety of uses and applications in the fields of materials science[61], physics[49, 57], chemistry[6, 8], biology[69] and medicine[55]. Table 2.4 lists the primary advan-



(a) (b)

Figure 2.23: a) Undulator schematic with (1) magnets, (2) electron beam, (3) synchrotron radiation b)Wiggler schematic showing the Halback array (1), electron beam path (2), synchrotron radiation (3), parallel mirrors used to create a a resonant cavity to trap the produced photons.

tages of it as a light source and Table 2.4 shows the experiments that are a mainstay at any typical synchrotron facility.

Table 2.4: The Many Advantages of Synchrotron Radiation

Property	Comment
Brightness/Intensity	Light produced is billions of times brighter than conventional sources.
Energy Spectrum	Can produce light from the infrared to hard X-ray wavelengths.
Beam Collimation	No need for collimators as the beam has a naturally high collimation.
Polarization	Naturally high polarization of the light.
Pulsed Emission	Time resolved studies on the scale of ≤ 1 ns are possible in modern sources

One of the great advantages of working in a synchrotron facility is that your beamline station will be right next to other beamlines each with their own specialty. Should you ever have need of looking at your samples with a new technique or have general experimental setup questions then answers may be only a short walk away. The collaborative environment that is found in synchrotron facilities is a great strength. In the following sections I will present a brief description of the common experiments listed in Table 2.4. These descriptions are not meant to be an exhaustive introduction to the techniques (with the exception of the section on tomography) but just a general introduction to them. Since the bulk of the data presented in this dissertation was produced through synchrotron X-ray tomography experiments, Section 2.2, will cover the tomography experiment using synchrotron radiation and will be a full description.

Table 2.5: Experimental Beamlines Found at Most Synchrotrons

Experiment	Use
Protein Crystallography	Structure characterization of proteins
Powder Diffraction	Structural characterization of materials
Small Angle X-ray Scattering	Provides shape and size information of macromolecules
X-ray Absorption Spectroscopy	Coordination information of atoms in materials
Tomography	Non destructive technique for 3D information of materials.
Lithography	Microfabrication technique.

X-ray Crystallography

X-ray crystallography is an experiment that determines the arrangements of atoms inside a crystal. The arrangement of atoms is calculated from the way X-rays are scattered from the electrons found around the atoms inside the crystal. When the X-rays scatter from the electrons they scatter in directions predicted by Bragg's law and a diffraction pattern is formed. This pattern can then be converted into a three dimensional electron density map. The 3D electron density map can then be used to derive the atom's coordinates and their chemical bonds. [77] Figure 2.24 shows a general workflow of the crystallography experiment.

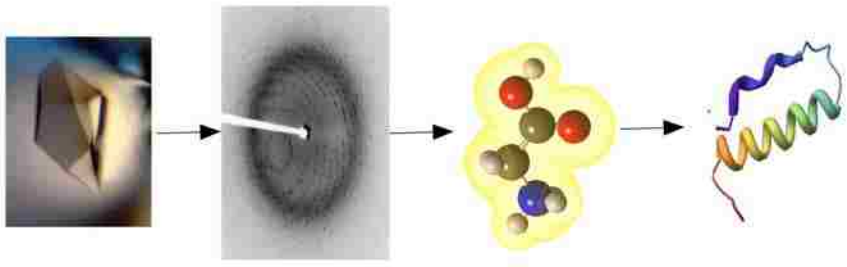


Figure 2.24: A) A single crystal is need for the experiment. Growth of single crystals can be the hardest part. [80] B) X-ray diffraction pattern formed due to Bragg's law. [81] C) Diffraction pattern is used to calculate 3D electron density maps. D) From the electron density maps a model of the molecule can be derived. Many times at this step the experimentalist will continually refine his model by going back to the diffraction data and itierativly improve his/her model.

X-ray crystallography at synchrotrons is overwhelmingly used for protein structure determination [82]. While conventional sources can provide the necessary resolution to determine the structure of crystals with a small number of atoms (≤ 100) proteins have typically have over 10,000 atoms in a unit cell. The advantages of the high intensity of synchrotron light over conventional X-ray sources makes it the perfect source for protein structure characterization. Use protein X-ray crystallography for has become a wildly popular and looking at overall synchrotron radiation

usage protein X-ray crystallography is probably the most widely used.

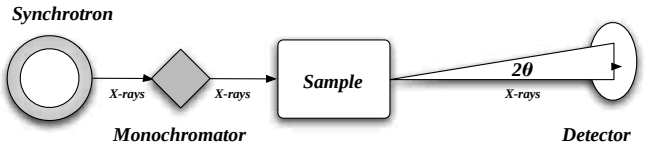
Small Angle X-ray Scattering

Small angle X-ray scattering (SAXS) measures the elastic scattering of X-rays at low range of angles. The angular scattered data can be used to derive properties of the sample being measured. Typically, macromolecules are used with this technique to calculate their shape, size, pore size and a variety of other properties. SAXS has unique advantages of having very little sample preparation, non-destructive and can accept a wide variety of samples such as soot nanoparticles, metallic glass, polymeric fibers and .[83, 84, 85] The experiment itself can be described as[69]:

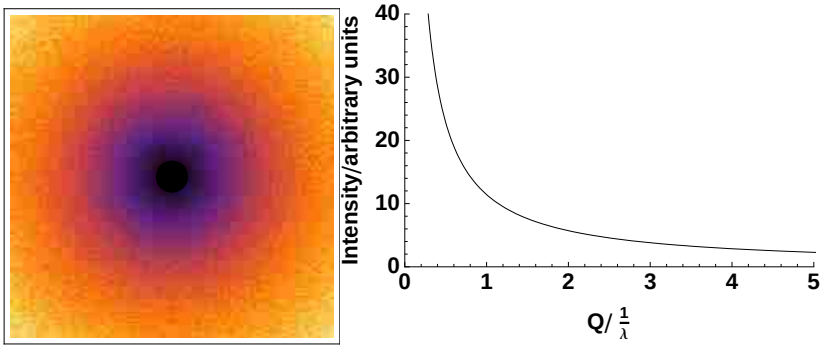
1. Expose sample to X-ray beam.
2. Scattered radiation is measured with a detector placed perpendicular to the primary X-ray beam.
3. Scattering data is measure at angles very small compared to the incidence of the primary X-ray beam.
4. Data is converted into 3D model.

Figure 2.25 shows the above list in a simplified graphical representation.

The major obstacle SAXS users have to overcome is being able to differentiate the scattered X-rays from the X-ray beam itself. An excellent analogy of the problem is one from astronomy, the sun's corona is normally not visible due to the brightness of the sun itself. During a lunar eclipse the main light of sun is blocked and the corona becomes visible.[86] This is roughly the same problem encountered by SAXS users, who have to block the primary X-ray beam. Conventional X-ray sources can be used for SAXS experiments and there are many techniques to produce a highly collimated and focused X-ray beam available.[72, 15, 67] However, the natural brilliance and high collimation of synchrotron light sources makes SAXS beamlines a common sight at all synchrotron facilities.



(a)



(b) (c)

Figure 2.25: a) Block diagram of the generalized SAXS experiment. b) Simulated two dimensional image of SAXS scattering experiment of a perfect sphere. c) Simulated spectra of a typical SAXS experiment, y-axis is in arbitrary units while the x-axis is in reciprocal wavelengths.

X-ray Absorption Spectrometry

When X-rays hit a sample they are either pass through, get absorbed or scattered.[87] For the X-rays that do not get scattered their intensity passing through the sample is governed by:

$$ln\left(\frac{I_0}{I}\right) = \mu x \rho \tag{2.31}$$

Where I_0 is the original beam intensity, μ is the linear absorption coefficient, x is the thickness of the sample, ρ is the density of the sample and I is final beam intensity. μ depends on the density of the material and the atoms that compose it. At certain energies the absorption increases dramatically, these energies of increased absorption are called edges. When an edge occurs the energy of the X-ray was enough to cause the excitation of a core electron to produce a photoelectron.[87] Figure 2.26 shows a graph that illustrates the edge concept.

This photoelectron's energy corresponds to the binding of the shell of the element left from. This information is useful in characterizing a wide variety of materials and the sample can be any phase, crystalline or amorphous and the technique is non-destructive. [87] XAS experiments themselves are typically divided into two types X-ray absorption near edge structure (XANES) and extended X-ray absorption fine structure (EXAFS).[15, 67] There is also a variation on this technique, called X-ray photoelectron spectrometery (XPS), that has developed into

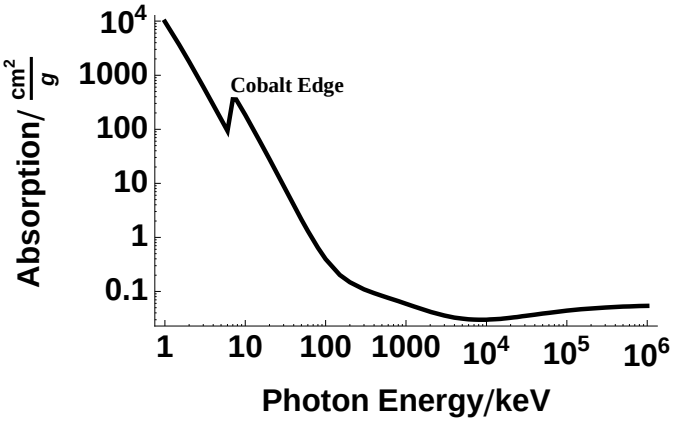


Figure 2.26: Absorption vs photon energy plot for the element Cobalt. Absorption spectra vary across all the elements elements, some have multiple edges and others have none. XAS a prized technique at probing the local electronic structure of materials.[88, 89, 90]

popular technique totally separate of XAS. XPS uses a constant energy for its X-rays rather than scanning through an absorption edge like XAS.[15] The ejected photoelectrons from the sample have their kinetic energy measured and since their kinetic energy is dependent on their binding energy different chemical species can be identified.

X-ray Lithography

X-ray lithography, which is closely related to photolithography, is a technique that has been developed to make integrated circuits[91, 92] and microelectromechanical (MEMS) devices[93]. Of all the beamlines and instruments found in synchrotron facilities the X-ray lithography facilities will be the ones most likely to see use by both industry[94] and academics. X-ray lithography benefitted from synchrotron light much like the previously mentioned techniques in that the high beam collimation and brilliance are the primary reasons to synchrotron facilities for X-ray lithography. With the help of synchrotron light's greater flux and brightness much deeper structures can be etched out creating three dimensional microstructures.[95, 96] This is known as Deep X-ray Lithography (DXRL) and is a core part of the LIGA process, which stands for X-ray Lithography Galvanoformung (Electroforming) and Abformung (Molding). DXRL makes use of a mask material that is transparent to X-rays but have a material that absorbs X-rays deposited on it in a desire pattern. The mask is then exposed to the synchrotron light at wavelengths of ≈ 0.1 nm, or less depending on the technique, and a pattern is etched out on a resist layer below the mask.[97] A resist layer is a chemical coating that will under a chemical reaction when it come into contact with the light.[98, 99] These resist layers specifically designed to have precise reactions that will benefit the final product produced. Once a pattern has been etched the electrodeposition can begin, this stage involves chemical deposition by electrolysis filling the voids with the desired atoms/compounds. This produces the required micro-structure for a one time fabrication but most fab projects have

the intention of making more than one device. After the electrodeposition a mold can be created by striping out out the resist layer still present in the new deposition layer.[91] The steps listed above a gross oversimplification of the process but they do cover the major steps of the technique. The basic setup of a X-ray lithography instrument is shown in Figure 2.27 and Figure 2.28 presents the workflow of a basic LIGA experiment.

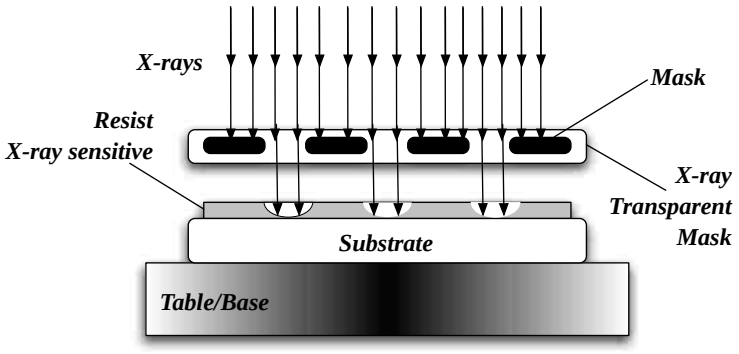


Figure 2.27: A block diagram of a Deep X-ray Lithography instrument. The table/base is actually one of the most important components since the mask can not touch the resist layer alignment of the two must be highly precise and thus the base that holds the sample must have high stability.[100] The substrate is typically some for of silicon wafer, silicone wafers as a substrate is well studied and understood making it an ideal choice. There are many types of resist layers regardless of the one used its purpose is to ensure that when X-rays come in contact with it there will be a favorable reaction.[98, 99] The mask is composed of two parts, one part is transparent to X-rays and surrounds the other component. It's main function is to maintain the shape of the mask by preventing deformations.[100] The other component is the material in the mask that is the X-ray shield for the experiment

Synchrotrons being used for nano-structured and micro-structured devices is very common and X-ray lithography as a part of LIGA is a popular technique but there a many other options available to create these devices.[91] While not every synchrotron will have a X-ray lithography beamline they typically will have some type of facility for producing microstructures as the advantages of synchrotron light are too great to be passed up.

All of the techniques presented in this section benefit and take advantage of a synchrotrons ability to produce intense, energy selectable X-rays. Much of the research achieved with these techniques could not be achieved with conventional X-ray sources.[69] Synchrotron light facilities have clearly moved beyond the moniker "parasitic" facilities and are a staple of modern materials research.

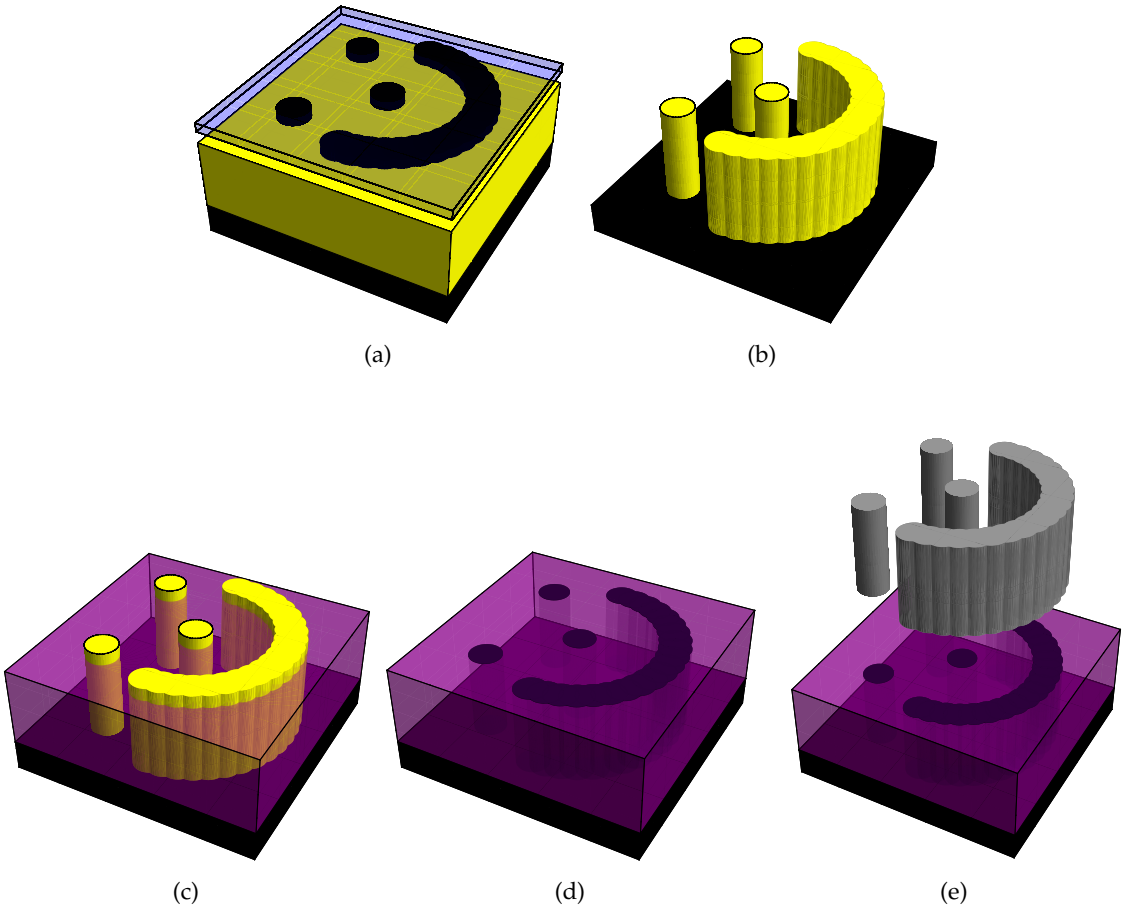


Figure 2.28: The base is not shown, the substrate is colored black,resist layer yellow, mask shown slightly above resist layer as blue, the transparent portion, and black, the X-ray blocking material. a) Exposure stage, X-rays are not shown but in the experiment they would hit perpendicular to the surface. b) After exposure and mask is removed from view c) Electrodeposition stage voids created in previous steps filled with desired atoms/compounds. d) Striping stage where the resist layer still left is removed this can be done chemically or via physical methods. e) The mold is now ready to be used to mass produce the device.[101]

2.5 Diffusion Theory

The subject of molecular diffusion is wide and deep, research into molecular diffusion finds itself in many fields like atmospheric sciences[102], catalyst design[103], nanotechnology[104], crystal growth[105] and fluid flow[106] are a few examples. A wide range of techniques like NMR[107], neutron/light scattering[108, 109], raman spectroscopy [110] and synthetic zeolites[111] have been used to quantify and measure diffusion properties of materials. Molecular diffusion is a cornerstone of many chemical processes and it is now surprise that bast amounts of research on the subject has been conducted.

On first glance, the subject of the mathematics of diffusion has far less research available in comparison[6]. The field of diffusion research is alive and well with a wide range of topics, solving complex boundary conditions found actual systems[112, 113] or abstract higher dimensional diffusion[114] to name just two examples. There lack of research on fundamental diffusion laws is simply due to the fact that it has already been thoroughly covered by another. John Crank released the first edition *The Mathematics of Diffusion* [115] in 1956 and has been called called the authoritative[7] source on the mathematics of diffusion. The book offers comprehensive solutions to the most common diffusion problems and is easily accessible. My pontification on Crank's mathematical contribution to diffusion theory is not without reason. The diffusion model used to describe the polymer blends presented in this research were lifted verbatim from Crank's book,[115] no other sources were used. What follows in this section is not meant to be a all encompassing introduction to the diffusion theory. It is far too large and has already been well covered by Crank. Rather, it will explain a polymer additive problem called "blooming" in terms of a diffusion model, taken from Crank, and justify the selected model and the assumptions used.

Blooming is a process of diffusion and/or nucleation of small molecules on the surface of a polymer blend. It is known that changes in the blend morphology in polymers, such as blooming, can lead to the reduction or even elimination of the desired polymer blend functions. Flame retardant blooming is cause for concern in industry and is also an interesting scientific problem when viewed as a diffusion process. Synchrotron X-ray tomography has developed into an excellent method for three-dimensional imaging of polymer blends and could be used to study the blooming of flame retardants in polymer blends. Before this process can be studied a general diffusion model for blooming must be chosen, consider Figure 2.29:

[6] For example a search for "Mathematics Diffusion" in the internet search engine Web of Science yielded 222 hits while a corresponding search of "Molecular Diffusion" yielded 31,231 hits.

[7] Google Scholar states that Crank's book on diffusion has been cited over 9,000 times. Crank himself appears to have been cited over 15,000 times.

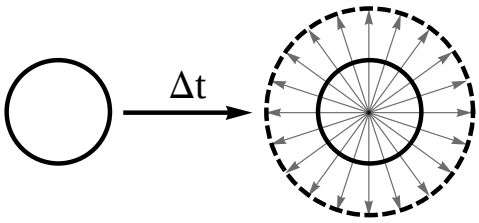


Figure 2.29: Graphical representation of the diffusion process in two dimensions. The diffusional process depicted is called a point source diffusion model due to uniform distribution of particles surrounding a central point.

The point source diffusion described in Figure 2.29 can be described as a 2D approximation of the blooming process. This process can be accurately described by the following model[115]:

$$C = \frac{M}{2\sqrt{\frac{\pi b^2}{4}}} e^{-\left(\frac{x^2}{b^2}\right)} + c \tag{2.32}$$

Where C is concentration, M is the total amount of substance that will be diffused at time t=0 and x=0, c is the value that the function will converge on as x goes to infinity and can be thought of as the average concentration. The value of b is known as the diffusion length and represents how far diffusion has propagated. There is a relationship between the diffusion length b and the diffusion coefficient[115]:

$$D = \frac{b^2}{4t} \tag{2.33}$$

$$b = \sqrt{4Dt} \tag{2.34}$$

Using Equation 2.34 we can derive a three dimensional version of Equation 2.32 that contains the diffusion coefficient [115]. This model is also called a point source model and can be defined as:

$$C = ae^{\left(\frac{-r^2}{4Dt}\right)} + c \tag{2.35}$$

Using this model we have the ability to calculate the diffusion constant for a given system.

The diffusion coefficient[8], D, is measured in units of length squared per time (i.e. $\frac{cm^2}{s}$) and is proportional to the velocity of the molecules that are diffusing.

Equation 2.35 is the model used to study the blooming of flame retardants in polymer blends. Many other models have been solved for hollow spheres, cylinders, torus, cuboids, etc. and with a variety of boundary conditions.[115] With so many other models available choosing the correct one becomes tricky. Picking a model without knowing what the structure the experimental data will take is haphazard and will likely result with a poor model. The data for the blooming experiments presented in this research (which will be discussed in full detail in later chapters) proved to be strikingly similar to the 3D point source model making the choice of Equation 2.35 easy.

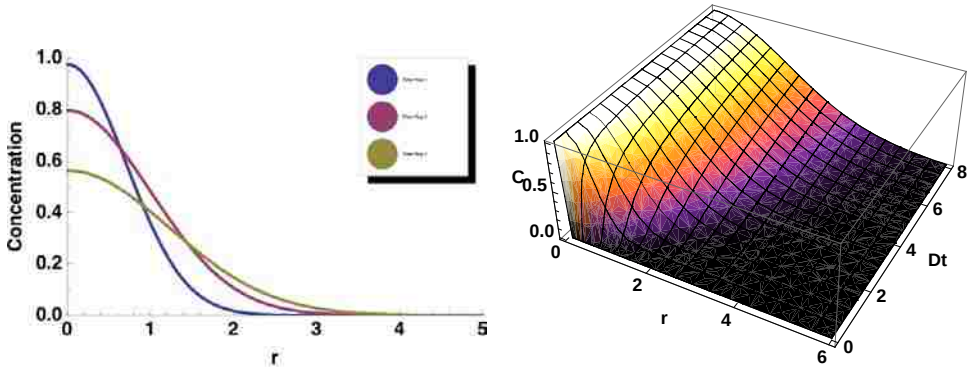
Now that we have general model to fit to our experimental data what can it tell us to expect when dealing with spherical diffusion of flame retardants in polymers. Figure 2.30 investigates the behavior of Equation 2.35 as time evolves.

Assuming that we have chosen the correct model, Figure 2.30 gives us a glimpse into how experimental data should behave as it diffuses over time. It also implies some conditions that are important for us to achieve a good fit. Our model will work best when the experimental data has diffusion domains that are spherical, non-overlapping and isolated from any objects that would perturb the concentration gradients in the domain. If those conditions are met then calculating diffusion coefficients from tomography data should be possible and accurate.

2.6 3D Image Processing

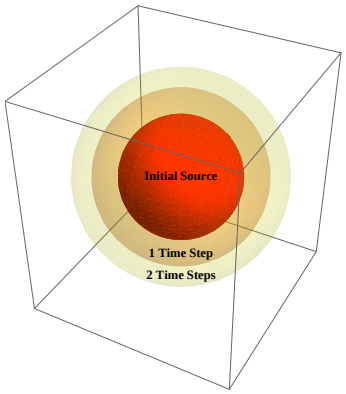
This section will cover the most common techniques and algorithms that are used in my research. These are not algorithms to used to reconstruct data from projections or convert absorption data to chemical concentrations. The algorithms presented in this are concerned with extracting desired data out of a chemical concentration volume. First a note on the way this section is presented. Since this section is devoted to algorithmic development computer code is needed. It was decided that instead of presenting actual code pseudocode would be presented instead. Actual code can be messy and overburdened with unnecessary syntax that hinders its readability. Whenever pseudocode is presented it will appear in the form:

[8]While not directly useful to the tasks at hand there is an interesting relationship for D that can be defined mathematically as $D = k_b T 6\pi\eta r$. Where ν is the viscosity, T temperature, k_b is Boltzmann's constant and r the radius. This equation is known as the Einstein-Stokes Relation and it shows that D depends on viscosity, temperature and the size of the molecules diffusing.



(a)

(b)



(c)

Figure 2.30: Various graphical depictions of the diffusion model for a spherical point source (Equation 2.35)over time. a) Each line represents a new time step in the model. Important features include the decreasing amplitude and increasing diffusion length parameters (a and b respectively) of the model as time evolves. b) 3D plot of the model with axes for r radius,C concentration and Dt the diffusion length. c) Graphical 3D representation of the model.

```
x = [1,2,3]
y = Sum(x)
If y != 6
        Print("Error")
Else
        Print(y)
```

The pseudocode tries to be as clear as possible if there is any ambiguity. If there any difficulty with reading the pseudocode itself every algorithm presented in this section has a fully functional version written in Mathematica or Matlab in the Appendix.

The concepts discussed in this section are concerned with digital image processing, or to be more specific 3D image processing. 3D image processing is concerned with taking spatial data and producing either new desired information or making the spatial data easier to manage.[116] The field of image processing is large and has many subfields such as computed tomography[117, 118], computer vision[119], morphological/geometric transformations[120], medical imaging[121, 122], face recognition[123, 124], and image restoration[125, 126, 127] are a few of the prominent fields. I won't try and cover them all, just the ones used in this research. The techniques I use in my research can be broadly defined as belonging to one of four classifications given in Table 2.6. The table gives a brief description of the algorithm and whether or not new data is generated in the process of using it. Please note that this list is not considered a standard, it is simply my own own views on the types of algorithms that I write and may or may not be similar to other lists.

Table 2.6: 3D Algorithm Classifications

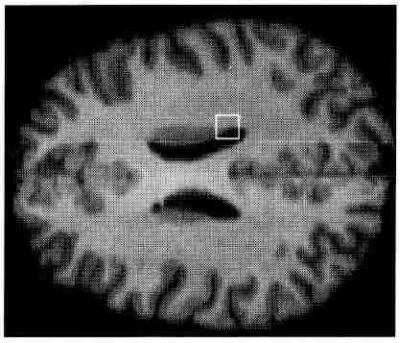
Classification	Description	New Data?
Object Extraction	Separates wanted/unwanted data	New 3D data
Pattern Recognition	Detects conditions set by user	No
Systemization	Organizes data in a more efficient way	New data typically 2D
Transform	Applies a operation that alters the data	Either

Before discussion of the main algorithms I employ, a basic definition of terms and concepts of some simple, though fundamental, algorithms are in order.

2.6.1 Thresholding

Typical two dimensional images are composed of individual elements known as pixels. Pixels can have a wide range of values from simple binary, 16 bit, 32 bit, grayscale, RGB values, etc.[116] Grayscale is common image format in it pixels can range in values between 0 and 255

with 0 representing white and 255 black. Figure 2.31 shows a grayscale image of a CT brain slice and some of its corresponding pixel values.



166	176	169	164	166	167	164	167
170	171	169	168	168	163	164	166
170	168	172	170	174	162	165	168
165	165	169	165	173	161	163	164
155	154	154	149	151	134	133	130
141	136	128	117	98	57	59	54
128	118	94	69	45	36	37	28
91	78	57	45	38	39	31	25

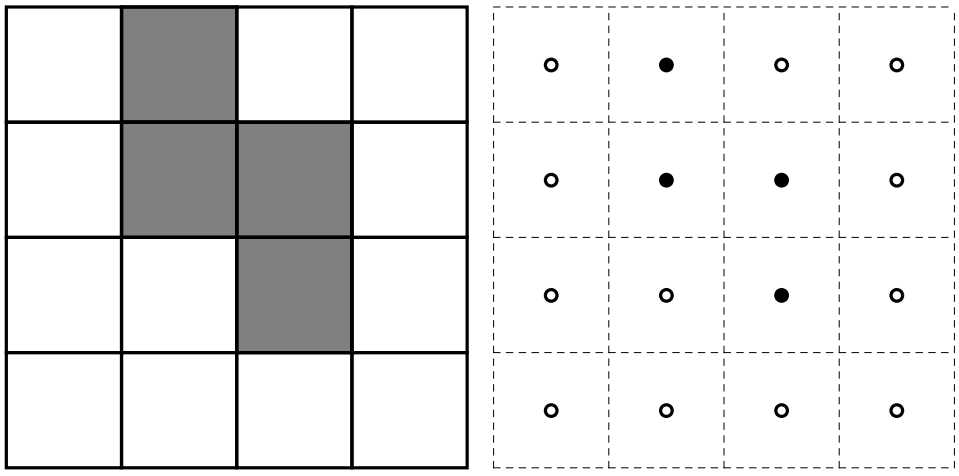
Figure 2.31: A digital image and its grey codes.[128]

Thresholding is perhaps the simplest image segmentation technique available. It is a transform algorithm since it used to reduce a regular image into a binary form of that image. An image that has been through a threshold algorithm has had each of individual pixels set to ”0” or ”1” if there value is greater or lower than some previously chosen threshold value. The effect of a thresholding algorithm can produce an entirely new 3D data set or overwrite the old data. This is left up to the discretion of the user but in practice overwriting data is considered bad form. The concept of foreground and background voxels becomes important here. A foreground voxel is typically associated with the value of ”1” while background voxels are ”0”.[129] Most often the object of interest is given the foreground values and everything else the background. Figure 2.32 shows two graphic representations of background and foreground pixels of a matrix defined as:

$$Image = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Figure 2.31 shows a entire image and a small section of its grey codes. A greyscale image such as this is easily used with thresholding algorithms to produce binary images. With the proper selection of a threshold value Figure 2.31 can be transformed into Figure 2.33.

Binary images have a number of advantages in image processing. They have small memory requirements due to their 8 bit nature, have special storage formats such as octree and raster encodings [130] that can further reduce the memory requirements and in certain cases speed up data analysis.



(a) (b)

Figure 2.32: a) Grey squares represent the foreground pixels and white squares the background pixels. b) Another representation of foreground/background pixels with black circles being foreground and white circles background pixels. This representation is more abstract than the first but is useful when discussing pixel connectedness and neighborhoods which is to be discussed later.

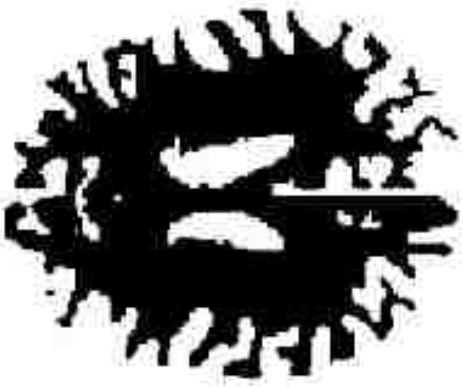


Figure 2.33: A binary image constructed by applying a threshold algorithm on the greyscale image from Figure 2.31.[128]

Thresholding itself can be categorized into six different groups such as histogram shape, spatial, local gray level, space clustering, entropy and object recognition methods.[14] Thresholding is the easiest segmentation available and is widely used as a first step in many other techniques.

2.6.2 Neighborhoods

Neighborhoods is defined as the collection of pixels surrounding a chosen center pixel. There are two ways to define a define a neighborhood in two dimensions with 4 or 8 of the nearby by pixels being included.[128] Figure 2.34 shows both definitions in graphically form.

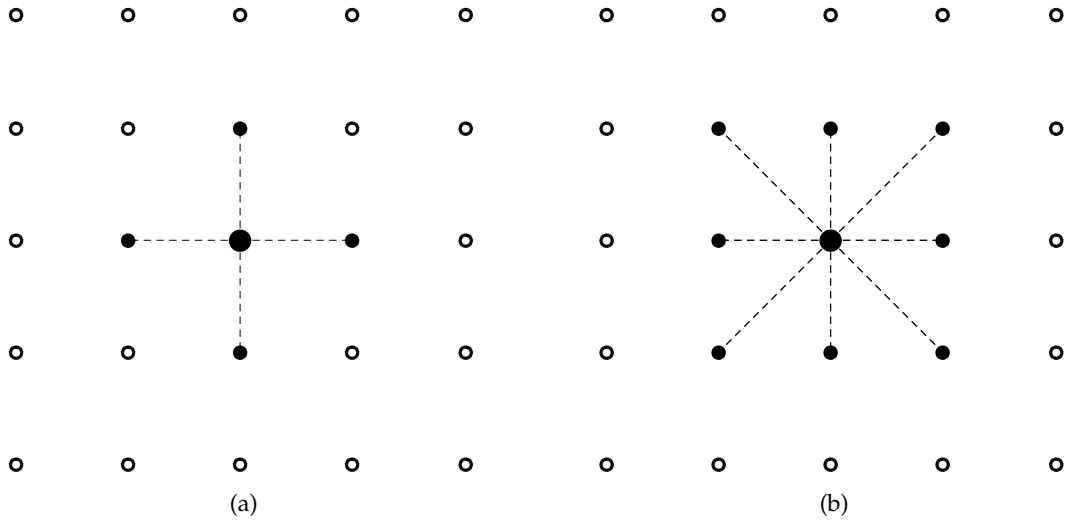


Figure 2.34: a) 4 nearest neighbor representation with the coordinates of the neighbors being set along the discrete horizontal and vertical axis.

When the dimensions of an image are increased to three and we have volume image rather than the traditional flat images the neighborhoods obviously become more complex. Since the term pixel is a two dimensional term the word voxel comes into play. A voxel is a data value in a 3D volume, which represents a value with coordinates in three dimensional space, this is analogous to a pixel in two dimensions. A visual representation of a voxel is shown in Figure 2.35. Just as there were neighborhoods in 2D, the 3D case is no exception. With the third dimension there are three types of neighborhoods that can be defined. Figure 2.36 displays all three possible representations. The selection of which neighborhood to use in a algorithm has a direct impact on the speed and performance of the algorithm. [129] Picking a 6 neighborhood structure will lead to less calculations done but you give up the extra information/precisiono of 18 and 26. give.

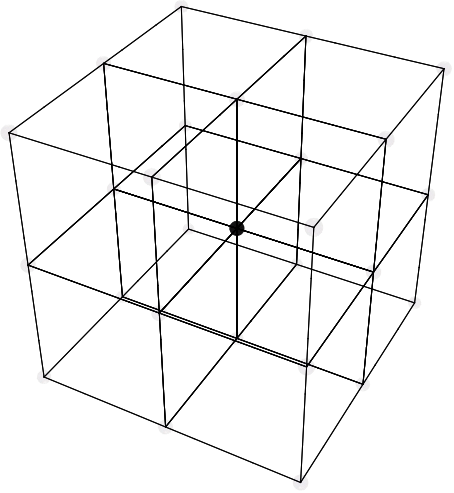


Figure 2.35: Visual representation of the central voxel as the dark sphere in the center of the 3×3×3 grid of light gray voxels.

2.6.3 Edge Detection

Edge detection is used in two and three dimensional image processing to find borders between objects.[129] This is a object extraction method and is widely studied technique with many variations on how to perform it.[116] Simple edge detection aims at labeling the pixels/voxels at which their values sharply change near their neighbors. Edges can be classified into two different types, viewpoint independent and viewpoint dependent.[128] Viewpoint independent edges are what would be traditionally thought of as images and they represent boundaries between objects, surfaces and shapes. Viewpoint dependent edges are found in rapidly changes images (i.e. movies). For the purposes of this research all the edges can be considered viewpoint independent since our data is static 3D volumes of data.

The Canny filter method is considered one of the oldest and is one of the most cited techniques for edge detection.[116] Other methods for calculating edges in images are many and diverse. [126, 5] For the purposes of this research a modern edge detection algorithm is overkill. Sufficient results can be obtained by thresholding our based on a chemical concentration or absorption value of our choosing to produce a binary image of the desired components. Once in a binary image format the edge detection algorithm becomes a nearly trivial loop as shown in pseudocode below[9].

```
1 for slice = 1 to numSlices
2         for row = 1 to numRows
3                 for col = 1 to numCols
4                         if Voxelat(vol,slice,row,col) == 1
```

[9]This pseudocode is for a 6 nearest neighbor check, 16 and 27 nearest neighbor versions are simply linear extensions and have more lines of code.

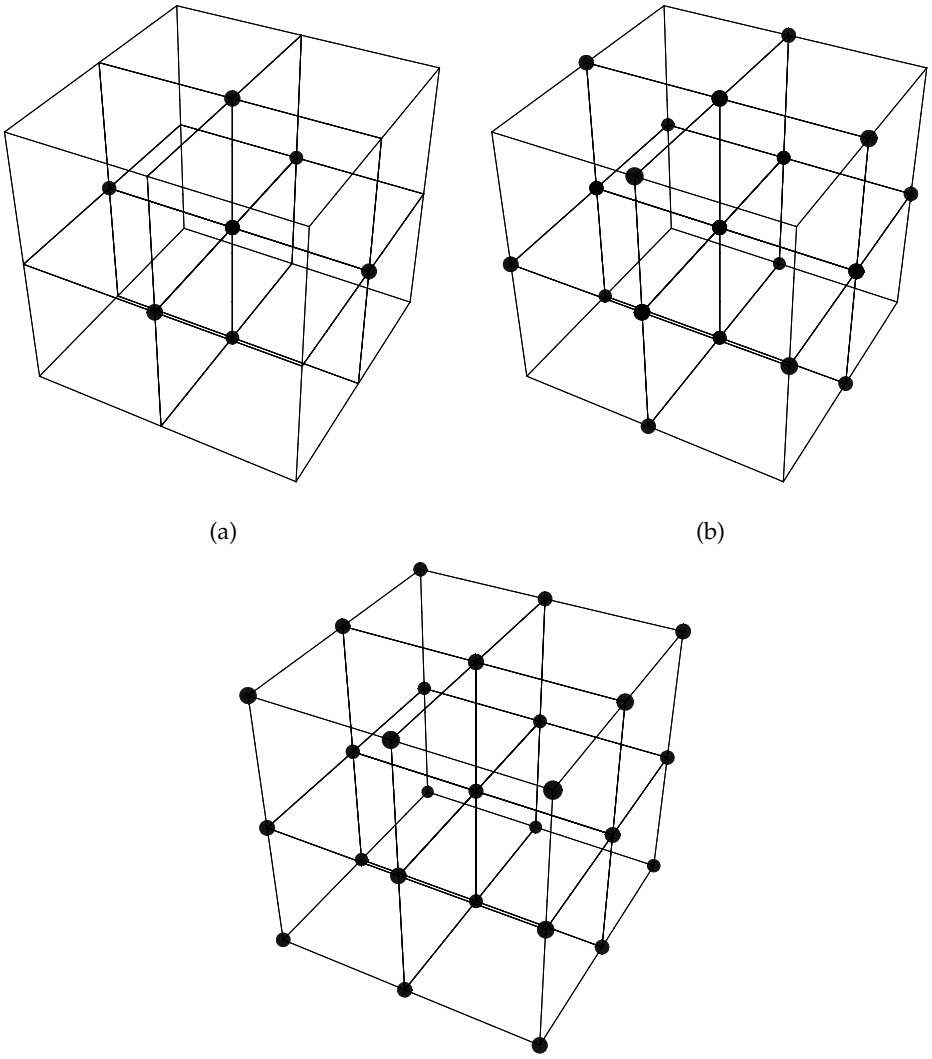


Figure 2.36: a) The 6 nearest neighbors of a central voxel. b) The 18 nearest neighbors. c) The 26 nearest neighbors of a voxels. Selection of which neighborhood to use can be of great importance in a image segmentation study.

```
5                                   and Pixelat(vol,slice,row,col-1) == 0
6                                   and Pixelat(vol,slice,row,col+1) == 0
7                                   or Pixelat(vol,slice,row-1,col) == 0
8                                   or Pixelat(vol,slice,row+1,col) == 0
9                                   or Pixelat(vol,slice-1,row,col) == 0
10                                  or Pixelat(vol,slice+1,row,col) == 0
11
12                                  SetVoxel(newVol,slice,row,col) = 1
```

2.6.4 Connected Components

Connected components is an object extraction algorithm. In fact one can argue that it is the most fundamental and basic type of object extraction available. In 3D, the connected component algorithm searches through a binary volume and identifies foreground voxels that are adjacent/connected to each other.[128] Where a connection/adjacency can be defined in the since of neighborhoods as discussed previously. All foreground voxels that are determined to be connected are labeled with a unique identifier.

The algorithm iterates through each volume element of the data by column, row and slice. If the element is not a member of the background voxels then we set the neighboring elements to the label value. We find the neighbors smallest label attached to one of it's other neighbors and assign it to the current element. If there are no neighbors, then this voxel is a unique island and we should increase the label value and continue. If there are neighbors with labels then the element is assigned the lowest equivalent label found from its neighbors. Pseudocode for this procedure is listed below:

```
1  label = 1
2  labels = zeros(sizeof(src))
3  for slice = 1 to numSlices
4          for row = 1 to numRows
5                  for col = 1 to numCols
6                          if Voxelat(src,slice,row,col) == 1
7                                  neighbors = getNeighborsVals(labels,slice,row,col)
8                                          if All neighbors in labels are empty
9                                                  linked(label) = label
10                                                 SetVoxel(labels,slice,row,col) = label
11                                                 label++
12                                         else
13                                                 L = neighbors
14                                                 SetVoxel(labels,slice,row,col) = min(L)
```

Connected components has found itself to be extremely useful in variety of fields.[116]

Due to its usefulness it has been extensively improved and modified for various tasks such as taste evaluation of sponge cake [131], automatic detection of neurological diseases [117] and computer aided diagnoses and surgery [121, 122]. Each of previously mentioned subjects requires a connected components analysis in order to work but each has separate requirements. Evaluating the size of air bubbles in sponge cake can be far less strict than computer aided surgery requires. Computer aided surgery on the other would require extremely fast computation but may not care about the efficiency of algorithm. A variety of modifications of the connected components algorithm can be found, such as a fast 2 pass version [132], a specialized noisy/fuzzy component labeling[133], competitive labeling that improves the accuracy of the shapes extracted [134] and a version that can extract components despite a distortion/shearing of the volume[135]. Each of these algorithms presents pros and cons in using them and the researcher must take care in choosing which when to apply to his/her data.

2.6.5 Collision Detection

In a collision detection algorithm the code checks for collisions/intersections between two or more objects. Large amounts of work has been put into algorithms for collision detection.[136] Chemical physics simulation [137], biomechanics simulations of human movement[138], robot movement AI[139] and all video games[140] require collision detection. Some systems, such as billiards or bowling, can be simulated as rigid body of motion with elastic collisions and systems such as this can have relatively simple algorithms. Systems such as air traffic flight control require a more complex algorithm.[141]

All collision detection algorithms can be divided into two four approaches: space-time volume intersection, swept volume, interference detection and trajectory parameterization.[136] In trajectory parameterization the collision detection algorithm calculates the instants of the collision before the objects ever actually intersect. This method uses the trajectories and positions of the objects and make it ideal for things like air traffic control.[141] Space-time and swept volume algorithms are designed for simulations that will react to a detected collision but the simulation will not end, examples include video games and multi-particle physics simulations.[142] The interference detection method works by checking for any collisions in a static environment and if none are found advance to the next step and check for more collisions.

In our case the volume intersection method is by far the best choice. Without discussing the exact nature of the data to presented in later chapters it is hard to justify the simplifications that I will use in the rest of the this section. For now just trust that the 3D image analysis required

by the polymer blend data does not require the prediction of real time collisions, just where the collision occurs. A simplification used to simplify this calculation is to treat all objects as if they were spheres. This simplification actually does not contradict the nature of the 3D data and this will be discussed in later chapters.

The collision detection algorithm used takes static objects labeled by the connected components algorithm and checks for collisions, if none our found we increase the radius of the object and re-check until a collision is found. Once a collision is found assign the save the radius and move on to the next object. All that this method requires is a list of our objects in the 3D volume, which is readily provided by the connected components algorithm, along with their volume and approximate spherical/cylindrical radius. None of the other chemical/volume data is required at this point to determine the points of collisions. Pseudocode for this collision detection algorithm is listed below:

```
#labels variable is the result from a connected component algorithm
for object = 1 to [Number of Objects]
        while r <= MaxRadiusThreshold
                coords = GetSphereCoordinates(object,r)
                list = VoxelsAt(labels,coords)
                        if list is empty
                                r++
                        else
                                objectRadius(i) = r
                                break
```

2.6.6 Domain Extraction and Neighbor Finding

Frequently when dealing with experimental 3D data we find that volume shows a extremely chaotic and messy system. In order to effectively fit our data to a chosen model or to do just simple calculations of phases of interest a algorithm has to be developed to specifically address the issues of that sample. The flame retardants in polymer blends tend to begin as highly concentrated ”lumps”. These lumps then diffuse over time and the desire to study those lumps have lead to the creation of the algorithm described in this section. We have to enforce certain conditions on our algorithm in order to ensure that data we segment out can be used by our diffusional model. The conditions are:

- Domains must be spherical. The closer they are to being perfect spheres the better. Our model is designed for spherical distribution, no other shapes.

- The domains can not overlap with other domains. Overlapping domains result in concentration gradients meeting and this is not accounted for in our model.
- Domains should be isolated from air bubbles, cracks and anything that would warp or enhance the concentration gradients in the domain.

With our restrictions listed, let's state the things that would improve our fit or improve our ability to work with the data.

We will have a lot of data, each of the hundreds (or thousands!) of domains will have parameters of label, radius, centroid, volume, voxel XYZ points, bounding box and many more. Let's assume that on average each volume initially has 1000 different domains (this isn't a unreasonable number for a first pass). Let's also assume that the average discrete volume of each of these domains is 100 voxels. Label, radius and volume are all scalars but the bounding box is a 6 component vector and with the average discrete volume being 100 voxel thats 100 XYZ points (total of 300 values) we need to store.

$$1000\ domains \times 3\ scalars \times 6\ components \times 300\ XYZ = 5.4 \times 10^5\ values$$

So we have on average 5.4×10^5 objects to keep track of. That number is actually an underestimate since later on we will define new variables that need to be tracked and saved. Keeping the objects saved in a logical manner is just half the fight, we also need to be able to quickly process the data for comparisons and cross references. A efficient, flexible and powerful data format/database is needed to keep track of all this data, ASCII tables will not do. If our restrictions on our domains don't trim down the number of domains to a manageable number we need to consider further steps to reduce the number of domains but in a way that improves the final result and is not arbitrarily removing domains.

One such removal criteria can be small domains. Small domains don't provide as much benefit as larger ones, there is simply less data to contribute. Computationally keeping track of small domains is wasteful, why store a domain that has only contributes 10 voxels. If possible, removing domains whose volume falls below a certain threshold would be desirable.

The domains that we will look at will have a set radius that was calculated at the boundary defined by the thresholding algorithm. If the domain is sufficiently isolated from other domains then an extension of this radius to include more voxels would be good. We obviously can't arbitrarily increase random domains radii. We will have to calculate a way to systematically increase the radii of domains that are isolated but not increase the radii to a point that they would overlap

other domains.

To calculate diffusion measurements around the pockets of high concentration in the sample while keeping in mind our restrictions and desires, a new algorithm was developed. This algorithm has the following features that attempt to address the restrictions and desires discussed above:

- All data stored in HDF5 format for long term storage and immediate analysis.
- Conditional test to remove domains with small volumes.
- Nearest neighbor algorithm developed to systematically increase radii to their maximum possible value.
- Conditional test determine if domain is spherical enough.
- Connected component algorithm does not allow for overlap to occur.
- Automatic identification of air bubbles and cracks.

How the Algorithm Works

This section covers the theory and logic behind the algorithm developed to extract domains of interest from the experimental data. The primary goal in the development of this algorithm was achieve excellent domain extraction but as a secondary goal it was the author's desire to write it in a versatile and multifaceted fashion. In other words each major part of the algorithm is completely interchangeable with another algorithm that performs the same function. For instance, this algorithm contains a connected components code section that is essentially the same one from Section 2.6.4. If the user desires to use his/her own version of the code then in theory all he/she should have to do is rewrite their code to output their data in the format that this algorithm reads. Effort was taken to make sure that all data input/output was clear, concise and easy to implement. This was done for future code reusability and code readability. The distinct separations between each major step makes reading and comprehending the code much easier. This design has been somewhat tested and seems to perform adequately in this manner. Below is a list that has the general steps that algorithm goes through in a typical run:

1. Using a basic threshold algorithm on the volume, identify the foreground voxels as having high bromine concentration and the everything else being background voxels.

2. Using a connected component algorithm on the binary volume. Count and label each set of voxels that are connected as one unique object or ”domain”.

3. For every labeled domain calculate the centroid, volume and a 1x3 vector that represents widths of the smallest enclosing bounding box. Assign these values to the labeled domain.

4. User determines if connected components results are acceptable. If they are not go back and threshold at a new value.

5. Remove all labeled domains that have volumes less than a selected threshold.

6. Remove all labeled domains whose bounding box widths don't approximate the lengths of a cube. This is a conditional meant to gauge how close the domain shape comes to a sphere.

7. For every labeled domain calculate the distances to all other domains.

8. For every labeled domain find its nearest neighbor.

9. For every labeled domain assign a radius value that is equal to half the distance to its nearest neighbor.

10. User decides if results are good. If they are store all data in a hierarchal format supported by HDF5.

Mathematica code that employs this method is given in Appendix 6.4. Pseudocode and a flow diagram for this method is shown in Figure 2.37.

Figure 2.38 shows a mathematica simulation of the nearest neighbor code (corresponds to steps 7-9) in the algorithm.

Most traditional 3D image processing algorithms are designed to be fully automatic and with the exception of starting it up for the first time one can leave most it unattended. The desire for full automation is certainly tempting, however, I have designed this algorithm to have manual input at 2 stages. The major reason for placing manual input sections into the code is the incredible range of samples that a typical tomography beamline can produce. I have personally imaged or analyzed polymer blends, geological rock samples, parrot beaks, cat claws, soil samples and many objects over the past few years. Each one of the samples was entirely different in terms of resolution, noise, contrast and size, all factors that have major impact on image analysis algorithms. Just changing X-ray energies in the tomography experiment can lead new phases showing up in a sample. Thus, the two manual input stages are provided so the user doesn't accidently waste

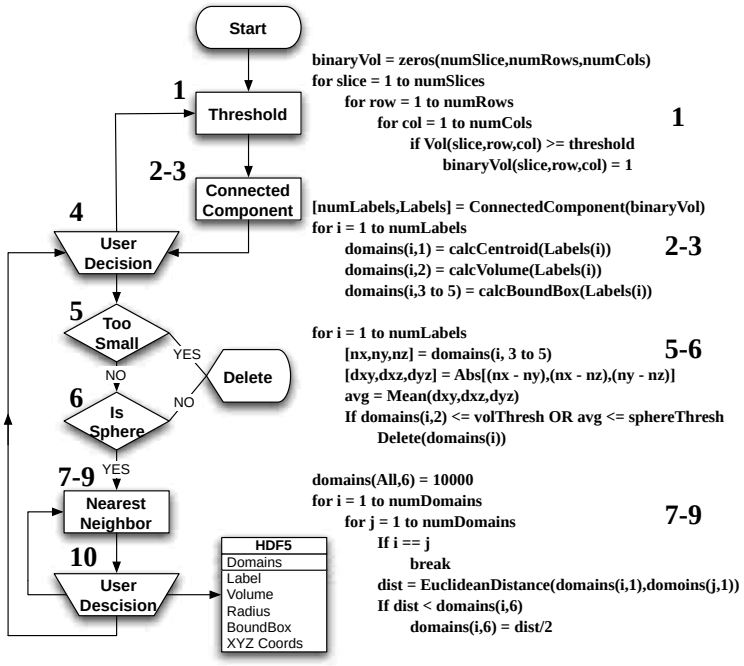


Figure 2.37: Flow diagram of the algorithm and the corresponding pseudocode. The numbers indicate which portions of code are executed in the flow diagram. The numbers are also directly related to List 10. In the flow diagram portion, rectangles represent code that can only produce new data, diamonds represent code that has the power to eliminate data, trapezoids do not represent code but rather the user supplying input to the algorithm about what it's next step should be. The HDF5 table simply represents saving the data in that format. There are two other miscellaneous objects, start and delete, they do not represent code and are self-explanatory.

computation time on bad data. The two manual steps are 4 and 10. At these stages the algorithm waits for the user to accept or reject the results that have been produced so far.

The first manual input occurs after the connected components code (step 3), this is the place that causes the most errors in our image analysis. If the connected component algorithm produced either too many or too few objects this would be cause for a rejection of the results. This typically means the threshold value was poorly chosen in step 1 and starting over with a new threshold value is in order. Step 9 occurs after the nearest neighbor code, there are no typical reasons for failure at this step. Since it is the last step before writing the results a manual input was placed here as a catch all, errors at this step are fairly rare. In the case of a error the user can choose to rerun the nearest neighbor code and hope for the best or he/she can go back to step 4 and double check their connected components results. If the user had bad connected component results and did not correct them that would certainly lead to failure at step 9. Otherwise, the best way to determine where the error is to perform a iterative process of running and rerunning the

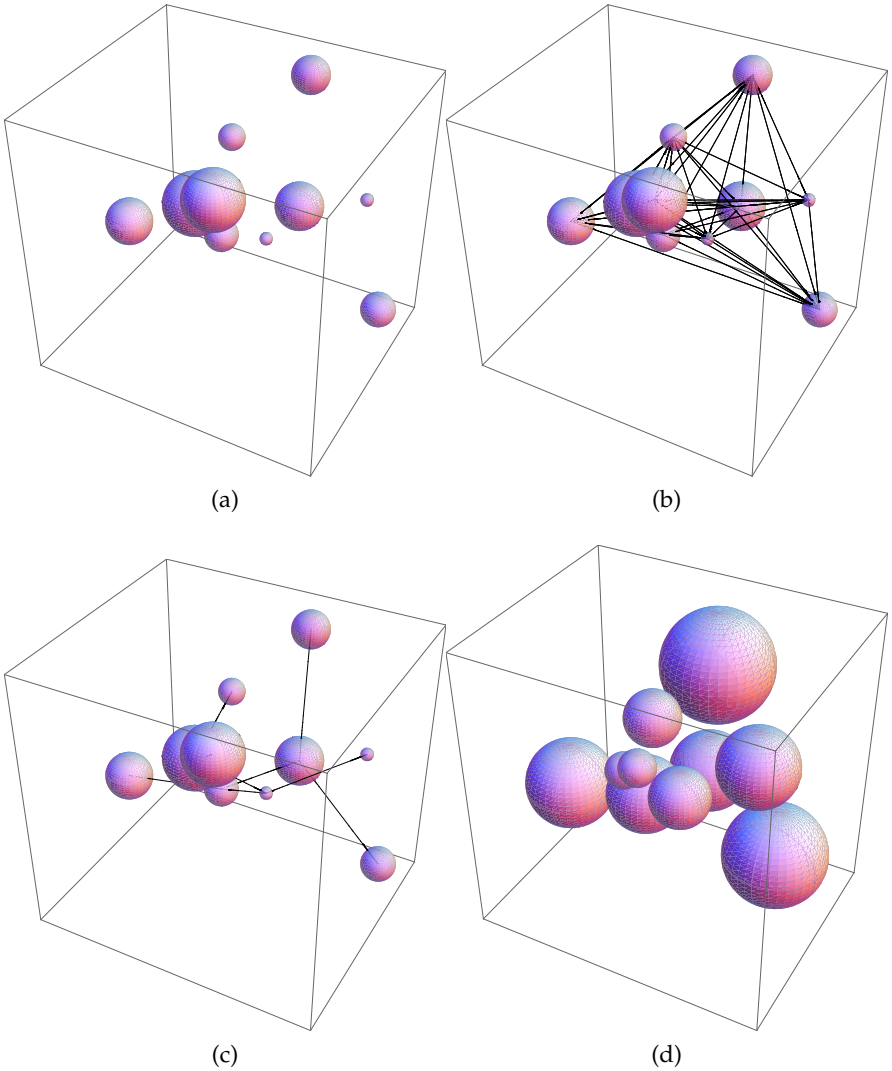


Figure 2.38: a) Randomly chosen sample domains with various radii, these are meant to represent the domains of high hexabromobenzene concentration in the sample. b) Lines are generated for every possible pair of domain connections. This is step 7 of List 10. c) The lines are then culled until the only connections that remain are the ones that represent the nearest neighbor domain pairs. Step 8 of List 10. d) The domain's radii are then set to half the distance to their respective nearest neighbor. Step 9 of List 10.

algorithm to find the error.

Once a user has become an expert with data analysis of particular samples he/she can of course automate the algorithm fully with a minimal rewrite of code.

2.7 Bibliography

[1] M. Lavaljeantet and H. Fischgold. History of Tomography. *Journal de Radiologie d Elctrologie et de Medecine*, 56, 1975.

[2] Pittsburgh Supercomputer Center. The Visible Human Project at the Pittsburgh Supercomputing Center. http://www.psc.edu/biomed/research/vh/pvb/ (Accessed September, 2003), 2003.

[3] H. Jastrow and L. Vollrath. Teaching and Learning Gross Anatomy Using Modern Electronic Media Based on the Visible Human Project. *Clinical Anatomy*, 16, 2003.

[4] J. T. Ratnanather, K. N. Botteron, T. Nishino, A. B. Massie, R. M. Lal, S. G. Patel, S. Peddi, R. D. Todd, and M. I. Miller. Validating Cortical Surface Analysis of Medial Prefrontal Cortex, 2001.

[5] S. Y. Wan, E. L. Ritman, and W. E. Higgins. Multi-Generational Analysis and Visualization of the Vascular Tree in 3D Micro-CT Images. 32, 2002.

[6] K. Ham, H. Jin, R. I. Al-Raoush, X. G. Xie, C. S. Willson, G. R. Byerly, Larry S. Simeral, M. L. Rivers, R. L. Kurtz, and L. G. Butler. Three-Dimensional Chemical Analysis with Synchrotron Tomography at Multiple X-Ray Energies: Brominated Aromatic Flame Retardant and Antimony Oxide in Polystyrene. *Chemistry of Materials*, 16, 2004.

[7] R. Marschallinger. Three-Dimensional Reconstruction and Modeling of Microstructures and Microchemistry in Geological Materials. *Scanning*, 20, 1998.

[8] L. G. Butler, J. W. Owens, F. K. Cartledge, R. L. Kurtz, G. R. Byerly, A. J. Wales, P. L. Bryant, E. F. Emery, B. Dowd, and X. G. Xie. Synchrotron X-Ray Microtomography, Electron Probe Microanalysis, and NMR of Toluene Waste in Cement. *Environmental Science & Technology*, 34, 2000.

[9] A. Elmoutaouakkil, G. Fuchs, P. Bergounhon, R. Pres, and F. Peyrin. Three-Dimensional Quantitative Analysis of Polymer Foams from Synchrotron Radiation X-Ray Microtomography. *J. Phys. D: Appl. Phys.*, 36, 2003.

[10] B. D. Smith. Cone-Beam Tomography - Recent Advances and a Tutorial Review. *Optical Engineering*, 29, 1990.

[11] Web of Science. http://apps.isiknowledge.com, 2008.

[12] F. Natterer and E. L. Ritman. Past and Future Directions in X-Ray Computed Tomography (CT). *International Journal of Imaging Systems and Technology*, 12, 2002.

[13] Charles L. Epstein. *Mathematics of Medical Imaging*. Prentice Hall, 2003.

[14] M. Slaney and K. C. Avinash. *Principles of Computerized Tomographic Imaging*. IEEE Press, 1988.

[15] Douglas A. Skoog, James F. Holler, and Timothy A. Nieman. *Principles of Instrumental Analysis*. Brooks Cole, 1997.

[16] Frank Natterer and Frank Wubbeling. *Mathematical Methods in Image Reconstruction*. SIAM, 2001.

[17] Yves Nievergelt. Elementary Inversion of Radon's Transform. *Siam Review*, 28, 1986.

[18] H. H. Barrett and H. Gifford. Cone-Beam Tomography with Discrete-Data Sets. *Physics in Medicine and Biology*, 39, 1994.

[19] M. L. Jan, C. Y. Chen, C. K. Yeh, T. R. Yeh, and M. T. Wang. 3D Image-Reconstruction Using Cone-Beam Tomography. *NDT & E International*, 27, 1994.

[20] K. E. Bennett, G. W. Faris, and R. L. Byer. Experimental Optical Fan Beam Tomography. *Applied Optics*, 23, 1984.

[21] D. Schuster. Filtered Backprojection for 3-Dimensional Radon Problems Summary. *Numerical Functional Analysis and Optimization*, 10, 1989.

[22] K. L. D'Amico, H. W. Deckman, J. H. Dunsmuir, B. P. Flannery, and W. G. Roberge. X-Ray Microtomography with Monochromatic Synchrotron Radiation. *Review of Scientific Instruments*, 60, 1989.

[23] P. Spanne and M. L. Rivers. Computerized microtomography using synchrotron radiation from the NSLS. *Nuclear Instruments & Methods in Physics Research Section B-Beam Interactions with Materials and Atoms*, B24-25, 1987.

[24] M. L. Rivers. Rivers, M. L. Tutorial Introduction to X-ray Computed Microtomography http://www-fp.mcs.anl.gov/xray-cmt/rivers/tutorial.html (accessed June 2003), 2003.

[25] Nist. Xcom: Photon Cross Sections Database, 2003.

[26] O. Heaviside. The Waste of Energy from a Moving Electron. *Nature*, 67, 1903.

[27] G. A. Schott. On the Radiation from Moving Systems of Electrons, and on the Spectrum of Canal Rays. *Philosophical Magazine*, 13, 1907.

[28] A. C. Thompson, David T. Attwood, Eric M. Gullikson, Malcolm R. Howells, Jeffrey B. Kortright, Arthur L. Robinson, and James H. Underwood. *X-Ray Data Booklet*. Lawerence Berkely National Laboratory, 2001.

[29] G. A. Schott. *Electromagnetic Radiation and the Mechanical Reactions Arising from it*. Cambridge University Press, 1912.

[30] J. Schwinger. Electron Radiation in High Energy Accelerators. *Physical Review*, 70, 1946.

[31] J. Schwinger. On Quantum-Electrodynamics and the Magnetic Moment of the Electron. *Physical Review*, 73, 1948.

[32] J. Schwinger. On the Classical Radiation of Accelerated Electrons. *Physical Review*, 75, 1949.

[33] A. A. Sokolov and I. M. Ternov. The Quantum Theory of the Radiating Electron. *Soviet Physics JETP-USSR*, 1, 1955.

[34] A. A. Sokolov. Further Discussion on the Quantum Theory of the Radiating Electron. *Soviet Physics JETP-USSR*, 3, 1956.

[35] A. A. Sokolov and A. N. Matveev. The Problem of the Radiation of Ultra High Velocity Electrons Moving in a Constant Magnetic Field. *Soviet Physics JETP-USSR*, 3, 1956.

[36] D. W. Kerst. A 20-Million Electron-Volt Betatron or Induction Accelerator. *Review of Scientific Instruments*, 13, 1942.

[37] J. P. Blewett. Synchrotron Radiation - Early History. *Journal of Synchrotron Radiation*, 5, 1998.

[38] D. Iwanenko and I. Pomeranchuk. On the Maximal Energy Attainable in a Betatron. *Physical Review*, 65, 1944.

[39] Physics Time Capsule: The Betatron. http://physics.illinois.edu/history/Betatron.asp, 2008.

[40] J. P. Blewett. Synchrotron Radiation - 1873 to 1947. *Nuclear Instruments & Methods in Physics Research Section A-Accelerators Spectrometers Detectors and Associated Equipment*, 266(0168-9002), 1988.

[41] V. I. Veksler and L. M. Kovrizhnykh. Cyclic Acceleration of Particles in High-Frequency Fields. *Soviet Physics Jetp-Ussr*, 8, 1959.

[42] E. M. McMillan. The Relation between Phase Stability and 1st-Order Focusing in Linear Accelerators. *Physical Review*, 80, 1950.

[43] H. C. Pollock. Combination of Betatron and Synchrotron for Electron Acceleration. *Physical Review*, 69, 1946.

[44] H. C. Pollock, R. V. Langmuir, F. R. Elder, J. P. Blewett, and A. M. Gurewitsch. Design of a 70-MeV Synchrotron. *Physical Review*, 70, 1946.

[45] F. R. Elder, A. M. Gurewitsch, R. V. Langmuir, and H. C. Pollock. Radiation from Electrons in a Synchrotron. *Physical Review*, 71, 1947.

[46] H. C. Pollock. The Discovery of Synchrotron Radiation. *American Journal of Physics*, 51, 1983.

[47] F. R. Elder, A. M. Gurewitsch, R. V. Langmuir, and H. C. Pollock. A 70-MeV Synchrotron. *Journal of Applied Physics*, 18, 1947.

[48] F. R. Elder, R. V. Langmuir, and H. C. Pollock. Radiation from Electrons Accelerated in a Synchrotron. *Physical Review*, 74, 1948.

[49] J. P. Blewett. Synchrotron Light and Its Uses. *IEEE Transactions on Nuclear Science*, 26(0018-9499), 1979.

[50] J. P. Blewett, L. Blumberg, A. J. Campillo, R. P. Dinardo, H. C. Hsieh, S. Krinsky, A. Luccio, C. Pellegrini, J. Schuchman, P. Z. Takacs, and A. Vansteenbergen. Free-Electron Laser Experiment at the NSLS 700 MeV Electron Storage Ring. *IEEE Transactions on Nuclear Science*, 28(0018-9499), 1981.

[51] J. P. Blewett. Space-Charge Limit in a Relativistic Migma. *Nuclear Instruments & Methods in Physics Research Section A-Accelerators Spectrometers Detectors and Associated Equipment*, 271(0168-9002), 1988.

[52] E. M. McMillan. A History of the Synchrotron. *Physics Today*, 37, 1984.

[53] S. Doniach, K. Hodgson, I. Lindau, P. Pianetta, and H. Winick. Early Work with Synchrotron Radiation at Stanford, 1997.

[54] I. M. Ternov. Use of Specific Properties of Synchrotron Radiation .7. Synchrotron Radiation Research at Moscow-State-University. *Nuclear Instruments & Methods*, 152, 1978.

[55] D. W. Fry. The Synchrotron Accelerator - Its Potentialities as a Generator of X-Rays and Electrons of 10-50 MeV Energies for Medical Use. *British Journal of Radiology*, 22, 1949.

[56] K. R. Lea. Highlights of Synchrotron Radiation. volume 43. Elsevier Science, 1978.

[57] A. A. Sokolov, I. M. Ternov, F. A. Korolev, Mikhaili.Vv, and V. R. Khalilov. Properties of Synchrotron Radiation and Its Application. *Izvestiya Vysshikh Uchebnykh Zavedenii Fizika*, 1972.

[58] J. S. Nodvick and D. S. Saxon. Suppression of Coherent Radiation by Electrons in a Synchrotron. *Physical Review*, 96, 1954.

[59] D. W. Lynch. Tantalus, a 240 MeV Dedicated Source of Synchrotron Radiation, 1968-1986, 1997.

[60] M. Kihara. Development of Synchrotron Radiation Storage Rings. *Journal of Synchrotron Radiation*, 5, 1998.

[61] G. R. Neil. Trends and Opportunities in Light Source Development. *Nuclear Instruments & Methods in Physics Research Section a-Accelerators Spectrometers Detectors and Associated Equipment*, 483, 2002.

[62] J. L. Laclare. Light Source Performance Achievements. *Nuclear Instruments & Methods in Physics Research Section A-Accelerators Spectrometers Detectors and Associated Equipment*, 467, 2001.

[63] H. Kitamura. Recent Trends of Insertion-Device Technology for X-Ray Sources. *Journal of Synchrotron Radiation*, 7, 2000.

[64] Light Sources. http://lightsources.org/cms/, 2008.

[65] The World of Synchrotron Radiation. http://www.srs.ac.uk/srs/SRworldwide/index.htm, 2008.

[66] Advanced Light Source: Synchrotron Light Sources of the World. http://www-als.lbl.gov/als/synchrotron_sources.html.

[67] James William Rohlf. *Modern Physics from a to Z0*. Wiley, 1994.

[68] H. Wiedemann. *Synchrotron Radiation*. Springer, 2002. 1.

[69] G. Margaritondo. *Elements of Synchrotron Light: For Biology, Chemistry, and Medical Research*. Oxford University Press, 2002.

[70] M. H. R. Donald, M. R. Harold, N. M. King, B. G. Loach, J. R. M. Maidment, C. W. Planner, G. H. Rees, and J. V. Trotman. A Superconducting Synchrotron Design Study at the Rutherford Laboratory. *IEEE Transactions on Nuclear Science*, 18, 1971.

[71] Synchrotron Schematic. http://en.wikipedia.org/wiki/Image:Schema_de_principe_du_synchrotron.jpg, 2008.

[72] R. A. Dunlap. *Experimental Physics - Modern Methods*. Oxford University Press, 1988. October.

[73] D. J. Thompson, R. Coisson, M. Ericksson, J. Leduff, A. Hofmann, D. Husmann, G. Mulhaupt, M. W. Poole, M. Renard, M. Sommer, V. P. Suller, S. Tazzari, and F. H. Wang. The All Wiggler Synchrotron Radiation Source. *IEEE Transactions on Nuclear Science*, 28, 1981.

[74] E. M. McMillan. *The Synchrotron - a Proposed High Energy Particle Accelerator*, volume 68. American Physical Society, 1945.

[75] E. M. McMillan. *The Origin of the Synchrotron*, volume 69. American Physical Society, 1946.

[76] H. Wiedemann. Brightness of Synchrotron Radiation from Electron Storage-Rings. *Nuclear Instruments & Methods*, 172, 1980.

[77] J. R. Helliwell. Synchrotron X-Radiation Protein Crystallography - Instrumentation, Methods and Applications. *Reports on Progress in Physics*, 47, 1984.

[78] H. Wiedemann. Design of Low Emittance Storage-Rings. *Nuclear Instruments & Methods in Physics Research Section A-Accelerators Spectrometers Detectors and Associated Equipment*, 246, 1986.

[79] H. Wiedemann. Storage Ring Design Optimization for FEL Operation. *Journal De Physique*, 44, 1983.

[80] Single Crystal. http://en.wikipedia.org/wiki/Single_crystal, 2008.

[81] Single Crystal Diffraction Pattern. http://www.esrf.eu/UsersAndScience/Experiments/SCMatter/ID13/p 2008.

[82] Zygmunt Derewenda. Advances in Protein Crystallography - Fourth Annual Meeting - 24-25 January 2007, San Francisco, Ca, Usa, 2007.

[83] X. J. Liu, X. D. Hui, G. L. Chen, and M. H. Sun. In Situ Synchrotron Saxs Study of Nanocrystallization in Zr65Ni25Ti10 Metallic Glass. *Intermetallics*, 16, 2008.

[84] S. F. Ran, X. H. Zong, D. F. Fang, B. S. Hsiao, B. Chu, P. M. Cunniff, and R. A. Phillips. Studies of the Mesophase Development in Polymeric Fibers during Deformation by Synchrotron Saxs/Waxd. *Journal of Materials Science*, 36, 2001.

[85] S. di Stasio, J. B. A. Mitchell, J. L. LeGarrec, L. Biennier, and M. Wulff. Synchrotron Saxs (in Situ) Identification of Three Different Size Modes for Soot Nanoparticles in a Diffusion Flame. *Carbon*, 44, 2006.

[86] Paul G. Hewitt. *Conceptual Physics*. Addison Wesley, 2005.

[87] R. P. Madden. *X-Ray Spectroscopy*. McGraw-Hill Book Company, 1974.

[88] E. D. L. Rienks, J. W. Bakker, A. Baraldi, S. A. C. Carabineiro, S. Lizzit, C. J. Weststrate, and B. E. Nieuwenhuys. Synchrotron XPS Study of the No-Co Reaction on Pt(100). *Surface Science*, 532, 2003.

[89] K. Roodenko, M. Gensch, J. Rappich, K. Hinrichs, N. Esser, and R. Hunger. Time-Resolved Synchrotron XPS Monitoring of Irradiation-Induced Nitrobenzene Reduction for Chemical Lithography. *Journal of Physical Chemistry B*, 111, 2007.

[90] Camilla Nordhei, Astrid Lund Ramstad, and David G. Nicholson. Nanophase Cobalt, Nickel and Zinc Ferrites: Synchrotron XAS Study on the Crystallite Size Dependence of Metal Distribution. *Physical Chemistry Chemical Physics*, 10, 2008.

[91] C. Linder, N. F. Derooij, R. P. Peters, N. Unal, and R. Ruprecht. Manufacturing Technologies for Microsystems. *Technisches Messen*, 60, 1993.

[92] A. Heuberger. Synchrotron Lithography for High-Resolution. In *Journal of Vacuum Science & Technology A-Vacuum Surfaces and Films*, volume 5, 1987.

[93] H. L. Huber, H. Betz, and A. Heuberger. Synchrotron Lithography - the Way to Submicron Features with Single Layer Resists. *Polymer Engineering and Science*, 26, 1986.

[94] E. Weihreter. Superconducting Synchrotron Radiation Sources for Industrial Applications. *Nuclear Instruments & Methods in Physics Research Section B-Beam Interactions with Materials and Atoms*, 68, 1992.

[95] D. Munchmeyer and J. Langen. Manufacture of 3-Dimensional Microdevices Using Synchrotron Radiation. *Review of Scientific Instruments*, 63, 1992.

[96] D. Chinn, P. Ostendorp, M. Haugh, R. Kershmann, T. Kurfess, A. Claudet, and T. Tucker. Three Dimensional Imaging of LIGA-Made Microcomponents. *Journal of Manufacturing Science and Engineering-Transactions of the ASME*, 126, 2004.

[97] J. T. Kim, S. P. Han, and M. Y. Jeong. A Modified DXRL Process for Fabricating a Polymer Microstructure. *Journal of Micromechanics and Microengineering*, 14, 2004.

[98] K. Kadel, W. K. Schomburg, and G. Stern. X-Ray Masks with Tungsten Absorbers for Use in the LIGA Process. *Microelectronic Engineering*, 21, 1993.

[99] J. Trube and A. Heuberger. Application of Organic Resist Materials in Synchrotron Lithography. *Angewandte Makromolekulare Chemie*, 183, 1990.

[100] J. Schulz, K. Bade, M. Guttmann, L. Hahn, A. Janssen, U. Kohler, P. Meyer, and F. Winkler. Ensuring Repeatability in LIGA Mold Insert Fabrication. *Microsystem Technologies-Micro-and Nanosystems-Information Storage and Processing Systems*, 10, 2004.

[101] G. Baumeister, R. Ruprecht, and J. Hausselt. Replication of LIGA Structures Using Microcasting. *Microsystem Technologies-Micro-and Nanosystems-Information Storage and Processing Systems*, 10, 2004.

[102] A. Garcia Munoz. Formulation of Molecular Diffusion in Planetary Atmospheres. *Planetary and Space Science*, 55, 2007.

[103] Eivind Johannessen, Gang Wang, and Marc-Olivier Coppens. Optimal Distributor Networks in Porous Catalyst Pellets. I. Molecular Diffusion. *Industrial & Engineering Chemistry Research*, 46, 2007.

[104] Y. Y. Kievsky, B. Carey, S. Naik, N. Mangan, D. Ben-Avraham, and I. Sokolov. Dynamics of Molecular Diffusion of Rhodamine 6g in Silica Nanochannels. *Journal of Chemical Physics*, 128, 2008.

[105] H. Youk, R. List, and T. Ola. The Growth of Ice Crystals by Molecular Diffusion. *Journal of the Atmospheric Sciences*, 63, 2006.

[106] C. R. Huang, A. Denny, and N. W. Loney. Molecular Diffusion in the Laminar Sub-Layer during Turbulent Flow in a Smooth Tube. *Chemical Engineering Science*, 59, 2004.

[107] Yoav Shrot and Lucio Frydman. The Effects of Molecular Diffusion in Ultrafast Two-Dimensional Nuclear Magnetic Resonance. *Journal of Chemical Physics*, 128, 2008.

[108] R. Mukhopadhyay and S. Mitra. Molecular Diffusion and Confinement Effect: Neutron Scattering Study. *Indian Journal of Pure & Applied Physics*, 44, 2006.

[109] G. R. Palmer and D. B. Sattelle. Laser-Light Scattering Diffusion Measurements on Brain Microtubule Protein from Dogfish (Squalus-Acanthus) and Beef. *Journal of Cell Science*, 52, 1981.

[110] J. H. Yoon, T. Kawamura, S. Takeya, S. K. Jin, Y. Yamamoto, T. Komai, M. Takahashi, A. V. Nawaby, and Y. P. Handa. Probing Fickian and Non-Fickian Diffusion of CO2 in Poly(Methyl Methacrylate) Using in Situ Raman Spectroscopy and Microfocus X-Ray Computed Tomography. *Macromolecules*, 37, 2004.

[111] Ryo Nagumo, Hiromitsu Takaba, and Shin-ichi Nakao. High-Accuracy Estimation of 'Slow' Molecular Diffusion Rates in Zeolite Nanopores, Based on Free Energy Calculations at an Ultrahigh Temperature. *Journal of Physical Chemistry C*, 112, 2008.

[112] Arturo de Pablo, Guillermo Reyes, and Ariel Sanchez. Blow-up for a Heat Equation with Convection and Boundary Flux. *Proceedings of the Royal Society of Edinburgh Section a-Mathematics*, 138, 2008.

[113] Fengde Chen, Yuming Chen, and Jinlin Shi. Stability of the Boundary Solution of a Nonautonomous Predator-Prey System with the Beddington-Deangelis Functional Response. *Journal of Mathematical Analysis and Applications*, 344, 2008.

[114] Mauricio Bogoya. A Nonlocal Nonlinear Diffusion Equation in Higher Space Dimensions. *Journal of Mathematical Analysis and Applications*, 344, 2008.

[115] J. Crank. *The Mathematics of Diffusion*. Oxford University Press, 1975.

[116] J. C. Russ. *The Image Processing Handbook*. CRC, 2006.

[117] E. D. Seletchi and O. G. Duliu. Image Processing and Data Analysis in Computed Tomography. *Romanian Journal of Physics*, 52, 2007.

[118] Gabor T. Herman. *Discrete Tomography*. Birkhauser, 1999.

[119] Bin Shao and Honwei Xin. A Real-Time Computer Vision Assessment and Control of Thermal Comfort for Group-Housed Pigs. *Computers and Electronics in Agriculture*, 62, 2008.

[120] Guillermo Sapiro. *Geometric Partial Differential Equations and Image Analysis*. Cambridge University Press, 2001.

[121] Qiang Niu, Xiaoyi Chi, Ming C. Leu, and Jorge Ochoa. Image Processing, Geometric Modeling and Data Management for Development of a Virtual Bone Surgery System. *Computer Aided Surgery*, 13, 2008.

[122] Heber MacMahon. *Advanced Image Processing and Computer-Aided Diagnosis - Are We There yet?*, volume 23. Lippincott Williams & Wilkins, 2008.

[123] Nojun Kwak. Feature Extraction for Classification Problems and Its Application to Face Recognition. *Pattern Recognition*, 41, 2008.

[124] Daniel Riccio and Jean-Luc Dugelay. Geometric Invariants for 2D/3D Face Recognition. *Pattern Recognition Letters*, 28, 2007.

[125] Jr. Johnson, C. Richard, Ella Hendriks, Igor J. Berezhnoy, Eugene Brevdo, Shannon M. Hughes, Ingrid Daubechies, Jia Li, Eric Postma, and James Z. Wang. Image Processing for Artist Identification. *IEEE Signal Processing Magazine*, 25, 2008.

[126] S. Kim. PDE-Based Image Restoration: A Hybrid Model and Color Image Denoising. *IEEE Transactions on Image Processing Transactions on Image Processing*, 15, 2006.

[127] S. M. Chao and D. M. Tsai. Astronomical Image Restoration Using an Improved Anisotropic Diffusion. *Pattern Recognition Letters*, 27, 2006.

[128] Gabriele Lohmann. *Volumetric Image Analysis*. Wiley-Teubner, 1998.

[129] R. C. Gonzalez, R. E. Woods, and S. L. Eddins. *Digital Image Processing Using Matlab*. Prentice Hall, 2004.

[130] D. Meagher. Geometric Modeling Using Octree Encoding. *Computer Graphics and Image Processing*, 19, 1982.

[131] K. Kato, K. Yamamoto, and N. Ogawa. Proposal for New Image Processing Application for Taste Evaluation of Sponge Cake Using 3D Range Sensor. *Imaging Science Journal*, 56, 2008.

[132] Lifeng He, Yuyan Chao, and Kenji Suzuki. A Run-Based Two-Scan Labeling Algorithm. *IEEE Transactions on Image Processing*, 17, 2008.

[133] Ming-Hu Ha and Xue-Dong Tian. An Improved Algorithm of Optical Formula Extraction with Fuzzy Classification. *International Journal of Pattern Recognition and Artificial Intelligence*, 22, 2008.

[134] Binh-Minh Bui-Xuan, Michel Habib, and Christophe Paul. Competitive Graph Searches. *Theoretical Computer Science*, 393, 2008.

[135] Umut Ozertem, Deniz Erdogmus, and Robert Jenssen. Mean Shift Spectral Clustering. *Pattern Recognition*, 41, 2008.

[136] P. Jimenez, F. Thomas, and C. Torras. 3D Collision Detection: A Survey. *Computers & Graphics-UK*, 25, 2001.

[137] S. Redon, A. Kheddar, and S. Coquillart. Fast Continuous Collision Detection between Rigid Bodies. *Computer Graphics Forum*, 21, 2002.

[138] Sujeong Kim, Stephane Redon, and Young J. Kim. Continuous Collision Detection for Adaptive Simulation of Articulated Bodies. *Visual Computer*, 24, 2008.

[139] M. Perez-Francisco, A. P. del Pobil, and B. Martinez-Salvador. Parallel Collision Detection between Moving Robots for Practical Motion Planning. *Journal of Robotic Systems*, 18, 2001.

[140] Anderson Maciel and Suvranu De. An Efficient Dynamic Point Algorithm for Line-Based Collision Detection in Real Time Virtual Environments Involving Haptics. *Computer Animation and Virtual Worlds*, 19, 2008.

[141] J. Hershberger. Kinetic Collision Detection with Fast Flight Plan Changes. *Information Processing Letters*, 92, 2004.

[142] M. L. Gavrilova and J. Rokne. Collision Detection Optimization in a Multi-Particle System. *Computational Science-ICCS 2002, Pt III, Proceedings*, 2002.

Chapter 3

3D Chemical Distribution of a Flame Retardant and Synergist in a Polymer Blend

3.1 Introduction

The dream of 3D chemical analysis for material science investigations is coming true with new advances in data acquisition and image processing.[1, 2, 3, 4, 5] Recently, synchrotron X-ray tomography has developed into a fast, reliable method for three-dimensional imaging of materials at the micrometer distance scale.[6, 7, 8, 9, 10, 11] While invasive microtome-and-image still has applications in geology [12], biology [13], and the Visible Human Project [14], full 3D imaging methods are needed in materials science for reasons of sample throughput and data quality. In this work, we investigate the 3D concentration distribution of a flame retardant throughout a polymer blend, especially at the interface with fiberglass reinforcement bundles.

Three-dimensional, and even higher dimensionality, imaging is used to investigate a variety of polymer structure and processing issues. For the new process of high shear rate thermoplastic powder injection molding, synchrotron X-ray tomography of silica particles within the matrix shows flow effects beyond those predicted with Newtonian flow models, indicative of wall slip or surface roughness in the mold.[15] A laboratory micro-focus X-ray source yielding images with near 4 μm voxel size at 25 keV was used to image voids and cracks in fiber-reinforced composite laminates.[16] To more easily visualize cracks, an X-ray dye was forced into the sample, thus enabling observation of cracks with sizes on the order of 20% of the voxel size. X-ray absorption tomography can yield inadequate contrast for blends with similar elemental composition; in that case, X-ray refraction differences can sometimes be used to enhance contrast, as shown for a blend of polystyrene and polymethylmethacrylate, imaged at 17.7 keV with 3.14 μm voxel size.[17] A

high-strength, thermal-insulating foam, consisting of glass hollow microspheres imbedded in an epoxy resin was imaged in 3D at 9.7 keV.[18] The microsphere mean diameter was 35 μm, which is resolved nicely with the 0.7 μm voxel size. The images clearly show the three phase system: glass, air, and resin. The images were then analyzed to yield the microsphere size distributions, both sphere diameter and wall thickness. Interestingly, refraction effects, use to advantage by Momose et al., [17] created minor analysis problems for the microsphere detection. X-ray fluorescence tomography was used to study uptake of Pt(II) and Pt(IV) anticancer drugs in growing tissues; concentration data was presented as a radial concentration profile through the near-spherical tumor sample.[19] In a recent work in electron microscopy tomography, multi-energy images were used to separate two components of interest, multiwalled carbon nanotubes and a nylon matrix; the authors note the 4D character of the work, with three spatial dimensions and one chemical analysis dimension.[20] TEM and electron microscopy tomography of nickel oxide nanoparticles in an ordered mesoporous material showed the unique advantages of 3D imaging.[21] The pore structure in polyvinyl chloride foams has been studied with tomography and a mean intercept technique [22]. A laboratory X-ray source was used to image CO_2-foamed samples of polymethylmethacrylate. Eight samples were measured after different exposure times, and the foam thickness growth rate about the solid PMMA core was shown to be a result of non-Fickian CO_2 diffusion.[23] The structures of star-terpolymers contain interpenetrating columns, with diameters on the order of 20 nm, are imaged by reactive metal staining followed by electron microscopy tomography.[24, 25]

Some flame retardants exhibit interesting dissolution or precipitation properties, depending upon blend compatibility. We previously used 3D tomography to assess the dissolution and molecular diffusion of BT-93 into polystyrene, where extremely slow diffusion rates, on the order of 10^{-17} m^2/s can be detected.[26] Also, we performed a number of simulated tomography experiments to explore instrumentation parameters critical to accurate 3D chemical composition mapping with synchrotron X-ray tomography. The experimental parameters included CCD camera specifications and multi-spectral X-ray imaging, and how these parameters affect the composition maps.[27]

Fiberglass-reinforced thermoplastics have utility in applications and selecting the proper blend for a task is important. In some applications, there is the potential for ignition [28], hence the blending with a flame retardant [29, 30, 31]. One high performance flame retardant is a short-chain, partially-brominated polystyrene (**FR**), (Figure 3.1), enhanced with a synergist, antimony(III) oxide, Sb_2O_3.

In a previous investigation of a flame retardant/polystyrene blend [10], we developed some of the techniques used herein, especially the concept of imaging with X-ray energies that

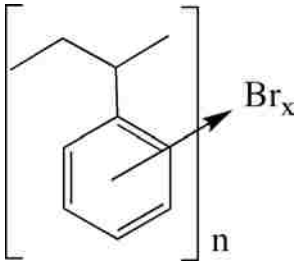


Figure 3.1: Chemical structure of brominated polystyrene (x = 3) used as a flame retardant (**FR**).

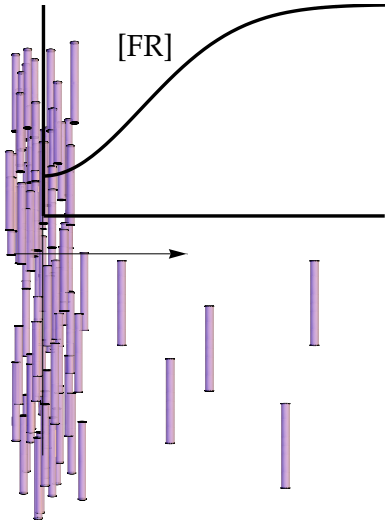


Figure 3.2: Small glass fibers will assemble into crude bundles in the extruded polymer sample. In tomography, the concentration of **FR** is measured in cubic volume elements (voxels). The volume elements can be partially occupied by SiO_2, which should cause **FR** concentration to decrease within the bundles.

span important features in the Br and Sb absorbances. In that work, the flame retardant Saytex BT-93, a brominated aromatic, combined with a synergist, antimony(III) oxide, were found to have a rather heterogeneous distribution in the polystyrene blend. In contrast, in this work, we find the short-chain, partially-brominated polystyrene to be much more evenly distributed throughout the polymer matrix. However, the introduction of a fiberglass reinforcement creates an additional step in the data analysis, and also allows an assessment of the enhanced **FR** concentration distribution in the close vicinity of the reinforcement. This sample vividly demonstrates the need for multi-energy tomography as not all X-ray energies yield high contrast images.

3.2 Experimental Section

3.2.1 Sample Preparation

A short chain, partially brominated polystyrene flame retardant (Figure 3.1 with x = 3), and its synergist, antimony(III) oxide, were blended in a twin-screw reactor at 285 °C with a dry mixture of nylon, teflon, and fiberglass, and extruded as a rod. The nylon and fiberglass were supplied as the composite material, Zytel-70G43L. From the formulation in the 1000 g batch prepara-

tion and the Br and Sb analyses, the sample composition (vol%) is: FR,10.86; Sb_2O_3, 1.42; nylon, 63.84; SiO_2, 23.56; and teflon, 0.31. More details are given in Table S6.0 (Appendix C). The vol% values are calculated from the component densities, assuming additive volumes.

3.2.2 Synchrotron X-ray Tomography

Tomography was done at the Advanced Photon Source (APS), Argonne National Laboratory, at bending magnet 13-BM-D in the GeoSoilEnviroCARS (GSECARS) group.[32] The X-ray beam was defined with a double Si-111 crystal monochromator yielding X-rays in the range of 8 to 65 keV with a beam size up to 50 mm wide and 5 mm high at the sample rotation stage.[33] Transmitted X-rays are converted to light with a 500 μm thick Ce:YAG scintillator and imaged with a 10X Mitutoyo microscope objective focused onto a 12-bit CCD camera (Roper Scientific MicroMAX 5 MHz) 1300x1100 pixels (pixel size 7 μm), a combination of optical magnification and binning yields 3.26 μm$\times$ 3.26 μm resolution. The sample was turned into a cylinder with diameter at Z=300 (see Figure 3.4) of approximately 1.52 mm, and mounted with clay atop the tomography sample rotation stage. A total of 720 images was collected as the sample was rotated from 0° to 179.5° by 0.5°, then 0.25° to 179.75° by 0.5°. Over the course of the imaging experiment, the X-ray flux at the sample was stabilized to within a few percent with two mechanisms: The APS beam current was held constant by operation in "top-up mode". The orientation of the second crystal of the monochromator was fine-tuned based on feedback from an ion-chamber X-ray flux indicator mounted immediately upstream of the sample. The sample was imaged at 12, 13.4, 17, 25, 30.43, 30.53, and 40 keV with an X-ray bandpass of $\Delta E/E = 10^{-4}$; the Br and Sb K-edges are at 13.47 and 30.49 keV, respectively.

Each CCD image was converted to an absorbance image with white field and average dark count correction. White field images were collected at intervals of 50 CCD images. The average dark field intensity was 100 counts. Sinograms were constructed; efforts were made to reduce zingers and ring artifacts. The slice reconstruction was done with a fast Fourier transform algorithm following a re-gridding from polar to cartesian coordinate systems.[33] The slices were assembled into a single reconstructed volume of 650x650x515 volume elements, each element containing a value of the voxel absorbance.

The 2D FFT-based reconstruction introduces scale ($\leq$6.5%), offset ($\leq 4\times10^{-4}$), and voxel size ($\leq$ 2.7%) errors in the voxel absorbances, and these were corrected by rescaling, Eq. 3.1, such that projections of representative slices yield absorbances in agreement with the original absorbance images. Note: when calculating projections from the 3D data sets, only the voxels within the reconstructed Radon-defined circle were used; the absorbance values for positions lying be-

tween the inscribed circle and the square defined by data storage were not used as these areas are not defined by the Radon transform. The correction factors were independently determined for each sample at each X-ray energy, as listed in Table S6.0.

$$\mathbf{A}_{xyzE}^{expt} = \left[\mathbf{A}_{xyzE}^{raw} \underrightarrow{voxel\ size} \mathbf{A}_{xyzE}\right] \times scale_E + offset_E \tag{3.1}$$

3D compositions are calculated from the experimental voxel absorbance and the voxel linear attenuation coefficients for the pure components. The NIST XCOM database reports mass attenuation coefficients [34] and these are converted to voxel linear attenuation coefficients for 3.26 μm voxels, as listed in Table 3.1. The voxel linear attenuation coefficients are equal to the mass attenuation coefficient (cm^2/g) $\times$ density (g/cm^3) $\times$ voxel dimension (cm).

Table 3.1: Calculated Voxel Linear Attenuation Coefficients of Pure Components for 3.26 μm Voxels.

energy/keV	nylon	SiO_2	**FR**	Sb_2O_3	teflon
12	0.000645	0.00795	0.0153	0.139	0.00285
13.4	0.000480	0.00578	0.0113	0.104	0.00207
17	0.000270	0.00289	0.0409	0.0545	0.00106
25	0.000135	0.000996	0.0146	0.0194	0.000412
30.43	0.000107	0.000604	0.00853	0.0115	0.000282
30.53	0.000107	0.000600	0.00845	0.0629	0.000280
40	0.0000866	0.000333	0.00403	0.0314	0.000190

From the corrected reconstructed volumes, a cuboid subvolume of $200 \times 200 \times 500$ voxels was selected for further analysis at the coordinate listed in Table S6.0. The subvolume was chosen to exclude air, however a small internal air-filled crack is captured in the subvolume and these air-filled voxels, 0.0703 % of the subvolume, were excluded by a binary air mask from further calculation.

3.2.3 3D Composition Calculation

The 3D chemical concentration distributions of **FR**, Sb_2O_3 and SiO_2 were calculated for all voxels in the subvolume (excluding the air crack voxels). A least-squares fit to the voxel linear

attenuation coefficients of Table 3.1 was made based on a Beer's law absorbance model [35]:

$$A^{calc}_{xyzE} = [\mathbf{FR}]_{xyz} \times A^{\mathbf{FR}}_E + [Sb_2O_3]_{xyz} \times A^{Sb_2O_3}_E + [SiO_2]_{xyz} \times A^{SiO_2}_E + \left(1 - [\mathbf{FR}]_{xyz} - [Sb_2O_3]_{xyz} - [SiO_2]_{xyz}\right) \times A^{matrix}_E \quad (3.2)$$

The X-ray absorbance for the nylon/teflon blend, A^{matrix}_E, is defined as a weighted average:

$$A^{matrix}_E = \frac{[nylon] \times A^{nylon}_E + [teflon] \times A^{teflon}_E}{[nylon] + [teflon]} \quad (3.3)$$

The parenthetical term in Eq. 3.2 restricts application of the model to voxels completely filled with sample, i.e., no cracks, voids, or surface voxels. The concentrations are determined with a minimization of difference between the experimental X-ray voxel absorbances and the calculated absorbances. The fit parameters, $[\mathbf{FR}]_{xyz}$, $[Sb_2O_3]_{xyz}$, and $[SiO_2]_{xyz}$, are constrained to have physically reasonable values between 0 and 100 vol%, inclusive. These 3D chemical compositions of $[\mathbf{FR}]_{xyz}$, $[Sb_2O_3]_{xyz}$, and $[SiO_2]_{xyz}$ of a $200 \times 200 \times 500$ cuboid were then used to analyze the radial concentration distribution about the fiberglass bundles.

3.2.4 Scanning Electron Microscopy

The sample was freeze-fractured in liquid N_2, and then coated with a 10 nm thick carbon layer for microscopy with a JEOL-840 SEM. The secondary electron images were acquired at 20 kV.

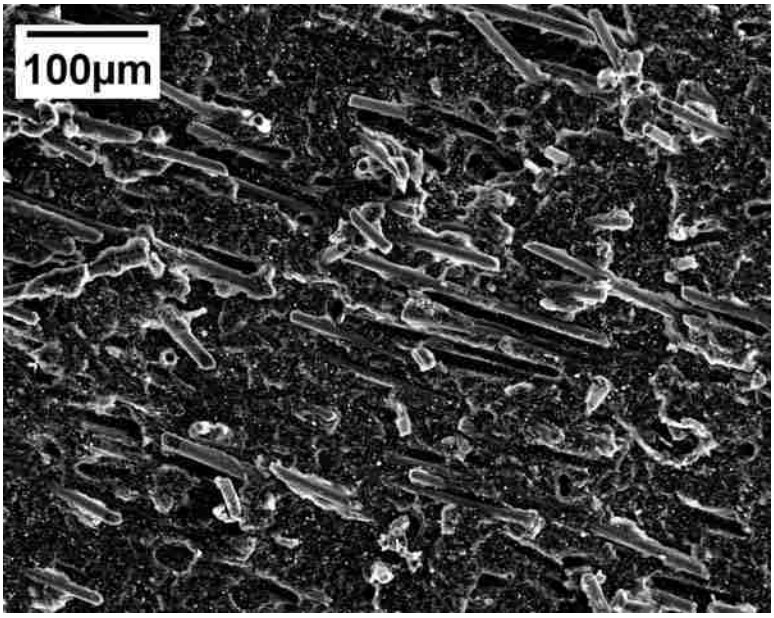


Figure 3.3: SEM image of freeze-fractured of the sample showing the embedded fiberglass in a nylon matrix. Each glass fiber is ≈10μm in diameter [36] and ≈100μm in length.

3.3 Results

3.3.1 X-ray Absorbance and Tomographic Reconstructions

The absorbance images of the sample at two representative X-ray energies below and above the Br K-edge, 13.4 and 25 keV, respectively, are shown in Figure 3.4. The light-colored striations observed at 13.4 keV are due to bundles of glass fibers; at this energy, SiO_2 has greater attenuation, 0.00578, than any other component, including the well-mixed **FR** in nylon. The bundle sizes are much larger than an individual fiber, probably consisting of 10-30 fibers. Interestingly, the bundles disappear at 25 keV and this will be attributed to accumulation of **FR** near the SiO_2 fibers, giving each bundle the same X-ray attenuation as **FR**/nylon. As noted in Figures 3.4c and 3.5, the fiberglass bundles are roughly aligned along the tomography Z-axis, which also corresponds to the extrusion axis of the cylindrical sample.

3.3.2 3D Binary Mask of SiO_2-Rich Voxels

Based on the sample formulation, some voxels will be completely filled with SiO_2 while other voxels may contain an FR/Sb_2O_3/nylon blend. To aid the analysis, a 3D binary mask is defined based on the voxels with high SiO_2 concentrations, as calculated in a later section. The 3D binary fiberglass mask is based on an isosurface, with island removal, using a threshold of [SiO_2]

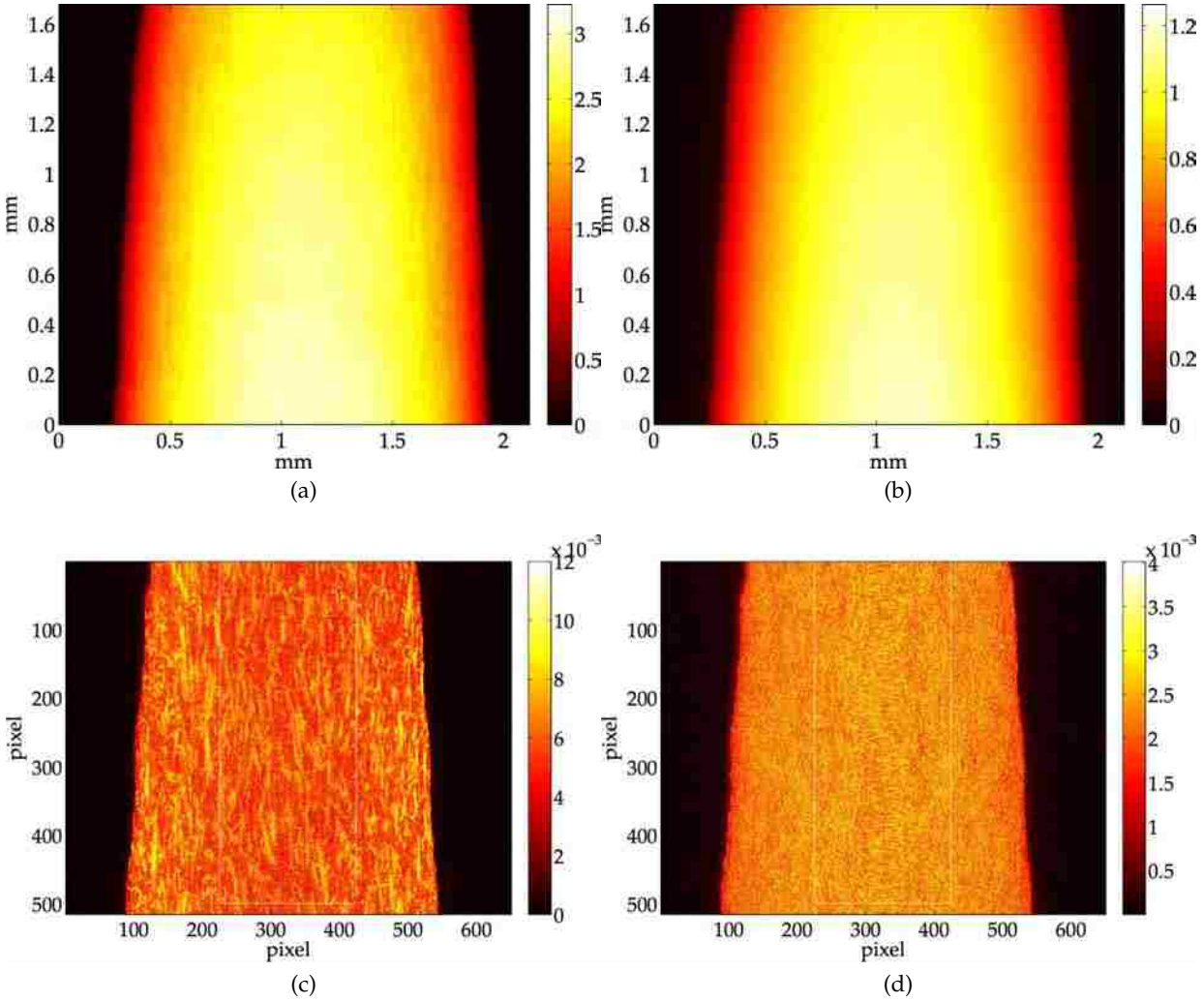


Figure 3.4: A comparison of 2D absorbance images, (a) 13.4 keV and (b) 25 keV, with XZ-slices from 3D reconstructions, (c) 13.4 keV and (d) 25 keV, respectively. The colorbars for the 2D images (a,b) are total absorbance in the sample while the colorbars for the slices (c,d) describe absorbance in the 3.26 μm voxels. The white rectangle (c,d) outlines the cuboid subvolume chosen for analysis. The light striations (a,c) are the fiberglass bundles which have, at 13.4 keV, more X-ray absorbance than the FR/nylon blend.

= 45 vol%. This mask shows fiberglass bundles, Figure 3.5, along with a few exposed ends of line segments representing the principal axis of some selected fiber bundles. The structure of the mask agrees with SEM images of a freeze-fractured sample, Figure 3.3. Line probes show most bundles have a diameter on the order of 20-30 μm. Ten bundles were selected for further analysis; bundle coordinates are given in Table S6.0.

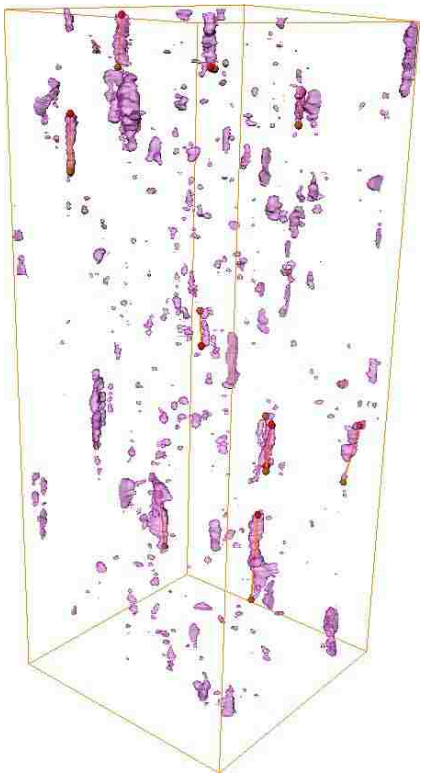


Figure 3.5: A $200 \times 200 \times 500$ cuboid binary mask for fiberglass based on the reconstructed volume for SiO_2. This mask is conservative with 12.58% of the cuboid voxels selected, compared to the 23.56 vol% of fiberglass in the sample based on formulation.

3.3.3 Voxel Absorbance Values

The motivating question in this 3D chemical analysis is the distribution of the **FR** and Sb_2O_3 synergist within the polymer blend and in the vicinity of the fiberglass bundles. A preliminary blending assessment can be obtained from an analysis of the average voxel absorbance as a function of X-ray energy, as shown in Figure 3.6. The average values are shown pairwise; polymer-rich voxels with filled symbols and SiO_2-rich with empty symbols, where voxel identification is based on the binary fiberglass mask, Figure 3.5.

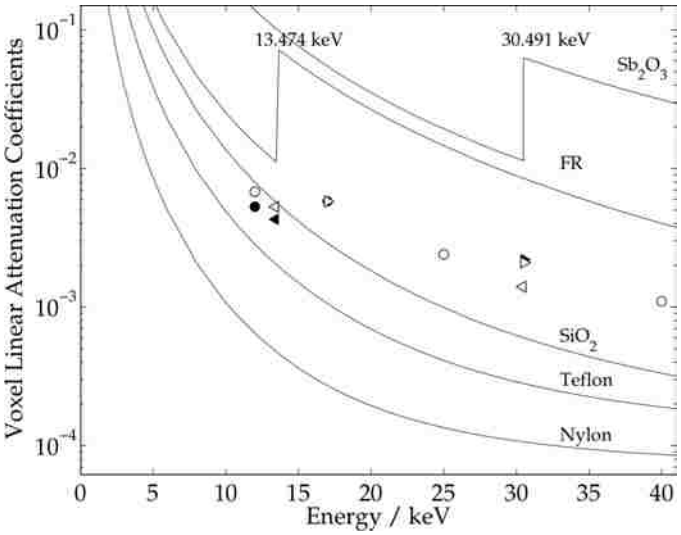


Figure 3.6: The traces show the calculated voxel linear attenuation coefficients for the pure components (see Table 3.1). The average voxel absorbance values for the two sets of voxels identified by the binary fiberglass mask: the polymer-rich voxels are represented by filled (●, ◀ , ▶) symbols and the average values for the SiO_2-rich voxels by open (○, ◁, ▷) symbols, which at some X-ray energies overlap the filled symbols.

The average voxel absorbance shows the expected increases at Br and Sb K-edges. An interesting feature is the contrast change between SiO_2-rich and polymer-rich voxels. At X-ray energies below the Br K-edge, SiO_2-rich voxels are more absorbing, as noted in Figures 3.4(a,c) and 3.6. Above the Br K-edge, both SiO_2-rich and polymer-rich voxels have the same average X-ray absorbance as shown in Figures 3.4(b,d) and 3.6. Later, this feature will be attributed to enhanced **FR** and Sb_2O_3 concentrations near the fiberglass bundles.

To assess the signal-to-noise of the reconstructed volumes, a histogram of the absorbance values of air-filled voxels is plotted in Figure 3.7. These voxels are defined by a 40 x200x300 cuboid outside the sample, but contained within the radius of reconstructed data, i.e, the poorly defined voxels at the corners of the slices were excluded. In the noise-free limit, all histograms of air-filled voxels would be delta functions centered at zero absorbance. Here, at X-ray energies at which the sample has the greatest average absorbance, the X-ray flux at the scintillator is smallest, causing a greater contribution from shot noise.[35]

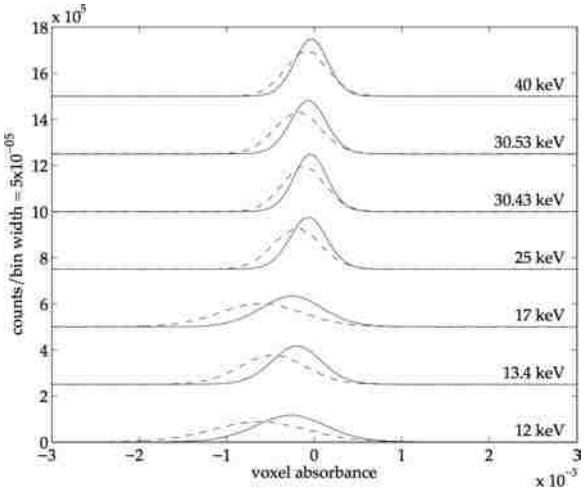


Figure 3.7: Histogram of air-filled voxels. The standard deviation is determined by photon counting statistics and noise introduced by the back projection reconstruction. Dashed lines show apparent absorbance prior to scale and offset corrections, Eq.3.1

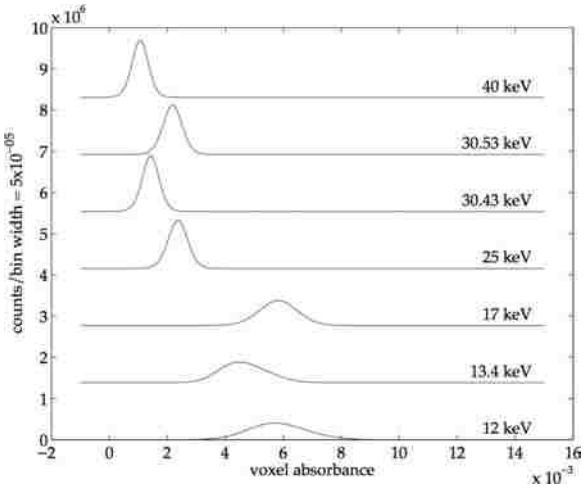


Figure 3.8: Voxel absorbances in the sample-filled $200 \times 200 \times 500$ cuboid for the polymer-rich voxels, selected based on the binary fiberglass mask, Figure 3.5. Overall, the sample becomes more transparent at higher X-ray energies, but with the Br and Sb K-edge absorbances evident at 17 and 30.53 keV, respectively.

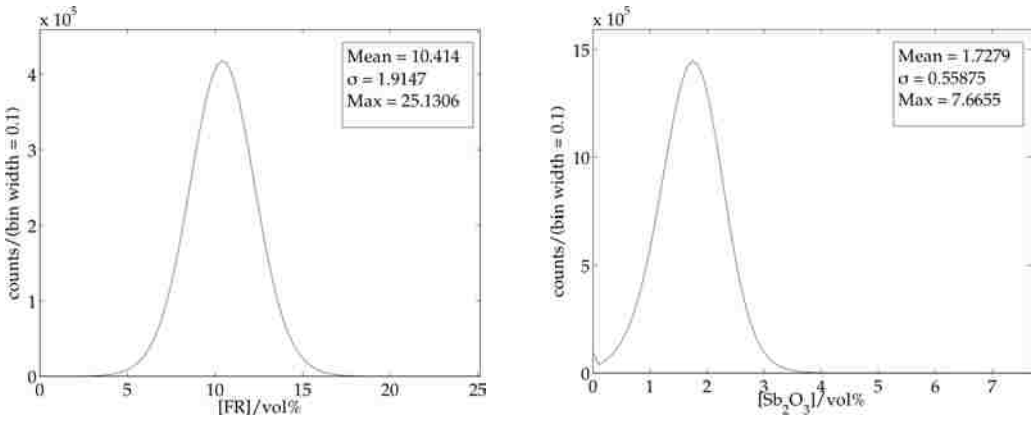
The absorbance histograms, Figure 3.8, of the polymer-rich voxels, chosen with the aid of the binary fiberglass mask, show the expected decrease in absorbance with increasing X-ray energy. The mean voxel absorbances at each X-ray energy are plotted in Figure 3.6 as filled symbols, and compared with the voxel linear attenuation coefficients for voxels fully occupied by **FR**, Sb_2O_3, or nylon. The widths of each mode are about twice that measured for air-filled voxels, Figure 3.7. In summary, the single, relatively narrow mode at all X-ray energies, as well as the mean absorbance values, indicate good blending of both **FR** and Sb_2O_3 in nylon.

3.3.4 3D Chemical Composition

The calculated vol% concentrations of **FR**, Sb_2O_3, and SiO_2 are obtained from the multi-energy X-ray tomography data and the three-parameter model, Equation 3.2, for almost all of the cuboid's 2×10^7 voxels, excepting the 0.07% of cuboid voxels identified as an internal air-filled crack. The calculated SiO_2 vol% concentration cuboid was converted into a binary fiberglass mask using a threshold value of [SiO_2] = 45 vol%, as shown in Figure 3.5; 12.58% of the cuboid voxels are thus identified as predominantly SiO_2. The binary fiberglass mask is conservative, as the sample formulation shows a fiberglass concentration of 23.56 vol%; the difference between the formulation and the binary mask is due to the small size of the glass fibers, about 10×100 μm, relative to the voxel size of 3.26 μm, creating many voxels only partially occupied by SiO_2. After application of the binary fiberglass and air-filled crack masks, 87.35% of the cuboid voxels are left as polymer-rich voxels, and appropriate for analysis of the **FR** and Sb_2O_3 3D chemical distributions.

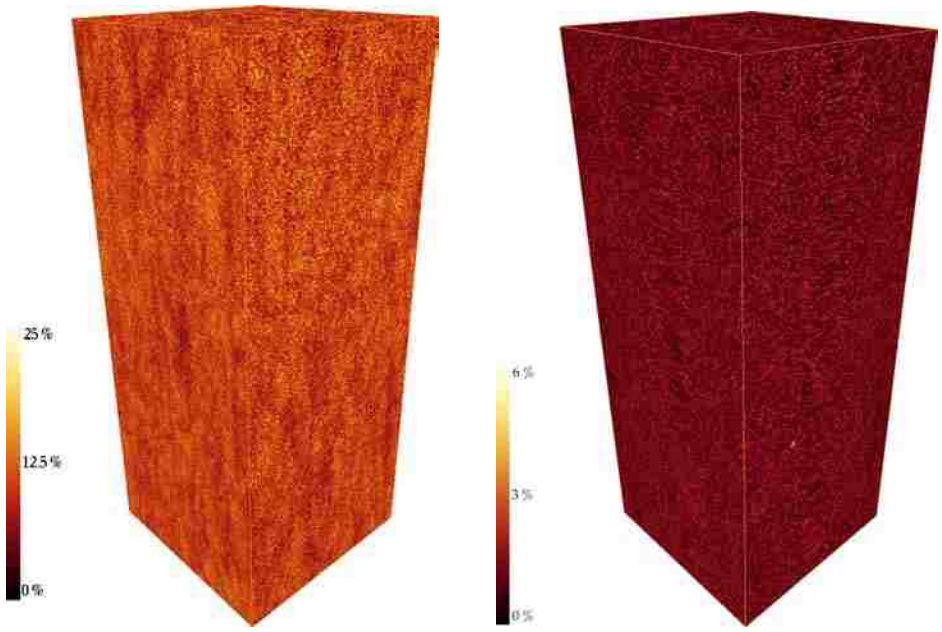
The calculated vol% concentrations of **FR** and Sb_2O_3 are shown as histograms in Figure 3.9. The mean FR concentration in the polymer-rich voxels is 10.41 vol% (σ = 1.9 vol%) compared with chemical analysis of 10.86 vol% for the entire sample. Similarly tomography shows [Sb_2O_3] = 1.72 vol% (σ = 0.56 vol%) while chemical analysis gives 1.42 vol%. Contributions to the standard deviation come from shot noise in the tomography experiment [35], from voxel-to-voxel fluctuations in the chemical concentration [10], and from the voxels partially or fully occupied by glass fibers, yet not removed from analysis by the binary fiberglass mask.

A 3D view of a well-blended mixture should look rather bland, as seen for both **FR** and Sb_2O_3 in Figure 3.10. Here, the binary fiberglass mask has not been applied and all calculated voxel concentrations are shown. The dark structures in **FR** are due to the fiberglass bundles. More importantly, the **FR** is well-blended throughout the sample, with no indication of regions of excessively high or low **FR** concentrations, and similarly for Sb_2O_3. A 3D view of the SiO_2 concentration is shown in Figure 3.5 as an isosurface at [SiO_2] = 45 vol%.



(a) (b)

Figure 3.9: Concentrations of (a) **FR** and (b) Sb_2O_3 in the 3.26 μm voxels in the cuboid. The mean values agree with sample formulation and the relatively small standard deviations indicate good blending with nylon.



(a) (b)

Figure 3.10: 3D renderings of the concentrations for (a) **FR** and (b) Sb_2O_3 for the $200 \times 200 \times 500$ cuboid subvolume with 3.26 μm voxels. The colorbars indicate concentrations in vol%.

3.3.5 Radial Concentration about Fiber Bundles

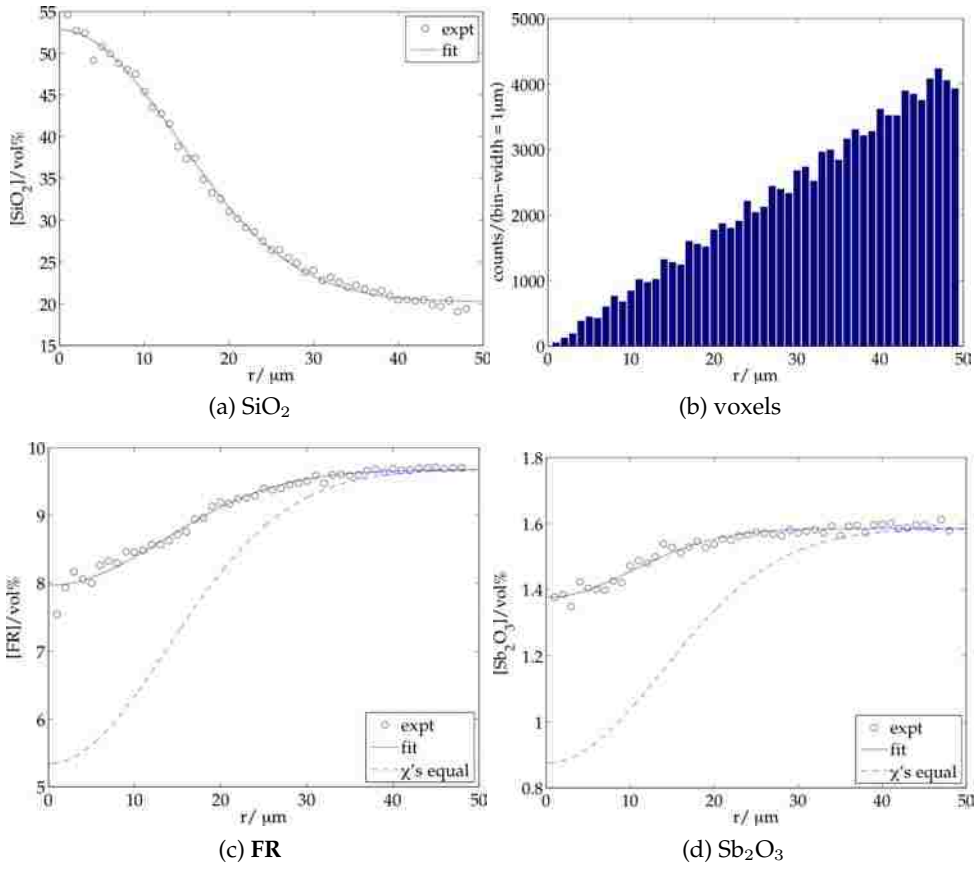
Figure 3.2 shows a simplistic model for **FR** concentrations near the fiberglass bundles. More sophisticated models such as the Flory-Huggins theory for polymer blends notes the importance of component interaction enthalpies.[37] In Figure 3.2, we assume no special affinity or aversion of **FR** with SiO_2, but simply $\chi_{FR:SiO_2} = \chi_{FR:nylon}$, and similarly for Sb_2O_3, where χ is the enthalpy term. Yet, the curious change in image structure with increasing X-ray energy—see

Figure 3.4(c,d) and also note the equivalence in mean voxel absorbance for SiO_2-rich and polymer-rich voxels in Figure 3.6—suggest an enhanced concentration of **FR** and Sb_2O_3 within the fiberglass bundles which implies an enthalpy-driven concentration gradient. How might we use the tomography results to assess **FR** the concentration in the vicinity of the fiberglass bundles?

Starting with the binary fiberglass mask, Figure 3.5, we identified ten well-formed bundles, all relatively isolated. Each bundle is cylindrical, roughly 40 μm in diameter. The line segments defining the cylinder axis are listed in Table S6.0. All voxels surrounding the cylinder axes up to a radius of 50 voxels were located and assessed for collision with neighboring glass fibers, again using the binary fiberglass mask. A collision will remove a voxel from further consideration, as well as other voxels along that same radial vector away from the cylinder axis. The voxels retained were sorted by Euclidean distance from a cylinder axis, and the corresponding voxel concentrations were used to generate the radial concentrations plots shown in Figure 3.11. A simple histogram of the number of voxels contributing to a radial concentration plot about a cylindrical axis would normally show an r^2 increase in voxel count. However, due to collisions with neighboring fiberglass bundles, the r dependence is closer to linear, as shown in Figure 3.11b. The voxel selection procedure is described in the supplementary material.

The SiO_2 radial concentration, Figure 3.11a, is averaged over the ten fiber bundles and shows several features: (**1**) The average fiber bundle radius is about 20 μm, in agreement with Figure 3.5. (**2**) The fibers are loosely arranged into bundles, as the maximum SiO_2 concentration is 52.8 vol%, indicating significant polymer flow into the fiber bundles, consistent with the SEM imaging, Figure 3.3. (**3**) At r = 50 μm, the SiO_2 concentration drops to about 20 vol%, close to the sample formulation value of 23.31 vol%. The SiO_2 radial concentration is fitted to a Gaussian, $[SiO_2]_r = ae^{-(r/b)^2} + c$, and the coefficients are listed in Table S6.0; the value of b is 19.5 μm.

The radial concentrations of **FR** and Sb_2O_3 in the vicinity of the ten selected fiber bundles are plotted in Figures 3.11c and 3.11d, respectively. As expected, the concentrations decrease at small r, when voxels lying within a fiber bundle are inspected for **FR** and Sb_2O_3 concentrations; many of these voxels are partially occupied with SiO_2. However, the decrease in **FR** and Sb_2O_3 concentrations is not as dramatic as predicted based upon the volume of SiO_2 present. Shown in both figures are predicted concentrations assuming a simple space-filling model applies, e.g. Figure 3.2. For both **FR** and Sb_2O_3, the concentrations are near doubled within the fiber bundles, relative to the space-filling model. One possible explanation is the hydrophobic coatings applied to the glass fibers leads to precipitation of **FR** and Sb_2O_3 onto the fibers.



(a) SiO_2 (b) voxels

(c) **FR** (d) Sb_2O_3

Figure 3.11: The radial concentration of (a) SiO_2 as a function of distance from the fiber bundle axes, sampled over 10 fiber bundles. Collision detection reduces the number of sampled voxels (b) at large r. The radial concentrations for (c) **FR** and (d) Sb_2O_3, and comparisons with a space-filling model based on the radial concentration of SiO_2.

3.4 Conclusion

Three-dimensional chemical analysis is shown here as an effective method for studying a polymer blend, even with a composite material reinforcement. An analysis procedure was developed to generate radial concentrations measurements about cylindrical composites, measurements which reveal unique information about the polymer blend structure.

3.5 Bibliography

[1] F. Natterer and E. L. Ritman. Past and future directions in X-ray computed tomography (CT). *International Journal of Imaging Systems and Technology*, 12(4):175–187.

[2] Gabor T. Herman and Attila Kuba. *Discrete Tomography*. Birkhäuser, Boston.

[3] Charles L Epstein. *Mathematics of Medical Imaging*. Prentice Hall.

[4] Gabriele Lohmann. *Volumetric Image Analysis*. Wiley-Teubner, New York.

[5] Guillermo Sapiro. *Geometric Partial Differential Equations and Image Analysis*. Cambridge University Press.

[6] P. Spanne, J. F. Thovert, C. J. Jacquin, W. B. Lindquist, K. W. Jones, and P. M. Adler. Synchrotron computed microtomography of porous media: topology and transports. *Phys. Rev. Lett.*, 73(14):2001–4.

[7] W. B. Lindquist, S.-M. Lee, D. A. Coker, K. W. Jones, and P. Spanne. Medial Axis Analysis of Void Structure in Three Dimensional Tomographic Images of Porous Media. *Journal of Geophysical Research B*, 101:8297–8310.

[8] L. G. Butler, J. W. Owens, F. K. Cartledge, R. L. Kurtz, G. R. Byerly, A. J. Wales, P. L. Bryant, E. F. Emery, B. Dowd, and X. G. Xie. Synchrotron X-ray microtomography, electron probe microanalysis, and NMR of toluene waste in cement. *Environmental Science & Technology*, 34(15):3269–3275.

[9] C. G. Schroer, P. Cloetens, M. Rivers, A. Snigirev, A. Takeuchi, and W. B. Yun. High-resolution 3D imaging microscopy using hard x-rays. *MRS Bulletin*, 29(3):157–165.

[10] K. Ham, H. Jin, R. I. Al-Raoush, X. G. Xie, C. S. Willson, G. R. Byerly, Larry S. Simeral, M. L. Rivers, R. L. Kurtz, and L. G. Butler. Three-Dimensional Chemical Analysis with Synchrotron Tomography at Multiple X-ray Energies: Brominated Aromatic Flame Retardant and Antimony Oxide in Polystyrene. *Chemistry of Materials*, 16(4032-42).

[11] M. A. Le Gros, G. McDermott, and C. A. Larabell. X-ray tomography of whole cells. *Current Opinion in Structural Biology*, 15(5):593–600.

[12] R Marschallinger. Three-dimensional reconstruction and modeling of microstructures and microchemistry in geological materials. *Scanning*, 20(2):65–73.

[13] T. Muller-Reichert, H. Hohenberg, E. T. O'Toole, and K. McDonald. Cryoimmobilization and three-dimensional visualization of *C-elegans* ultrastructure. *Journal of Microscopy-Oxford*, 212:71–80.

[14] Pittsburgh Supercomputer Center. The Visible Human Project at the National Library of Medicine `http://www.nlm.nih.gov/research/visible/visible\_human.html` (accessed August, 2008).

[15] R. Heldele, S. Rath, L. Merz, R. Butzbach, M. Hagelstein, and J. Hausselt. X-ray tomography of powder injection moulded micro parts using synchrotron radiation. *Nuclear Instruments & Methods in Physics Research Section B-Beam Interactions with Materials and Atoms*, 246(1):211–216.

[16] P. J. Schilling, B. P. R. Karedla, A. K. Tatiparthi, M. A. Verges, and P. D. Herrington. X-ray computed microtomography of internal damage in fiber reinforced polymer matrix composites. *Composites Science and Technology*, 65(14):2071–2078.

[17] A. Momose, A. Fujii, H. Kadowaki, and H. Jinnai. Three-dimensional observation of polymer blend by X-ray phase tomography. *Macromolecules*, 38(16):7197–7200.

[18] E. Maire, N. Gimenez, V. Sauvant-Moynot, and H. Sautereau. X-ray tomography and three-dimensional image analysis of epoxy-glass syntactic foams. *Philosophical Transactions of the Royal Society a-Mathematical Physical and Engineering Sciences*, 364(1838):69–88.

[19] R. A. Alderden, H. R. Mellor, S. Modok, M. D. Hall, S. R. Sutton, M. G. Newville, R. Callaghan, and T. W. Hambley. Elemental tomography of cancer-cell spheroids reveals incomplete uptake of both Platinum(II) and Platinum(IV) complexes. *Journal of the American Chemical Society*, 129:13400–13401.

[20] M. H. Gass, K. K. K. Koziol, A. H. Windle, and P. A. Midgley. Four-dimensional spectral tomography of carbonaceous nanocomposites. *Nano Letters*, 6(3):376–379.

[21] H. Friedrich, J. R. A. Sietsma, P. E. de Jongh, A. J. Verkleij, and K. P. de Jong. Measuring location, size, distribution, and loading of NiO crystallites in individual SBA-15 pores by electron tomography. *Journal of the American Chemical Society*, 129(33):10249–10254.

[22] A. Elmoutaouakkil, G. Fuchs, P. Bergounhon, R. Péres, and F. Peyrin. Three-dimensional quantitative analysis of polymer foams from synchrotron radiation x-ray microtomography. *J. Phys. D: Appl. Phys.*, 36:A37–43.

[23] J. H. Yoon, T. Kawamura, S. Takeya, S. K. Jin, Y. Yamamoto, T. Komai, M. Takahashi, A. V. Nawaby, and Y. P. Handa. Probing fickian and non-fickian diffusion of CO2 in poly(methyl methacrylate) using in situ Raman spectroscopy and microfocus X-ray computed tomography. *Macromolecules*, 37(25):9302–9304.

[24] K. Yamauchi, K. Takahashi, H. Hasegawa, H. Iatrou, N. Hadjichristidis, T. Kaneko, Y. Nishikawa, H. Jinnai, T. Matsui, H. Nishioka, M. Shimizu, and H. Fukukawa. Microdomain morphology in an ABC 3-miktoarm star terpolymer: A study by energy-filtering TEM and 3D electron tomography. *Macromolecules*, 36(19):6962–6966.

[25] A. Takano, S. Wada, S. Sato, T. Araki, K. Hirahara, T. Kazama, S. Kawahara, Y. Isono, A. Ohno, N. Tanaka, and Y. Matsushita. Observation of cylinder-based microphase-separated structures from ABC star-shaped terpolymers investigated by electron computerized tomography. *Macromolecules*, 37(26):9941–9946.

[26] H. A. Barnett, K. Ham, and L. G. Butler. Synchrotron X-ray tomography for 3D chemical diffusion measurement of a flame retardant in polystyrene. *Nuclear Instruments & Methods in Physics Research Section a-Accelerators Spectrometers Detectors and Associated Equipment*, 582(1):202–204.

[27] K. Ham, H. A. Barnett, and L. G. Butler. Burning issues in tomography analysis. *Computing in Science & Engineering*, 10(2):78–81.

[28] M. Simonson. Effect of FR enclosures on the fire behavior of TV sets. *Polymeric Materials Science and Engineering*, 83:90–91.

[29] D. De Schryver, S. D. Landry, and J. S. Reed. Latest developments on the flame retardancy of engineering thermoplastics - SAYTEX HP-7010 (brominated polystyrene) in glass filled engineering thermoplastics. *Polymer Degradation and Stability*, 64(3):471–477.

[30] A. Litzenburger. Criteria for, and examples of optimal choice of flame retardants. *Polymers & Polymer Composites*, 8(8):581–592.

[31] T. Randoux, J. C. Vanovervelt, H. Van den Bergen, and G. Camino. Halogen-free flame retardant radiation curable coatings. *Progress in Organic Coatings*, 45(2-3):281–289.

[32] P. Spanne and M. L. Rivers. Computerized microtomography using synchrotron radiation from the NSLS. *Nucl. Instrum. Methods Phys. Res., Sect. B*, B24-25(Pt. 2):1063–7.

[33] M. L. Rivers. Rivers, M. L. Tutorial Introduction to X-ray Computed Microtomography `http://www-fp.mcs.anl.gov/xray-cmt/rivers/tutorial.html` (accessed August 2008).

[34] NIST. "XCOM: Photon Cross Sections Database" `http://physics.nist.gov/PhysRefData/Xcom/Text/XCOM.html` (accessed August 2008).

[35] K. Ham and L. G. Butler. Algorithms for three-dimensional chemical analysis via multi-energy synchrotron X-ray tomography. *Nuclear Instruments & Methods in Physics Research Section B-Beam Interactions with Materials and Atoms*, 262(1):117–127.

[36] J. A. Davis and T. T. Laskowski. Materials Analysis of Fiberglass-Reinforced Plastics by Scanning Electron Microscopy. *Microbeam Analysis*, 7(7):193–196.

[37] C. C. Hsu and J. M. Prausnit. Thermodynamics of Polymer Compatibility in Ternary-Systems. *Macromolecules*, 7(3):320–324.

Chapter 4

Synchrotron X-Ray Tomography for 3D Chemical Diffusion Measurement of a Flame Retardant in Polystyrene

4.1 Introduction

There is significant interest in the chemical process known as ”blooming”, diffusion and nucleation of small molecules on the surface of a polymer blend. Even food materials are subject to bloom, for instance, the appearance of cocoa butter on chocolate is a diffusional process. These extremely slow processes call into question the models for small molecule motion through glassy polymers. Recently, synchrotron X-ray tomography has developed into an excellent method for three-dimensional imaging of polymer blends at the micrometer distance scale.[1] To investigate the chemical process of blooming, new procedures are developed using X-ray tomography to assess the flame retardant distribution in the sample. More expensive flame retardants such as BT-93 ^{TM}are known to be stable in polystyrene and polypropylene, and are known as non-bloomers. In this preliminary work we study the dissolution of BT-93 ^{TM}in polystyrene.

4.2 Experimental Section

The blend was made from 1057 g polystyrene and 55.638 g Saytex^{TM} BT-93, a brominated phthalimide dimer, in a screw extruder at 220 °C, 450 psi and 39 rpm, yielding ≈2.35 mm diameter cylinders. Two of the samples were then separately annealed in vacuum at 90 °C for 1 and 5 days. All samples were imaged at room temperature using synchrotron X-ray tomography at 9 μm resolution at 12 and 14.5 keV.

A constrained least-squares fit for the BT-93 and polystyrene concentrations to the voxel linear attenuation coefficients was made based on a Beer's law absorption model (Eq 4.1). The attenuation coefficients A_E^{BT93} and $A_E^{Polystyrene}$, are calculated from molecular formula and density.

$$A_{xyzE}^{calc} = [BT93]_{xyz} \times A_E^{BT93} + \left(1 - [BT93]_{xyz}\right) \times A_E^{Polystyrene} \tag{4.1}$$

To calculate diffusion measurements around the pockets of high concentration in the sample, new methods had to be developed. First a simple thresholding of the data is needed to separate the high concentration areas from the low concentration areas. The area's of high concentration are irregularly shaped domains, the domains are identified and then counted/labeled with a connected components algorithm.[2] For each labeled domain, its center-of-mass and distance/direction to its nearest neighbor is calculated.

Before we can calculate concentration gradients, the air bubble impurities must be noted. To identify the air bubbles, a procedure similar to the domain labeling was taken. Another thresholding of the values near zero absorption will identify the air bubbles. Connected components algorithm was also used to count and label the air bubbles. The center of mass was then calculated for each air bubble but no distance measurements were needed. Instead the coordinates of the air bubbles were compared with the coordinates of the labeled domains. If any of them overlapped, then those voxels were removed from the diffusion data. Voxels in between the center of a labeled domain and an air bubble are valid but those voxels that are between the air bubble and edge of the domain are not. Care was taken to exclude these voxels as well. Finally, concentration gradients are plotted centered at each labeled domain's center-of-mass and extending halfway to the nearest neighbor, but excluding voxels near air bubbles.

4.3 Results and Discussion

The concentration gradients are initially assessed with a simple Gaussian fit. [3] The fitted data follows expected trends for a diffusion process. As annealing time increases the bulk average concentration increases (represented by c in the fit). The a value decreases with anneal time indicating that the concentration domains have a lower average value. These trends are displayed in Figure 4.2. Three-dimensional chemical diffusion analysis via synchrotron x-ray tomography is shown here as an effective method for studying the dissolution process. The study of the diffusion of the BT-93 concentrations about high concentration domains shows that anneal time has a definite effect on the diffusion rates. The major focus for future work will continue to study this

sample but a longer anneal times to determine the full extent of the ”blooming” process for flame retardants such as decabromo diphenyl oxide.

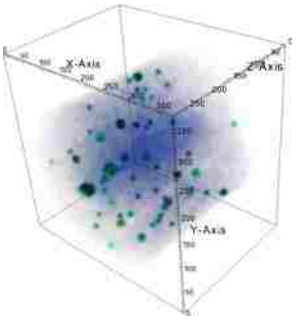


Figure 4.1: XY slice of sample composition (Z = 230) of the non-annealed sample in volume percents of BT-93. Bright spots correspond to high concentrations of the BT-93 and indicate an imperfect blending process.

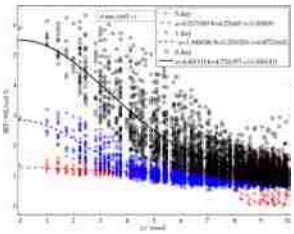


Figure 4.2: A plot of the BT-93 concentrations a function of distance about the center of high concentration domains dispersed throughout the sample and the exponential fits applied to the data.

4.4 Bibliography

[1] K. Ham, H. Jin, R. I. Al-raoush, X. G. Xie, C. S. Willson, G. R. Byerly, L. S. Simeral, M. L. Rivers, R. L. Kurtz, and L. G. Butler. Three-dimensional Chemical Analysis With Synchrotron Tomography At Multiple X-ray Energies: Brominated Aromatic Flame Retardant and Antimony Oxide In Polystyrene. *Chemistry Of Materials*, 16(4032-42), 2004.

[2] G. Lohmann. *Volumetric Image Analysis*. Wiley-Teubner, New York, 1998.

[3] J. Crank. *The Mathematics Of Diffusion*. Oxford University Press, 1975.

Chapter 5

Measurement of Diffusion Constants with Synchrotron X-ray Tomography

5.1 Introduction

Flame retardants in high-impact polystyrene have many applications including the manufacture of CRT cases and laser printers; these cases can include to 30 wt% bromine in the form of brominated aromatics. Phase stabilities, rates of diffusion, and diffusion activation energies are largely unknown. Situations in which the observer notices changes in surface color and feel, the result of flame retardant diffusion and nucleation on the surface: "blooming". Even food materials are subject to bloom, for instance, the white powder on chocolate is a diffusional process.

Blooming is a transportation process of mass[1], in this case small molecules of hexabromobenzene, being transported onto the surface of a polymer blend from the interior space of the polymer. It is known that changes in the blend morphology in polymers, such as blooming, can lead to the reduction or even elimination of the desired polymer blend functions. Flame retardant blooming is cause for concern in industry and is also an interesting scientific problem when viewed as a diffusion process. Synchrotron X-ray tomography [2] has developed into an excellent method for three-dimensional imaging of materials [3, 4], and has shown particularly good results with polymer blends.[5, 6, 7] Herein, we estimate the sensitivity of synchrotron X-ray tomography to diffusion coefficients of a hexabromobenzene/o-terphenyl blend. These extremely slow processes call into question the models for small molecule motion through glassy polymers. Bloom is one example of small molecule diffusion through rubbery and glassy polymers. Diffusion through rubbers is currently understood as a cooperative process coupled to polymer chain segmental motion. Computer simulations will help us interpret the images and guide the im-

provement of techniques for image analysis of diffusion processes.

5.2 Experimental and Computational Section

5.2.1 Sample Preparation

The blend was made from 3.974 g o-terphenyl (density 1.1 $\frac{g}{cm^3}$) and 0.20 g hexabromobenzene (density 3.543 $\frac{g}{cm^3}$), for an end result of 4.8 % by weight mixture. The sample mixture was divided into four approximately equal parts. Then each sample was separately baked at 50°C, 53°C, 56°C and 59 °C for 2 hours. All samples were imaged at room temperature using synchrotron X-ray tomography at 9 μm resolution at 30 keV.

5.2.2 Tomography Reconstruction

Tomography was done at the Center for Advanced Microstructures and Devices (CAMD), Louisiana State University, at the tomography beamline. The X-ray beam energy was selected with a double crystal monochromator yielding X-rays with 30 keV energy. A total of 720 images was collected as the sample was rotated from 0° to 179.5° by 0.25°. The procedure followed is the procedure listed in Section 2.2 and is described in the literature. [5, 8] The order of angle acquisition was not linear but determined by the golden ratio.[8] This technique provides a safety net, in case of beam failures a reasonable reconstruction may still be possible without all 720 angles. The four samples were studied at the single X-ray energy 30 keV but each sample was imaged at two different heights.

The reconstruction followed the typical reconstruction for computerized tomographic imaging. [9] This method is described in full in Section 2.1 but in short, each CCD image was converted to an absorption image with white field and average dark count correction. White field images were collected at intervals of 30 CCD images. Sinograms were constructed, efforts were made to reduce artifacts, and the transmission values were converted to line integrals of the absorption. The slice reconstruction was done with a filtered back-projection algorithm [10], iradon, from the Matlab Image Analysis toolbox. The slices were assembled into a single reconstructed volume of 461×461×161 volume elements with resolutions of $8.798\times10^{-4}\,\frac{cm}{voxel}$, each sample was collected at two heights effectively doubling the volume size for each. Figure 5.1 shows a slice from one of the reconstructed volumes and Figure 5.2 shows a 3D image of a segmented reconstruction volume.

For the purpose of calculating chemical concentrations, the sample contains a single im-

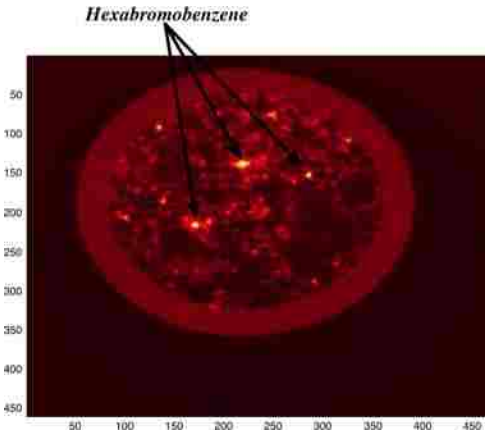


Figure 5.1: A 461×461 XY slice from the reconstructed volume for the 50 ^{o}C sample. Areas that contain voxels that have high hexabromobenzene concentration are labeled.

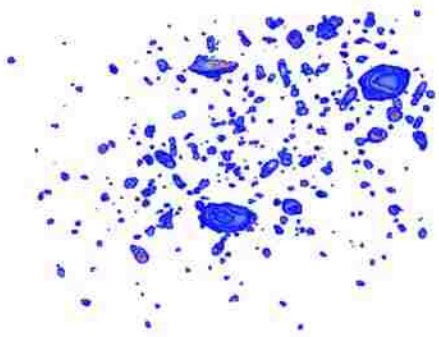


Figure 5.2: Domains of high concentration hexabromobenzene segmented out of a 461×461×161 reconstructed volume for the 50 ^{o}C sample. Segmentation was done with a simple threshold algorithm as described in Chapter 2. Various ranges of sizes and shapes are shown indicating a need to ”clean up” the data before fitting.

portant element: Br, in the hexabromobenzene. The simple nature of this two-component mixture makes calculating the volume percent of hexabromobenzene at a voxel a trivial matter of algebra:

$$\begin{aligned} Abs^{sample} &= \left([C_6Br_6] \times Abs_E^{C_6Br_6}\right) + \left(oT] \times Abs_E^{C_6Br_6}\right) \\ Abs^{sample} &= \left([C_6Br_6] \times Abs_E^{C_6Br_6}\right) + \left((1 - [C_6Br_6]) \times Abs_E^{oT}\right)] \\ [C_6Br_6] &= \frac{Abs^{sample} - Abs_E^{oT}}{Abs_E^{C_6Br_6} - Abs_E^{oT}} \end{aligned} \tag{5.1}$$

where oT stand for o-terphenyl, Abs^X is the absorption of X at that energy and voxel and $[X]$ is the concentration of X at that voxel. For the diffusion studies no effort was taken to convert the absorption data to volume percents as done in previous chapters. This was done partly because the absorption data can easily be used for tracking the displacement of the bromine in the o-terphenyl. Equation 5.1 is included for completeness.

5.2.3 3D Image Analysis

Figure 5.2 shows an extremely chaotic and messy system of hexabromobenzene domains. In order to effectively fit our data to the chosen diffusion model we have to select the data we want. Creating a three-dimensional image processing algorithm will allow us to extract the voxels of interest out of our volume in a automatic and reliable fashion.[11] In order to extract the correct voxels that will work with our chosen diffusion model we will need to enforce the following conditions on our algorithm:

- Domains of selected voxels must be as spherical as possible.
- The domains can not overlap with other domains.
- Domains should be isolated from air bubbles, cracks etc..

The rational behind these conditions are simple. Since we are employing a diffusion model based on a spherical function the restriction of spherical domains is logical. Domains that overlap would have concentration gradients that oppose each other and this is not allowed in our model. Isolation from cracks and air bubbles is needed since their presence would alter the concentration gradients within the domains. These conditions are satisfied with the algorithm presented in Section 2.6.6.

Table 5.1: Results of Domain Extraction Algorithm

Properties	50 ^{o}C	53 ^{o}C	56 ^{o}C	59 ^{o}C
Number of Domains	74	88	68	72
Mean Volume ($\times 10^{-6}$ cm^3)	0.0512	0.0758	0.0525	0.0767
Mean Vol%	33.45	36.02	32.46	30.89
Vol% σ	3.014	2.96	2.52	3.21

The results of image analysis algorithm are summarized in Table 5.1. The results generated were of good quality with little need to do repeat analysis. All results were saved down to a HDF5 format and imported into Mathematica for all further analysis.

5.2.4 Diffusion Measurements

For modeling the diffusion of the flame retardant in the polymer we chose a three dimensional point source model as described in Section 2.5. The model used is a three-dimensional point source model as described by Crank [1]:

$$C = ae^{\frac{-r^2}{4Dt}} + c \tag{5.2}$$

Where C is concentration, c is the value that the function will converge on as r goes to infinity and can be thought of as the average concentration, D is the diffusion coefficient. The fit was performed in Mathematica using the NonlinearRegress function from the Nonlinear Regression package. This function also allowed the calculation of error associated with each parameter that was fit, a,D and c. The t value in Equation 5.2 was the time of the bake intervals, 7200 seconds (2 hours).

5.3 Results and Discussion

Concentration gradients are plotted centered at each labeled domain's center-of-mass and extending halfway to the nearest neighbor. The points are then fit to the Equation 5.2. Figure 5.3 shows the four samples hexabromobenzene concentrations as a function of distance about the center of mass of their domains and the plot of the exponential fits.

Figure 5.4 shows all four exponential fits plotted on the same axis. This shows that the effect of raising the temperature does cause changes in the samples diffusion process. The expo-

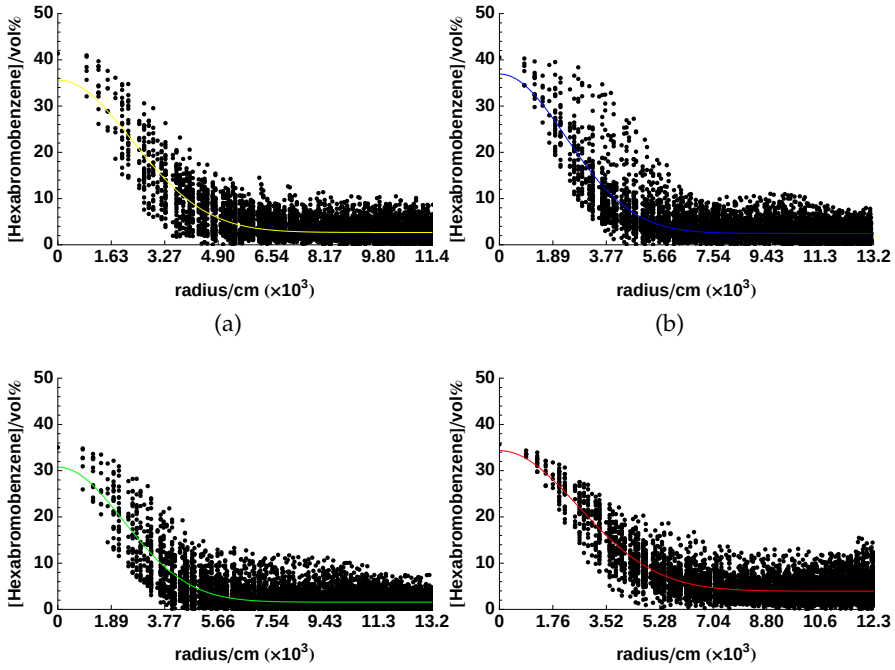


Figure 5.3: a) Hexabromobenzene concentrations as a function of distance about the center of mass for the sample baked at 50 ^{o}C b) 53 ^{o}C . c) 56 ^{o}C d) 59 ^{o}C

nential fit results are listed in Table 5.2.

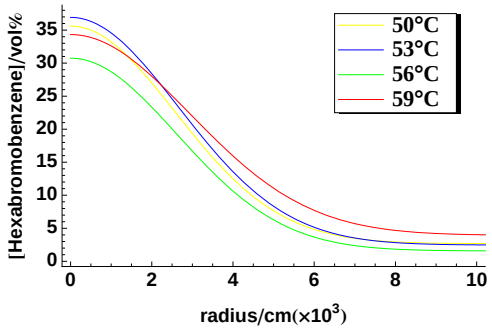


Figure 5.4: All four exponential fits plotted on the same axis. Decreases in the a fit parameter follow expected trends with the exception of the 53 ^{o}C sample but the difference is minor. Diffusion coefficients get progressively larger with increasing temperature indicating that diffusion is occurring.

Table 5.2: Fit Results with Error Reports

^oC	Regression	a	D	c
50	Estimate	32.9392	4.68926×10^{-9}	2.67611
	Error	0.358835	1.12878×10^{-10}	0.0255998
	CI	(32.2358, 33.6426)	(4.468×10^{-9}, 4.91053×10^{-9})	(2.62593, 2.72629)
53	Estimate	34.4687	5.35533×10^{-9}	2.45985
	Error	0.328795	1.10777×10^{-10}	0.0196567
	CI	(33.8242, 35.1132)	(5.13819×10^{-9}, 5.57247×10^{-9})	(2.42132, 2.49838)
56	Estimate	31.0002	6.40302×10^{-9}	1.66769
	Error	0.325398	1.4595×10^{-10}	0.0191292
	CI	(30.3624, 31.638)	(6.11693×10^{-9}, 6.6891×10^{-9})	(1.6302, 1.70519)
59	Estimate	30.385	7.9803×10^{-9}	3.93877
	Error	0.345672	2.17508×10^{-9}	0.0329525
	CI	(29.7074, 31.0626)	(7.55393×10^{-9}, 8.40668×10^{-9})	(3.87417, 4.00336)

5.4 Conclusions

The concentration gradients are initially assessed with a simple Gaussian fit. The fitted data follows expected trends for a diffusion process. As annealing time increases the bulk average concentration increases (represented by c in the fit). The a value decreases with anneal time indicating that the concentration domains have a lower average value. These trends are displayed in Figure 5.4. Three-dimensional chemical diffusion analysis via synchrotron x-ray tomography is shown here as an effective method for studying the dissolution process. The study of the diffusion of the hexabromobenzene concentrations about high concentration domains shows that temperature has a definite effect on the diffusion rates.

5.5 Bibliography

[1] J. Crank. *The Mathematics of Diffusion*. Oxford University Press, 1975.

[2] K. L. D'Amico, H. W. Deckman, J. H. Dunsmuir, B. P. Flannery, and W. G. Roberge. X-Ray Microtomography with Monochromatic Synchrotron Radiation. *Rev. Sci. Instrum.*, 60(7, Pt. 2A):1524–6, 1989.

[3] L. G. Butler, J. W. Owens, F. K. Cartledge, R. L. Kurtz, G. R. Byerly, A. J. Wales, P. L. Bryant, E. F. Emery, B. Dowd, and X.G. Xie. Synchrotron X-Ray Microtomography, Electron Probe Microanalysis, and NMR of Toluene Waste in Cement. *Environmental Science & Technology*, 34(15):3269–3275, 2000.

[4] R. Marschallinger. Three-Dimensional Reconstruction and Modeling of Microstructures and Microchemistry in Geological Materials. *Scanning*, 20(2):65–73, 1998.

[5] K. Ham, H. Jin, R. I. Al-raoush, X. G. Xie, C. S. Willson, G. R. Byerly, Larry S. Simeral, M. L. Rivers, R. L. Kurtz, and L.G. Butler. Three-Dimensional Chemical Analysis with Synchrotron Tomography at Multiple X-Ray Energies: Brominated Aromatic Flame Retardant and Antimony Oxide in Polystyrene. *Chemistry of Materials*, 16(4032-42), 2004.

[6] H. A. Barnett, K. Ham, and L.G. Butler. Synchrotron X-Ray Tomography for 3DChemical Diffusion Measurement of a Flame Retardant in Polystyrene. *Nuclear Instruments & Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 582(1):202–204, 2007.

[7] A. Momose, A. Fujii, H. Kadowaki, and H. Jinnai. Three-Dimensional Observation of Polymer Blend by X-Ray Phase Tomography. *Macromolecules*, 38(16):7197–7200, 2005.

[8] K. Ham and L.G. Butler. Algorithms for Three-Dimensional Chemical Analysis Via Multi-Energy Synchrotron X-Ray Tomography. *Nuclear Instruments & Methods in Physics Research Section B-Beam Interactions with Materials and Atoms*, 262(1):117–127, 2007.

[9] M. Slaney and K.C. Avinash. *Principles of Computerized Tomographic Imaging*. IEEE Press, 1988.

[10] Frank Natterer and Frank Wubbeling. *Mathematical Methods in Image Reconstruction*. SIAM, Philadelphia, PA, 2001.

[11] Gabriele Lohmann. *Volumetric Image Analysis*. Wiley-Teubner, New York, 1998.

Chapter 6

Conclusions and Expectations

In this chapter I give my expectations on where tomography research will head in the future and my summary of research done so far.

6.1 Fourth Generation Light Sources

Third-generation facilities are in their prime, however, intense effort develop fourth generation light sources has already begun. [1] These next generation sources will offer vastly enhanced performance, delivering ultra-bright X-rays whose properties are similar to lasers. Laser-like properties would include fully coherent beams, and the X-rays could be pulsed on time scales of 10-100 fs with some techniques further reducing pulse duration.[2] New phenomena in condensed matter physics involving atoms and molecules will be able to be explored with this new source.[3]

The fourth-generation source will have hard X-rays (wavelengths of $\leq$ 1 nm) generated from devices such as free-electron laser (FEL) systems. FEL systems are based on the use of very long undulators in synchrotron particle accelerator rings.[4] The basic concept of undulators used in third generation sources was covered in Section 2.4.3 and Figure 2.23a shows the basic design of an undulator. FEL tabletop instruments are already available today[5] and are actively used but by combining their technology with synchrotron storage rings the fourth generation sources will result in greatly improved radiation source for materials science research. Fourth generation light sources are already coming online.[6] This is next logical progression of light source technology and fourth generation light sources will generate exciting new research in the following years.

6.2 Tomography

Currently tomography projects in polymers, materials science, and environmental science, biological science, civil and environmental engineering, geology, and medical physics are all being done at LSU. From the type of applications reported in the literature, our universities own research in the field and the rate of publication, one can conclude that tomography imaging is only going to grow with time.

6.2.1 Nanoscale Tomography

Three-dimensional tomographic imaging of materials on the micron scale has been proven as a excellent technique. However, micrometer measurements are longer considered special or unique. Nanotechnology is the new buzz word and for good reason it's still an evolving field and the advances it has already made are impressive. Well what about tomography at the nanoscale, there should be a strong desire to get real 3D spatial information out of nanoscale materials.[7] Tomography has gown in the past few decades, no longer does it encompasses a few experimental procedure with minor variations. Traditional computed X-ray tomography systems are now commercially available as standalone instruments. Lambda tomography [8, 9], thermoacoustic tomography[10], and neutron tomography[11] are all new exciting tomography experiments that are tackling specific problems in a variety of fields. Exciting news that atom probe[12], electron[13] and focused ion beam tomography (FIB)[14, 15] have all already breached the nanometer resolution is great news for tomographers and the scientific community that uses these methods.

6.2.2 Atom Probe Tomography

Atom probe tomography (APT) was first demonstrated in 1986 when the first prototype was built by Michael Miller who was working off of a patent by J.A. Panitz.[16]

By combining time-of-flight mass spectroscopy and field ion microscopy, APT can generate 1D compositional streams of the atoms in the sample being analyzed and 3-D reconstructions are easily generated from this data. The instrument can detect over 1×10^8 atoms for a typical sample.[17] The APT experiment starts with sample fabrication into a sharp tip, then placed in ultra high vacuum at low temperature (normally $\leq$100 K). The atoms at the apex of the tip are ionized, either by a positive pulsed voltage or a laser. These ions are repelled from the tip and detected with a position sensitive detector. By measuring the time between the ionization and the

impact on the detector a high resolution chemical profile can be calculated via the mass-to-charge ratio.[18] Figure 6.1 shows a typical APT result in condensed form; the thousands of plot points are representative of $\approx 10^8$ atom measurements.

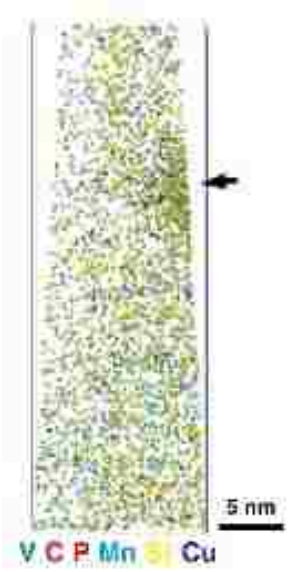


Figure 6.1: Three-dimensional atom probe tomography chemical map of a sample containing vanadium, carbon, phosphorus, magnesium, silicone, and cooper. [19])

6.2.3 Electron Tomography

Electron tomography is an electron microscopy experiment modified to produce a 3D image. Electron tomography can be used to study just about any sample but has a major limitation in that the sample must be a thin rod. The technique is destructive and is highly susceptible to noise degradation. However, resolutions of ≈5-20 nm can be obtained if proper conditions are meet.[20, 21]

Data collection for electron tomography collects data on thin rods by tilting the around one axis in ranges of $\pm 70^o$. Complete data collection required by the transform would require tilting ranges of $\pm 90^o$ but due to mechanical sample holder limitations the higher angle are not accessible. Electron tomography has seen extensive use in characterizing multi-protein primary structures in the fields of molecular biology[22]; Figure 6.2 shows the application of ET to virus structure.

6.2.4 Focused Ion Beam Tomography

Like atom probe tomography, focused ion beam tomography is slightly different to most other modern techniques that have tomography in their titles. Most other tomography experi-

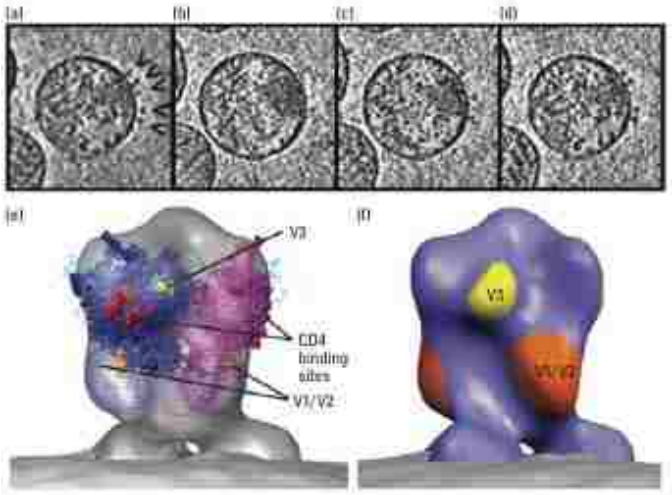
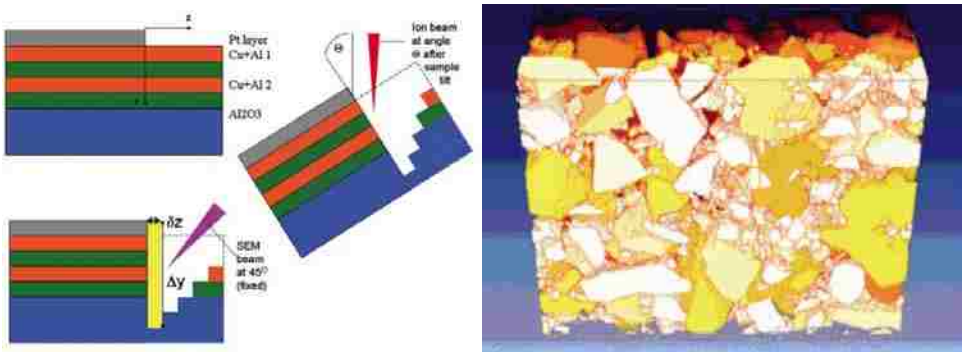


Figure 6.2: Electron tomography of virus a-d) Tomographic data slices of hydrated-virus. Some proteins have been labeled with arrowheads in image (a). e) Surface of averaged densities calculated from the volume. Certain biologically active sites are labeled and a superimposed X-ray crystallography structure is also shown. f) Surface colored regions of image (e). [23])

ments yield 3D information about a sample from projection data generated from a source interacting with a sample, while the FIB tomography experiment actually cuts the sample in manner that resembles the act of physically slicing the sample with a knife. Obviously, this is a nanoscale knife that cuts in multiple orientations and is destructive to the sample. Figure 6.3a displays the principles of the experiment.

Focused ion beam tomography has its roots in metallography.[23] Metallography is a technique in which one prepares a metal surface for analysis by optical microscopy by grinding and polishing. Over the years, and with the advent of electron microscopy, this technique has improved to reach nanoscale resolution. The biological and materials science communities modified the technique to produce 3D images via multiple cutting of the materials. Today, researchers using FIB as the source can reach resolutions of ≤ 50 nm.[24] FIB tomography has been widely used in the semiconductor industry [25] to look for subsurface deformation. Ceramics research has also used the technique to image grain shapes in materials.[26] Figure 6.3a shows a cement sample imaged with FIB tomography to study grain shapes.

The three fields listed above: atom probe tomography, electron tomography, and focused ion beam tomography all represent excellent examples of current state of the art tomography techniques that reach nanoscale resolution. However, none of the techniques can be used to study the polymer blend systems investigated in this dissertation. Atom probe tomography requires metallic samples to work[17], electron tomography has the strict requirement of thin samples that



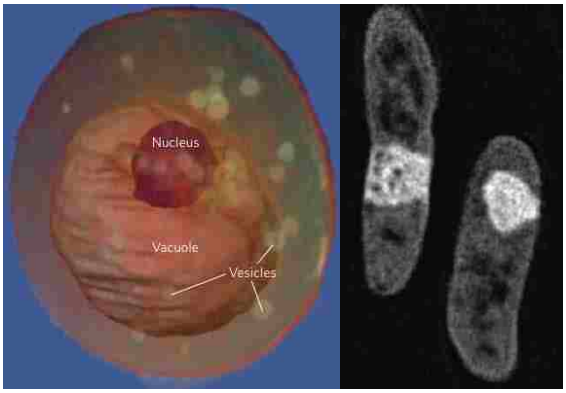
(a) (b)

Figure 6.3: Principles of a FIB tomography experiment. For this figure a Cu-Al metallic mulit-layer is used for the sample. a) Sample preparation with the platinum protection layer. b) Destructive milling of sample to generate 3D volume data. This is the setup used with a double beam FIB experiment c) Titling over a range of angles is required for single beam FIB experiments. b) Visualization of a system of cement particles/grains imaged with FIB tomography. [23])

are optimally $\leq$500 μm thick[21], and focused ion beam tomography is destructive[24] and that obviously hinders dynamic measurements of samples. Finally, all three of these techniques require ultrahigh vacuums to operate, which in and of itself is not deal breaker for dynamic tomography measurements , but does add another layer of complexity to an already complicated experiment.

The experimental X-ray tomography experiments presented in this dissertation offer the best method for dynamically studying polymer blends. With that in mind, can we reach nanoscale resolutions that would enable investigation of dynamic chemical concepts such as diffusion and nucleation? Recent advances in X-ray microscopy instrumentation[27] has lead to tomographic resolutions of $\approx$ 60 nm. [28] This research leads to whole yeast [29] and bacterium [30] cells to be imaged at this resolution as seen in Figure 6.4.

With the success of these cell experiments, both the Advanced Light Source and Advanced Photon Source, at Lawrence Berkeley National Laboratory and Argonne National Laboratory respectively, have begun the construction/upgrade of tomography beamlines that will have resolutions on the nanometer scale. [28] Already, workshops on the applications of nanotomography have been held. While most of the attention is focused on the advances nanotomography will bring cell and molecular biology, there are no obstacles to using the technique to investigate materials and dynamic processes.



(a) (b)

Figure 6.4: a) Whole, hydrated yeast cell (diameter ≈ 5 μm). Notable cellular structures such as the nucleus, vacuole, and vesicles are labeled. b) Tomographic slice from reconstructed volume data for two $Escherichiacoli$ bacteria, whose width is ≈ 0.5 μm. Both experiments acquired data at 517 eV with 45 projected images covering angles up to 180^o in 4^o intervals with exposure times of 1-3 seconds. Resolution of both experiments was ≈ 60 nm [28])

6.2.5 High Throughput X-Ray Tomography

High-throughput, optimum equipment usage presents a queuing problem akin to super-computer queues, but with a more complex parameter space. Advanced imaging resources are scarce, exceedingly expensive, and in high demand. To maximize experiment throughput, beamlines are mostly automated and enhanced with sample changers. Still, for maximum performance, experienced users and staff scientist will manually adjust optics, slits, monochromator, interferometers, scintillators, and detectors. For example, one sample may need imaging with exactly 1 micron voxels, while another sample may need imaging over a range of interferometer settings. For a given sample tray that may have 96-samples ready for imaging, how does the automation software merge the beamline manual settings with the requirements of each sample? If there is a ideal solution that would automate a 96 sample sample tray with multiple resolution requirements, how much would it cost and would it be justified?

With commercial tools such as National Instruments LabView and Matlab's Simulink and Data Acquisition Toolboxes available to us, creating a robust automated experimental system is certainly feasible. The ultimate question that arises is that does the full automation of a beamline provide a significant boost to efficiency and productivity? The more dead time a beamline has, the less useful a fully automated beamline becomes. As previously stated the field of tomography is only going to grow, so the likelihood of beamlines being less used than they currently are is a remote one. Significant effort into building beamlines fully automated from the start seems logical

and prudent.

6.2.6 Neutron Tomography

Neutron tomography is just beginning to receive attention from the scientific community.[31] Applications for neutron imaging experiments have so far mostly been introductory studies of neutron radiography with little real world application.[32, 33] Already, researchers are using neutron imaging methods to explore several different types of problems[31, 34]:

- Water distribution in sedimentary formations
- Oil flow in operating internal combustion engines
- Water plugs in hydrogen-oxygen fuel cells
- Swords and helmets in archeological studies
- The structure of explosive devices

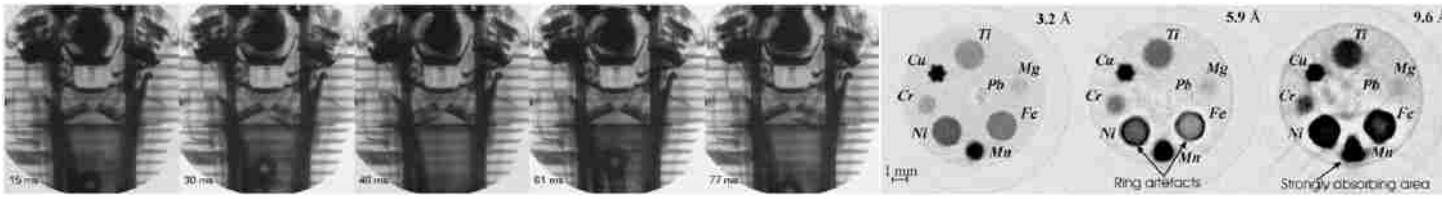
The reported applications typically are examples of preliminary evaluations of neutron radiography or tomography to a new area of science or engineering. As the field has progresses several interesting imaging experiments can be envisioned that can investigate dynamic processes, perform chemical analysis via energy selectivity of materials being imaged with neutrons, and phase contrast imaging are all possible.

Engineering problems such as the operation of internal combustion engines can be studied with dynamic high-speed neutron radiography.[35] A major strength X-ray tomography is its ability to selectively image certain elements across their K-edges.[36, 37] The idea of using variable wavelengths of neutrons to extract chemical information from the experiment has been suggested[38] but little to no work has been done on it. Phase contrast enhancement in neutron tomography has been demonstrated successfully [39, 40]. Figure 6.5 shows three results of current neutron imaging experiments. A variety of samples have already been investigated[39, 40, 41, 38, 42], some of the samples include:

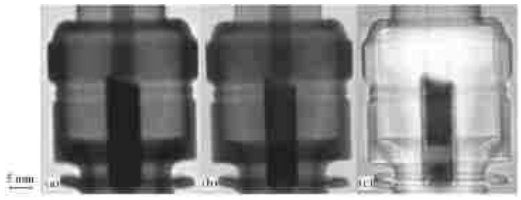
- Insects
- Silicon test objects
- Syringe needles

- Aluminum foams and castings
- Wall material used in nuclear reactors

Ultimately, I see neutron tomography and imaging as a complementary method to X-ray tomography. Both techniques are able to provide chemical spatial information. The evolution from brominated flame retardants to phosphorus-based systems, systems difficult to study with X-ray, thus neutron tomography will be the preferred technique to do analysis with this next generation of flame retardants. Both techniques have a bright future both in terms of independent and complementary use as a tool to do materials analysis, and polymers blend research will certainly benefit from both.



(a) (b)



(c)

Figure 6.5: a) Series of selected time frames from a dynamic imaging experiment of piston positions. b) A test sample used to illustrate the wavelength selectivity in neutron imaging. c) Phase contrast experiment of a spark plug at two different neutron wavelengths with the third image indicating the success of the phase contrast technique to show unseen new features in the sample.

6.3 New Mathematics and Computational Tools

Future tomography research is going to rely heavily on new mathematics, computational methods and advanced computer hardware. In the following sections I cover the areas of math and technology that will be crucial for future topographers to perform their research.

6.3.1 New Mathematics and Algorithms for Reconstruction and 3D Image Analysis

New mathematics and algorithms are needed for 3-D image acquisition and analysis. 3D data comes from many disciplines and the mathematics that govern tomography and 3D image analysis needs to be advanced. The complexities of the mathematics argue for developing collaborations between mathematicians and scientists that will produce new math and algorithms. In order to address this issue workshops have been held. [43] Topics covered were wide ranging and approximately 120 participants were at the conference. Issues that were discussed relating to the research presented in this dissertation included:

- The combination of multiple volume into a single volume.
- Automatic 3D image analysis of simple geometric shapes.

Some issues did not get discussed but are still an issue. Raw data rates at synchrotron X-ray tomography beamlines can exceed 10 MB/s, and this data must be processed rapidly into normalized projection data, ordered into sinograms, and then converted into image data where the typical back projection reconstruction algorithm is an O(N3) problem. With projections approaching 2k$\times$2k, small computer clusters are needed to generated reconstructed slice images at rates comparable to the data acquisitions. The desire for a much faster reconstruction algorithm that does not sacrifice much in terms of accuracy and resolution would be highly desired. 3D image processing algorithms are still a relatively new discipline so scientists, not all of whom are parallel computing experts, must run high level software such as cluster versions of Mathematica (GridMathematica) or Matlab rather than MPI-optimized C/Fortran code. New 3D image processing algorithms that are more efficient or simply faster would be of great use to everyone. Post reconstruction image processing also adds a large amount of data overhead that must be organized and tracked in an efficient manner what is best way to keep track of this data in a way that will foster open data sharing.

The overall conference was considered a success but a stronger presence of experimentalists was desired. In the future more conferences like the previously mentioned is necessary to help foster connections between mathematicians and scientists so that new mathematics and algorithms can be developed at a more rapid rate.

Table 6.1: Visualization Software

Name	Commercial?	Comments
Amira	yes	expensive, algorithms are blackbox
VisIt	open source	excellent support for cluster use, interface not as refined as others
ITK	open source	no interface just a library, popular among core comp sci researchers
Osirix	open source	mac only, primarily for radiologists
Maya	yes	mainly used for animation, not much scientific support

6.3.2 High Level Hierarchal Data Formats

As previously stated there is a lot of data associated with the typical tomography experiment. In a effort to reduce the amount of data many experimentalist throw away absorption images and sinograms once the reconstruction has completed. Even with this trimming the reconstructed volume and any post processing data generated can still be quite large. Tomography beamlines and laboratory systems employ a wide range of data formats in order to manage all this data. Formats range from the simple block-of bytes to more complex hierarchal data types. We currently prefer HDF5, however, it is still problematic and is not well supported by a number of 3D visualization tools. The major issue with picking a data format is that there is no standard that researchers can all agree on. Currently many groups take one of the formats listed above and modify it to suit their particular needs. This modification while useful to the original group makes it very hard to share their data collaboratively. A standard format or set of guidelines could be adopted by tomography research community but no attempts to do this have been attempted yet. Alternatively, GUI-based data conversion software for rapid translation of various formats to the widely used formats such as block-of-bytes, HDF5, and netCDF would find much success.

6.3.3 Visulization Tools

Scientific visualization of data has traditionally be done through a multitude of graphs and plots. These graphs produced by the scientific community are useful to the community itself but to anyone outside the field the graphs would rapidly lose meaning. With the advent of 3D graphics on personal computers it has become possible to render a wide variety of scientific data in manner that is accessible to experts and non-experts alike. A variety of 3D visualization tools such as Amira, OsiriX, VisIt, VTK, Maya and many others. Some are commercial and others open source all have advantages and disadvantages, Table 6.1 lists some of the common viz tools compares them.

Until a vastly superior product comes out most people will find themselves using one or two visualization software packages.

6.3.4 Data Storage

Long term storage of nearly raw tomography data is often done with the normalized projection data, which is a series of 2D images, often 500 to 2,000 images with each image on the order of 0.5 to 10 MB. As multiple tomography experiments can be done every day, and instruments often run 24/7, long-term storage requirements on the order of 10 - 100 TB are needed. LSU has recently acquired funding for Petashare[44] project which aims to provide a innovative distributed data archival, analysis and visualization cyberinfrastructure for data intensive collaborative research. PetaShare will enable transparent handling of underlying data sharing, archival and retrieval mechanisms, and will make data available to scientists for analysis and visualization on demand. Petashare is an excellent step towards the data storage demands that tomography experimentalists will require.

6.4 Remarks

Various X-ray tomography experiments have been performed in this research to study the blending properties of polymer blends and the diffusion properties. Experiments were done at the Center for Advanced Microstructures and Devices tomography beamline and the Advanced Photon Source (APS), Argonne National Laboratory, at bending magnet 13-BM-D in the GeoSoilEnviroCARS (GSECARS) group. This research has concentrated on developing new algorithms for complex data analysis in the three dimensions.

A fiberglass-reinforced polymer blend with a new-generation flame retardant was studied with multi-energy synchrotron X-ray tomography to assess blend homogeneity. The sample was difficult to image due to low x-ray contrast between the fiberglass reinforcement and the polymer blend. New procedures were developed to find and mark the fiberglass, then assess the flame retardant distribution near the fiber.

By using three-dimensional chemical analysis techniques another polymer blending problem called blooming was investigated. To investigate the chemical process of blooming, new procedures are developed to assess the flame retardant distribution as a function of annealing time in the sample. The chemical distribution was measured and the data was ft to a generalized diffu-

sion equation. To calculate diffusion measurements around the pockets of high concentration in the sample, new methods had to be developed.

The diffusion properties of hexabromobenzene and o-terphenyl was studied using X-ray tomography. The diffusion properties were compared with molecular dynamics computer simulations. The computer simulations helped us interpret the images and guide the improvement of techniques for image analysis of diffusion processes.

Multi-spectral X-ray tomography is shown as an excellent technique to study polymer blends. Near micrometer resolution tomography along with spatial concentration distribution analysis can give new insights into the science of polymer blends. The procedures under development here can be applied to a wide variety of materials science problems, ranging from studies of cornified tissues in biological samples to pollutant flow in environmental samples.

6.5 Bibliography

[1] A. C. Thompson, David T. Attwood, Eric M. Gullikson, Malcolm R. Howells, Jeffrey B. Kortright, Arthur L. Robinson, and James H. Underwood. *X-Ray Data Booklet*. Lawerence Berkely National Laboratory, 2001.

[2] W.A. Barletta and H Winick. Introduction To Special Section On Future Light Sources. *nuclear Instruments & Methods in Physics Research Section a-Accelerators Spectrometers Detectors and Associated Equipment*, 500(1-3):1–10, MAR 11 2003.

[3] Richard B. Fenner, Rodney E. Gerig, J.Murray Gibson, Efim Gluskin, Gabrielle G. Long, Dennis M. Mills, and WilliamG. Ruzicka. The Advanced Photon Source Looks To The Future. *nuclear Instruments & Methods in Physics Research Section a-Accelerators Spectrometers Detectors and Associated Equipment*, 582(1):5–10, NOV 11 2007.

[4] H Wiedemann. Storage Ring Design Optimization For Fel Operation. *journal De Physique*, 44(NC-1):201–209, 1983.

[5] J.P. Blewett, L. Blumberg, A.J. Campillo, R.P. Dinardo, H.C. Hsieh, S. Krinsky, A. Luccio, C. Pellegrini, J. Schuchman, P.Z. Takacs, and A Vansteenbergen. Free-electron Laser Experiment At The NSLS 700 Mev Electron Storage Ring. *IEEE Transactions on Nuclear Science*, 28(3):3166–3168, 1981.

[6] S. Benson, G. Biallas, J. Boyce, D. Bullard, J. Coleman, D. Douglas, F. Dylla, R. Evans, P. Evtushenko, A. Grippo, C. Gould, J. Gubeli, D. Hardy, C. Hernandez-garcia, K. Jordan, J. M. Klopf, W. Moore, G. Neil, T. Powers, J. Preble, D. Sexton, M. Shinn, C. Tennant, R. Walker, S. Zhang, and G.P. Williams. The 4th Generation Light Source At Jefferson Lab. *nuclear Instruments & Methods in Physics Research Section a-Accelerators Spectrometers Detectors and Associated Equipment*, 582(1):14–17, NOV 11 2007.

[7] M. Stampanoni, G. Borchert, and R Abela. Towards Nanotomography With Asymmetrically Cut Crystals. *nuclear Instruments & Methods in Physics Research Section a-Accelerators Spectrometers Detectors and Associated Equipment*, 551(1):119–124, OCT 1 2005.

[8] Eric Todd Quinto. Tomographic Reconstructions From Incomplete Data - Numerical Inversion Of The Exterior Radon-transform. *inverse Problems*, 4(3):867–876, AUG 1988.

[9] Eric Todd Quinto and Ozan Oktem. Local Tomography In Electron Microscopy. *SIAM Journal on Applied Mathematics*, 68(5):1282–1303, 2007.

[10] Y.V. Gulyaev, K.M. Bograchev, I.P. Borovikov, Y.V. Obukhov, and Vi Pasechnik. Passive Thermoacoustic Tomography: Methods and Approaches. *Journal of Communications Technology and Electronics*, 43:1061–1067, 1998.

[11] W. Treimer, U. Feye-treimer, and C Herzig. On Neutron Tomography. *Physica B*, 241:1197–1203, 1997.

[12] Mk Miller. Interface Analysis With The Three-dimensional Atom Probe. *surface and Interface Analysis*, 31(7):593–598, JUL 2001.

[13] G. Mobus and Bj Inkson. Element Specific Nanoscale Electron Tomography. *Electron Microscopy and Analysis 2001*, pages 267–270, 2001.

[14] J. R. Jinschek, K. J. Batenburg, H. A. Calderon, R. Kilaas, V. Radmilovic, and C. Kisielowski. 3-D Reconstruction Of The Atomic Positions In A Simulated Gold Nanocrystal Based On Discrete Tomography: Prospects Of Atomic Resolution Electron Tomography. *Ultramicroscopy*, 108(6):589–604, MAY 2008.

[15] S. P. Ahrenkiel, P. R. Yu, J. E. Murphy, J. M. Nedeljkovic, and B.S. Donohoe. Nanoparticle Shape and Configuration Analysis By Transmission Electron Tomography. *journal of Microscopy-Oxford*, 230(3):382–387, JUN 2008.

[16] Mk Miller. The Development Of Atom Probe Field-ion Microscopy. *materials Characterization*, 44(1-2):11–27, JAN-FEB 2000.

[17] Simon P. Ringer, David J. Larson, Michael P. Moody, Michael K. Miller, and ThomasF. Kelly. Introduction: Special Issue On Atom Probe Tomography. *microscopy and Microanalysis*, 13(6):407, DEC 2007.

[18] M.K. Miller and Ea Kenik. Atom Probe Tomography: A Technique For Nanoscale Characterization. *microscopy and Microanalysis*, 10(3):336–341, JUN 2004.

[19] R.C. Thomson and Mk Miller. Atom Probe Characterisation Of High Temperature Materials. *materials Science and Technology*, 16(10):1199–1206, OCT 2000.

[20] W Baumeister. Electron Tomography: Towards Visualizing The Molecular Organization Of The Cytoplasm. *current Opinion in Structural Biology*, 12(5):679–684, OCT 2002.

[21] W. Baumeister, R. Grimm, and J Walz. Electron Tomography Of Molecules and Cells. *trends in Cell Biology*, 9(2):81–85, FEB 1999.

[22] W. Baumeister and Ac Steven. Macromolecular Electron Microscopy In The Era Of Structural Genomics. *trends in Biochemical Sciences*, 25(12):624–631, DEC 2000.

[23] Gunter Mobus and BeverleyJ. Inkson. Nanoscale Tomography in Materials Science. *Materials Today*, 10:18–25, 2007.

[24] A.J. Kubis, G.J. Shiflet, D.N. Dunn, and R Hull. Focused Ion-Beam Tomography. *Metallurgical and Materials Transactions a-Physical Metallurgy and Materials Science*, 35A:1935–1943, 2004.

[25] T. S. Yeoh, N. A. Ives, N. Presser, G. W. Stupian, M. S. Leung, J. L. Mccollum, and F.W. Hawley. Focused Ion Beam Tomography of a Microelectronic Device with Sub-2-Nm Resolution. *Journal of Vacuum Science & Technology B*, 25:922–925, 2007.

[26] C. Holzapfel, W. Schaef, M. Marx, H. Vehoff, and F. Muecklich. Interaction of Cracks with Precipitates and Grain Boundaries: Understanding Crack Growth Mechanisms through Focused Ion Beam Tomography. *Scripta Materialia*, 56:697–700, 2007.

[27] G.C. Yin, M.T. Tang, Y.F. Song, F.R. Chen, K.S. Liang, F.W. Duewer, W.B. Yun, C.H. Ko, and Hpd Shieh. Energy-Tunable Transmission X-Ray Microscope for Differential Contrast Imaging with Near 60 Nm Resolution Tomography. *Applied Physics Letters*, 88, 2006.

[28] David Attwood. Microscopy - Nanotomography Comes of Age, 2006. Nature.

[29] C.A. Larabell and Ma L.E.Gros. X-Ray Tomography Generates 3-D Reconstructions of the Yeast, Saccharomyces Cerevisiae, at 60-Nm Resolution. *Molecular Biology of the Cell*, 15:957–962, 2004.

[30] M.A. LeGros, G. Mcdermott, and Ca Larabell. X-Ray Tomography of Whole Cells, 2005. Current Opinion In Structural Biology.

[31] Peter Vontobel, Eberhard H. Lehmann, Rene Hassanein, and Gabriel Frei. Neutron Tomography: Method and Applications. *Physica B-Condensed Matter*, 385:475–480, 2006.

[32] S. Baechler, N. Kardjilov, M. Dierick, J. Jolie, G. Kuhne, E. Lehmann, and T Materna. New Features in Cold Neutron Radiography and Tomography - Part 1: Thinner Scintillators and a Neutron Velocity Selector to Improve the Spatial Resolution. *Nuclear Instruments & Methods in Physics Research Section a-Accelerators Spectrometers Detectors and Associated Equipment*, 491:481–491, 2002.

[33] N. Kardjilov, S. Baechler, M. Basturk, M. Dierick, J. Jolie, E. Lehmann, T. Materna, B. Schillinger, and P Vontobel. New Features in Cold Neutron Radiography and Tomography - Part II: Applied Energy-Selective Neutron Radiography and Tomography. *Nuclear Instruments & Methods in Physics Research Section a-Accelerators Spectrometers Detectors and Associated Equipment*, 501:536–546, 2003.

[34] M.I. Silvani, R.T. Lopes, Efo D.E.Jesus, G.L. D.E.Almeida, and Af Barbosa. Evaluation of a Computer Aided Neutron Tomographic System Incorporating a Gaseous Position Sensitive Detector. *Nuclear Instruments & Methods in Physics Research Section a-Accelerators Spectrometers Detectors and Associated Equipment*, 505:568–572, 2003.

[35] P. Vontobel, G. Frei, J. Brunner, A.E. Gildemeister, and M Engelhardt. Dynamic Imaging with a Triggered and Intensified Ccd Camera System in a High-Intensity Neutron Beam. *Nuclear Instruments & Methods in Physics Research Section a-Accelerators Spectrometers Detectors and Associated Equipment*, 542:148–153, 2005.

[36] K. Ham and L. G. Butler. Algorithms for Three-Dimensional Chemical Analysis Via Multi-Energy Synchrotron X-Ray Tomography. *Nuclear Instruments & Methods in Physics Research Section B-Beam Interactions with Materials and Atoms*, 262(1), 2007.

[37] H. A. Barnett, K. Ham, and L.G. Butler. Synchrotron X-Ray Tomography for 3d Chemical Diffusion Measurement of a Flame Retardant in Polystyrene. *Nuclear Instruments & Methods in Physics Research Section a-Accelerators Spectrometers Detectors and Associated Equipment*, 582:202–204, 2007.

[38] E. Lehmann, K. Lorenz, E. Steichele, and P Vontobel. Non-Destructive Testing with Neutron Phase Contrast Imaging. *Nuclear Instruments & Methods in Physics Research Section a-Accelerators Spectrometers Detectors and Associated Equipment*, 542:95–99, 2005.

[39] P.J. Mcmahon, B.E. Allman, D.L. Jacobson, M. Arif, S.A. Werner, and Ka Nugent. Quantitative Phase Radiography with Polychromatic Neutrons. *Physical Review Letters*, 91, 2003.

[40] B.E. Allman, P.J. Mcmahon, K.A. Nugent, D. Paganin, D.L. Jacobson, M. Arif, and Sa Werner. Imaging - Phase Radiography with Neutrons. *Nature*, 408:158–159, 2000.

[41] N. Kardjilov, S.W. Lee, E. Lehmann, I.C. Lim, C.M. Sim, and P Vontobel. Improving the Image Contrast and Resolution in the Phase-Contrast Neutron Radiography. *Nuclear Instruments & Methods in Physics Research Section a-Accelerators Spectrometers Detectors and Associated Equipment*, 542:100–105, 2005.

[42] M. Basturk, N. Kardjilov, H. Rauch, and P Vontobel. Neutron Imaging of Fiber-Reinforced Materials. *Nuclear Instruments & Methods in Physics Research Section a-Accelerators Spectrometers Detectors and Associated Equipment*, 542:106–115, 2005.

[43] L. G. Butler, EricTodd Quinto, and Gestur Olafsson. New Mathematics and Algorithms for 3-D Image Analysis. 2006.

[44] MehmetBalman, IbrahimSuslu, and TevfikKosar. Distributed Data Management with Petashare. In *MG '08: Proceedings of the 15th ACM Mardi Gras conference*, pages 1–1, New York, NY, USA, 2008. ACM.

Appendix A Permission Form

This is a License Agreement between Heath A Barnett (”You”) and Elsevier (”Elsevier”). The license consists of your order details, the terms and conditions provided by Elsevier, and the payment terms and conditions.

This agreement is for licensed content publication of Nuclear Instruments and Methods in Physics Research Section A and the licensed content title *Synchrotron X-ray tomography for 3D chemical diffusion measurement of a flame retardant in polystyrene*.

Customer name	Heath A Barnett
License Number	1994870771096
License date	Jul 23, 2008
Licensed content publisher	Elsevier Limited
Licensed content author	H.A. Barnett, K. Ham and L.G. Butler
Licensed content date	11 November 2007
Volume number	582
Issue number	1
Pages	3
Type of Use	Thesis / Dissertation
Portion	Full article
Expected publication date	Dec 2008
Elsevier VAT number	GB 494 6272 12

Appendix B Algorithms

Reconstruction Algorithm

This is a matlab program for taking raw data from a tomography experiment and reconstructing a volume dataset. Much of this code is automatic corrections such centering, white/dark field issues, cropping, and data logging. Anything done to the data is recorded and stored in a structured variable.

```
% FirstReconstruction_v31.m
% June 27, 2007
% This program is modified from FirstReconstruction_v24_kh_matlab74.m for use of the
% data set with larger field of view than the acquision column size
% This version can reconstructed tomography data sets that have been
% interrupted with beam dumps.
% Important data is read from the footer with a revised ReadFITS.m file
%
% New: Updates to PROC.angle_to_file_seq   and  PROC.angle
%      These update recognize that some raw data cannot be used due to a
%      lack of appropriate white fields (see the "otherwise" option in
%      PROC.raw_with_white).
%Also updated with streak removal
%line 86 is for AirRoundObject value

clear; clc;
% Locate the date files: *.fits files of white, dark, and raw images
[FILE.filename, FILE.filepath] = uigetfile('*.fits', 'Click on any *.fits data file');
% FILE.filename = 'shell25.dark.00001.000.000.fits';
% FILE.filepath = '/Users/tomog9/Desktop/Cat_Satuday/';

%how many whites taken at a given angle
PROC.white_average = 3;
% how often whites taken, after every 30 images
PROC.white_frequency = 30;


% Within that directory, find all the white, dark, and raw *.fits files
original_DataPath = pwd;        cd(FILE.filepath);        FILE.filenames = dir;        cd(original_DataPath);
[numberOfFiles, lengthOfFilename] = size(FILE.filenames);
% Find the mean dark count in a dark field image.  Use this mean count Find
% any dark file.  Calculate the mean counts in the dark field.
% Set the count_threshold value at 1.2 * dark field count.   Any raw or
% white field less than threshold is bad data and ignored.
for k = 1:numberOfFiles
    temp = FILE.filenames(k).name;
        if findstr(temp, '.fits') & findstr(temp, 'fits'),
                if findstr(temp, '.dark')
```

```
            [data, PARAMETERS] = ReadFITS(FILE.filepath, temp);
            FILE.count_threshold = 1.2*PARAMETERS.mean_count;
            break,
        end;
    end;
end;
% Get file names and mean_counts for dark, white, and raw files.
FILE.white = {}; FILE.dark = {}; FILE.raw = {};
for k = 1:numberOfFiles
        temp = FILE.filenames(k).name;
        if findstr(temp, '.fits')& findstr(temp, 'fits'),
                if findstr(temp, '.dark'),
            index = length(FILE.dark) + 1;
                        FILE.dark(index) = {temp};
                elseif findstr(temp, '.white'),
            [data, PARAMETERS] = ReadFITS(FILE.filepath, temp);
            if PARAMETERS.mean_count >= FILE.count_threshold,
                index = length(FILE.white) + 1;
                FILE.white(index) = {temp};
                FILE.white_counts(index) = PARAMETERS.mean_count;
                ACQ.white_file_number(index) = index;
                ACQ.white_sequence_number(index) = PARAMETERS.sequence_number;
            end;
                elseif findstr(temp, '.raw'),
            [data, PARAMETERS] = ReadFITS(FILE.filepath, temp);
            if PARAMETERS.mean_count >= FILE.count_threshold,
                index = length(FILE.raw) + 1;
                FILE.raw(index) = {temp};
                FILE.raw_counts(index) = PARAMETERS.mean_count;
                ACQ.raw_file_number(index) = index;
                ACQ.angle(index) = PARAMETERS.angle;
                ACQ.raw_sequence_number(index) = PARAMETERS.sequence_number;
            end;
                 end;
        end;
end;
FILE.dark(1) = [];
[x, FILE.numberWhiteFiles] = size(FILE.white);
[x, FILE.numberDarkFiles] = size(FILE.dark);
[x, FILE.numberRawFiles] = size(FILE.raw);
ACQ.NumberOfProjections = length(FILE.raw) - 1;
ACQ.naxis1 = PARAMETERS.NAXIS1;     ACQ.columns = PARAMETERS.columns;
ACQ.naxis2 = PARAMETERS.NAXIS2;     ACQ.rows = PARAMETERS.rows;
ACQ.exptime = PARAMETERS.exptime;
ACQ.ccd_xoffset = PARAMETERS.ccd_xoffset;   % fix *.fits header/footer and ReadFITS.m
ACQ.ccd_yoffset = PARAMETERS.ccd_yoffset;   % fix *.fits header/footer and ReadFITS.m
ACQ.PixelToMicron = (4.5);                % fix *.fits header/footer and ReadFITS.m
PROC.AirAroundObject = 15;%5;%1;%15;%0;%15;%5;%15;
ACQ.X_axis = ACQ.ccd_xoffset+(1:ACQ.columns);   ACQ.Y_axis = ACQ.ccd_yoffset+(1:ACQ.rows);

% Display the first white image
disp('*** Display a white image (uncropped).  Find the useful region of illumination.');
frameIndex = 1; h0 = figure(frameIndex); clf;
        set(h0, 'Position',[1,10,560, 420],'Color', [1,1,1], 'Name', 'White Field');
[raw_white, PARAMETERS] = ReadFITS(FILE.filepath, char(FILE.white(1)));
[ACQ.rows, ACQ.columns] = size(raw_white);
First_Crop_Rectangle = [ACQ.ccd_xoffset, ACQ.ccd_yoffset, ACQ.columns, ACQ.rows];
max_counts = max(max(raw_white));        min_counts = min(min(raw_white));
mean_counts = round(mean(reshape(raw_white,ACQ.rows*ACQ.columns,1)));
std_counts = round(std(reshape(raw_white,ACQ.rows*ACQ.columns,1)));
CLIM = mean_counts + 5*[-std_counts, std_counts];
subplot(211); imagesc(ACQ.X_axis,ACQ.Y_axis,raw_white, CLIM); colormap(gray);  axis('ij'); colorbar('horiz');
line1 = [char(FILE.white(1)),'; Rows, Columns = ',mat2str([ACQ.rows, ACQ.columns])];
line2 = ['Counts (min, max) = ',mat2str([min_counts,max_counts]),...
         '; Counts (mean, std) = ',mat2str([mean_counts, std_counts])];
h2 = title( {line1;line2} );
set(h2, 'Interpreter', 'none'); hold on;
[hot_row,hot_column] = find(raw_white> (4*std_counts + mean_counts));
```

```
hot_row = hot_row + ACQ.ccd_yoffset;  hot_column = hot_column + ACQ.ccd_xoffset;
    for k = 1:length(hot_row)
        h5=plot(hot_column(k),hot_row(k),'yo'); set(h5,'MarkerSize',10);
    end;

subplot(223);
vp_left = raw_white(1:ACQ.rows, round(ACQ.columns/4));   % vertical profile at left quartile of white
vp_mid = raw_white(1:ACQ.rows, 2*round(ACQ.columns/4)); % vertical profile at middle of white
vp_right = raw_white(1:ACQ.rows, 3*round(ACQ.columns/4)); % vertical profile at right quartile of white
plot( ACQ.Y_axis, vp_left, 'r-',  ACQ.Y_axis, vp_mid, 'k-',  ACQ.Y_axis, vp_right, 'b-');
xlabel('CCD row');        ylabel('counts');        drawnow;    clear vp_left vp_mid vp_right
title('vertical profiles: left(red), middle(black), right(blue)');

subplot(224);
[n,x] = hist(reshape(raw_white, ACQ.rows*ACQ.columns, 1), 100);
bar(x,log10(n+1));
xlabel('counts');        ylabel(['log_{10}(frequency), bin width = ',num2str(x(2)-x(1))]);        drawnow;
title('histogram of pixel values');        drawnow;
movieData(frameIndex) = getframe(h0);  frameIndex = frameIndex + 1;

%  Show the raw images at 0 and 180 degrees
disp('** Display raw 0deg and raw 180deg images');
h0 = figure(frameIndex); clf;
        set(h0, 'Position',[10,10,560, 420],'Color', [1,1,1], 'Name', 'Raw Images: 0, 180 deg, and last raw');
[raw_0deg, PARAMETERS] = ReadFITS(FILE.filepath, char(FILE.raw(1)) );
subplot(211);        imagesc(ACQ.X_axis,ACQ.Y_axis,raw_0deg, [min(min(raw_0deg)), CLIM(2)]);
    colormap(gray); axis('ij');   % colorbar('horiz');
    h2 = title([char(FILE.raw(1))]);        set(h2, 'Interpreter', 'none');

hp_top    = raw_0deg(  round(ACQ.rows/4), 1:ACQ.columns);   % horizontal profile at top quartile of raw
hp_mid    = raw_0deg(2*round(ACQ.rows/4), 1:ACQ.columns);   % horizontal profile at middle of raw
hp_bottom = raw_0deg(3*round(ACQ.rows/4), 1:ACQ.columns);   % horizontal profile at bottom quartile of raw
subplot(212);  plot( ACQ.X_axis, hp_top, 'r-',  ACQ.X_axis, hp_mid, 'k-',  ACQ.X_axis, hp_bottom, 'b-')
title('top(red), middle(black), bottom(blue)');        ylabel('counts');
        movieData(frameIndex) = getframe(h0);  frameIndex = frameIndex + 1;

h0 = figure(frameIndex); clf;
        set(h0, 'Position',[10,10,560, 420],'Color', [1,1,1], 'Name', 'Raw Images: 0, 180 deg, and last raw');
[raw_180deg, PARAMETERS] = ReadFITS(FILE.filepath, char(FILE.raw(2)) );
subplot(211);        imagesc(ACQ.X_axis,ACQ.Y_axis,raw_180deg, [min(min(raw_0deg)), CLIM(2)]);
    colormap(gray); axis('ij');  % colorbar('horiz');
    h2 = title([char(FILE.raw(2))]);        set(h2, 'Interpreter', 'none');

hp_top    = raw_180deg(  round(ACQ.rows/4), 1:ACQ.columns);   % horizontal profile at top quartile of raw
hp_mid    = raw_180deg(2*round(ACQ.rows/4), 1:ACQ.columns);   % horizontal profile at middle of raw
hp_bottom = raw_180deg(3*round(ACQ.rows/4), 1:ACQ.columns);   % horizontal profile at bottom quartile of raw
subplot(212);        plot( ACQ.X_axis, hp_top, 'r-',  ACQ.X_axis, hp_mid, 'k-',  ACQ.X_axis, hp_bottom, 'b-');
clear hp_top  hp_mid  hp_bottom
title('top(red), middle(black), bottom(blue)');        xlabel('CCD column');        ylabel('counts');
        movieData(frameIndex) = getframe(h0);  frameIndex = frameIndex + 1;

h0 = figure(frameIndex); clf;
        set(h0, 'Position',[10,10,560, 420],'Color', [1,1,1], 'Name', 'Raw Images: 0, 180 deg, and last raw');
[raw_180deg, PARAMETERS] = ReadFITS(FILE.filepath, char(FILE.raw(length(ACQ.raw_sequence_number))) );
subplot(211);        imagesc(ACQ.X_axis,ACQ.Y_axis,raw_180deg, [min(min(raw_0deg)), CLIM(2)]);
    colormap(gray); axis('ij');  % colorbar('horiz');
    h2 = title([char(FILE.raw(length(ACQ.raw_sequence_number)))]);        set(h2, 'Interpreter', 'none');

hp_top    = raw_180deg(  round(ACQ.rows/4), 1:ACQ.columns);   % horizontal profile at top quartile of raw
hp_mid    = raw_180deg(2*round(ACQ.rows/4), 1:ACQ.columns);   % horizontal profile at middle of raw
hp_bottom = raw_180deg(3*round(ACQ.rows/4), 1:ACQ.columns);   % horizontal profile at bottom quartile of raw
subplot(212);        plot( ACQ.X_axis, hp_top, 'r-',  ACQ.X_axis, hp_mid, 'k-',  ACQ.X_axis, hp_bottom, 'b-');
clear hp_top  hp_mid  hp_bottom
title('top(red), middle(black), bottom(blue)');        xlabel('CCD column');        ylabel('counts');
        movieData(frameIndex) = getframe(h0);  frameIndex = frameIndex + 1;

% Calculate dark image
% Use multiple dark files to get an average dark image
```

```
PROC.dark = zeros(ACQ.rows, ACQ.columns);
for k = 1:FILE.numberDarkFiles
    temp = ReadFITS(FILE.filepath, char(FILE.dark(k)) );
    PROC.dark = PROC.dark + temp;
end;
PROC.dark = PROC.dark/FILE.numberDarkFiles;
disp('** Display the averaqe dark image');
h0 = figure(frameIndex); clf;
        set(h0, 'Position',[10,10,560, 420],'Color', [1,1,1], 'Name', 'Average dark image');
subplot(311);
max_counts = max(max(temp));        min_counts = min(min(temp));
mean_counts = round(mean(reshape(temp,ACQ.rows*ACQ.columns,1)));
std_counts = round(std(reshape(temp,ACQ.rows*ACQ.columns,1)));
CLIM = mean_counts + 5*[-std_counts, std_counts];
imagesc(ACQ.X_axis, ACQ.Y_axis,temp, CLIM );   colormap(gray);  axis('ij'); colorbar('horiz');
line1 = [char(FILE.dark(1)),'; Rows, Columns = ',mat2str([ACQ.rows, ACQ.columns])];
h2 = title(line1);  set(h2, 'Interpreter', 'none'); hold on;
max_counts = max(max(PROC.dark));        min_counts = min(min(PROC.dark));
mean_counts = round(mean(reshape(PROC.dark,ACQ.rows*ACQ.columns,1)));
std_counts = round(std(reshape(PROC.dark,ACQ.rows*ACQ.columns,1)));
CLIM = mean_counts + 5*[-std_counts, std_counts];
subplot(312); imagesc(ACQ.X_axis, ACQ.Y_axis,PROC.dark, CLIM); colormap(gray);  axis('ij'); colorbar('horiz');
line2 = ['Average dark:  Counts (min, max) = ',mat2str([min_counts,max_counts]),...
         '; Counts (mean, std) = ',mat2str([mean_counts, std_counts])];
h2 = title( line2 );   set(h2, 'Interpreter', 'none'); hold on;
subplot(313);  [n,x] = hist(reshape(PROC.dark, ACQ.rows*ACQ.columns, 1), 100);
bar(x,log10(n+1));
xlabel('counts');        ylabel(['log_{10}(frequency), bin width = ',num2str(x(2)-x(1))]);        drawnow;
title('histogram of pixel values');        drawnow;
        movieData(frameIndex) = getframe(h0);  frameIndex = frameIndex + 1;

% *************** PROC.raw_with_white *****************
% Build pointer between raw images to white image.   For example, with
% PROC.white_average = 3 and PROC.white_frequency = 30, then the first raw
% image is at sequence number = 4.  Raw #4 will use white #1,2,3,34,35,36.
% Also, it will use 96.67% of #1,2,3 and 3.33% of #34,35,36; this is the
% "ratio" between the "white_front" and "white_back" white fields.
PROC.raw_with_white = zeros(max([max(ACQ.white_sequence_number),max(ACQ.raw_sequence_number)]), (2*PROC.white_ave
for index_raw_seq =  ACQ.raw_sequence_number        % ([1:50,400:426])
    white_front = ACQ.white_sequence_number(find(ACQ.white_sequence_number < index_raw_seq));
        if length(white_front) >= PROC.white_average,
            white_front = white_front( (length(white_front)-PROC.white_average+1):length(white_front) );
        end;
    white_back  = ACQ.white_sequence_number(find(ACQ.white_sequence_number > index_raw_seq));
        if length(white_back) >= PROC.white_average,
            white_back = white_back( 1:PROC.white_average );
        end;
    white_back_sequence_flag = 1;
        if isempty(white_back) == 0 & length(white_back) >= 2,
            for k = 2:length(white_back),
                if white_back(1) + (k-1) == white_back(k),
                    white_back_sequence_flag = white_back_sequence_flag +1;
                end;
            end;
        end;
    test_conditions = mat2str([~isempty(white_back),...
                        white_back_sequence_flag == PROC.white_average,...
                        (min(white_back) - max(white_front)) - 1 == PROC.white_frequency,...
                        (min(white_back) - max(white_front)) < (PROC.white_frequency+1)]);
%     disp(['seq=',num2str(index_raw_seq),'  white_front = ',mat2str(white_front),'  white_back = ',mat2str(white
%     disp(['          test_conditions=', test_conditions]);
    switch test_conditions
        case {'[true true true false]'}
            % Normal white_front, raw, white_back
            %   Use white_front and white_back and compute ratio
%             disp(['          test_conditions=', test_conditions]);
            ratio = 1 - (index_raw_seq - max(white_front))/(PROC.white_frequency+1);
            PROC.raw_with_white(index_raw_seq,:) = [white_front, white_back, ratio];
```

```
        case {'[true true false true]'}
            % The range of raw images does not match PROC.white_frequency
            %   Use white_front and set ratio = 1
%             disp(['          test_conditions=', test_conditions]);
            PROC.raw_with_white(index_raw_seq,:) = [white_front, white_front, 1];
        case {'[true true false false]'}
            % The range of raw images does not match PROC.white_frequency
            %   Use white_back and set ratio = 1
%             disp(['          test_conditions=', test_conditions]);
            if(min(white_back)-index_raw_seq) < PROC.white_frequency
            PROC.raw_with_white(index_raw_seq,:) = [white_back, white_back, 1];
            else PROC.raw_with_white(index_raw_seq,:)=[white_front, white_front,1];
            end
        case {'[false false]'}
            % The white_back is missing or incorrect length
            %   Use white_front and set ratio = 1
%             disp(['          test_conditions=', test_conditions]);
            PROC.raw_with_white(index_raw_seq,:) = [white_front, white_front, 1];
       otherwise
            % skip this data: set white_front = white_back = ratio = 0
            disp(['          PROBLEM: test_conditions=', test_conditions, ' seq=',num2str(index_raw_seq)]);
            PROC.raw_with_white(index_raw_seq,:) = zeros(1,(1+2*PROC.white_average));
   end;
end;
clear test_conditions
%break
% *************** end: PROC.raw_with_white ***************

% *************** PROC.control ***************
% seq, white_file, counts(white), raw_file, counts(raw), angle(raw),...
%           white_front, white_back, ratio
PROC.control = zeros(length(PROC.raw_with_white), (5+size(PROC.raw_with_white,2)));
for index_seq = 1:length(PROC.control),
    PROC.control(index_seq,1) = index_seq;
    index_white = find(ACQ.white_sequence_number == index_seq);
        if isempty(index_white),
            PROC.control(index_seq,2:3)= 0;
        else,
            PROC.control(index_seq,2)= index_white;
            PROC.control(index_seq,3)= FILE.white_counts(index_white);
        end;
    index_raw = find(ACQ.raw_sequence_number == index_seq);
        if isempty(index_raw),
            PROC.control(index_seq,4:6)= 0;
        else,
            PROC.control(index_seq,4)= index_raw(1);
            PROC.control(index_seq,5)= FILE.raw_counts(index_raw(1));
            PROC.control(index_seq,6)= ACQ.angle(index_raw(1));
        end;
    index_white = 7:(6+size(PROC.raw_with_white,2));
    PROC.control(index_seq,index_white) = PROC.raw_with_white(index_seq,:);
    % If "case otherwise" found in generation of PROC.raw_with_white, then
    % eliminate that raw image from PROC.control
    if sum(PROC.raw_with_white(index_seq,:)) == 0, PROC.control(index_seq,4)= 0; end;
end;

% *************** end: PROC.control ***************

% *************** PROC.angle_to_file_seq ***************
% Connect angles to raw file index and sequence numbers of the raw images
% Recall, PROC.control = [seq, white_file, counts(white), raw_file, counts(raw), angle(raw),...
%           white_front, white_back, ratio]
index = find( PROC.control(:,4) > 0);
    temp_sequence = PROC.control(index,1);
    temp_file     = PROC.control(index,4);
    temp_angle    = PROC.control(index,6);
    [temp_angle, index] = sort(temp_angle);
    temp_file = temp_file(index);
```

```
311     temp_sequence = temp_sequence(index);
312
313
314
315 %***********need to find duplicated angles and remove********
316 disp('**** find duplicated angle and remove ***')
317
318         index_skip = [];
319     for i=2:length(temp_angle)
320         if temp_angle(i-1) == temp_angle(i),
321         index_skip = [index_skip, (i-1)];
322         end;
323     end;
324
325    [temp_angle, index] = sort(temp_angle);
326
327
328 %break
329  PROC.angle_to_file_seq=[]
330 ind=find(index_skip)
331
332 if ind ~= 0
333
334 index_keep=[];
335    [temp_angle, index] = sort(temp_angle);
336  k=0;
337       for i= 1:length(index)
338           if i ~= index_skip
339               k=k+1;
340              PROC.angle_to_file_seq(k,:) = [temp_angle(i), temp_file(i), temp_sequence(i)];
341           end;
342
343       end;
344 else
345   for i=1:length(index)
346    PROC.angle_to_file_seq(i,:)=[temp_angle(i), temp_file(i), temp_sequence(i)];
347
348   end;
349 end;
350
351 %PROC.angle_to_file_seq = [temp_angle, temp_file, temp_sequence];
352     clear temp*
353 % *************** end: PROC.angle_to_file_seq ***************
354
355 % *************** Display: PROC.angle_to_file_seq ***************
356 h0 = figure(frameIndex); clf
357 h4 = gca; set(h4,'Visible', 'off')
358     set(h0, 'Position',[50,10,560, 420],'Color', [1,1,1], 'Name', 'PROC.angle_to_file_seq');
359 xtext = -0.1; ytext = 1; yshift = -0.04;
360 h2 = text(xtext, ytext, ['PROC.angle_to_file_seq']);
361     set(h2, 'HorizontalAlignment', 'left','FontName', 'Courier','FontSize',14, 'Interpreter','none');
362 ytext = ytext+yshift;
363 h2 = text(xtext, ytext, ['row   deg        file #     sequence # ']);
364     set(h2, 'HorizontalAlignment', 'left','FontName', 'Courier','FontSize',12);
365 for k=1:25
366     a = num2str(k,'%02i');
367     b = num2str(PROC.angle_to_file_seq(k,1),'%07.3f\t');
368     c = num2str(PROC.angle_to_file_seq(k,2),'%05i\t');
369     d = num2str(PROC.angle_to_file_seq(k,3),'%05i');
370     h3 = text(xtext, ytext+k*yshift, [a,'    ', b,' ',  c,'    ',d]);
371         set(h3,'FontName', 'Courier','FontSize',12);
372 end;
373         movieData(frameIndex) = getframe(h0);  frameIndex = frameIndex + 1;
374 % *************** End Display: PROC.angle_to_file_seq ***************
375 % *************** Display: PROC.raw_with_white ***************
376     h0 = figure(frameIndex); clf
377     set(h0, 'Position',[50,10,560, 420],'Color', [1,1,1], 'Name', 'PROC.raw_with_white');
378     h4 = gca; set(h4,'Visible', 'off')
```

```
    xtext = [-0.1,0.5]; ytext = 1; yshift = -0.04; m = 0;
    h2 = text(xtext(1), ytext, ['PROC.raw_with_white']);
    set(h2, 'HorizontalAlignment', 'left','FontName', 'Courier','FontSize',14, 'Interpreter','none');
    ytext = ytext+yshift;

for j = 1:2,
    h2 = text(xtext(j), ytext, ['row   low        high       ratio ']);
        set(h2, 'HorizontalAlignment', 'left','FontName', 'Courier','FontSize',10);
    for k=1:25,
        m = m+1;
        a = num2str(m,'%02i');                                  index = 1:PROC.white_average;
        b = mat2str(PROC.raw_with_white(m,index));%matlab74,'%05i\t');  index = (PROC.white_average+1):(2*PROC.wh
        c = mat2str(PROC.raw_with_white(m,index));%matlab74,'%05i\t');  index = 2*PROC.white_average + 1;
        d = num2str(PROC.raw_with_white(m,index),'%05.3f');
        h3 = text(xtext(j), ytext+k*yshift, [a,'    ', b,'    ',  c,'    ',d]);
           set(h3,'FontName', 'Courier','FontSize',10);
    end;
end;
clear a b c d
movieData(frameIndex) = getframe(h0);  frameIndex = frameIndex + 1;
% *************** End Display: PROC.raw_with_white ***************
% *************** Display: PROC.control ***************
    h0 = figure(frameIndex); clf
        set(h0, 'Position',[50,10,560, 420],'Color', [1,1,1], 'Name', 'PROC.control');
    h4 = gca; set(h4,'Visible', 'off')
    xtext = -0.1; ytext = 1; yshift = -0.04;
    h2 = text(xtext(1), ytext, ['PROC.control']);
        set(h2, 'HorizontalAlignment', 'left','FontName', 'Courier','FontSize',14, 'Interpreter','none');
    ytext = ytext+yshift;
% Recall, PROC.control = [seq, white_file, counts(white), raw_file, counts(raw), angle(raw),...
%           white_front, white_back, ratio]
    h2 = text(xtext, ytext, ['seq [white]     [raw]                    [front,back]   ratio ']);
        set(h2, 'HorizontalAlignment', 'left','FontName', 'Courier','FontSize',12);
for k=1:25
    a = num2str(PROC.control(k,1),'%02i');
    b = num2str(PROC.control(k,2:3),'%5d  %7.1f');
    c = num2str(PROC.control(k,4:6),'%5d  %7.1f  %7.3f');
    d = mat2str(PROC.control(k,7:(6+2*PROC.white_average)));%matlab74,'%5d ');
    e = num2str(PROC.control(k,(7+2*PROC.white_average)),'%7.3f');
    h3 = text(xtext, ytext+k*yshift, [a,' [', b,'] [',  c,']    ',d,'  ',e]);
        set(h3,'FontName', 'Courier','FontSize',12);
end;
clear a b c d e
movieData(frameIndex) = getframe(h0);  frameIndex = frameIndex + 1;
% *************** End Display: PROC.control ***************

% Find the left and right edges of the object
disp('*** Find the left and right edges of the object.');
displayAngles = [0:10:180];
PROC.ObjectEdgeLeft = []; PROC.ObjectEdgeRight = [];
for k = 1:length(displayAngles)
    % seq, white_file, counts(white), raw_file, counts(raw), angle(raw),...
    %           white_front, white_back, ratio
    index_row = min(find(PROC.angle_to_file_seq(:,1) >= displayAngles(k)));
    index_file = PROC.angle_to_file_seq(index_row,2);
    index_seq = PROC.angle_to_file_seq(index_row,3);
        [raw_image, PARAMETERS] = ReadFITS(FILE.filepath, char(FILE.raw(index_file)));
    index_white = PROC.control(index_seq,(7:(6+2*PROC.white_average +1)));
    disp(['*** White file sequence numbers; ratio: ',mat2str(index_white,3)]);
        index_white_front = index_white(1:PROC.white_average);
        index_white_back  = index_white(PROC.white_average + (1:PROC.white_average));
        ratio = index_white(length(index_white));
    PROC.white_front = zeros(ACQ.rows, ACQ.columns);
    PROC.white_back = zeros(ACQ.rows, ACQ.columns);
    for j = index_white_front
        index_white_file = find(ACQ.white_sequence_number == j);
        temp = ReadFITS(FILE.filepath, char(FILE.white(index_white_file)));
        PROC.white_front = PROC.white_front + temp;
```

```
    end;
    PROC.white_front = PROC.white_front/PROC.white_average;
    for j = index_white_back
        index_white_file = find(ACQ.white_sequence_number == j);
        temp = ReadFITS(FILE.filepath, char(FILE.white(index_white_file)));
        PROC.white_back = PROC.white_back + temp;
    end;
    PROC.white_back = PROC.white_back/PROC.white_average;
    PROC.white = PROC.white_front*ratio + PROC.white_back*(1-ratio);
        abs_image = real(log( (PROC.white - PROC.dark)./(raw_image - PROC.dark) ));
        h0 = figure(frameIndex); clf;
             set(h0, 'Position',[10,10,560, 420],'Color', [1,1,1], 'Name', 'Absorption Images (and edges)');
    subplot(211);  imagesc(ACQ.X_axis, ACQ.Y_axis, abs_image); colormap(gray); axis('ij');  colorbar('vert');
        h2 = title([char(FILE.raw(index_file)), '  ',num2str(PARAMETERS.angle),' deg']);
        set(h2, 'Interpreter', 'none');
        abs_sum = sum(abs_image);
        abs_sum_threshold = 0.1*(max(abs_sum) - min(abs_sum)) + min(abs_sum);
         CenterOfGravity =  round(abs_sum.*ACQ.X_axis/abs_sum);
    index = find( abs_sum >= abs_sum_threshold);
    PROC.ObjectEdgeLeft  = [PROC.ObjectEdgeLeft;  ACQ.X_axis(min(index))];
    PROC.ObjectEdgeRight = [PROC.ObjectEdgeRight; ACQ.X_axis(max(index))];
    subplot(212);        plot(ACQ.X_axis, abs_sum, 'k-');        xlabel('CCD column');        ylabel('column sum
       title(['Left = ',num2str(PROC.ObjectEdgeLeft(length(PROC.ObjectEdgeLeft))),...
        ';  Center = ',num2str(CenterOfGravity),';  Right = ',...
        num2str(PROC.ObjectEdgeRight(length(PROC.ObjectEdgeRight)))]);
    drawnow;
        movieData(frameIndex) = getframe(h0);  frameIndex = frameIndex + 1;
end;
PROC.ObjectEdgeLeft = min(PROC.ObjectEdgeLeft);   PROC.ObjectEdgeRight = max(PROC.ObjectEdgeRight);
index_col_low = find(ACQ.X_axis <= (PROC.ObjectEdgeLeft - PROC.AirAroundObject));
index_col_high = find(ACQ.X_axis >= (PROC.ObjectEdgeRight + PROC.AirAroundObject));
index_columns = [index_col_low, index_col_high];
PROC.air_columns = [index_col_low, index_col_high];
%PROC.ObjectEdgeLeft=20;
clear index*


% Make absorption images and write to disk
disp('*** Make absorption images and write to disk');
frameIndex_fixed = frameIndex;
index_raw_good = PROC.control(find(PROC.control(:,4) > 0),1);
index_white_front_old = zeros(1, PROC.white_average);  index_white_back_old = zeros(1, PROC.white_average);
for index_proc_control_row = index_raw_good',
    index_raw_seq  = PROC.control(index_proc_control_row, 1);
    index_raw_file = PROC.control(index_proc_control_row, 4);
    index_white    = PROC.control(index_proc_control_row,(7:(6+2*PROC.white_average +1)));
    disp(['*** White file sequence numbers; ratio: ',mat2str(index_white,3)]);
        index_white_front = index_white(1:PROC.white_average);
        index_white_back  = index_white(PROC.white_average + (1:PROC.white_average));
    if sum(index_white_front ~= index_white_front_old) > 0,
        PROC.white_front = zeros(ACQ.rows, ACQ.columns);
        for j = index_white_front,
            index_white_file = find(ACQ.white_sequence_number == j);
            temp = ReadFITS(FILE.filepath, char(FILE.white(index_white_file)));
            PROC.white_front = PROC.white_front + temp;
        end;
        PROC.white_front = PROC.white_front/PROC.white_average;
        index_white_front_old = index_white_front;
    end;
    if sum(index_white_back ~= index_white_back_old) > 0,
        PROC.white_back  = zeros(ACQ.rows, ACQ.columns);
        for j = index_white_back,
            index_white_file = find(ACQ.white_sequence_number == j);
            temp = ReadFITS(FILE.filepath, char(FILE.white(index_white_file)));
            PROC.white_back = PROC.white_back + temp;
        end;
        PROC.white_back = PROC.white_back/PROC.white_average;
```

```
        index_white_back_old = index_white_back;
    end;
    PROC.white = PROC.white_front*ratio + PROC.white_back*(1-ratio);
    [raw_image, PARAMETERS] = ReadFITS(FILE.filepath, char(FILE.raw(index_raw_file)));
    abs_image = real(log( (PROC.white - PROC.dark)./(raw_image - PROC.dark) ));
% FUDGE:  adjust abs_image offset
abs_image_offset = mean(mean(abs_image(1:ACQ.rows, PROC.air_columns)));
abs_image = abs_image - abs_image_offset;
    temp = char(FILE.raw(index_raw_file));
    temp = temp( 1:(length(temp)-4));
    index = findstr(temp, '.raw.');
    temp(index:(index+4)) = '.abs.';
    temp = [temp,'mat'];
    disp(['*** Absorption filename: ', temp]);
    FILE.abs(index_raw_file) = {temp};
    eval(['save ', [FILE.filepath,temp], ' abs_image']);
    if mod(index_raw_file,10)==1,
            h0 = figure(frameIndex_fixed); clf;
                set(h0, 'Position',[10,10,560, 420],'Color', [1,1,1], 'Name', 'Absorption Images and Histograms '
            subplot(211);  imagesc(ACQ.X_axis, ACQ.Y_axis, abs_image); colormap(gray); axis('ij');  colorbar('ver
            h2 = title([char(FILE.raw(index_raw_file)), '  ',num2str(PARAMETERS.angle),' deg']);
            set(h2, 'Interpreter', 'none');
            subplot(212);
            temp = reshape(abs_image, ACQ.columns*ACQ.rows, 1);
            hist(temp, 100);
            xlabel('abs');        ylabel('counts');
               title(['min abs = ',num2str(min(temp)),';  max abs = ',num2str(max(temp))]);
            drawnow;
           % movieData(frameIndex) = getframe(h0);  frameIndex = frameIndex + 1;
    end;

end;

movieData(frameIndex) = getframe(h0);  frameIndex = frameIndex + 1;

clear index* temp abs* framIndex_fixed

% save 1_FirstReconstruction
% movie2avi(movieData, [FILE.filepath, 'FirstRecon_Tuesday1140AM', '.avi']);
%
% break;

disp('*** Initialize sinograms to zeros: size ACQ.columns by length(ACQ.angles) ');
disp('*** Create filenames and store in FILE.sino');
disp('*** Create variable names and store in PROC.sino');
t0 = clock;
temp = char(FILE.abs(1));
index = findstr(temp,'.');
temp = temp(1:index(1));
for index_sino = ACQ.ccd_yoffset - 1 + (1:ACQ.rows)
    k = index_sino - ACQ.ccd_yoffset + 1;
    FILE.sino(k) = {[temp, 'sino.', num2str(index_sino),'.mat']};
    eval(['sino_',num2str(index_sino),' = zeros(length(ACQ.angle), ACQ.columns);']);
    PROC.sino(k) = {['sino_',num2str(index_sino)]};
end;
clear index_sino k

disp(['*** Time required to initialize all sinograms = ',num2str(etime(clock,t0))]);
disp('*** Calculate all s (large RAM requirements)');
t0 = clock;
for index_angle = 1:length(PROC.angle_to_file_seq(:,1))
    index_file = PROC.angle_to_file_seq(index_angle, 2);
    load([FILE.filepath, char(FILE.abs(index_file))]);   % loads abs_image
    for index_sino = ACQ.ccd_yoffset - 1 + (1:ACQ.rows)
        index_row = index_sino - ACQ.ccd_yoffset + 1;
        abs_row = abs_image(index_row,:);
        eval(['sino_',num2str(index_sino),'(index_angle,:) = abs_row;']);
        index_file = index_sino - ACQ.ccd_yoffset + 1;
```

```
        save([FILE.filepath, char(FILE.sino(index_file))], char(PROC.sino(index_file)))   % saves sinogram
        eval(['temp = ', char(PROC.sino(index_file)),';']);
        h0 = figure(frameIndex_fixed); clf;
        set(h0, 'Position',[10,10,560, 420],'Color', [1,1,1], 'Name', 's');
        imagesc(ACQ.X_axis, PROC.angle_to_file_seq(:,1), temp);   colormap(gray); axis('ij');  colorbar('vert');
        xlabel('column');        ylabel('theta (deg)');
        h2 = title([char(PROC.sino(index_file))]);    set(h2, 'Interpreter', 'none');
        movieData(frameIndex) = getframe(h0);  frameIndex = frameIndex + 1;
        clear sino_*
    end;
end;

clear index_angle

disp(['*** Time required to compute all sinograms = ',num2str(etime(clock,t0))]);
disp('*** Store sinograms to disk');
t0 = clock;
%ACQ.rows=20;
% for index_sino = ACQ.ccd_yoffset - 1 + (1:ACQ.rows)
%     index_file = index_sino - ACQ.ccd_yoffset + 1;
%     save([FILE.filepath, char(FILE.sino(index_file))], char(PROC.sino(index_file)))   % saves sinogram
% end;
clear index_sino index_file

disp(['*** Time required to store all sinograms = ',num2str(etime(clock,t0))]);
t0 = clock;
frameIndex_fixed = frameIndex;
% for index_sino = ACQ.ccd_yoffset - 1 + (1:ACQ.rows)
%     index_file = index_sino - ACQ.ccd_yoffset + 1;
%     eval(['temp = ', char(PROC.sino(index_file)),';']);
%     h0 = figure(frameIndex_fixed); clf;
%         set(h0, 'Position',[10,10,560, 420],'Color', [1,1,1], 'Name', 's');
%     imagesc(ACQ.X_axis, PROC.angle_to_file_seq(:,1), temp);   colormap(gray); axis('ij');  colorbar('vert');
%     xlabel('column');        ylabel('theta (deg)');
%     h2 = title([char(PROC.sino(index_file))]);    set(h2, 'Interpreter', 'none');
%     movieData(frameIndex) = getframe(h0);  frameIndex = frameIndex + 1;
% end;
disp(['*** Time required to display all s = ',num2str(etime(clock,t0))]);
clear sino_* index_sino index_file temp

save 2_sino

clear FILE.abs*

% % Vertical streak removal
disp('*** Vertical streak removal ');
%index_sino = ACQ.ccd_yoffset - 1 + (1:ACQ.rows);
X_axis=1:ACQ.columns; %1:ACQ.columns;
frameIndex_fixed=frameIndex;
for index_sino = ACQ.ccd_yoffset - 1 + (1:ACQ.rows)
    index_file = index_sino - ACQ.ccd_yoffset + 1;

   load([FILE.filepath, char(FILE.sino(index_file))]);   % read sinogram
    eval(['sinogram = ',char(PROC.sino(index_file)),';']);
    sinogram_title = char(PROC.sino(index_file));
   % eval(['clear ',char(PROC.sino(index_sino(j)))]);
  h0 = figure(frameIndex_fixed); clf;
  %eval(['sum_sinogram = sum(sino_', num2str(index_sino(j)), ')/ACQ.NumberOfProjections;']);
       %sum_sinogram = sum(sinogram);
    %subplot(411);  plot(X_axis, sum_sinogram, 'k-');
    subplot(411);imagesc(sinogram);colormap(gray);%drawnow;%plot(X_axis, sinogram, 'k-');
%     ts=size(sinogram);
%     factor=zeros(ts(1), ts(2));

    %med_sinogram=sum(medfilt2(sinogram, [1 3]));
    med_sinogram=medfilt2(sinogram, [1 3]);

   % subplot(412);        plot(X_axis, med_sinogram, 'k-');
```

```
651     subplot(412);imagesc(med_sinogram);colormap(gray);%drawnow;%plot(X_axis, med_sinogram, 'k-');
652     %title('median filtering and column sum');
653         title('median filtering');
654 
655     diff=sum(sinogram- med_sinogram);
656     subplot(413);plot(X_axis, diff, 'k-');%drawnow;
657     title(num2str(index_sino));
658 
659     [srow, scolumn]=find(abs(diff) > 4*std2(diff));
660 
661     %ind=find(abs(diff) > 5*std2(diff));
662     %factor(ind)=1;
663     %factor2=1 - factor;
664 
665 %         for k = 1:length(PROC.angle_to_file_seq(:,1))
666 %             eval(['new_sinogram(k,:) = sinogram(k,:) - diff./(length(PROC.angle_to_file_seq(:,1)));']);
667 %     end;
668      %new_sinogram=sinogram;
669      %med_sinogram = medfilt2(med_sinogram, [1 3]);
670      sinogram(:,scolumn) = med_sinogram(:,scolumn);
671     %new_sinogram=sinogram .*factor2 + (medfilt2(med_sinogram, [1 3]) .*factor);
672 
673         subplot(414);imagesc(sinogram);        xlabel('CCD column');
674         title(['filtered sinogram']); %drawnow;
675 
676     %set(h0, 'Position',[10,10,560, 420],'Color', [1,1,1], 'Name', 'Streak Removal');
677         %movieData(frameIndex) = getframe(h0);  frameIndex = frameIndex + 1;
678  %clear sinogram
679 
680 % DC correction after vertical streak removal
681 disp('*** Repeat DC correction after vertical streak removal ');
682 %for j = 1:ACQ.rows-1
683 
684          % eval(['sinogram = ',char(PROC.sino(j)),';']);
685  for k2=1:length(PROC.angle_to_file_seq(:,1))
686     % for k = 1:ACQ.NumberOfProjections
687     if PROC.AirAroundObject~=0
688                 DC_offset = mean([sinogram(k2, 1:PROC.AirAroundObject), sinogram(k2, (ACQ.columns-PROC.AirAroundO
689     else
690         DC_offset=0;
691     end;
692         sinogram(k2,:) = sinogram(k2,:) - DC_offset;
693  end;
694         %eval(['PROC.sino',num2str(j),'  = new_sinogram;']);
695   %  index_sino=ACQ.ccd_yoffset-1+j
696     eval(['sino_',num2str(index_sino),'= sinogram;']);
697     %  eval(['clear ',char(PROC.sino(j))]);
698 
699      save([FILE.filepath, char(FILE.sino(index_file))], char(PROC.sino(index_file)))
700     clear sino_* med_sinogram sinogram diff DC_offset
701 end;
702 movieData(frameIndex) = getframe(h0);  frameIndex = frameIndex + 1;
703 
704 clear index_sino index_file k2 k %factor factor2
705 
706 % Autocentering based on center of gravity of object
707 disp('*** Autocentering based on center of gravity of object ');
708 CenterOfGravity = [];
709 PROC.angle = PROC.angle_to_file_seq(:,1)';
710 for index_sino = ACQ.ccd_yoffset - 1 + round(linspace(5,ACQ.rows-5,15))
711    % for index_sino = ACQ.ccd_yoffset - 1 + round(linspace(5,ACQ.rows-5,7))
712     index_file = index_sino - ACQ.ccd_yoffset + 1;
713     load([FILE.filepath, char(FILE.sino(index_file))]);   % read
714     eval(['sinogram = ',char(PROC.sino(index_file)),';']);
715 
716     sinogram_title = char(PROC.sino(index_file));
717     eval(['clear ',char(PROC.sino(index_file))]);
718         h0 = figure(frameIndex); clf;
```

```
        set(h0, 'Position',[10,10,560, 420],'Color', [1,1,1], 'Name', 'Sinogram');
        imagesc(ACQ.X_axis, PROC.angle, sinogram);  colorbar('vert')
        h2 = title(sinogram_title);  set(h2, 'Interpreter', 'none');    drawnow;
        movieData(frameIndex) = getframe(h0);   frameIndex = frameIndex + 1;
    CenterOfGravityVector = zeros(1,length(PROC.angle));

   if sum(sum(sinogram)) > 0.00001 % to avoid the air only rows.

    for k3 = 1:length(PROC.angle)

         CenterOfGravityVector(k3) = ACQ.ccd_xoffset + sum(sinogram(k3,:).*(1:ACQ.columns))/sum(sinogram(k3,:));
    end;
        h0 = figure(frameIndex); clf;
        set(h0, 'Position',[10,10,560, 420],'Color', [1,1,1], 'Name', 'Autocentering');
    subplot(311);  plot(PROC.angle, CenterOfGravityVector, 'k.'); hold on;
    coef = [(ACQ.X_axis(round(length(ACQ.X_axis)/2)))  100   0.01];
    new_coef = lsqcurvefit('sinusoidal_function', coef, PROC.angle,  CenterOfGravityVector);
    y = sinusoidal_function(new_coef, PROC.angle);
    CenterOfGravity = [CenterOfGravity, new_coef(1)];
    plot(PROC.angle, y, 'k-');
    title(['Expt. center of gravity (.) and sinusoidal fit (-); Center = ',...
        num2str(CenterOfGravity(length(CenterOfGravity)))]);
    subplot(312);  plot(PROC.angle, (CenterOfGravityVector - y), 'k-');        % plot residuals
    title(['Residuals (= expt CG - fit); ACQ.columns = ',num2str(ACQ.columns)]);        drawnow;
        movieData(frameIndex) = getframe(h0);     frameIndex = frameIndex + 1;
   end;

end;



clear index_sino index_file k3


%save 2_sino

%break

PROC.MeanCenterOfGravity = sort(CenterOfGravity);


if length(CenterOfGravity) > 2
%PROC.MeanCenterOfGravity = mean(PROC.MeanCenterOfGravity(3:(length(PROC.MeanCenterOfGravity)-2 )));
PROC.MeanCenterOfGravity = mean(PROC.MeanCenterOfGravity(2:(length(PROC.MeanCenterOfGravity)-2 )));
else
    PROC.MeanCenterOfGravity=mean(PROC.MeanCenterOfGravity);
end;

PROC.FieldOfView=2*(max(abs(PROC.MeanCenterOfGravity-(PROC.ObjectEdgeLeft-PROC.AirAroundObject)),...
    abs(PROC.MeanCenterOfGravity-(PROC.ObjectEdgeRight+PROC.AirAroundObject))));

PROC.FieldOfView = ceil(PROC.FieldOfView/5)*5 + 1;



disp(['*** Best field-of-view: ', num2str(PROC.FieldOfView)]);

% if (round(PROC.MeanCenterOfGravity-PROC.FieldOfView/2) > ACQ.ccd_xoffset)
%     xmin=round(PROC.MeanCenterOfGravity-PROC.FieldOfView/2);
%     new_ACQ.ccd_xoffset=xmin;
% else
%     xmin = find(ACQ.X_axis == round(PROC.MeanCenterOfGravity - PROC.FieldOfView/2));
%     new_ACQ.ccd_xoffset= xmin; %ACQ.ccd_xoffset;
% end
%need to look at the above lines more carefully

xmin=round(PROC.MeanCenterOfGravity-PROC.FieldOfView/2);
new_ACQ.ccd_xoffset = xmin;
```

```
787
788 % index_column = xmin:(xmin + PROC.FieldOfView - 1);
789 % index_row = 1:ACQ.NumberOfProjections;
790
791 extension = xmin+ PROC.FieldOfView -1;
792
793 if extension > max(ACQ.X_axis)
794     sino_extensionR=zeros(length(PROC.angle_to_file_seq(:,1)),extension-max(ACQ.X_axis));
795 else
796     sino_extensionR=[];%zeros(length(PROC.angle_to_file_seq(:,1)),0);
797 end;
798
799
800 if new_ACQ.ccd_xoffset < ACQ.ccd_xoffset
801      sino_extensionL=zeros(length(PROC.angle_to_file_seq(:,1)), ACQ.ccd_xoffset-xmin);
802 else
803     sino_extensionL=[]; % zeros(length(PROC.angle_to_file_seq(:,1)),0);
804 end;
805
806
807 new_ACQ.X_axis = new_ACQ.ccd_xoffset - 1 + (1:PROC.FieldOfView);
808 new_ACQ.X_axis(1)
809 PROC.sino_crop_row = 1:length(PROC.angle_to_file_seq(:,1));
810 PROC.sino_crop_column= new_ACQ.X_axis;%:extension - 1;% - ACQ.ccd_xoffset:(extension-1-ACQ.ccd_xoffset);
811 %PROC.X_axis = new_ACQ.X_axis(PROC.sino_crop_column);
812 PROC.X_axis = new_ACQ.X_axis;%:extension-1;%PROC.sino_crop_column;
813 PROC.angle = PROC.angle(PROC.sino_crop_row);
814 %break
815
816 % Crop all sinograms
817 t0 = clock;
818 for index_sino = ACQ.ccd_yoffset - 1 + (1:ACQ.rows)
819     index_file = index_sino - ACQ.ccd_yoffset + 1;
820     load([FILE.filepath, char(FILE.sino(index_file))]);  % read
821     eval(['sinogram = ',char(PROC.sino(index_file)),';']);
822
823
824 test_conditions2 = mat2str([isempty(sino_extensionL==0),...
825                         isempty(sino_extensionR==0)]);
826
827    switch test_conditions2
828          %Field of View needs to extend to left and right
829        case {'[false false]'}
830            %             disp(['          test_conditions=', test_conditions]);
831            sinogram = horzcat(sino_extensionL, sinogram, sino_extensionR);
832        case {'[false true]'}
833             % The range of raw images does not match PROC.white_frequency
834             %   Use white_front and set ratio = 1
835 %             disp(['          test_conditions=', test_conditions]);
836             sinogram=horzcat(sino_extensionL, sinogram);
837        case {'[true false]'}
838             % The range of raw images does not match PROC.white_frequency
839             %   Use white_back and set ratio = 1
840 %             disp(['          test_conditions=', test_conditions]);
841             sinogram=horzcat(sinogram, sino_extensionR);
842
843        case {'[true true]'}
844             %
845             sinogram=sinogram(PROC.sino_crop_row, xmin-ACQ.ccd_xoffset-1+(1:length(PROC.sino_crop_column)));
846
847    end;
848
849  %   FILE.sino(index_file)=horzcat(FILE.sino(index_file), sino_extension);
850     %eval([char(PROC.sino(index_file)), '= ',...
851      %       char(PROC.sino(index_file)),'(PROC.sino_crop_row, PROC.sino_crop_column);']);
852     eval([char(PROC.sino(index_file)), '= sinogram ;']);
853     %eval(['sinogram = ', char(PROC.sino(index_file)),';']);
854     save([FILE.filepath, char(FILE.sino(index_file))], char(PROC.sino(index_file)))   % saves
```

```
        h0 = figure(frameIndex_fixed); clf;
        set(h0, 'Position',[10,10,560, 420],'Color', [1,1,1], 'Name', 's');
    imagesc(PROC.X_axis, PROC.angle, sinogram );
        colormap(gray); axis('ij');  colorbar('vert');
        xlabel('column');        ylabel('theta (deg)');
        h2 = title([char(PROC.sino(index_file))]);    set(h2, 'Interpreter', 'none');
    movieData(frameIndex) = getframe(h0);  frameIndex = frameIndex + 1;
end;
disp(['*** Time required to crop all sinograms = ',num2str(etime(clock,t0))]);

clear sino_* index_sino index_file sinogram

% Make variable names for reconstructed slices
temp = char(FILE.sino(1));  index = findstr(temp,'.');  temp = temp(1:index(1));

for index_slice = ACQ.ccd_yoffset - 1 + (1:ACQ.rows)
    k = index_slice - ACQ.ccd_yoffset + 1;
    FILE.slice(k) = {[temp, 'slice.', num2str(index_slice),'.mat']};
    PROC.slice(k) = {['slice_',num2str(index_slice)]};
end;

% Convert s to reconstructed slices
t0 = clock; frameIndex_fixed = frameIndex;
for index_slice = ACQ.ccd_yoffset - 1 + (1:ACQ.rows)
    index_file = index_slice - ACQ.ccd_yoffset + 1;
    load([FILE.filepath, char(FILE.sino(index_file))]);   % read
    eval([' sinogram= ', char(PROC.sino(index_file)),';']);
    slice = iradon(sinogram', PROC.angle, 'linear', 'Hann', PROC.FieldOfView)';
    eval([char(PROC.slice(index_file)), ' = slice;']);
    save([FILE.filepath, char(FILE.slice(index_file))], char(PROC.slice(index_file)))   % saves
    h0 = figure(frameIndex_fixed); clf;
        set(h0, 'Position',[100,10,560, 420],'Color', [1,1,1], 'Name', 'Reconstructed Slice');
        imagesc(PROC.X_axis, PROC.X_axis, slice); colorbar('vert');
        slice_title = char(FILE.slice(index_file));
        h2 = title(slice_title);
            set(h2, 'Interpreter', 'none', 'FontSize', 18);     drawnow;
        movieData(frameIndex) = getframe(h0);    frameIndex = frameIndex_fixed + 1;
    clear sino_* slice_*
end;

clear index_slice index_file

disp(['*** Time required to reconstruct all slices = ',num2str(etime(clock,t0))]);


temp = char(FILE.slice(1));
index = findstr(temp,'.');
temp = temp(1:index(1));
volume = zeros(size(slice,1), size(slice,2), length(FILE.slice));
t0 = clock;
for index_slice = 1:length(FILE.slice)
    load([FILE.filepath, char(FILE.slice(index_slice))]);   % read slice
    eval(['slice = ', char(PROC.slice(index_slice)),';']);
        clear slice_*
    volume(:,:,index_slice) = slice;
end;
disp(['*** Time required assemble volume from slices = ',num2str(etime(clock,t0))]);
size(volume)
cd(FILE.filepath);
save volume.mat volume
clear volume

save 1_FirstReconstruction
movie2avi(movieData, [FILE.filepath, 'FirstRecon_Wed_145PM', '.avi']);
```

Linear Least Squares Fit

This matlab algorithm takes the absorption data that was created in the reconstruction code and converts it into chemical concentrations by a linear least squares fit. The linear equation used is based of a form of Beer's law. This code has less data logging and stored variables as opposed to the previous code. This is due to the fact that typically this code is run on clusters or at minimum the slices are divided up on local machines for this reason simplicity in the code is desired.

```
clear;clc;close('all')

load lbutler_vols
load NIST_data
tic
for i = 1:200
        display(['On Slice ', num2str(i)])
    for j = 1:200
        for k = 1:500
        A_expt = [FR_12__cube(i,j,k),FR_13p4__cube(i,j,k),FR_17__cube(i,j,k),FR_25__cube(i,j,k),FR_30p43__cube(i,
        X0 = [0.2,0.02,0.7];    t1 = clock;
        foptions = optimset('TolX', 1e-5, 'Display', 'off','LargeScale', 'off');
        c = [(c_C8H5Br3-c_Other),(c_Sb2O3-c_Other),(c_SiO2-c_Other)];
        d = A_expt' - c_Other;
        [X]  = lsqlin(c, d,[],[],[],[],[0,0,0]',[1,1,1]',X0,foptions);
        FR(i,j,k) = X(1);
        Sb2O3(i,j,k) = X(2);
        SiO2(i,j,k) = X(3);
        end
    end
end
t = toc;
clear FR* mask
save('results.mat')
```

Fiber Coordinate Points Generation

A mathematica script used to calculate points surrounding selected fiber bundles. Vast majority of the code is definition of simple functions that are later used in conjunction to produce the desired list of points. Point coordinates are exported to .mat files since the next step in done in Matlab.

```
Define Functions

px,py,pz: position vector (location of one end of the cylinder axis)
ax,ay,az: axis vector (vector running along the cylinder axis)
R: radius of the cylinder.

Needs["Histograms`"]
(* Needs["Geometry`Rotations`"]
Needs["BarCharts`"];Needs["Histograms`"];Needs["PieCharts`"]
Needs["BarCharts`"];Needs["Histograms`"]
```

```
Needs["Graphics`Shapes`"] *)

funcCylinderStart[data_] := {data[[1]], data[[2]], data[[3]]};
funcCylinderEnd[data_] := {data[[4]], data[[5]], data[[6]]};
funcCylinderRadius[data_] := data[[7]];
funcCylinderVector[data_] := Module[{},
   funcCylinderEnd[data] - funcCylinderStart[data]  ];
funcCylinderLength[data_] := Norm[ funcCylinderVector[data] ] // N;
funcCylinderVectorNorm[data_] :=
  Module[{},
   funcCylinderVector[data]/Norm[ funcCylinderVector[data] ]  // N];
(* http://mathworld.wolfram.com/VectorNorm.html *)

funcPlane[center_, norm_] := Module[{ x, y, z, a, b, c, d},
   eq = Simplify[Dot[norm, ({x, y, z} - center )]];
   a = Coefficient[eq, x];
   b = Coefficient[eq, y];
   c = Coefficient[eq, z];
   d = Simplify[eq - (a x + b y + c z) ] // N;
   Chop[{a, b, c, d}] // N ];

funcDistanceStart[data_, point_] := Module[{x0, distance},
   x0 = funcCylinderStart[data];
   distance =
    Sqrt[(x0[[1]] - point[[1]])^2 + (x0[[2]] - point[[2]])^2 + (x0[[3]] -
         point[[3]])^2 ] // N];
funcDistanceEnd[data_, point_] := Module[{x0, distance},
   x0 = funcCylinderStart[data] + funcCylinderVector[data][[All, 1]];
   distance =
    Sqrt[(x0[[1]] - point[[1]])^2 + (x0[[2]] - point[[2]])^2 + (x0[[3]] -
         point[[3]])^2 ] // N];
funcDistance[data_, point_] := Module[{x1, x2, x0},
   x1 = funcCylinderStart[data];
   x2 = funcCylinderEnd[data];
   x0 = point;
   Norm[Cross[(x2 - x1), (x1 - x0)]] / Norm[(x2 - x1)] // N];
(* The equation for the distance between a point and a line is at
http://mathworld.wolfram.com/Point-LineDistance3-Dimensional.html  *)

funcFractionalProjection[data_, point_] :=
  Module[{x0, vectorCylinder, lengthCylinder},
   x0 = funcCylinderStart[data];
   vectorCylinder = funcCylinderVector[data];
   lengthCylinder = funcCylinderLength[data];
   fractionalProjection =
    Round[100*Dot[vectorCylinder, point - x0]/lengthCylinder^2] // N];

funcFindPlanePoints[data_, center_, radius_] :=
  Module[{ norm, a, b, c, d, xPoints, yPoints, zPoints, allPoints, span},
   span = Ceiling[1.5 * radius];
   norm = funcCylinderVectorNorm[data];
   {a, b, c, d} = funcPlane[center, norm] ;
   (* Print[a,b,c,d]; *)
   (* Print[ Ordering[Abs[norm ],-1][[1]] ]; *)

   If[Ordering[Abs[norm ], -1][[1]] == 3, (* fiber aligned with Z-axis *)

    xPoints = Table[center[[1]] + i, {i, -span, span}];
    yPoints = Table[center[[2]] + i, {i, -span, span}];
    zPoints =
     Round[Table[ (-d - a xPoints[[i]] - b yPoints[[j]])/
       c, {i, 1, Length[xPoints]}, {j, 1, Length[yPoints]}]];
    allPoints = Table[Module[{distance},
       point = {xPoints[[i]], yPoints[[j]], zPoints[[i, j]]};
       distance = funcDistance[data, point];
       {xPoints[[i]], yPoints[[j]], zPoints[[i, j]], distance}],
      {i, 1, Length[xPoints]}, {j, 1, Length[yPoints]}];
    allPoints = Flatten[allPoints, 1];
```

```
    allPoints = Select[allPoints, #[[4]] <= radius & ],
    (* Print["not Z"] *) ];
   If[Ordering[Abs[norm ], -1][[1]] == 2, (* fiber aligned with Y-axis *)

    xPoints = Table[center[[1]] + i, {i, -span, span}];
    zPoints = Table[center[[3]] + i, {i, -span, span}];
    yPoints =
     Round[Table[ (-d - a xPoints[[i]] - c zPoints[[j]])/
       b, {i, 1, Length[xPoints]}, {j, 1, Length[zPoints]}]];
    allPoints = Table[Module[{distance},
       point = {xPoints[[i]], yPoints[[i, j]], zPoints[[j]]};
       distance = funcDistance[data, point];
       {xPoints[[i]], yPoints[[i, j]], zPoints[[j]], distance}],
      {i, 1, Length[xPoints]}, {j, 1, Length[zPoints]}];
    allPoints = Flatten[allPoints, 1];
    allPoints = Select[allPoints, #[[4]] <= radius & ],
    (* Print["not Y"]*)];
   If[Ordering[Abs[norm ], -1][[1]] == 1, (* fiber aligned with X-axis *)

    yPoints = Table[center[[2]] + i, {i, -span, span}];
    zPoints = Table[center[[3]] + i, {i, -span, span}];
    xPoints =
     Round[Table[ (-d - b yPoints[[i]] - c zPoints[[j]])/
       a, {i, 1, Length[yPoints]}, {j, 1, Length[zPoints]}]];
    allPoints = Table[Module[{distance},
       point = {xPoints[[i, j]], yPoints[[i]], zPoints[[j]]};
       distance = funcDistance[data, point];
       {xPoints[[i, j]], yPoints[[i]], zPoints[[j]], distance}],
      {i, 1, Length[yPoints]}, {j, 1, Length[zPoints]}];
    allPoints = Flatten[allPoints, 1];
    allPoints = Select[allPoints, #[[4]] <= radius & ],
    (* Print["not X"]*)];
   allPoints];
(* The equation for the normal vector to a plane is
 http://mathworld.wolfram.com/NormalVector.html  *)

funcFindAllPoints[data_, radius_] :=
  Module[{x1, x2, topPlane, bottomPlane, xRange, yRange, zRange,
    vectorCylinder, lengthCylinder, allPoints, temp},
   x1 = funcCylinderStart[data];
   x2 = funcCylinderEnd[data];
   TableForm[bottomPlane = funcFindPlanePoints[data, x1, radius]];
   TableForm[topPlane = funcFindPlanePoints[data, x2, radius]];
   xRange = {Min[ {Min[topPlane[[All, 1]] ], Min[bottomPlane[[All, 1]] ] }],
      Max[ {Max[topPlane[[All, 1]] ], Max[bottomPlane[[All, 1]] ] }] };
   yRange = {Min[ {Min[topPlane[[All, 2]] ], Min[bottomPlane[[All, 2]] ] }],
      Max[ {Max[topPlane[[All, 2]] ], Max[bottomPlane[[All, 2]] ] }] };
   zRange = {Min[ {Min[topPlane[[All, 3]] ], Min[bottomPlane[[All, 3]] ] }],
      Max[ {Max[topPlane[[All, 3]] ], Max[bottomPlane[[All, 3]] ] }] };
   vectorCylinder = funcCylinderVector[data];
   lengthCylinder = Norm[funcCylinderVector[data]];
   allPoints = {};
   For[i = Min[xRange], i <= Max[xRange], i++,
    For[j = Min[yRange], j <= Max[yRange], j++,
     For[k = Min[zRange], k <= Max[zRange], k++,
       x0 =  {i, j, k};
       count = count + 1;
       distanceToCylinder =
        Norm[Cross[(x2 - x1), (x1 - x0)]] / Norm[(x2 - x1)] // N;
       If[distanceToCylinder <= radius,
        fractionalProjection = funcFractionalProjection[data, x0];
        allPoints =
         Append[allPoints, {i, j, k, distanceToCylinder,
           fractionalProjection, -99}]];
       ];];];
   temp = Sort[allPoints, #1[[4]] > #2[[4]] &];
   temp = Sort[temp, #1[[5]] < #2[[5]] &]];

```

```
funcFindSpherePoints[dataSphere_] :=
  Module[{radius, center, span, xPoints, yPoints, zPoints, distances,
    allPointsInCube, allPointsInSphere},
   radius = dataSphere[[4]];
   center = dataSphere[[{1, 2, 3}]];
   span = Ceiling[ radius];
   xPoints = Table[center[[1]] + i, {i, -span, span}];
   yPoints = Table[center[[2]] + i, {i, -span, span}];
   zPoints =  Table[center[[3]] + i, {i, -span, span}];
   distances = Table[
      Sqrt[(center[[1]] - xPoints[[i]])^2 + (center[[2]] -
          yPoints[[j]])^2  + (center[[3]] - zPoints[[k]])^2 ],
      {i, 1, Length[xPoints]}, {j, 1, Length[yPoints]}, {k, 1,
       Length[zPoints]}] // N;
   allPointsInCube = Flatten[ Table[
      {xPoints[[i]], yPoints[[j]],  zPoints[[k]], distances[[i, j, k]] },
      {i, 1, Length[xPoints]}, {j, 1, Length[yPoints]}, {k, 1,
       Length[zPoints]}], 2];
   indexByDistance = Ordering[ allPointsInCube[[All, 4]] ];
   TableForm[allPointsInCube = allPointsInCube[[indexByDistance, All]] ,
                                    TableHeadings -> {None, {"i", "j", "k", "d"}}];
   TableForm[
    allPointsInSphere = Select[allPointsInCube, #[[4]] <= radius & ] ,
                                    TableHeadings -> {None, {"i", "j", "k", "d"}}] ;
   allPointsInSphere];

drawCylinder[data_] := Module[{g1, g2},
   g1 = Graphics3D[{Opacity[0.5],
      Cylinder[{ funcCylinderStart[data], funcCylinderEnd[data]},
       funcCylinderRadius[data]]}];
   g2 = Graphics3D[{Thickness[Tiny],
      Line[{ funcCylinderStart[data], funcCylinderEnd[data]}]}];
   Show[{g1, g2}]   ];

drawSphereSolid[dataSphere_] := Module[{x, y, z, r},
   {x, y, z} = dataSphere[[{1, 2, 3}]];
   r = dataSphere[[4]];
    Graphics3D[{Opacity[0.5], Sphere[{x, y, z}, r] } ]] ;

cylinderData =  {{101, 178, 201, 178, 177, 155}, {179, 62, 318, 62,
    61, 251}, {164, 159, 148, 159, 161, 60},
   {161, 190, 201, 190, 190, 136}, {190, 175, 373, 175, 174,
    274}, {174, 161, 286, 161, 156, 175},
   {156, 95, 376, 156, 95, 344}, {95, 127, 373, 143, 145, 434}, {66,
    191, 351, 66, 66, 397}, {161, 66, 24, 161, 161, 66},
   {11, 161, 154, 11, 11, 243}, {130, 11, 64, 132, 130, 145}, {145,
    132, 306, 146, 145, 379}};
Do[
 radius = 15;
 spherePoints =
  funcFindSpherePoints[dataSphereAll[[indexCylinder, All]] ];
  allPoints =
  funcFindAllPoints[ cylinderData[[indexCylinder]], radius] ;

 (* Find the overlapping points of cylinder and sphere *)

 TableForm[
  overlapPoints =
   Intersection[spherePoints[[All, {1, 2, 3}]],
    allPoints[[All, {1, 2, 3}]] ] ];

 (* Find update the allPoints list in column #6.  Change #6 from -99 \
to Round[fractional projection] *)

 For[index = 1, index <= Dimensions[overlapPoints][[1]], index++,
   point = overlapPoints[[index]];
    fractionalProjection =
    funcFractionalProjection[cylinderData[[indexCylinder]], point];
```

```
   indexAllPoints =
    Flatten[Position[allPoints[[All, {1, 2, 3}]], point]];
   (* Print[indexAllPoints, "  ", allPoints[[indexAllPoints,All]]];  *)

      allPoints[[indexAllPoints, 6]] = fractionalProjection; ]

  (* Find update the allPoints list in column #6.
  Change #6 from -99 to Round[fractional projection] *)

  allPointsRemaining = {};
 For[index = 0, index <= 100, index++,
   diskOverlapPoints = Select[allPoints, #[[6]] == index & ] ;
   diskRadius = Min[   diskOverlapPoints[[All, 4]]  ];
   diskCylinderPoints = Select[allPoints, #[[5]] == index & ] ;
   diskCylinderPoints =
    Select[diskCylinderPoints, #[[4]] <= diskRadius & ] ;
   allPointsRemaining =
    Join[allPointsRemaining, diskCylinderPoints]; ]
  TableForm[ allPointsRemaining ];
 Export[ToFileName[{"/Users/heath/"},
   "Fiber" <> ToString[indexCylinder] <> "_coords.mat"],
  allPointsRemaining];
 , {indexCylinder, Length[cylinderData]}]
```

Collision Detection

This algorithm takes input from the mat files created by the Fiber Coordinate Points mathematica script. It then calculates all possible collisions between the fiber bundles and its surrounding area. Collisions are defined as other fibers whose volume is larger than than 3×3×3 and are connected. Voxels that pass the collision detection are saved for use in the RDF plot algorithms listed after this one.

```
clear; clc;
load /Users/heath/RDF/bundle_mask.mat
load /Users/heath/RDF/lbutler_fit_results.mat Sb2O3
load /Users/heath/RDF/march_12_new_fiber_coords.mat
FR = Sb2O3;
clear Sb2O3;
for i = 1:length(points)
points(i).sort_all = points(i).allpoints;
end

for i = 1:length(points);
    disp(['============Fiber ',num2str(i),'============'])
    points(i).sort_all(:,7) = 0; % Will be the volume percent at that [i,j,k]
    points(i).sort_all(:,8) = 0; % Will be the collision detection number for the 26 neighbor sum at [i,j,k]
    for n = 1:length(points(i).sort_all(:,7))
        if points(i).sort_all(n,1) < 0 | points(i).sort_all(n,2) < 0 | ...
            points(i).sort_all(n,3) < 0 |points(i).sort_all(n,1) > 199 | ...
            points(i).sort_all(n,2) > 199 | points(i).sort_all(n,3) > 499
        else %Excludes points outside of bounds
            points(i).sort_all(n,7) = FR(points(i).sort_all(n,1)+1,points(i).sort_all(n,2)+1,points(i).sort_all(n
            %Adding one to coords due to Mathematica being base 0
            tocheck = funcFindNeighbors([points(i).sort_all(n,1:3) 1]); % User function used to create the [i,j,k
            clear maskList
                for m = 1:length(tocheck)
                    maskList(m) = mask(tocheck(m,1),tocheck(m,2),tocheck(m,3));
```

```
                end
            points(i).sort_all(n,8) = sum(maskList);
        end

    end

    bundle_r = 4; % Our "soft" radius for our fiber bundles, we are guessing
                  % that after this radius the SiO2 concentration should drop
                  % off as it is away from the bundles.


    for n = 1:100 % Loop through all 100 disks that were defined in column 5
        safe_r = bundle_r; % Dummy value meant to be overwritten on first pass
        all_disk_index = find(points(i).sort_all(:,5)==n); % Find all points in a disk n
        disk_max_r = max(points(i).sort_all(all_disk_index,4)); % Find the max distance from the points to the
        points(i).sort_all(all_disk_index,6) = disk_max_r; % Column 6 now acquires the value for the point %furth
        fiber(i).disk(n) = mean(points(i).sort_all(all_disk_index,6)); % Take the avg of the points in disk n and
        disk_index = find(points(i).sort_all(:,5)==n & points(i).sort_all(:,8) >= 9 & points(i).sort_all(:,4) >=

        % Find all points that match the criteria of being in the correct
        % disk with few collsions (less than 8) and are outside our "soft"
        % radius
        if isempty(disk_index) == 1
            safe_r = 15;
        else
            safe_r = min(points(i).sort_all(disk_index,4));
        end

        disp(['n = ', num2str(n),'   safe_r = ', num2str(safe_r), '  length(disk_index) = ', num2str(length(disk_
        % Loop over points in disk n and determin the point with the min
        % distance to the center of the fiber... there might be a problem
        % with my logic here.
        points(i).sort_all(all_disk_index,6) = safe_r; % Overwrite max dist if there is a min distance for that d
        exceed_safe_r = find(points(i).sort_all(all_disk_index,4) > safe_r);
        points(i).sort_all(exceed_safe_r,:) = []; % Find all points that exceed the safe r value and delete them.
    end


end

 for i = 1:length(fiber)
     close all
     bar(fiber(i).disk)
     axis([1 100 1 15])
     title(['Fiber ',num2str(i)])
     drawnow;
     movie_data(i) = getframe(gcf);
     numpoints(i) = length(points(i).sort_all);
end
movie2avi(movie_data,'fiber_disks_bar.avi')

clear movie_data
bar(numpoints)
drawnow;
save B_results_apr30_Sb2O3.mat points

```

RDF Plots

Simple script to generate RDF plots.

```
%%
clear; clc; close all;

warning off all
pix2micron = 3.26;
binwidth = 1;
%%
for k = 1:3
    clear points
    switch k
        case{1}
                load /Users/heath/RDF/B_results_apr17_FR.mat
        case{2}
                load /Users/heath/RDF/B_results_apr17_Sb2O3.mat
        case{3}
                load /Users/heath/RDF/B_results_apr17_SiO2.mat
    end
    %%
        r_and_conc = [];
        for i = 1:length(points)
            index = find(points(i).sort_all(:,7) > 0);
            temp = [points(i).sort_all(index,4), points(i).sort_all(index,7)];
            r_and_conc = [r_and_conc; temp];

        end
        r_and_conc(:,1) = r_and_conc(:,1)*pix2micron;
        r_and_conc = sortrows(r_and_conc,1);


        binaxis = 0:binwidth:(binwidth*ceil(max(r_and_conc(:,1)))/binwidth);


        mean_conc = zeros(size(binaxis));
        std_conc = zeros(size(binaxis));

        for j = 1:length(binaxis)-1
            index = find( r_and_conc(:,1) > binaxis(j) & r_and_conc(:,1) <= binaxis(j+1) );
            binsize(j) = length(index);
            mean_conc(j) = mean(r_and_conc(index,2));
            std_conc(j) = std(r_and_conc(index,2));
        end

        cutoff = length(binsize);%+min([find(std_conc == 0, 1, 'first'), find(isnan(std_conc) == 1, 1, 'first')])
        conc_bin = mean_conc(1:cutoff-1);
    switch k
        case{1}
                FR_conc_bin = conc_bin;
        case{2}
                Sb2O3_conc_bin = conc_bin;
        case{3}
                SiO2_conc_bin = conc_bin;
    end
        std_bin = std_conc(1:cutoff-1);

        figure(k)
        h1=errorbar(binaxis(1:length(conc_bin))+binwidth,conc_bin,std_bin,'-ko');
        set(h1, 'MarkerSize', 7, 'MarkerFaceColor','White', 'MarkerEdgeColor','Black');
        set(gca, 'FontName', 'Palatino', 'FontSize', 18)
        L1=xlabel(['r/ \mum']);
        L2=ylabel(['[FR]/vol% (bin width = ',num2str(binwidth),'\mum)']);
        set(gca, 'FontName', 'Palatino', 'FontSize', 18)
        set(L1, 'FontName', 'Palatino', 'FontSize', 18)
        set(L2, 'FontName', 'Palatino', 'FontSize', 18)
        xt = get(gca,'XTick');
        set(gca,'XTickLabel',ceil(xt))

end

```

```
figure
scatter(binaxis(1:length(conc_bin))+binwidth,(FR_conc_bin./(100-SiO2_conc_bin)*100),'ko');


figure
scatter(binaxis(1:length(conc_bin))+binwidth,(Sb2O3_conc_bin./(100-SiO2_conc_bin)*100),'ko');
```

Domain Extraction

Mathematica script that takes concentrations and fits the data to an gaussian equation.

```
DomainArea = Import["Domain_TRIM_2_10S.h5", {"Datasets", "/Area"}];
DomainCentroids =
  Import["Domain_TRIM_2_10S.h5", {"Datasets", "/Centroids"}];
DomainBox = Import["Domain_TRIM_2_10S.h5", {"Datasets", "/Box"}];

(*nx, ny, nz = the length of the sides of the bounding box around the \
Domain bubble. dxy, dxz, dyz is the difference between sides and would = \
0 in a perfect cube, meaning a perfectly spherical bubble*)
todel = Reap[For[i = 1, i <= Length[DomainBox], i++,
   nx = DomainBox[[i, 4]];
   ny = DomainBox[[i, 5]];
   nz = DomainBox[[i, 6]];
   dxy = Abs[nx - ny];
   dxz = Abs[nx - nz];
   dyz = Abs[ny - nz];
   avg = Mean[{dxy, dxz, dyz}];
   If[avg >= 8, Sow[i], Continue]
   ]]
todel =  todel[[2]][[1]];
todel = Partition[todel, 1];
DomainArea = Delete[DomainArea, todel];
DomainCentroids = Delete[DomainCentroids, todel];
DomainBox = Delete[DomainBox, todel];
{Length[DomainArea], Length[DomainCentroids], Length[DomainBox]}

(* adding half the distance between centroids as a radius to form \
spheres that touch tangentially*)

toadd = Reap[For[i = 1, i <= Length[DomainBox], i++,
    px = DomainBox[[i, 1]] + DomainBox[[i, 4]]/2;
    py = DomainBox[[i, 2]] + DomainBox[[i, 5]]/2;
    pz = DomainBox[[i, 3]] + DomainBox[[i, 6]]/2;

    d1 = EuclideanDistance[
      DomainCentroids[[i, All]], {px, py, DomainBox[[i, 3]]}];
    d2 = EuclideanDistance[
      DomainCentroids[[i, All]], {px, DomainBox[[i, 2]], pz}];
    d3 = EuclideanDistance[
      DomainCentroids[[i, All]], {DomainBox[[i, 1]], py, pz}];

    avg = Mean[{d1, d2, d3}];
    Sow[avg]
    (*
    Print[ToString[i],"  ",ToString[avg],"  ",ToString[{px,py,
    pz}],"  ",ToString[{d1,d2,d3}],"  ", ToString[DomainCentroids[[i,
    All]]]]
    *)
    ]
   ];
```

```
DomainRadius = toadd[[2]][[1]];
DomainCentroids =
  Table[Flatten[{DomainCentroids[[i]], DomainRadius[[i]]}], {i, 1,
    Length[DomainCentroids]}];

pDomain = {0, 0}; minDist = 0;
Domainalldist = Reap[For[i = 1, i <= Length[DomainCentroids] + 1, i++,
    Sow[{pDomain, minDist}];
    minDist = 10000;
    For[j = 1, j <= Length[DomainCentroids] + 1, j++,
     If[i != j,
      dist =
       EuclideanDistance[DomainCentroids[[i, 1 ;; 3]],
        DomainCentroids[[j, 1 ;; 3]]];
      If[dist <= minDist, minDist = dist; pDomain = {i, j}, Continue],
      Continue
      ]
     ]
    ]
   ];

Domainalldist = Domainalldist[[2]][[1]];
Domainalldist = Drop[Domainalldist, 1];
TableForm[Domainalldist];

Domainnewdist = Reap[For[i = 1, i <= Length[Domainalldist], i++,
    Sow[Domainalldist[[i, 2]]/2]
    ]];
Domainnewdist = Drop[Domainnewdist, 1];
Domainnewdist = Flatten[Domainnewdist];
Length[Domainnewdist];
```

Diffusion Analysis

Mathematica script that takes concentrations and fits the data to an gaussian equation.

```
P1 = {55, 209};
P2 = {396, 209};
dist = EuclideanDistance[P1, P2] Voxel;
NMR = 3 Milli Meter;
reso = NMR/dist;
resolution = Convert[reso, Centi Meter/Voxel] // N
res = resolution Voxel/(Centi Meter)
model = a Exp[-(r/b)^2] + c ;

t = Convert[2 Hour, Second]/Second;
maxR = Round[Max[RC50C[[All, 1]]]];
xtick = Range[0, maxR, maxR/7];
ntick = SetPrecision[xtick*res*1000, 3];
ticks = Table[{xtick[[i]], ntick[[i]]}, {i, 1, Length[ntick]}];
model = a Exp[-(r^2/(Sqrt[4*d*t]))] + c ;

answer = NonlinearRegress[RC50C,
   model, {{a, 30}, {d, 0.0004}, {c, 1}}, r,
   RegressionReport -> {BestFitParameters, ParameterCITable},
   MaxIterations -> 1000];
regress50C = answer[[2, 2, 1]]\[Transpose];
fit50C = BestFitParameters /.answer;
G1 = Plot[Evaluate[model /. fit50C], {r, 0, maxR},
   PlotStyle -> {Yellow}];
p1 = ListPlot[RC50C, PlotRange -> {{0, maxR}, {0, 50}},
   PlotStyle -> {Black}];
```

```
Show[p1, G1,
 LabelStyle ->
  Directive[FontFamily -> "Helvetica", FontSize -> 12, Bold],
 Frame -> {{True, False}, {True, False}},
 FrameLabel -> {"radius/cm (\[Times]\!\(\*SuperscriptBox[\"10\", \
\"3\"]\))", "[Hexabromobenzene]/vol%"},
 FrameTicks -> {ticks, Automatic}]
Export["/Users/heath/Dissertation/diss/figures/flame4/50C_RDF_flame4.\
pdf", %]


maxR = Round[Max[RC53C[[All, 1]]]];
xtick = Range[0, maxR, maxR/7];
ntick = SetPrecision[xtick*res*1000, 3];
ticks = Table[{xtick[[i]], ntick[[i]]}, {i, 1, Length[ntick]}];

answer = NonlinearRegress[RC53C,
   model, {{a, 30}, {d, 0.0004}, {c, 1}}, r,
   RegressionReport -> {BestFitParameters, ParameterCITable},
   MaxIterations -> 1000];
regress53C = answer[[2, 2, 1]]\[Transpose];
fit53C = BestFitParameters /.answer;
G2 = Plot[Evaluate[model /. fit53C], {r, 0, maxR},
   PlotStyle -> {Blue}];
p2 = ListPlot[RC53C, PlotRange -> {{0, maxR}, {0, 50}},
   PlotStyle -> {Black}];
Show[p2, G2,
 LabelStyle ->
  Directive[FontFamily -> "Helvetica", FontSize -> 12, Bold],
 Frame -> {{True, False}, {True, False}},
 FrameLabel -> {"radius/cm (\[Times]\!\(\*SuperscriptBox[\"10\", \
\"3\"]\))", "[Hexabromobenzene]/vol%"},
 FrameTicks -> {ticks, Automatic}]
Export["/Users/heath/Dissertation/diss/figures/flame4/53C_RDF_flame4.\
pdf", %]


maxR = Round[Max[RC56C[[All, 1]]]];
xtick = Range[0, maxR, maxR/7];
ntick = SetPrecision[xtick*res*1000, 3];
ticks = Table[{xtick[[i]], ntick[[i]]}, {i, 1, Length[ntick]}];

answer = NonlinearRegress[RC56C,
   model, {{a, 30}, {d, 0.0004}, {c, 1}}, r,
   RegressionReport -> {BestFitParameters, ParameterCITable},
   MaxIterations -> 1000];
regress56C = answer[[2, 2, 1]]\[Transpose];
fit56C = BestFitParameters /.answer;
G3 = Plot[Evaluate[model /. fit56C], {r, 0, maxR},
   PlotStyle -> {Green}];
p3 = ListPlot[RC56C, PlotRange -> {{0, maxR}, {0, 50}},
   PlotStyle -> {Black}];
Show[p3, G3,
 LabelStyle ->
  Directive[FontFamily -> "Helvetica", FontSize -> 12, Bold],
 Frame -> {{True, False}, {True, False}},
 FrameLabel -> {"radius/cm (\[Times]\!\(\*SuperscriptBox[\"10\", \
\"3\"]\))", "[Hexabromobenzene]/vol%"},
 FrameTicks -> {ticks, Automatic}]
Export["/Users/heath/Dissertation/diss/figures/flame4/56C_RDF_flame4.\
pdf", %]


maxR = Round[Max[RC59C[[All, 1]]]];
xtick = Range[0, maxR, maxR/7];
ntick = SetPrecision[xtick*res*1000, 3];
ticks = Table[{xtick[[i]], ntick[[i]]}, {i, 1, Length[ntick]}];

```

```
answer = NonlinearRegress[RC59C,
   model, {{a, 30}, {d, 0.0004}, {c, 1}}, r,
   RegressionReport -> {BestFitParameters, ParameterCITable},
   MaxIterations -> 1000];
regress59C = answer[[2, 2, 1]]\[Transpose];
fit59C = BestFitParameters /.answer;
G4 = Plot[Evaluate[model /. fit59C], {r, 0, maxR},
   PlotStyle -> {Red}];
p4 = ListPlot[RC59C, PlotRange -> {{0, maxR}, {0, 50}},
   PlotStyle -> {Black}];
Show[p4, G4,
 LabelStyle ->
  Directive[FontFamily -> "Helvetica", FontSize -> 12, Bold],
 Frame -> {{True, False}, {True, False}},
 FrameLabel -> {"radius/cm (\[Times]\!\(\*SuperscriptBox[\"10\", \
\"3\"]\))", "[Hexabromobenzene]/vol%"},
 FrameTicks -> {ticks, Automatic}]
Export["/Users/heath/Dissertation/diss/figures/flame4/59C_RDF_flame4.\
pdf", %]
```

Appendix C Chapter 3 Supplemental Information

Chapter 3 Supplemental Information

Sample Composition

The sample composition, Table S6.0, is based first on the polymer formulation, but then update with elemental analysis for Br and Sb to get revised wt% composition. Then, the pure component densities, Table S6.0 are used to generate vol%.

Table S0: Sample Compositions in wt% and vol%

components	density (g/cm^3)	formulated wt%	analysis	revised wt%	vol%
nylon	1.15	42.62	–	46.46	63.84
flame retardant	2.22	20.71	(10.73 wt% Br)	15.26	10.86
Sb_2O_3	5.67	6.12	(4.19 wt% Sb)	5.10	1.42
teflon (6C)	2.2	0.400	–	0.44	0.313
SiO_2	2.197	30.05	–	32.76	23.56

Table S0: Sources of Component Densities

components	density (g/cm^3)	source
nylon (70G43L)	1.150	`http://webbook.nist.gov/`
SiO_2	2.197	`http://webbook.nist.gov/`
FR	2.22	`http://www.albemarle.com/Products_and_services`
Sb_2O_3	5.67	`http://webbook.nist.gov/`
teflon (6C)	2.200	`http://webbook.nist.gov/`

Correction Factors for Tomography Volumes

The following Matlab code snippet is pulled directly from the actual algorithm that applies the correction factors to the X-ray absorbance data. There are three factors to be applied a gain, offset, and voxel expansion factor which are listed in Table S6.0. The order in which the corrections are applied are non-trivial due to the two dimensional interpolation at the last step. The code that is omitted from deal with read/write routines for data preservation, no changes other than the ones presented here occurred.

```
scale_factor = 1e-6; \% to convert from integer 16 to float data type
for k = 1:cube_length(3)
    sliceXY_raw = nc{'VOLUME'}( NZ_vector(k) , : , : );
    sliceXY = imresize(sliceXY_raw, voxel_size_corr, 'bilinear');
    rect = [1, 1, (NX-1), (NX-1) ];
    sliceXY = imcrop(sliceXY, rect);
    sliceXY_corr =  sliceXY * scale_factor * slope_corr + offset_corr;
    ...
```

Table S0: Correction Factors for the Reconstructed Tomography Volumes

energy/keV	$scale_E$	$offset_E$	$voxel\ size_E$
12	1.0571	0.0003930	1.0271
13.4	1.0584	0.0002970	1.0275
17	1.0658	0.0003909	1.0272
25	1.0589	0.0001417	1.0271
30.43	1.0624	0.0000838	1.0270
30.53	1.0611	0.0001307	1.0270
40	1.0647	0.0000618	1.0269

Coordinates of Subvolume

Table S6.0 lists the origin coordinates used to extract the 200x200x500 subvolume from the full 650x650x515 volume.

Table S0: Subvolume Origin Coordinates (within original data volume, base index-0)

i_0	j_0	k_0
225	220	0

Coordinates of Selected Glass Fiber Bundles

Within the binary fiberglass mask, 10 fiber bundles were selected. Line segments were drawn through these fiber bundles, where the line segments are defined by the start and end

coordinates listed in Table S6.0

Table S0: Fiber Coordinates (within subvolume, base index-1)

Fiber #	Start (i,j,k)	End (i,j,k)
1	73, 79, 465	69, 79, 499
2	66, 159, 464	71, 160, 499
3	24, 164, 395	23, 162, 434
4	105, 51, 143	107, 56, 67
5	109, 149, 82	108,154, 129
6	86, 35, 231	82, 30, 196
7	70, 21, 199	71, 19, 233
8	168, 16, 160	163, 2, 212
9	60, 75, 306	58, 72, 282
10	124, 45, 429	125, 43, 451

Radial Concentration Parameters

The SiO_2, **FR**, and Sb_2O_3 radial concentrations are fitted to a Gaussian, $[X]_r = ae^{-(r/b)^2} + c$.

Table S0: Radial Concentration Parameters

$[X]_r$	a/vol%	b/μm	c/vol%
SiO_2	32.6	19.5	20.2
FR	-1.70	18.9	9.67
Sb_2O_3	-0.209	14.5	1.58

Radial Concentration: Voxel Identification and Distance to Fiber Axis

The strategy for determining the radial concentration distribution of flame retardant about a glass fiber assumes moderately-isolated, linear fibers. Some glass fibers are well isolated and these are chosen as representative of the fiber/polymer interface. A fiber axis is defined with start and end xyz coordinates chosen manually; application of automatic cylinder-identification algorithm to the binary fiberglass mask was unsuccessful. Shown in Figure S6.0 is a sketch of the analysis sequence from fiber identification to voxel identification for the radial concentration analysis.

The following mathematical definitions were obtained from `http://www.mathworld.com` and were used in developing the collision detection algorithm which in turn in used extensively generating the radial distribution graphs.

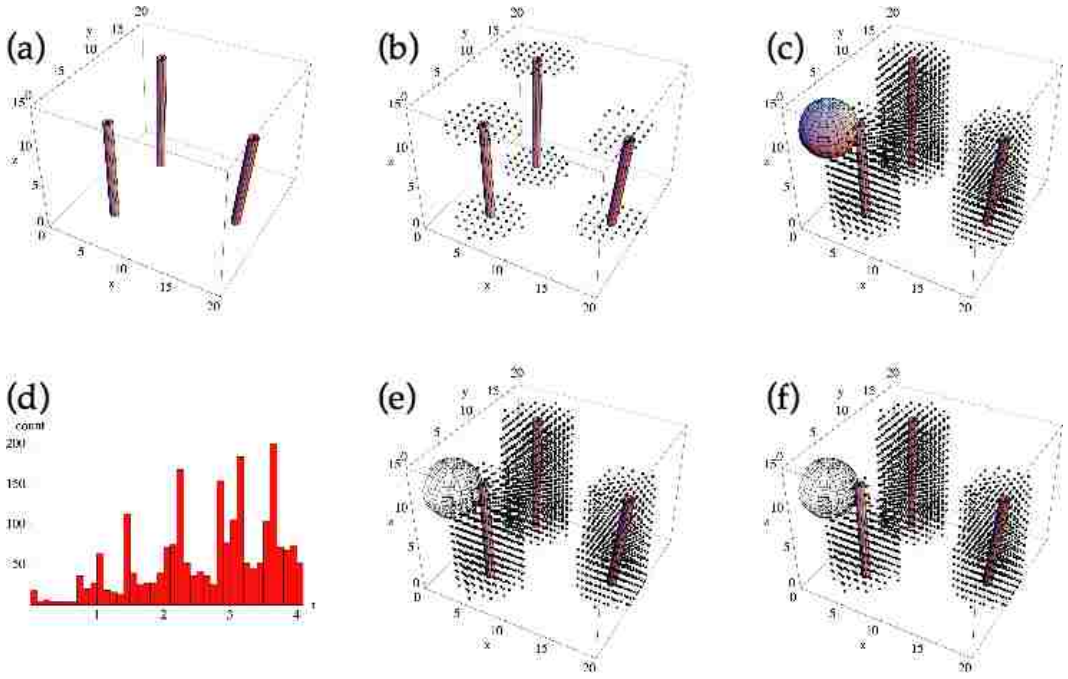


Figure S0: Simulation of fiber analysis in a 20x20x15 volume: (a) initial fibers; (b) voxels identified at fiber ends, lying within the search radius from the fiber axis; (c) a cylinder of voxels about each fiber axis, now showing collision with a sphere; (d) the count of voxels versus distance from the fiber axes, showing the expected r^2 increase in voxel count; (e) elimination of voxels shared with the sphere; (f) use of a "safe radius" and "stacked disks" to remove other voxels from the close vicinity of the sphere, so as to prevent analysis of regions between neighboring glass fibers.

vector norm: `http://mathworld.wolfram.com/VectorNorm.html` Let a line be specified by two points $\mathbf{x_1} = (x_1, y_1, z_1)$ and $\mathbf{x_2} = (x_2, y_2, z_2)$, then the L^2 norm is

$$|\vec{x}| = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \tag{6.1}$$

normal vector to a plane: `http://mathworld.wolfram.com/NormalVector.html` The equation of a plane with a normal vector $\mathbf{n} = (a, b, c)$ passing through the point (x_0, y_0, z_0) is given by

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} \begin{bmatrix} x - x_0 \\ y - y_0 \\ z - z_0 \end{bmatrix} = a(x - x_0) + b(y - y_0) + c(z - z_0) = 0 \tag{6.2}$$

distance from point to a line: `http://mathworld.wolfram.com/Point-LineDistance3-Dimens.html` Let a line be specified by two points $\mathbf{x_1} = (x_1, y_1, z_1)$ and $\mathbf{x_2} = (x_2, y_2, z_2)$ and the point off the line as $\mathbf{x_0} = (x_0, y_0, z_0)$. Then, the distance from the point to the line is

$$d = \frac{|(\mathbf{x_2} - \mathbf{x_1}) \times (\mathbf{x_1} - \mathbf{x_0})]^2}{|\mathbf{x_2} - \mathbf{x_1}|^2} \tag{6.3}$$

The leftmost fiber in Figure S6.0a is parallel to the z-axis; the fiber axis is defined with points $\mathbf{x_1} = (5, 5, 0)$ and $\mathbf{x_2} = (5, 5, 12)$, hence the L^2 norm is $|\vec{x}| = (0, 0, 1)$. Let us assume a search radius $r_{search} = 4$. Then, the next step is to find voxels surrounding $\mathbf{x_1}$ that lie within the search radius. In a nested loop over $|\Delta x|$ and $|\Delta y| \leq r_{search}$ centered about $\mathbf{x_1}$ and in the plane $z = 0$, we accept all voxels xyz with distances less than r_{search} from the fiber axis. In this work, Euclidean distances are used.

Among the 49 voxels found at $d \leq r_{search}$ are, for example, $(1, 5, 0)$ at $d = 4$ and $(2, 3, 0)$ at $d = 3.6055$. All 49 voxels are plotted as the lower disk on the leftmost fiber in Figure S6.0c. A similar disk of 49 voxels caps the other end of the fiber, centered at $\mathbf{x_2} = (5, 5, 12)$. The minimum and maximum xyz values in these two disks define a volume of possible voxels, many of which have $d \leq r_{search}$. In a nested loop, voxels within this volume are tested and 637 voxels are found as shown about the leftmost fiber in Figure S6.0c. The other fibers are processed similarly, and a histogram of the number of identified voxels versus radial distance shows the expected r^2 increase in count, as shown in Figure S6.0d.

We next turn to the issue of collision detection, which has four objectives. First, as a radial concentration plot is generated, we wish to exclude voxels that are too close to adjacent fibers. Second, it is important to eliminate from calculation a line of voxels that might penetrate between two adjacent fibers. Third, in this dense array of glass fibers, we need to retain as many voxels as possible for radial concentration plots of [**FR**], [Sb_2O_3], and [SiO_2] along a radial direction perpendicular to the fiber axis. Therefore, the concepts of "stacked disks", each having a maximum "safe radius", as shown in Figure S6.0, address the first two issues. Fourth, we require a definition of a "collision" that can be adjusted so as to reduce sensitivity to speckle noise that would otherwise lead to false collisions. Figure S6.0e shows collision of some voxels with a neighboring shape, and the resulting "safe radius". Figure S6.0f shows the "safe radius" has been applied to all voxels in the uppermost of the "stacked disks". The thickness of each stacked disk is arbitrary; here, we have chosen a thickness of 10% of the fiber axis length.

There are a number of possible definitions of a "collision". The simplest possible definition is a union of two sets, as shown in Figure S6.0E in which the eliminated voxels were contained in both the cylindrical set of voxels about the fiber and in the sphere representing a collision. A more sophisticated collision definition addresses the fourth issue mentioned above, the problem of speckle noise and false collisions. Each voxel to be tested for collision exists in the center of a $3 \times 3 \times 3$ cube yielding 26-neighboring voxels sharing faces, edges, or vertices with the central voxel. Speckle noise can cause one or more of the 27-voxels be members of both the experimental binary fiberglass mask and of the cylindrical set of voxels surrounding a fiber. On the basis of trial-and-error with our data, a true collision is defined as threshold value of one-half or more voxels shared with the mask. Special care is needed with the threshold value for voxels near the boundaries of the $200 \times 200 \times 500$ subvolume and this is done with a threshold scaled to the number of valid voxels within the potential 26-neighbors.

Vita

Heath Barnett was born October 4, 1980, in Urania, Louisiana. He enrolled in Louisiana Tech University and earned a Bachelor of Science degree in chemistry in May of 2003. He entered the graduate program at Louisiana State University in August of 2003. His research there focused on chemical analysis of polymer blends via synchrotron X-ray tomography. In addition, he taught introductory general chemistry, mentored LSU undergraduate biology majors in the use of Amira. He traveled to the Advanced Photon Source at Argonne National Laboratory to perform tomography experiments. His work also included modifying and improving the LSU synchrotron beamline at the Center for Advanced Microstructures and Devices. Graduate work at Louisiana State University proved to be a rewarding and career-defining experience due to his intensive participation in interdisciplinary tomography research projects which, included fields of chemistry, mathematics, physics, and biology. Presently, he is working at Avoyelles in the renewable fuels group helping design computer interfaces to reactor designs.