

MSc Toni Uusimäki

# Electron Tomography of Porous Materials and Magnetic Nanoparticles

DOCTORAL THESIS

For obtaining the academic degree of  
Doktor der Naturwissenschaften



Graz University of Technology  
Graz University of Technology

Supervisor:  
Ao.Univ.-Prof. Dipl.-Ing. Dr.techn. Gerald Kothleitner  
Institute for Electron Microscopy and Nanoanalysis  
Centre for Electron Microscopy  
(FELMI-ZFE)

Graz, October 2015



Deutsche Fassung:  
Beschluss der Curricula-Kommission für Bachelor-, Master- und Diplomstudien vom 10.11.2008  
Genehmigung des Senates am 1.12.2008

## EIDESSTÄTTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz, am .....  
.....  
(Unterschrift)

Englische Fassung:

## STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

.....  
date  
.....  
(signature)



## Abstract

Electron tomography, as carried out in a transmission electron microscope is a method to reveal the three dimensional structure of the sample at the nanometer scale. It is based on tilting the sample and recording subsequent images at different projections angles. Using specific reconstruction algorithms the density distribution of the sample can then be reproduced. In this thesis, electron tomography has been implemented for material science specimens and more rigorously to porous media infiltrated with magnetic nanoparticles. The volume and spatial distribution along with the knowledge of the demagnetizing factors were then used within a magnetic Monte Carlo simulation to predict the magnetic response of the nanoparticle assembly. The local curvature of nanoparticles within the template, known to be a critical geometrical parameter influencing material properties, was extracted with two distinctive methods. Furthermore, new capabilities needed for image analysis and processing of the tilt series had to be implemented for improved alignments and segmentation. A new method to align the tilt series without depending on markers was written for obtaining high quality reconstructions. Also a comparison was made between different scanning TEM acquisition modes such as incoherent bright field and high angle annular dark field imaging modes with respect to resolution and contrast changes.

## Abstrakt

Elektronentomographie ist eine Methode, die es erlaubt, die dreidimensionale Struktur einer Probe mit Nanometer auflösung abzubilden. Bei dieser Methode wird die Probe in einem Transmissionselektronenmikroskop über einen weiten Bereich gedreht, was die Abbildung der Probe in allen Drehwinkeln erlaubt. Die Dichteverteilung der Probe kann durch die Anwendung von speziellen Rekonstruktionsmechanismen berechnet werden. In der vorliegenden Arbeit wurde Elektronentomographie verwendet, um ein poröses Medium, welches mit magnetischen Nanopartikeln infiltriert wurde zu untersuchen. Auf Basis der Volums- und Winkelverteilung sowie der Demagnetisierungsfaktoren wurde in einem Monte Carlo Simulationsprogramm eine Vorhersage der magnetischen Antwort der Nanopartikel berechnet. Weiters wurde eine neue Methode, die keine Marker zur Rekonstruktion der aufgenommenen Bilder benötigt etabliert, was eine qualitative hochwertige Rekonstruktion ermöglicht. Die lokale Krümmung der Nanopartikel innerhalb ihrer Umgebung wurde mit zwei unterschiedlichen Methoden durchgeführt. Zwei Bildgebungsmethoden (incoherent bright field und high angle annular dark field) wurden in Bezug auf ihre Auflösung und Kontrastumkehr anhand einer Halbleiterprobe charakterisiert.

Though I was careful never to mention it,  
I began to see a new dimension in everything  
that happened.

Hunter S. Thompson

## Acknowledgements

In the early 2009 my supervisor, Professor Gerald Kothleitner was enjoying his holidays skiing in a mountain when he had a distressed phone call:

“Good morning professor Kothleitner! I am on my way to Graz with a big truck packed with my furniture. Could you please talk to this gentleman here, who only speaks German and I don’t quite understand what he is saying. He is a policeman and I think he wants to throw me in jail...”

The four years I spent in Graz working with this thesis went pretty much the same way: barely coping, seemingly understanding and hardly progressing. Fortunately the red line through my thesis found its’ shape eventually. There are many people to whom I should be grateful for...

For laying down the basic grounding and offering the opportunity to accomplish this work I am embedded to Professors Gerald Kothleitner, Ferdinand Hofer, Werner Grogger and the whole institute of FELMI-ZFE along with its’ people of excellent expertise within their field. Not to mention one of the best infrastructure of electron microscopy around the world. For sanity and Zen-like state of mind I am forever grateful to my family, especially my children Eva, Aida and Elle. May the knowledge of German language they acquired in Graz help them through their lives. The FP7 project CopPeR and Austrian Cooperative Research (ACR) are appreciated for financing this work.

The “Getting Things Going” award falls to Dr. Matthew Weyland to whom I owe many thanks for guidance and introduction to the wonderful world of electron tomography and scripting. For the latter also, I can’t thank enough Dr. Bernhard Schaffer for his patience and time. Much appreciated as well is the work and help from Dr. Meltem Sezen for being a friend and preparing my samples with the FIB as well as Martina Dienstleder. For helpful discussions and guidance I am grateful to Dr. Christian Gspan and Dr. Harald Plank within the CopPeR project; and Dr. Petra Granitzer, Dr. Klemens Rumpf, Dr. Kalliopi Trohidou and Professor Heinz Krenn around the topic of porous silicon and magnetic nanoparticles. Many kind words to Manuel Paller for the lab work and to Ulrike Stürzenbecher for her help with the Austrian bureaucracy.

The memory of the universe will certainly remember the many moods and situations that took place in our office, a.k.a. the Party room! Dr. Wernfried Haas, Dr. Stefanie Fladischer and Dipl.-Ing. Arno Meingast created and took part in this spectacle, which I will always remember. And if I don’t, the universe will.

Along the way many others have helped to accomplish this work. Dr. Robin Harmon, Dipl.-Ing. Bernd Kraus and Dr. Paul Thomas from Gatan Inc. helped solving many problems that I encountered. Dr. Mauro Porcu, Walter Mascherbauer and Klaus Dubovy from FEI are acknowledged for the knowledge they passed on concerning the electron microscopes. The practice behind energy filtered imaging was kindly demonstrated to me by Dr. Thomas Haber and Dr. Franz Schmidt. One must also not forget the power of forums as it comes to problem solving, few to be

named are the Matlab forum, DMSUG, 3dem, 3dmod, Avizo and Amira forums as well as some specific C++ forums. I am grateful for these nameless and faceless people who spend their time helping others!

For reviewing this thesis I would like to give my kind regards to Dr. Matthew Weyland, Dr. Bernhard Schaffer, Dr. Peter Ercius, Dr. Petra Granitzer and Dr. Kalliopi Trohidou.

# Contents

Abstract	v
Acknowledgements	viii
1 Introduction	1
2 Electron Microscopy	4
2.1 Build-up of a TEM	4
2.2 Electron-Matter Interactions	7
2.3 Imaging Modes	10
2.3.1 Conventional TEM	10
2.3.2 Scanning TEM	11
2.3.2.1 High Angle Annular Dark Field	11
2.3.2.2 Incoherent Bright Field	12
2.4 STEM Calibrations	13
2.5 Scanning Electron Microscope & Focused Ion Beam	15
3 Magnetism and Materials	17
3.1 Magnetism in Reduced Dimensions	17
3.1.1 Energy of Magnetic Nanoparticles	19
3.1.2 Monte Carlo Simulation of Magnetic Nanoparticle Assemblies	22
3.2 Porous Silicon Infiltrated with Magnetite Nanoparticles	24
3.3 Copper Interconnects	26
3.4 TEM Sample Preparation	29
4 Electron Tomography	34
4.1 Three Dimensional Metrology	34
4.1.1 X-Ray Tomography	35
4.1.2 FIB/SEM/SIMS Serial Sectioning	36
4.1.3 Atom Probe Tomography	36
4.2 Electron Tomography Calibrations	37
4.3 Alignment Methods	39
4.4 Reconstruction Methods	41
4.4.1 Weighted Back Projection	43
4.4.2 Simultaneous Iterative Reconstruction Technique	45
4.4.3 Other Reconstruction Methods	48

4.5	Resolution . . . . .	48
4.5.1	Fourier Shell Correlation . . . . .	49
4.5.2	Spectral Signal to Noise Ratio . . . . .	49
4.6	Segmentation and Visualization . . . . .	50
5	Results	52
5.1	Magnification Calibration Using Auto-Correlation . . . . .	52
5.2	New Alignment Methods . . . . .	53
5.2.1	Manual Projection Based Alignment . . . . .	53
5.2.2	Projection Re-projection Comparison Alignment . . . . .	54
5.3	ITK Segmentation . . . . .	57
5.4	Resolution Assessment . . . . .	59
5.5	Local Curvature Distribution . . . . .	64
5.6	Monte Carlo Simulations . . . . .	72
5.6.1	Demagnetizing Factors . . . . .	72
5.6.2	Comparison and Discussion . . . . .	76
6	Summary and Outlook	81
7	Appendices	84
7.1	EFTEM Tomography . . . . .	84
7.2	Digital Micrograph Scripts . . . . .	88
7.3	Insight Toolkit Implementation . . . . .	95
7.4	Amira Modules . . . . .	107
7.5	Matlab Code . . . . .	119
8	References	129
	List of Abbreviations . . . . .	141
	List of Symbols . . . . .	142
	List of Figures . . . . .	143



# 1 Introduction

“The determination of the effects of interparticle interactions in an actual particle assembly, usually characterized by three degrees of disorder, i.e. topological disorder, volume distribution and random distribution of easy axes, is an extremely complex task. Therefore, modeling necessarily implies some approximations. “

(J. L. Dormann, et al., 1999)

Simulating the magnetic response of a nanoparticle assembly commonly requires assumptions to be made in order to acquire the three Dormann's unknowns into the Hamiltonian equation of the model [1]. Not forgetting the actual spatial distribution of the particles, that strongly influences the short and long range magnetic interactions between the particles. These four quantitative distributions have been hitherto unavailable to magnetic simulations. With the application of electron tomography (ET) as used with a transmission electron microscope (TEM), it is possible to directly extract the three dimensional morphology of the sample at the nanometer scale. Volume and spatial distributions as well as topology of the nanoparticles can be quantitatively characterized, giving access to the first two of Dormann's unknowns. This information can subsequently be used for magnetic Monte Carlo (MC) simulations to yield the magnetic hysteresis and zero field cooling / field cooling vs. temperature curves. Additionally, the detailed knowledge of the topology of the particles allows the extraction of the demagnetizing factor distribution within the assembly, such that the demagnetization energy can be explicitly implemented into the Hamiltonian equation to be minimized. As a proof of concept, a porous silicon (pSi) sample infiltrated with magnetite nanoparticles was characterized, and the simulations were compared to the measured magnetic curves.

A series of images, tilted with respect to the electron beam in the optical axis of the microscope, acts as a basis to obtain a high quality reconstruction provided that the images are aligned properly beforehand. A tilt series acquired from TEM samples lacking fiducial markers can be increasingly difficult to align. Hence spatial registration necessitates novel sophisticated image filtering and segmentation functionality, which needed to be implemented. Here, a new kind of semi-automatic iterative alignment method is proposed, which compares the original tilt series to a re-projected version from an initial reconstruction. By using the open source library of image manipulation algorithms known as Insight Toolkit (ITK), this alignment method proved to be very effective when fiducial markers cannot be deposited onto the sample. In addition this toolkit provides multiple helpful functions to image filtering, registration, segmentation and deconvolution techniques.

As ET enables direct access to the topology and spatial distribution of the nanoparticles, it also classifies the tortuosity, size and interconnectivity of the mesoporous silicon as well as the particle-template adsorption sites. The local curvature distribution of the particles' adsorption sites is of high interest when optimizing the design of any adsorbed or partly embedded particles on or within a template e.g. in catalysts. The curvature of the template can be straightforwardly calculated using a triangulated surface acquired from the segmentation of the 3D tomogram. However, the local curvature around the particle can vary significantly depending on how the curvature is extracted. Here two different methods to obtain the local curvature are proposed and discussed. Additionally 3D imaging grants access to the activity and accessibility of the nanoparticles to the reactant species within the nanocomposite system.

Integrated circuit industry is pushing the miniaturization of devices to ever-decreasing scales. TEM proves to be crucial for the characterization of the sample at the nanoscale. Yet the third dimension in conventional imaging is lost because of the projection characteristics of the imaging method and the overlapping of densely patterned features may produce inaccurate interpretations. ET coupled with high angle annular dark field (HAADF) mode provides anew interesting type of metrology to extract the full three dimensional structure of the sample. Here ET was applied to copper interconnects to characterize the sample. However, HAADF signals become ambiguous for very dense and thick samples. A possible solution is to use incoherent bright field (IBF) imaging, where contrast remains monotonic. Here, a comparison of the ET results between IBF and HAADF modes, in terms of contrast and resolution, will be given.

In chapter 2 a basic introduction to electron microscopy (EM) is given in terms of imaging and electron interactions with matter. Also, microscope related necessary calibrations are discussed together with available imaging modes. More focus is put to scanning TEM (STEM) mode since it was mainly used within this thesis. Chapter 3 explains the sample preparation processes, materials used in this work and the magnetic behavior of nanoparticles. Magnetism in reduced dimensions will be introduced, yet restricting to only the most important aspects, which are crucial to understand the results. ET is explored in chapter 4 with a short history. After the tomography related calibration protocols, the four basic steps for an ET investigation are discussed: acquisition, alignment, reconstruction and segmentation.

The experimental results are given in chapter 5 focusing on the local curvature, demagnetizing factors and the MC simulations. Other results, such as the automatic magnification calibration method for TEM is given in chapter 5.1. Two different methods to align the tilt series and the segmentation implementations used within this thesis are explored in chapter 5.2. Segmentation routines using ITK are given in chapter 5.3. Two pSi samples, prepared in room temperature and cryo conditions, are compared in chapter 5.4 in terms of resolution and submerging ratio. A copper interconnect sample imaged using HAADF and IBF mode, are additionally compared, focusing on contrast reversal and resolution.

Conclusions and ideas about future work are discussed in chapter 6. Relevant software code is shown in the appendices chapter 7 to reproduce the work in this thesis and to help others with similar problems. Additionally proving feasibility, a qualitative 3D investigation of a flash memory device is given, by using energy filtering TEM with a short introduction.

The tilt series alignment method, ITK implementation, local curvature, demagnetizing factors and magnetic simulations were published in a peer-reviewed journal [2]. Within this publication, Dr. P. Granitzer and Dr. K. Rumpf fabricated the sample and made the magnetic measurements; Dr. M. Sezen prepared the sample TEM lamella using FIB; Dr. G. Margaris and Dr. K. Trohidou performed the magnetic simulations and their analysis. The author of this thesis did the TEM tomography acquisition, wrote the new software to align the tilt series as well as the ITK implementation to segmentate the reconstruction. The author also wrote the software for the two distinct methods to acquire the local curvature and did its analysis, as well as the ellipsoidal fitting that led to the demagnetizing factor distribution.

## 2 Electron Microscopy

TEM is a superior technique when it comes to resolution. For light microscopy, the resolution limit is in the order of the used wavelength of light, whereas for 200 kV accelerated electrons, the wavelength is as small as 0.00251 nm. The wavelength depends on the acceleration voltage used and is the theoretical resolution limit of the TEM, but the magnetic lens aberrations will deteriorate this. Since the imaging consists of transmitted electrons through the specimen, it follows that sample preparation is an important aspect in TEM. To ensure transparency the samples have to be as thin as possible commonly ranging from 5 nm to 300 nm - depending on the acceleration voltage, the sample and the level of resolution needed. [3]

The images acquired with a TEM are always projection images (in imaging mode) such that the electron intensity/signal within the image is integrated through depth, giving two dimensional information of a three dimensional object. Different imaging modes and contrast mechanisms exist explaining the intensity variations in the image. In addition to the conventional TEM (cTEM), where the whole field of view is illuminated with a parallel beam, one can also use the microscope in scanning transmission mode (STEM), where the electron beam is focused onto the sample and rastered to form an image. Because of ET's advantages in STEM mode for materials science, as discussed in chapter 2.3 - within this thesis HAADF and IBF are explored in more detail. In addition to imaging, TEM can also be used in diffraction mode, where a diffraction pattern can be recorded to acquire information about the crystal structure of the material.

### 2.1 Build-up of a TEM

A TEM can be divided into three distinctive parts consisting of an illumination, an imaging and a projection system. The whole column must be under high vacuum to minimize the electron scattering. With the TF20 (figure 2.1) this is accomplished with a turbo pump, backed with a scroll pump and an ion getter pump, which introduces fewer vibrations when acquiring images.

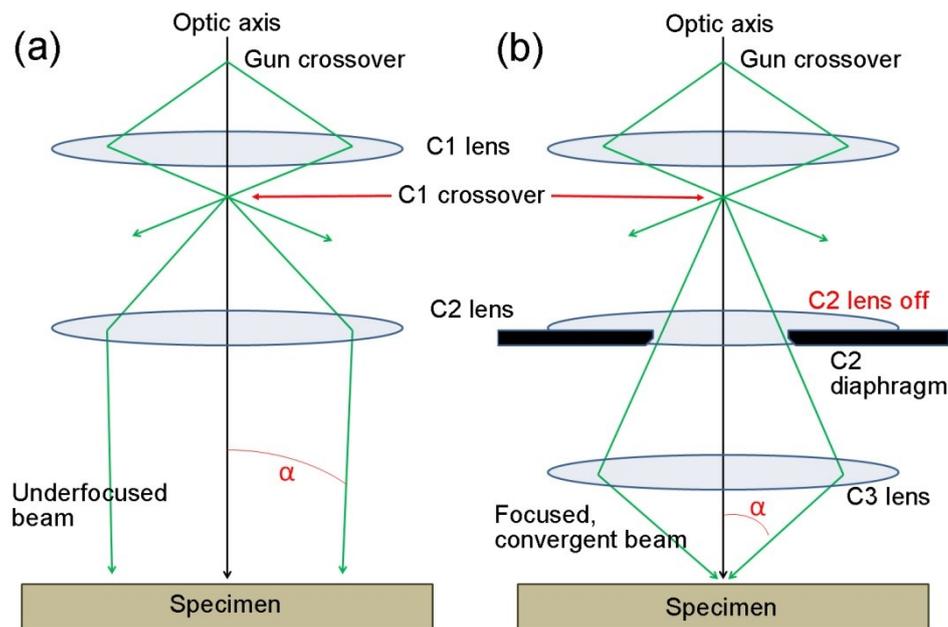


**Figure 2.1:** Tecnai TF20 located at Felmi-ZFE, TU Graz. The TF20 is equipped with a Schottky FEG, Wien type monochromator, EDAX sapphire Si(Li) detector for energy dispersive X-ray measurements, Gatan GIF Quantum with Dual EELS for energy loss measurements, Gatan Ultrascan CCD and Fischione HAADF detector model 3000.

The heart of the illumination system is the electron gun, traditionally made either using a tungsten needle or LaB<sub>6</sub> crystal. The sourcing of electrons from these materials is based on thermionic emission, where a barrier known as the work function  $\Phi$  of the material is surpassed by heating such that electrons can escape from the surface. However most modern microscopes use a field emission gun (FEG) made from tungsten material and has a shape of a sharp needle. The tip is coated with zirconium oxide to lower the work function. Schottky FEGs can be operated at relatively low vacuum at elevated temperature, and cold FEGs by ambient temperatures but only in ultra-high vacuum. The latter provides best performance in terms of brightness, spatial and temporal coherence, but is not stable over time and needs occasional flashing to clear the tip from contamination. The intensity and brightness of the electron source can be varied by the positive extraction voltage anode situated below the tip creating a strong electric field that attracts the electrons and allows tunneling from the tip. These electrons are further accelerated by an acceleration voltage usually in the range of 80-300 kV. High voltages give better theoretical resolution and higher transmission characteristics but also introduce knock-on damage. The current trend is to achieve atomic resolution with low acceleration voltages for beam sensitive materials [4]. Radiolysis and electrostatic damage can also occur.

As a part of the illumination system, the condenser lenses are used to define the shape of the beam as it interacts with the sample, in a parallel or convergent way. The path of the electrons is controlled using magnetic lenses, deflectors and apertures. The magnetic field exerts a Lorentz force to the electrons and the apertures are used to limit the intensity and convergence angles of the beam onto the sample plane. Using different combinations of lens strengths at the two condenser and upper objective lenses, the angle of illumination can be changed from parallel to convergent mode.

In the TF20, which was mainly used in this thesis, the position of the beam crossover between the C1 and C2 lens (figure 2.2(a)) defines what portion of the beam will pass the condenser aperture (also gun lens voltage has an influence). As the crossover moves upwards, fewer electrons pass the aperture resulting in long acquisition times yet with a beam of better spatial coherence and smaller diameter. In parallel mode the C1 lens first forms an image of the gun crossover and the C2 lens under focuses the beam onto the sample. Alternatively the upper objective lens can be further used to create a parallel beam. In convergent mode, the C2 lens is only weakly excited and the objective lens (C3) focuses the beam onto the plane of the sample (figure 2.2(b)). The C1 crossover and condenser aperture also influences the convergence angle  $\alpha$  of the beam. [3]



**Figure 2.2:** (a) Parallel illumination for cTEM mode. (b) Focused illumination for STEM mode.

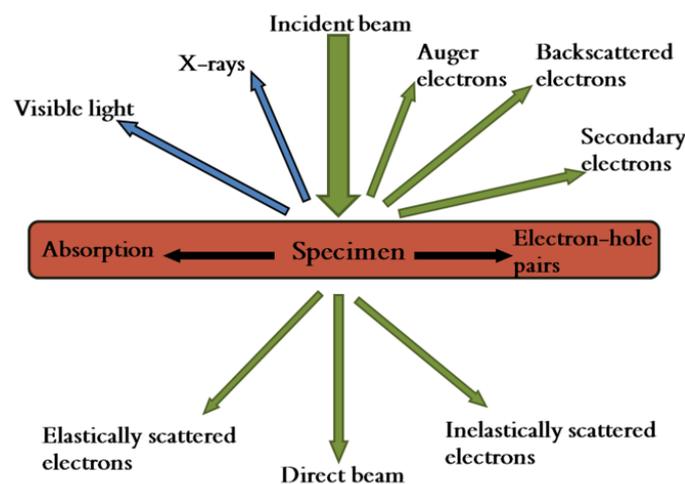
The sample holder is in the imaging system, situated between the two objective lens pole pieces and placed in the goniometer, which allows precise control of the sample displacement in x,y and z coordinates, where z is the height parallel to the optical axis of the microscope. The goniometer also allows tilting of the sample holder (max.  $\pm 80^\circ$ ). To avoid aberrations, the pole piece is as narrow as possible, which geometrically limits the tilt-range of the sample holder. The objective aperture is positioned in the back focal plane of the objective lens and in parallel mode, is used to limit the collection angle  $\beta$  of the transmitted electrons through the sample, reducing aberrations and improving the resolution and contrast. It can also be used to choose only certain excitations in the diffraction pattern and image the sample by the diffracted beam. This is called dark field (DF) imaging mode as opposed to bright field (BF) mode.

After the sample plane there are still the lower objective, diffraction, intermediate and two projection lenses that constitutes to the magnification of the image in parallel mode and to the choice of camera length in a diffracting convergent mode. By changing the strength of these lenses, one can switch between diffraction and imaging mode. In the former, the back focal plane of the objective lens acts as an object plane for the intermediate lens, and the image plane is the object in the latter. STEM is operated in diffraction mode, but since the beam is convergent, instead of

diffraction spots, a convergent beam electron diffraction (CBED) pattern is formed constituting as disks with a size and distance depending on the convergence angle and camera length respectively (and also on the sample). Scattered electrons can be collected by an annular dark field (ADF) detector or a HAADF detector in STEM mode. The camera length defines the angular range detected by the HAADF detector. Images and diffraction patterns (also CBED patterns) can be projected to the viewing screen or to a charge-coupled device (CCD) camera in TEM mode.

## 2.2 Electron-Matter Interactions

High-energy electrons can remove tightly bound inner-shell electrons from the atom. Analytical electron microscopy takes advantage of this by measuring the element characteristic signals like X-rays for energy dispersive spectroscopy (EDS) [5] [6] or electron energy losses in electron energy loss spectroscopy (EELS) [7] [8]. Some of the measurable signals are shown in figure 2.3. The screened Coulomb field of the nucleus can change the direction of the incoming electron without (elastic scattering) or with energy-loss (inelastic scattering). The term scattering implicitly embraces the idea of an electron as a particle. However, for diffraction the wave-nature of the electron should be considered. With a phase relation between the entering and transmitted electron waves, the signal is said to be coherent. Diffraction contrast typically arises from coherent elastically scattered electrons and incoherent elastically scattered electrons are used in HAADF mode, whereas incoherent inelastic signal is used in EELS.



**Figure 2.3:** Different signals generated by the high energy incident electron beam with the thin electron transparent specimen.

Electron waves diffracted from adjacent crystal lattices can have a path difference equal to the wavelength (multiplied by an integer number) of the electrons known as the Bragg law:

$$n\lambda_0 = 2d \sin\theta_B, \quad (1)$$

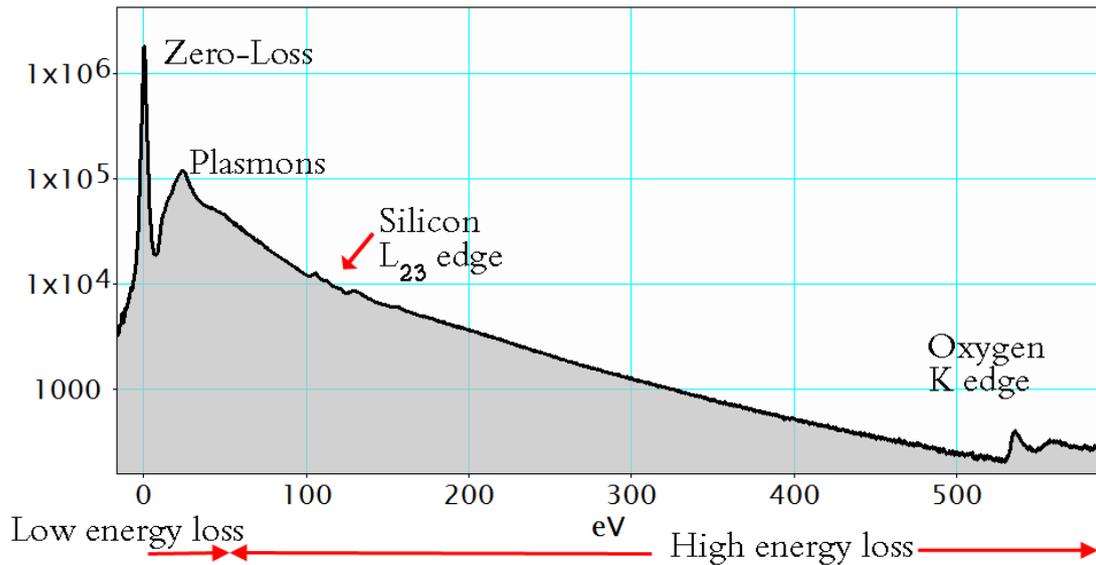
where  $n$  is the order integer,  $\lambda_0$  is the wavelength of the electrons,  $d$  is the crystal lattice distance and  $\theta_B$  is the Bragg angle. For ET in BF mode contrast differences depending on crystal orientations are problematic violating the projection criteria (See chapter 4). This interaction can be considered as electron-electron interaction and as such is rather weak giving low scattering angles.

Electron-nucleus interactions can result in large scattering angles and are considered incoherent. The Rutherford cross-section for electrons scattered elastically into angles  $> \theta$  is related as

$$\sigma \propto \left(\frac{Z}{E_0}\right)^2 \cot^2 \frac{\theta}{2}, \quad (2)$$

where  $Z$  is the atomic number,  $E_0$  is the energy of the incident electrons (in keV) and  $\theta$  is the scattering angle [3]. Cross-sections are used to calculate the probability that a scattering event will occur. Rutherford scattered electrons are interacting closely with the nucleus of the atoms having a small impact factor and hence are ideal for high resolution imaging. It is of interest also for ET since it is less affected by the diffraction contrast better fulfilling the projection criteria.

Inelastic interactions within the sample can be used to acquire information about the chemical composition. The incoming electron ejects an inner shell atomic electron ionizing the atom. This is visible as ionization edges (K,L,M,N...), whose energetic positions are characteristic to each element allowing its identification. An ionization event is followed by secondary transitions e.g. ejection of an auger electron or an emission of X-ray, and the ratio of the probability between the two is called fluorescence yield. Because EELS is not affected by the fluorescence yield limitation it is particularly useful as an elemental analytical method for the light elements, which are difficult to analyze with EDS. As a guideline, sample thicknesses for EELS should be well below 1 as measured in the units of  $t/\lambda$ , where  $t$  is the thickness and  $\lambda$  the inelastic free mean path of scattering. This is to avoid multiple scattering, accounting for significant spectral background and for altered edge intensities. Special thickness deconvolution schemes and analysis procedures can be applied to account for multiple scattering events and are paramount for thick samples [8]. The ratio  $t/\lambda$  is also known as scattering parameter, and can be conveniently measured by evaluating the zero-loss peak (ZLP) intensity and the total spectrum intensity. [9]



**Figure 2.4:** Typical EEL spectrum in logarithmic scale showing the ZLP, low-loss plasmons and oxygen ionization edge. The sample was  $\text{SiO}_2$ .

By using an energy dispersing element for example in the form of a post-column energy filter [10] [11] it is possible to energetically separate inelastically scattered electrons and detect them with a CCD camera. This dispersion produces an electron energy loss (EEL) spectrum, where the number of electrons detected is plotted against their energy loss. A typical EEL spectrum can be seen in figure 2.4 in logarithmic scale taken from a  $\text{SiO}_2$  sample and one can see the decrease in the inelastic inner shell cross-section of the oxygen K edge energy loss. The most prominent feature is the elastically scattered and/or quasi-elastically scattered electrons, comprising the ZLP at zero energy. The full width at half maximum (FWHM) of this peak is commonly used as a measure for the energy resolution. Collective oscillations of electrons, known as plasmons contribute to the low energy-loss region ( $< 50$  eV) and yield valence information of the sample [7]. The low loss spectrum depends on the optical properties of the sample and as such can be used to obtain the dielectric function of the material [12]. At high energy losses inner shell ionizations of elements occur, allowing analytical and quantitative elemental composition measurements. The shape of the ionization edge depends on the excited shell and density of states before and after the ionization event. A careful analysis of the energy loss near edge structure (ELNES) gives information about the electronic structure of the material. [8]

## 2.3 Imaging Modes

### 2.3.1 Conventional TEM

In cTEM the whole field of view on the sample is illuminated with a parallel beam as seen in figure 2.2(a). Using undiffracted electrons is known as bright field (BF) imaging. In BF mode the contrast is determined by the mass/thickness of the sample; with dense or thick samples, more electrons are scattered off axis or absorbed resulting in less intensity on the CCD camera. Crystalline areas can appear dark also if the lattice planes fulfill equation (1), as the electrons are intercepted by the objective aperture; this is known as diffraction contrast. At higher magnifications in thin and crystalline samples, phase contrast can dominate and the direct interpretation of images is hampered. This is because the contrast transfer function reverses repeatedly with frequency. [3]

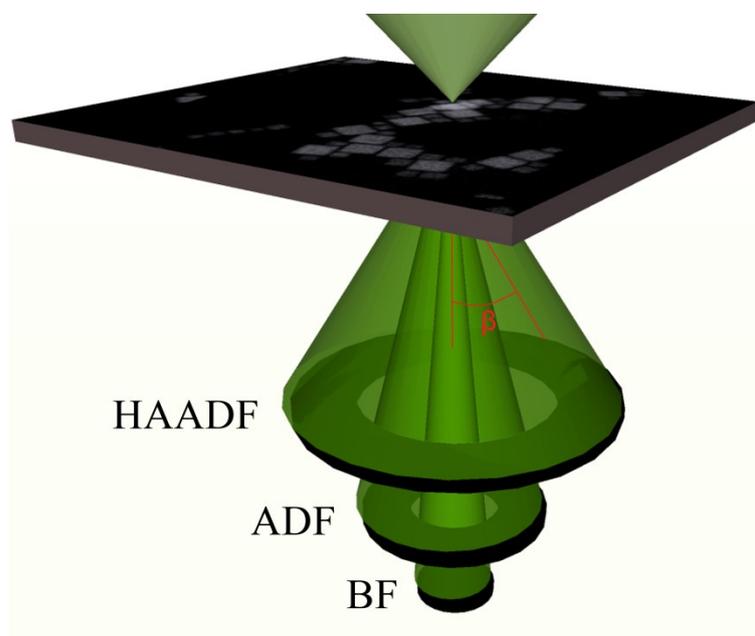
The main aberrations coming from the lenses are the chromatic and spherical aberrations. The former relates to the fact that electrons with different energies are focused on a different plane. The latter is because electrons further away from the optical axis experience a slightly different magnetic field and are again focused to a different plane. The according properties of a lens can be expressed by its chromatic aberration constant ( $C_C$ ) and its spherical aberration constant ( $C_S$ ). The effect of  $C_C$  can be minimized by energy filtering (by using only the ZLP).  $C_S$  correction needs specific hardware, which is available in new  $C_S$  aberration corrected TEM's (e.g. FEI Titan). For tomography in the materials science field with crystalline samples cTEM is problematic because of the Bragg reflections that violate the projection requirement. Biological samples are usually amorphous and with good sample preparation, cTEM tomography provides an excellent tool to characterize cells in all three dimensions at the nanometer scale.

### 2.3.2 Scanning TEM

Scanning transmission electron microscopy (STEM) is a technique where the electron beam is focused onto the sample (see figure 2.2(b)) and rastered across while the scattered intensity is measured for every point/pixel within the image. The spot size with modern  $C_S$  corrected microscopes can be well below one tenth of a nanometer [12]. In STEM mode the second condenser lens is weakly excited and the upper objective lens focuses the beam onto the sample. Beam deflectors are used to keep the beam tilt constant while rastering. The magnification in STEM mode depends only on the size of the scanned area and the size of the image displayed and the image is not affected by the imaging magnetic lens aberrations (unlike the probe). Depending on the angle of scattered electrons that is recorded, different contrast mechanisms and image modes are available i.e. BF, ABF, ADF or HAADF. Annular dark field (ADF) can be used to measure contrast from low angle scattered electrons, e.g. from diffracted crystalline areas. Annular bright field (ABF), where the central part of the BF detector is obscured, is useful for high resolution simultaneous imaging of light and heavy elements. [3] [13]

### 2.3.2.1 High Angle Annular Dark Field

For HAADF imaging, the detector is a disk with an inner annulus, situated below the sample. High angle forward scattered electrons are detected undergoing Rutherford scattering and thermal diffuse scattering [12]. The main advantage of HAADF imaging is the absence of coherence and the exclusion of coherently scattered Bragg electrons with low scattering angles contributing to diffraction contrast as illustrated in figure 2.5. Strong Bragg reflections at small angles are excluded and the weak high angle ones are averaged over. As seen in equation (2), HAADF contrast is related to mass-thickness, i.e.  $Z^2$  (or as found experimentally to  $Z^{1.5-1.7}$ ) and sample thickness, and hence giving ideal contrast for different chemistry of the sample [14]. The measured signal is related to average  $Z$  of the material, such that different chemical phases with same average  $Z$  does not produce contrast differences. The scattering angles hitting the detector can be controlled by post specimen lenses that define different camera lengths to be used. The contrast is directly interpretable, and the focus settings are unambiguous such that there is no need for through focus series as in cTEM.



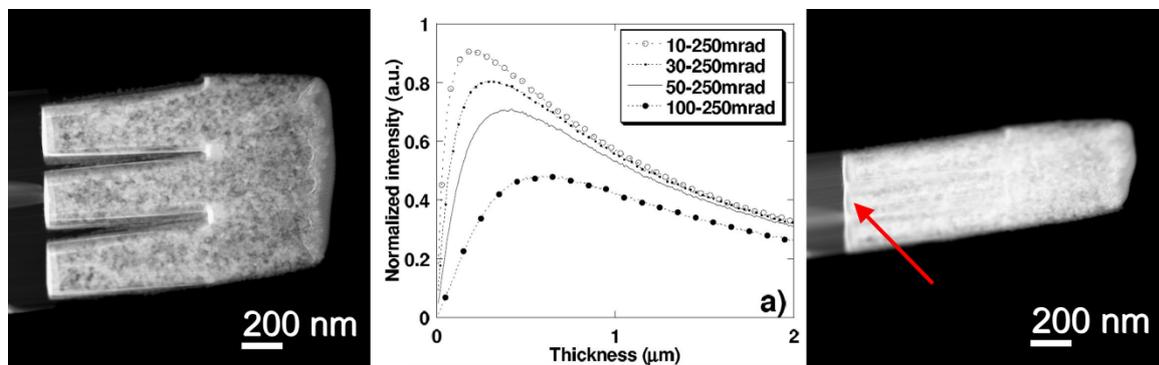
**Figure 2.5:** Schematics of the detectors used in STEM mode. The bright field (BF) detector collects the unscattered and low angle scattered ( $\beta < 10$  mrad) electrons. The ADF detector collects the medium angle scattered ( $10 \text{ mrad} < \beta < 50$  mrad) electrons and the HAADF ( $50 \text{ mrad} < \beta < 200$  mrad) collects the high angle scattered electrons. The values here are approximate and also depend on the convergence angle  $\alpha$  and the wavelength of the electrons.

HAADF imaging for tomography has many benefits: Image interpretation is more straightforward, since the intensity/signal is unambiguous, and approximately proportional to  $Z^2$ . Diffraction contrast arising from crystalline material under Bragg conditions at specific tilt angles is minimized, thus better fulfilling the projection requirement in tomography. In TEM mode, the gradual defocus differences between the minimum and maximum tilt angles causes in-plane rotation and a magnification gradient within the tilt series (Lorenz force). In HAADF-STEM mode

the post-sample magnetic fields are merely used to determine the camera length. The image alignment is hence easier since in-plane rotation and scaling changes does not occur. The possibility to use dynamic focusing, where the beam defocus is changed with every line scan of the image at the respective tilt angle, helps keeping the whole image in focus even at high tilt angles [15]. Finally the heat generated by the electron probe is dissipated quicker than in cTEM, resulting in unaltered samples needed for the long acquisition times in ET [16]. Knock-on damage can thereby be effectively avoided.

### 2.3.2.2 Incoherent Bright Field

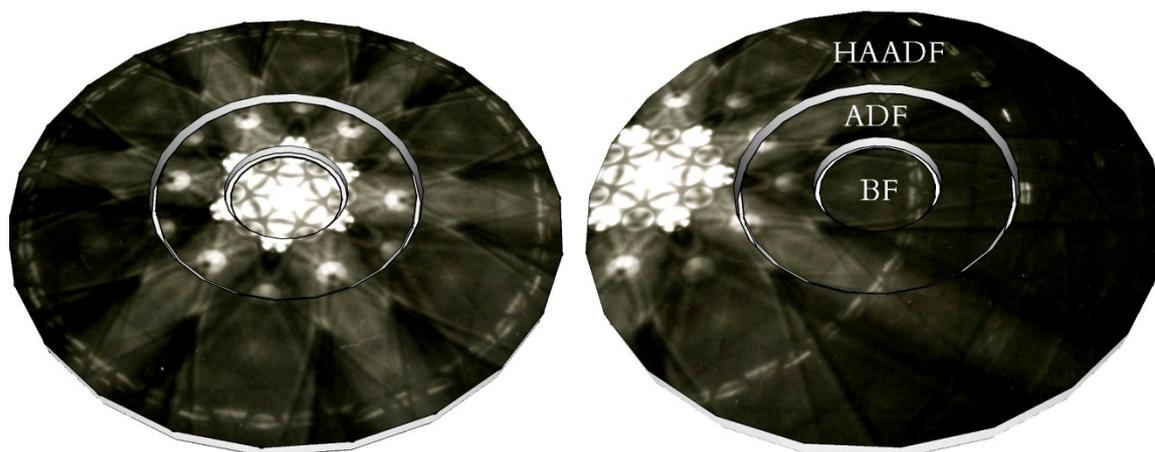
Although HAADF mode has multiple advances over cTEM tomography especially in material science, the HAADF signal is not a monotonic function when the sample becomes vastly too thick. After a certain material dependent cutoff thickness the intensity begins to decrease due to multiple scattering events within the sample, as illustrated in figure 2.6. Contrast reversal can cause misleading interpretations since the decreased intensity on thick areas appears as false voids in the reconstructions [17]. It should be remembered, that the projection thickness in a slab shaped sample at  $70^\circ$  tilt angle is about three times thicker as compared to  $0^\circ$  degrees. It would be beneficial to collect all high angle scattered and backscattered electrons but this is not applicable because of geometrical and practical restrictions in the small gap between the objective lens poles.



**Figure 2.6:** On the left a HAADF STEM image of a copper interconnect sample is imaged at -50 degrees tilt angle showing no contrast reversal on the vertical tantalum lines where as on the right imaged at -78 degrees vertical black lines are visible. In the middle simulated curves for 200 kV electrons traveling through a Cu sample are shown with different inner detection angles. The peak in the curves gives the cut-off thickness. The middle image was taken from [18].

Incoherent bright field (IBF) [17] [19] provides a solution for the contrast reversal problem with thick samples. In IBF, instead of collecting the high angle scattered electrons, the directly forward scattered beam is directed onto the HAADF detector with the diffraction shift coils as illustrated in figure 2.7. Although the geometry of the HAADF detector is not ideal for this process, it is more sensitive to the broad intensity range needed for the IBF signal. The camera length should be such

that all scattered electrons between 0 - 100 mrad are collected to suppress coherent diffraction contrast associated with low collection angles. The IBF signal can be understood as a complement of the ADF signal and the intensity is also scalable as  $Z^2$ . The signal becomes incoherent when the collection semi angle is about three times larger than the probe convergence semi angle, which can be varied by the choice of camera length and the size of the condenser aperture used respectively. [18]



**Figure 2.7:** The BF, ADF and HAADF detectors illuminated with a CBED pattern in STEM diffraction mode. The left image illustrates the normal HAADF mode, where the detector picks up high angle scattered signals. The right image depicts the arrangement in IBF mode, where the beam is translated such that the direct beam hits the HAADF detector.

## 2.4 STEM Calibrations

STEM mode calibrations are briefly introduced and explained here. Calibrations related to tomography are found in chapter 4.2.

### **Eucentric height:**

Calibration of the eucentric height practically means minimizing the lateral movement of the sample while tilting the stage, being crucial for a tomography tilt series acquisition. It can be found by wobbling the goniometer between  $-15^\circ$  and  $+15^\circ$  degrees and changing the z height such that the sample movement is minimal. A more reliable and reproducible method is offered in the FEI tomography software Xplore 3D [20], that uses cross-correlation (see chapter 4.3) and stage tilting to find the correct eucentric height. The automatic acquisition software Xplore 3D cannot cope with very large sample displacements that can happen when tilting the goniometer.

### **Beam tilt pivot points:**

The beam tilt pivot point alignment assures beam shift and beam tilt purity. Beam shift and tilt are connected to each other and it is important that while shifting the beam during the acquisition of an image, no beam tilting occurs. Beam tilt pivot points have to be aligned at the eucentric focus and can be corrected by changing the tilt values at a constant rate between two limits while adjusting the ratio of the deflection coils such that the two tilted spots coincide -both in x and y directions.

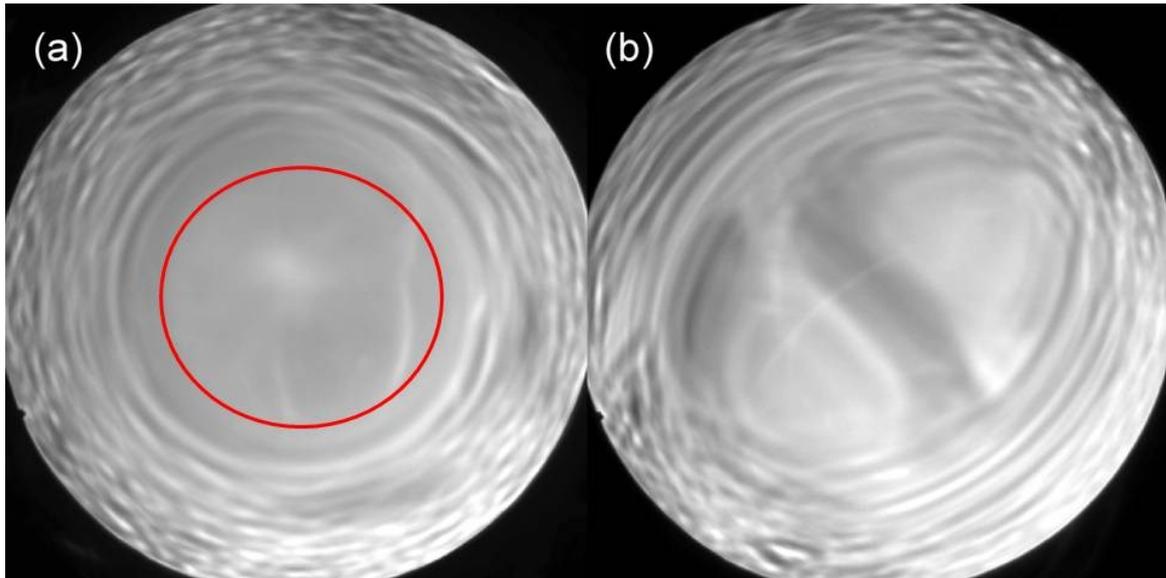
### **Rotation center:**

Rotation centering aligns the optical axis, as defined by the beam tilts and shifts, to the field of the objective lens, which is fixed. This can be accomplished by wobbling the defocus to over and under focus and changing the beam tilt such that the object is moving on-axis. The beam shift can be translated to the middle of the viewing screen via diffraction shift coils in diffraction-off mode.

All these alignments are interconnected and the actual calibration of the beam has to be done iteratively from low to high magnifications at given spot sizes and eucentric heights. The choice of camera length can be optimized such that the inner and outer scattering angles hitting the HAADF detector are ideal for the specific material being investigated. It can be calibrated by using the diffraction pattern of a known sample, e.g. evaporated polycrystalline aluminum.

### **Ronchigram:**

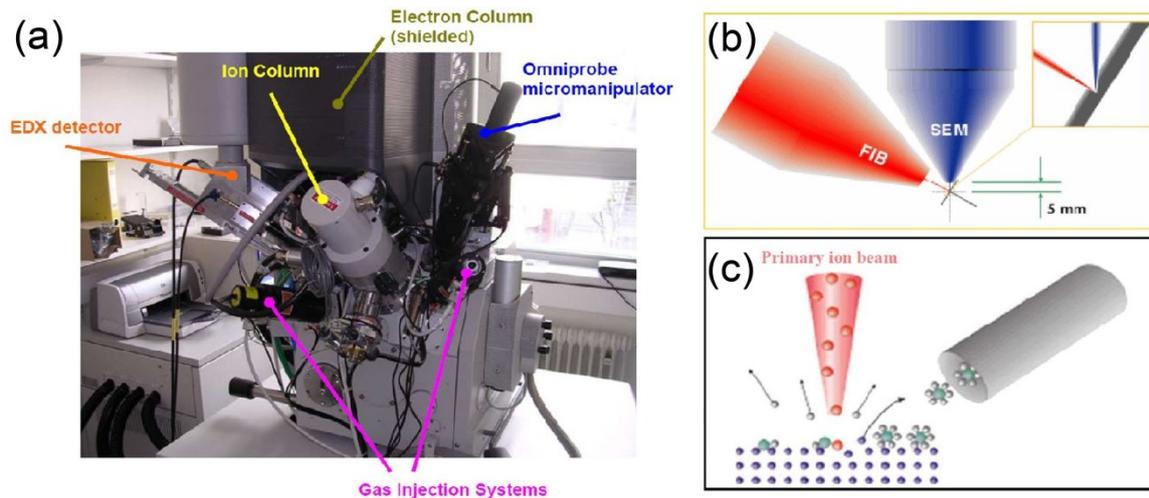
The final corrections to align the probe and to find the optical axis is to use a Ronchigram [21] a.k.a. shadow image (figure 2.8), with a stationary probe in the middle of the image fixed at an amorphous area. A CBED pattern can then be seen at the viewing screen or at the CCD camera. All the misalignments, aberrations and astigmatisms are shown in a Ronchigram simultaneously, being a representation of the whole probe forming optics of the microscope. The objective is to make the central area flat in contrast and as wide as possible. Then within the central region, all electrons have traversed the same point within the specimen, i.e. the beam is perfectly at focus. The radial streaking around the central area is caused by high angle electrons focused above the specimen (spherical aberration) and is called the ring of radial infinite magnification (not seen in figure 2.8), and the angular streaking is called the ring of azimuthal infinite magnification (as seen in figure 2.8). One can use x and y condenser lens stigmators to make the flat area as smooth and centro symmetric as possible. To optimize the probe, one then has to position the C2 aperture in the middle to exclude the azimuthal fringes. A Ronchigram can also be analyzed automatically to correct the aberrations of the microscope online. [12] [21] [22]



**Figure 2.8:** (a) Ronchigram with spherical symmetry and no astigmatism. Once the astigmatism has been corrected, a smaller C2 aperture can be centered in the middle excluding the rings. The red circle depicts the aberration free zone. (b) Ronchigram with astigmatism in the condenser lens.

## 2.5 Scanning Electron Microscope & Focused Ion Beam

In scanning electron microscopy (SEM) [23] [24] a focused electron beam is rastered across the surface of the sample, while the various signals that result from the interactions between the incoming beam and the sample surface are measured. The signal is coming mostly from the surface of the specimen within some interaction volume depending on composition, the acquired signal and the used high voltage of the accelerated electrons. Usually acceleration energies ranging from 50 V to 30 kV are used. Various signals can be measured, but for simple imaging of the surface topology usually secondary electrons (SE) are used. SEs are inelastic low-energy electrons that are knocked out of their shell around an atom by the incoming focused electron beam and provide the highest spatial resolution in SEM (below 1 nm). If contrast related to atomic number  $Z$  is required e.g. to separate different phases, also elastic backscattered electrons (BSE) can be used. SEMs are often used in combination with an EDS detector for chemical mapping and also crystalline orientation of the surface can be quantified with electron backscatter diffraction using the BSEs.



**Figure 2.9:** (a) FEI Nova 200 Nanolab Dual-Beam used for this work. (b) Configuration of the ion and electron beam inside the column. (c) Beam induced deposition of platinum. [25]

A focused ion beam (FIB) instrument [26] [27] is very similar to a SEM. Instead of electrons, ions are used for imaging, milling or deposition. Commonly gallium is used for milling with energies spanning from 5 to 50 keV. High energy ions knock out atoms from the sample on user defined areas and patterns. For deposition, the material is introduced to the chamber through controlled gas flow via a syringe just above the surface. Commonly platinum is used in a precursor organometallic gas (e.g.  $C_7H_{17}Pt$ ) in electron beam induced or ion beam induced chemical vapor deposition (EBID or IBID). Figure 2.9(c) illustrates the IBID of a platinum monolayer. The ion or electron beam causes the material to re-emit secondary electrons that are responsible for the decomposition of the precursor molecules and the platinum is deposited onto the surface.

SEM and FIB can be combined to create a dual-beam instrument for simultaneous imaging and deposition/milling. Figure 2.9(a) shows the dual-beam FIB-SEM instrument used in this work and (b) the configuration of it. The SEM and FIB columns are positioned at a specific angle to each other and the sample is positioned at the intersection of the beams. In addition to milling or sputtering material in site specific areas one can connect objects together with IBID. This is why FIB is a highly versatile instrument for TEM sample preparation as illustrated in chapter 3.4 for the copper interconnects and pSi samples. These samples were first roughly milled to acquire a lamella or pillar shape respectively and then transferred to a TEM grid using a micromanipulator. To reduce ion beam damage and the size of the amorphous layer on the sample surface the final milling should be done using low ion acceleration voltages with lower interaction volumes.

## 3 Magnetism and Materials

To understand the results given in chapter 5, here a basic introduction to the magnetic properties of magnetic nanoparticles and their assemblies is given. Since in magnetic Monte Carlo simulations the objective is to minimize the Hamiltonian equation, here main focus is on the energy terms. The materials explored in this thesis and their sample preparation methods are given in detail for the pSi and copper interconnects specimens. Additional materials are also shortly introduced, which are later used as examples to highlight some features in the following chapters.

### 3.1 Magnetism in Reduced Dimensions

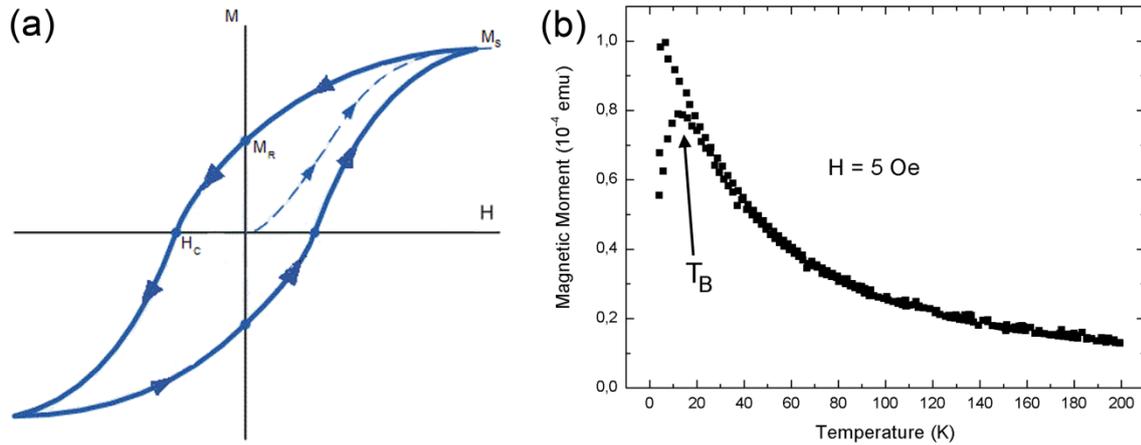
Magnetism of nanoparticles and their assemblies is different to that of the bulk behavior. Quantum confinement in reduced dimensions and high surface to volume ratios of the particles at the nanoscale are the cause [28]. In 1907 Pierre Weiss [29] [30] explained the hysteresis behavior of magnetic materials by assuming that there are magnetic domains within the material, which are separated by domain walls. Magnetic materials try to minimize the stray magnetic fields by fragmentation into domains. The energy stored in the domain walls increases proportionally to the surface area while the magnetostatic energy increases proportionally to the volume. Hence below a critical particle radius  $r_C$ , the introduction of a domain wall is energetically less favorable and the particles become single domain. The critical radius is given by

$$r_C = 9 \frac{\sqrt{AK_\mu}}{\mu_0 M_S^2}, \quad (3)$$

where  $A$  is the exchange constant ( $\text{Fe}_3\text{O}_4$ :  $1.28 \times 10^{-11} \text{ J/m}$ ),  $K_\mu$  is the uniaxial anisotropy constant ( $\text{Fe}_3\text{O}_4$ :  $-1.1 \times 10^4 \text{ J/m}^3$ ),  $\mu_0$  is the permeability constant and  $M_S$  is the saturation magnetization ( $\mu_0 M_S^2$  for  $\text{Fe}_3\text{O}_4$ :  $4 \times 10^5 \text{ J/m}^3$ ). For magnetite particles,  $r_C$  is 8.4 nm at room temperature. [31] [32] [33] [34]

A common way to measure the magnetic response of a material is to measure the hysteresis curve. A typical hysteresis curve is illustrated in figure 3.1(a). When increasing the magnetic field, the material's magnetization goes to saturation as all the internal spins are aligned; also known as saturation magnetization  $M_S$ . The Coercive field  $H_C$  is the external field needed to reduce the magnetization back to zero whereas remanent magnetization  $M_R$  is the magnetization at zero applied field. On bulk magnetic materials the hysteresis is a combined effect of the rotation of magnetization, magnetic anisotropies and magnetic domains. For particles below the critical radius, hysteresis is mainly caused by the coherent rotation of the magnetization within the particles as

assumed by the Stoner-Wohlfarth (SW) model [35]. The hysteresis curve in figure 3.1(a) has a typical shape for randomly oriented SW particles. Magnetic hysteresis curves are commonly measured using a vibrating sample magnetometer (VSM), where the sample is placed within a uniform magnetic field and coils are situated nearby, registering the magnetization of the vibrating sample while changing the strength of the uniform magnetic field ( $T = \text{constant}$ ).



**Figure 3.1:** (a) Typical magnetic hysteresis curve showing the magnetization  $M$  with respect to the magnetic field  $H$ , saturation magnetization  $M_s$ , remanent magnetization  $M_R$  and the coercive field  $H_c$ . (b) ZFC/FC curve of  $\text{Fe}_3\text{O}_4$  nanoparticles of 5 nm in size showing the peak at ZFC curve at  $T_B = 12\text{K}$  using a magnetic field of 5 Oe. (Image courtesy P. Granitzer)

The easy axis of a magnetic particle is the stable orientation of the magnetic moment minimizing the magnetic energy, and has two anti-parallel directions with an anisotropy barrier in between defining the hard axis. The magnetic moment of the particles can flip direction within the easy axis under the influence of temperature especially if  $K_\mu$  is small, such that the anisotropy barrier between the easy axis magnetization and hard axis magnetization is small. The time between two subsequent flips is called the Néel relaxation time and is given by:

$$\tau_N = \tau_0 e^{\frac{K_u V}{k_B T}}, \quad (4)$$

where  $\tau_N$  is the average time between the flips,  $\tau_0$  is the attempt time characteristic to the material ( $\text{Fe}_3\text{O}_4 \sim 10^{-10}$  s),  $V$  is the volume,  $k_B$  is the Boltzmann constant and  $T$  the temperature [30]. If the magnetization measurement time is longer than  $\tau_N$ , then the system appears to be non-magnetized since the fluctuations of the magnetization average out and the system is in a superparamagnetic state. It behaves like a paramagnet with no coercivity but instead of atomic spins, the assembly consists of nanoparticles with giant magnetic moments and as such is called superparamagnetic [36]. For comparable measurement times to  $\tau_N$ , the system comes to a blocked state when the thermal energy is not sufficient to flip the magnetization. This transition is characterized with a transition temperature and is called the blocking temperature  $T_B$ , which can be found using zero field cooling / field cooling (ZFC/FC) measurements.

The ZFC/FC curves are commonly acquired using a superconducting quantum interference device (SQUID) magnetometer, which can easily detect the small magnetization changes within samples [30]. SQUID is based on narrow insulating gaps (Josephson junctions that allow tunneling of electrons) between two superconductors that together form a ring. The magnetic flux penetrating through the ring creates resistance between the two superconductors, which can be measured. The sample is first cooled down without an external magnetic field. With decreasing temperature, the thermal energy becomes smaller and the particle moments are frozen in a random manner. The induced magnetization is measured while increasing the temperature with an applied magnetic field giving the ZFC curve. Then the sample is cooled again with an applied magnetic field and the magnetization is measured with decreasing temperature giving the FC curve. The peak of the ZFC gives  $T_B$  when all the moments are aligned. The point where the two curves collide is called the irreversibility temperature. After the  $T_B$  the thermal energy becomes sufficient to randomly flip the magnetization direction of the particles and the system becomes superparamagnetic [37]. A ZFC/FC curve of  $Fe_3O_4$  nanoparticles with an approximate diameter of 5 nm can be seen in figure 3.1(b), where  $T_B$  is found to be  $\sim 12$  K, which indicates that the particles are not interacting. It should be noted that  $T_B$  depends strongly on the measurement time [38].

### 3.1.1 Energy of Magnetic Nanoparticles

The total energy of a magnetic particle assembly can be described as follows

$$E_T = E_Z + E_k + E_D + E_\lambda + E_{exch} + E_{surf} + E_{shape}, \quad (5)$$

where  $E_Z$  is the Zeeman energy,  $E_k$  is the magnetocrystalline energy,  $E_D$  is the dipole-dipole energy,  $E_\lambda$  is the magnetoelastic energy,  $E_{exch}$  is the exchange energy,  $E_{surf}$  is the surface anisotropy energy and  $E_{shape}$  is the shape anisotropy energy [28]. The Zeeman energy depends on the dipole interactions with the external magnetic field and can be expressed as [39]

$$E_Z = -\mu_0 \vec{M} V \vec{B}. \quad (6)$$

The magnetocrystalline energy depends on the crystal structure of the material.  $Fe_3O_4$  has an inverse spinel crystal structure, where the  $\langle 111 \rangle$  direction is the easy axis of magnetization, which minimizes the energy. Within the SW model of coherent magnetization reversal, with uniformly magnetized particles having uniaxial anisotropy, magnetocrystalline energy can be approximated as

$$E_k = K_\mu V \sin^2 \theta_E - \mu_0 M H \cos(\theta_0 - \theta_E), \quad (7)$$

where  $K_\mu$  is the effective uniaxial anisotropy constant,  $\theta_E$  is the angle between the easy axis and the direction of magnetization and  $\theta_0$  is the angle between the easy axis and the magnetic field as seen in figure 3.2(a). The derivative with respect to  $\theta_E$  gives two energy minima ( $\theta_E = 0$  and  $\pi$ ), where the switching takes place from one minima to another. From this follows that  $K_\mu V$  is the

magnetocrystalline energy barrier that has to be overcome by the thermal energy for the switching of the magnetization [39]. For cubic crystals, the magnetocrystalline anisotropy energy can be written with direction cosines:

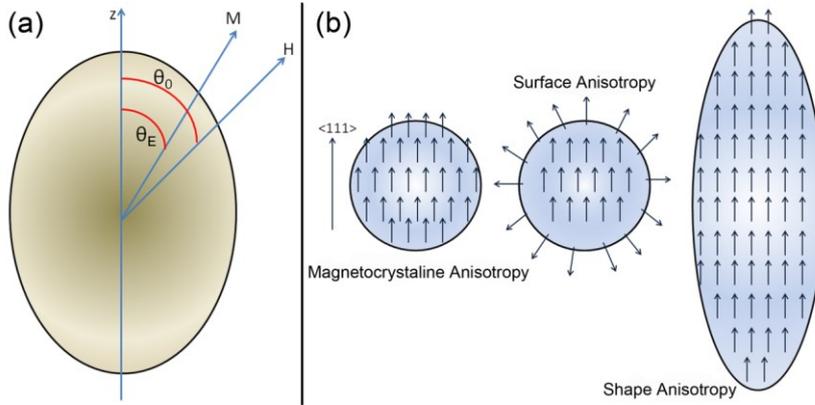
$$E_k = K_\mu V (\alpha_1^2 \alpha_2^2 + \alpha_2^2 \alpha_3^2 + \alpha_3^2 \alpha_1^2). \quad (8)$$

For magnetite in direction  $\langle 100 \rangle$  the direction cosines, for crystal angles a:  $0^\circ$ , b:  $90^\circ$  and c:  $90^\circ$ , are 1, 0 and 0 respectively, so  $E_k = 0$ . For  $\langle 111 \rangle$ , a:  $54.7^\circ$ , b:  $54.7^\circ$  and c:  $54.7^\circ$  so the direction cosines are all  $1/\sqrt{3}$  and as such  $E_k = K_\mu V/3$ . Since  $K_\mu$  is negative for magnetite, it follows that  $\langle 111 \rangle$  is the easy axis of magnetization and  $\langle 100 \rangle$  the hard axis.

The dipole-dipole potential energy comes from the direct interaction of the dipoles of two adjacent particles. It has a long range order compared to exchange interaction and can be expressed as

$$E_D = \frac{\mu_0 M_S^2 V_i V_j}{4\pi a^3} \frac{\hat{S}_i \hat{S}_j - 3(\hat{S}_i \hat{R}_{ij})(\hat{S}_j \hat{R}_{ij})}{R_{ij}^3}, \quad (9)$$

where  $a$  is the minimum interparticle distance,  $R_{ij}$  is the center-to-center distance between the particles and  $\hat{S}_i$  is the spin operator. Hats indicate unit vectors. [36]



**Figure 3.2:** (a) Magnetic nanoparticle with uniaxial anisotropy along the z-axis. The magnetization M is at angle  $\theta_E$  to the easy axis. (b) Magnetization of nanoparticles according to different anisotropies. For spinel-type particles the  $\langle 111 \rangle$  direction is the magnetocrystalline easy axis. The surface spins can be in different orientation than in the bulk: radial, canted or random. The shape anisotropy easy axis is the long axis of an ellipsoidal or nanorod particle, where the demagnetization energy is minimized.

The magnetoelastic energy  $E_\lambda$  may come from induced strain in the particles or from magnetostriction effects, where the shape of the particles is changed as they are exposed to a magnetic field. The exchange energy is usually explained using the Heisenberg model, where the Hamiltonian can be written as

$$E_{exch} = -2 \sum_{i>j} J_{ij} S_i S_j, \quad (10)$$

where  $J_{ij}$  is the exchange integral [39]. For ferromagnetic interaction  $J_{ij} > 0$  and antiferromagnetic  $J_{ij} < 0$ .  $E_{exch}$  and  $E_\lambda$  are not included in the Hamiltonian used within this thesis.

The surface anisotropy energy can be significant as the surface to volume ratio increases. The surface atoms of a nanoparticle are in different crystalline symmetry compared to the core atoms and the broken exchange bonds can lead to spin disorder or spin canting. This has been shown to reduce the saturation magnetization [40]. The energy can be written as

$$E_{surf} = K_{surf} \int \cos^2 \theta_S dS, \quad (11)$$

where  $K_{surf}$  is the surface anisotropy constant,  $\theta_S$  is the angle between the magnetization and the normal to the surface and  $dS$  is the surface element. [28]

The shape anisotropy energy arises from the long range magnetic dipole interactions in the particle, which generates a demagnetizing field within the particle and depends on the shape of the particle. The direction of the shape anisotropy easy axis is determined by the demagnetizing factors (DFs)  $N_a$ ,  $N_b$  and  $N_c$ , where  $N_a + N_b + N_c = 1$ . Single domain spherical particle has no shape anisotropy, which means that  $N_a = N_b = N_c = 1/3$ . The demagnetizing field is minimized along the long axis of an ellipsoidal or a nanorod particle (figure 3.2(b)) and defines the shape anisotropy easy axis for that particle. The energy can be described as [28]

$$E_{shape} = \frac{1}{2} \mu_0 V (N_a M_a^2 + N_b M_b^2 + N_c M_c^2). \quad (12)$$

The equations for the DFs of a general ellipsoid was solved by J. A. Osborn already in 1945 [41] and are defined as

$$N_a = 4\pi \frac{\cos \varphi \cos \vartheta}{\sin^3 \vartheta \sin^2 \alpha} [F(k, \vartheta) - E(k, \vartheta)], \quad (13)$$

$$N_b = 4\pi \frac{\cos \varphi \cos \vartheta}{\sin^3 \vartheta \sin^2 \alpha \cos^2 \alpha} \left[ E(k, \vartheta) - \cos^2 \alpha F(k, \vartheta) - \frac{\sin^2 \alpha \sin \vartheta \cos \vartheta}{\cos \varphi} \right] \quad (14)$$

and

$$N_c = 4\pi \frac{\cos \varphi \cos \vartheta}{\sin^3 \vartheta \cos^2 \alpha} \left[ \frac{\sin \vartheta \cos \varphi}{\cos \vartheta} - E(k, \vartheta) \right], \quad (15)$$

where  $N_a$ ,  $N_b$  and  $N_c$  are the DFs corresponding to the ellipsoid's semi axes a, b and c with  $a \geq b \geq c \geq 0$ .  $F(\cdot)$  and  $E(\cdot)$  are elliptic integrals of first and second kinds respectively. Also  $\cos \vartheta = c/a$ , where  $(0 \leq \vartheta \leq \pi/2)$ ,  
 $\cos \varphi = b/a$ , where  $(0 \leq \varphi \leq \pi/2)$  and  
 $\sin \alpha = \sin \varphi / \sin \vartheta = k$ , where  $(0 \leq \alpha \leq \pi/2)$ .

The DFs forms a tensor in the demagnetization energy term within the Hamiltonian equation to be minimized as introduced in the next chapter. As can be seen the DFs are a function of the ellipsoid's semi radii only and therefore can be solved if the particle's topology is known. This knowledge was acquired using electron tomographic analysis of the pSi sample infiltrated with magnetite nanoparticles and further used for magnetic MC modeling of the sample.

### 3.1.2 Monte Carlo Simulation of Magnetic Nanoparticle Assemblies

The Monte Carlo (MC) simulation technique for magnetic behavior of nanoparticles with the implementation of the Metropolis algorithm [42] used within this thesis has been written by K. Trohidou and M. Vasilakaki [43]. It was chosen here, since the influence of finite temperature can be studied and the Metropolis implementation overcomes local minima problems. The MC technique has proven to be a very powerful tool to study the microstructure of nanoparticle assemblies and to reproduce a qualitative agreement with experimental data. Hitherto the spatial and volume distribution of the particle assembly had to be assumed, but here the quantitative information from ET characterization of the magnetite particles was used for the simulations.

The idea in MC technique is to minimize energy in the chosen model for the Hamiltonian consisting of magnetic self-energies and interaction energies of the assembly by random fluctuations of effective spins of the particles (SW model). If the deviations of the spin direction within every iteration step are random, the system escapes from the metastable state responsible for hysteresis and the system always appears superparamagnetic. Hence the deviations in the spin direction are dependent from the history of the previous direction. If the difference in energy  $\Delta E < 0$ , then the new direction is accepted. If  $\Delta E > 0$ , then it is accepted only if a random number  $u$ , generated uniformly within the interval from 0 to 1, is smaller than the partition function  $\exp(-\Delta E/k_B T)$  of the canonical ensemble. In every MC step, all spins are considered once and the final result is averaged over 10 initial starting conditions for the spin directions. In ZFC/FC simulation the magnetic field is given and magnetization is calculated with respect to temperature while in magnetic hysteresis simulation the temperature is given and magnetization is calculated with respect to the magnetic field. [43]

The following theoretical considerations has been taken from the supplementary information of the author's publication [2] and written by G. Margaris. To model the assembly of nanoparticles we consider  $N_p$  magnetic particles (grains), with ellipsoidal shape. The magnetic particles are single domain and each of them is represented by a three dimensional classical spin [35] with magnetic moment  $\vec{m}_i = m_i \vec{s}_i$ ,  $i = 1, \dots, N_p$  of magnitude  $m_i = M_S V_i$  and direction  $\vec{s}_i$ , with  $|\vec{s}_i| = 1$ , where  $M_S$  is the saturation magnetization per unit volume and  $V_i$  is the particle volume. From the quantitative analysis of the reconstructed pSi volume,  $N_p=1042$  particles were extracted, along with the knowledge of the particle positions  $(x_i, y_i, z_i)$ , the particle dimensions (the lengths of the three semi-axes  $(a_i, b_i, c_i)$  of the ellipsoid), the particle volume and surface area, the 3 unit vectors along the principal axis of the fitted ellipsoid representing the particle  $\hat{a} = (a_x, a_y, a_z)$ ,  $\hat{b} = (b_x, b_y, b_z)$  and  $\hat{c} = (c_x, c_y, c_z)$  where a-axis is the major axis and c-axis is the minor axis. The demagnetizing tensor  $\mathbf{N}_i$  has eigenvalues  $N_a, N_b, N_c$  along the principal axes of the ellipsoid.

A uniaxial easy axis  $\hat{e}$  was assigned to each particle and the easy axes of the nanoparticles in the assembly were randomly distributed. The nanoparticles anisotropy includes contributions from the crystallographic and surface anisotropy. Ellipsoidal nanoparticles have also shape anisotropy,

which usually is included in an effective anisotropy constant. Within this thesis, due to the detailed knowledge of the particle morphology, the shape anisotropy was taken into account in the model explicitly, through the DFs. The demagnetizing field of the  $i$ -th particle is  $H = -M_S \mathbf{N}_i \vec{s}_i$  and the respective self-energy of the nanoparticle is  $E_{a,i} = \frac{1}{2} \mu_0 M_S^2 V_i \vec{s}_i \mathbf{N}_i \vec{s}_i$ . The energy of the assembly is

$$E = -\mu_0 H M_S \sum_{i=1}^N V_i \vec{s}_i \hat{e}_h - K_1 \sum_{i=1}^N V_i (\vec{s}_i \hat{e}_i)^2 - \frac{1}{2} \frac{\mu_0 M_S^2}{4\pi a^3} \sum_{\substack{i,j=1 \\ i \neq j}}^N V_i V_j \vec{s}_i \mathbf{D}_{ij} \vec{s}_j + \frac{1}{2} \mu_0 M_S^2 \sum_{i=1}^N V_i (\vec{s}_i \mathbf{N}_i \vec{s}_i), \quad (16)$$

where  $K_1$  is the anisotropy energy density,  $\hat{e}_h$  and  $\hat{e}_i$  are the directions of the magnetic field and the anisotropy axis of the  $i$ -th particle respectively,  $\mathbf{D}$  is the dipole-dipole interaction tensor  $\mathbf{D}_{ij} = \frac{3\hat{r}_{ij}\hat{r}_{ij} - 1}{r_{ij}^3}$ , where  $i \neq j$ ,  $r_{ij} = |\vec{r}_i - \vec{r}_j|$  and  $\hat{r}_{ij} = \vec{r}_{ij}/r_{ij}$ , are the interparticle distance and the direction vector joining sites  $i$  and  $j$  respectively, measured in units of the characteristic length  $\mathbf{a}$ . When particles are placed on a lattice structure,  $\mathbf{a}$  is usually taken to be the lattice constant. However in this thesis it was taken to be 1 nm, such that  $r_{ij}$  is a dimensionless quantity.

In the case of polydisperse assemblies we define the mean particle volume  $\langle V \rangle = \sum_{i=1}^{N_p} V_i / N_p$  and using the dimensionless magnetic moments  $n_i = V_i / \langle V \rangle$  the magnitude of the magnetic moment is expressed as  $m_i = M_S V_i = (M_S \langle V \rangle) n_i$ . All energy terms are rescaled by dividing with the mean anisotropy  $K_1 \langle V \rangle$  per particle, so in the calculations the reduced (dimensionless) parameters are used

$$h = \mu_0 \frac{M_S}{K_1} H, \quad k = 1, \quad g = \frac{\mu_0 M_S^2 \langle V \rangle}{4\pi K_\mu a^3} \quad \text{and} \quad d = \frac{\mu_0 M_S^2}{2K_1} \quad (17)$$

for the external magnetic field  $h$ , the nanoparticle anisotropies  $k$ , the dipole-dipole interaction strength  $g$  and the demagnetizing energy parameter  $d$  respectively. Here  $g$  have units of energy, so a division with  $K_1 \langle V \rangle$  is a dimensionless quantity. Then the reduced energy is written as

$$\begin{aligned} \varepsilon = \frac{E}{K_1 \langle V \rangle} = & -h \sum_{i=1}^N n_i \vec{s}_i \hat{e}_h \\ & - k \sum_{i=1}^N n_i (\vec{s}_i \hat{e}_i)^2 \\ & - \frac{1}{2} g \sum_{\substack{i,j=1 \\ i \neq j}}^N n_i n_j (\vec{s}_i \mathbf{D}_{ij} \vec{s}_j) + \frac{1}{2} d \sum_{i=1}^N n_i (\vec{s}_i \mathbf{N}_i \vec{s}_i) \end{aligned} \quad (18)$$

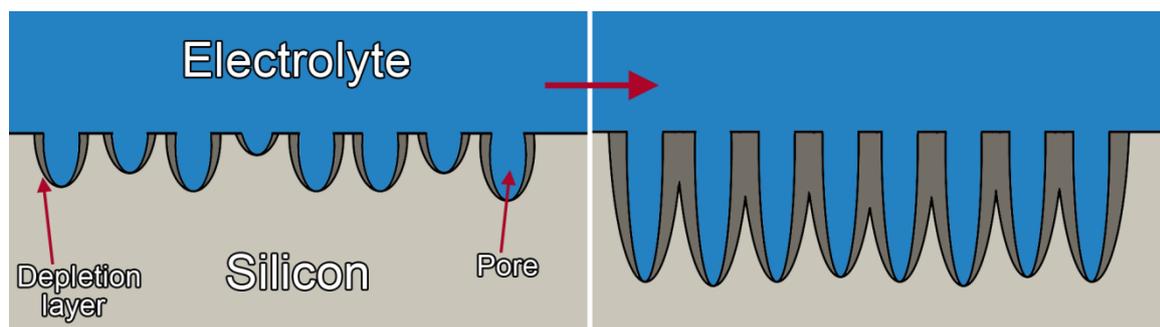
and the thermal energy is  $t_K = k_B / K_\mu \langle V \rangle$  and  $t = t_K T$ . The values of the other parameters are the values of the bulk Fe<sub>3</sub>O<sub>4</sub>, namely, the saturation magnetization  $M_S = 4.8 \times 10^5$  A/m [44] and the anisotropy energy density is  $K_1 = -1.3 \times 10^4$  J/m<sup>3</sup> [45].

## 3.2 Porous Silicon Infiltrated with Magnetite Nanoparticles

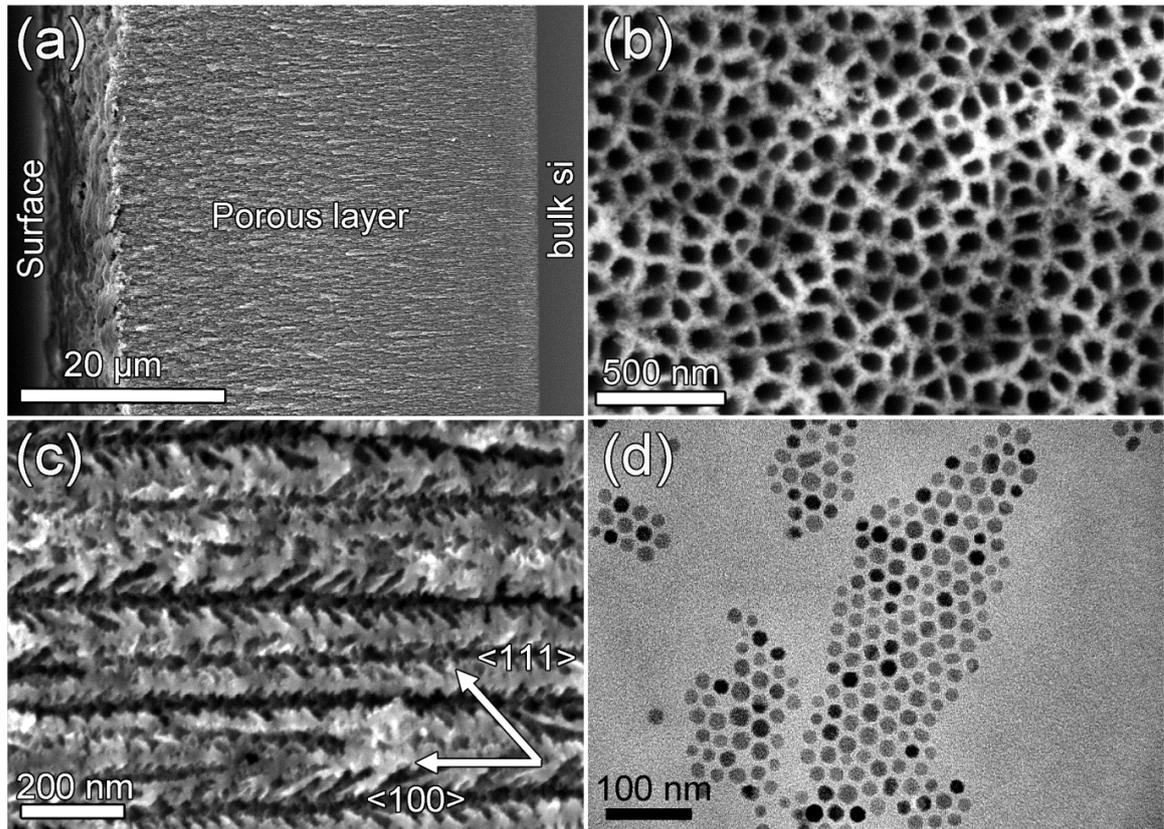
Porous silicon (pSi), can be produced by nanostructuring of a silicon wafer and is employed in various fields as in sensors, photovoltaics and energetic materials [46]. Beside its photo- [47] [48] and electroluminescence [49] in the visible range, also the specific surface chemistry [46] and controllable pore-size without pre-structuring is of high interest. The biodegradability [50] and bioactivity [51] renders this material also useful for biomedical applications. Due to the tunable size and the growth mechanism of the pores, as well as their quasi-regular arrangement [52], porous silicon is also utilizable as a template for the deposition of metals or nanoparticles. Soft magnetic materials, which possess low coercivity and remanence, are intensively studied in biomedicine for biodetection, separation and for a possible treatment for hyperthermia [53]. Much interest, as in this thesis, in the biomedical field has been put to magnetite ( $\text{Fe}_3\text{O}_4$ ) nanoparticles because of their superparamagnetic properties, sufficiently high saturation magnetization and well known preparation methods for monodisperse size distribution. [54] [55]

The pSi templates can be fabricated by anodization of a highly n-doped ( $10^{19} \text{ cm}^{-3}$ ) silicon wafer in aqueous hydrofluoric acid solution (10 wt%) [56]. By controlling the electrochemical parameters such as current density, electrolyte concentration and bath temperature, oriented pores are grown perpendicular to the surface of the (100) silicon wafer. The porosity and pore diameter are directly proportional to the current density but only up to a limiting current, after which electropolishing occurs and no longer pore formation. In addition to the main pores, smaller dendritic side pores (mean length smaller than 20 nm) are grown in the  $\langle 111 \rangle$  direction, forming a highly complex three dimensional structure consisting of dendritic, oriented pores which are separated from each other.

The growth of the pores is self-regulating. At the interface between the silicon surface and the electrolyte, a depletion layer is formed, leading to a paucity of holes needed for the dissolution of silicon. When the depletion layers of adjacent pores begin to overlap, the current flow is pinched off. As a consequence the dissolution advances mainly on the tip of the pore and no collapsing of the pores occurs (as seen in figure 3.3). The dendritic pores are grown more slowly due to the remaining free carriers in the silicon template. [57] [58]



**Figure 3.3:** The formation of pores in the anodization process. The pore growth in vertical direction is decreased due to depletion layers. The dendritic nanopores are not shown.



**Figure 3.4:** (a) SEM overview image of the cross-section of the whole structure. (b) Top view image of the pores. (c) A cross-section image of the pores before the infiltration of the nanoparticles, where the main pores are horizontal and the dendritic growth diagonal. [2] (d) Zero-loss filtered TEM image of magnetite nanoparticles with mean diameter of  $\sim 20$  nm. (Image courtesy: P. Granitzer)

The porous silicon was prepared at Institute of Physics, KFU by Dr. P. Granitzer and Dr. K. Rumpf (FWF project: P21155). For this work, a sample with an average pore diameter of 80 nm and a mean pore wall thickness of 40 nm was obtained using a current density of  $120 \text{ mA/cm}^2$ . A thickness of about  $35 \mu\text{m}$  of the porous layer was obtained during the adequate anodization procedure. Figure 3.4(a) shows a SEM overview image of the cross-section of the investigated sample and 3.4(b) the top surface, where the quasi-regular arrangement of the pores is imminent. In Figure 3.4(c), a higher magnification SEM image of the cross-section is shown before infiltration of the nanoparticles. Figure 3.4(d) is a zero-loss filtered TEM image, showing magnetite nanoparticles of about  $\sim 20$  nm mean diameter at different diffracting conditions (diffraction contrast). By infiltration of magnetite nanoparticles into the pores the magnetic properties can be controlled via particle-size and distances between each other.

Magnetite nanoparticles were synthesized in the Institute of Material Science in Madrid (CSIC). Magnetite nanoparticles with a diameter of 5 nm were prepared by high temperature decomposition in the presence of an organic precursor, namely iron acetylacetonate (acac). A mixture, consisting of 0.71 g of  $\text{Fe}(\text{acac})_3$  (2 mmol), 2.38 g of 1,2-hexadecanediol (10 mmol), 1.69 g of oleic acid (6 mmol), 1.6 g of oleylamine (6 mmol) and 20 ml of trioctylamine was prepared and heated to  $200^\circ\text{C}$  in the presence of nitrogen gas flow for 120 minutes while stirring. The mixture was allowed to reflux another 30 minutes at  $369^\circ\text{C}$  under nitrogen atmosphere and then cooled down to room

temperature. The powder was collected with a magnet after precipitation by ethanol and then dried under nitrogen gas flow. The final suspension was then acquired by mixing the powder with 20 mL of hexane and 0.05 mL of oleic acid. To avoid agglomeration and oxidation, oleic acid is added to the mixture to coat the particles approximately by a 2 nm layer. Therefore the minimum distance between the particles is about 4 nm and exchange interactions between the particles do not occur. Additionally it is reducing the effect of the surface anisotropy. [59] [56]

From previous studies with Fourier transform infrared spectroscopy (FTIR) and EDS it is also known that oleic acid and magnetite nanoparticles are present within the pSi template [58]. Patterns of X-ray diffractograms of Fe<sub>3</sub>O<sub>4</sub> nanoparticles furthermore reveal an inverse spinel structure with lattice parameters of 8.38 Å, which is in good agreement with published magnetite patterns [60].

### 3.3 Copper Interconnects

Within the semiconductor industry in the mid-1990s, copper was gaining more and more foothold as an interconnect material in integrated circuits instead of aluminum, since copper has a higher conductivity value. These interconnect vias are stacked within several planar metal layers; and connects the transistors to each other within the circuit functional block (local), provides clock and signal distribution (intermediate) and delivers ground and power to all junctions (global). The trench size increases from local to global interconnects to decrease resistance and all three of them together create a complex network known as the back-end-of-line (BEOL).

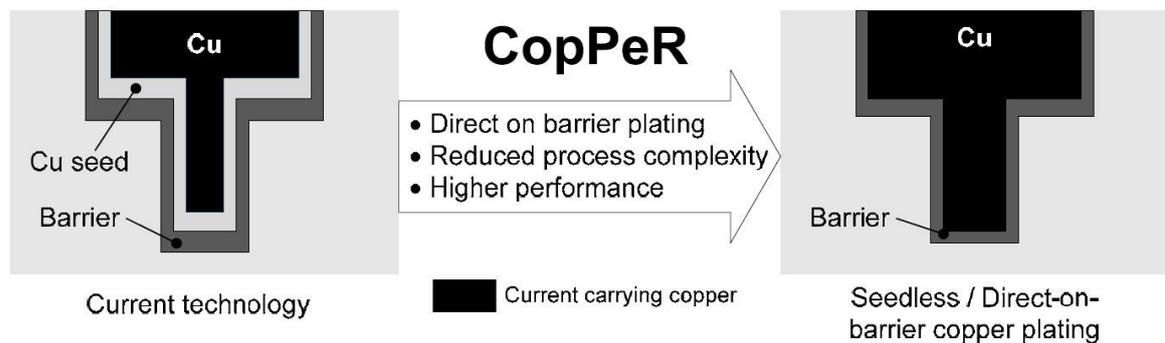
The down-scaling or miniaturization of the transistor feature sizes makes it possible to switch the transistors on and off faster and hence influences the clock frequency of the whole chip. The progress in down-scaling has led to a situation that currently the limiting factor in the clock frequency is the delay in the signal propagation in metal interconnects instead. This delay is proportional to the RC time constant, which depends on the dimensions and resistivity of the metal and the permittivity of the dielectric in between the trenches. Using low permittivity dielectrics and low resistivity metal as well as reducing the line lengths will reduce the time constant and speed up the clock frequency. Up to 50% reduction in RC time constant has been shown possible using copper ( $\rho = 1.7\mu\Omega cm$ ) instead of aluminum ( $2.7\mu\Omega cm$ ) [61].

Other advances of copper in relation to aluminum are better thermal conductivity and resistance to electromigration effects. The latter is a mass transport phenomena caused by the electrical current in the metal and can cause voids within the interconnect. The so called bamboo type grain structure is shown to reduce the electromigration effects, where the occurrences of grain boundaries parallel to the current are minimized. However within time, copper diffuses into the silicon substrate eventually compromising the device. This was prevented by a deposition of a tantalum and/or tantalum nitride liner by e.g. atomic layer deposition acting as a diffusion barrier between the silicon and copper. The paradigm shift from aluminum to copper also meant that the former subtractive etching procedures that worked for aluminum had to be replaced by so called damascene patterning scheme, where the vias are already etched in the dielectric and then filled with copper. Dual damascene process refers to simultaneous filling of both the vias and the trenches within the same copper deposition phase. [18] [62]

Electrochemical deposition is the common way of depositing copper into the vias but as the aqueous solvent oxidizes the tantalum, first a protective copper seed layer has to be applied on top of the tantalum by physical vapor deposition. This seed layer of copper is necessary for nucleation

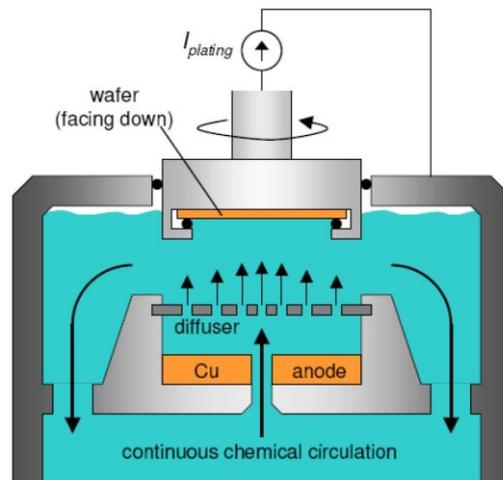
but has a higher resistivity value than the electrodeposited copper because of grain boundary scattering effects; therefore direct plating on tantalum would be advantageous. Since the seed layer is 5 to 10 nm thick, it is a problem for the sub 32 nm device scaling technology node as it takes too much space and raises the resistance.

A solution was suggested by using a different solvent in the electrochemical deposition process, non-aqueous ones e.g. ammonia or ionic liquids such that no oxidation takes place in the tantalum as seen in figure 3.5. This was accomplished within the CopPeR project (Copper interconnects for advanced performance and reliability), financed by the 7<sup>th</sup> Framework Programme of the European Commission. With the deposition of copper in liquid ammonia solvent there is no need for the extra copper seed layer, hence interconnect vias can be smaller, transistor count increases and chip performance improves. [63]



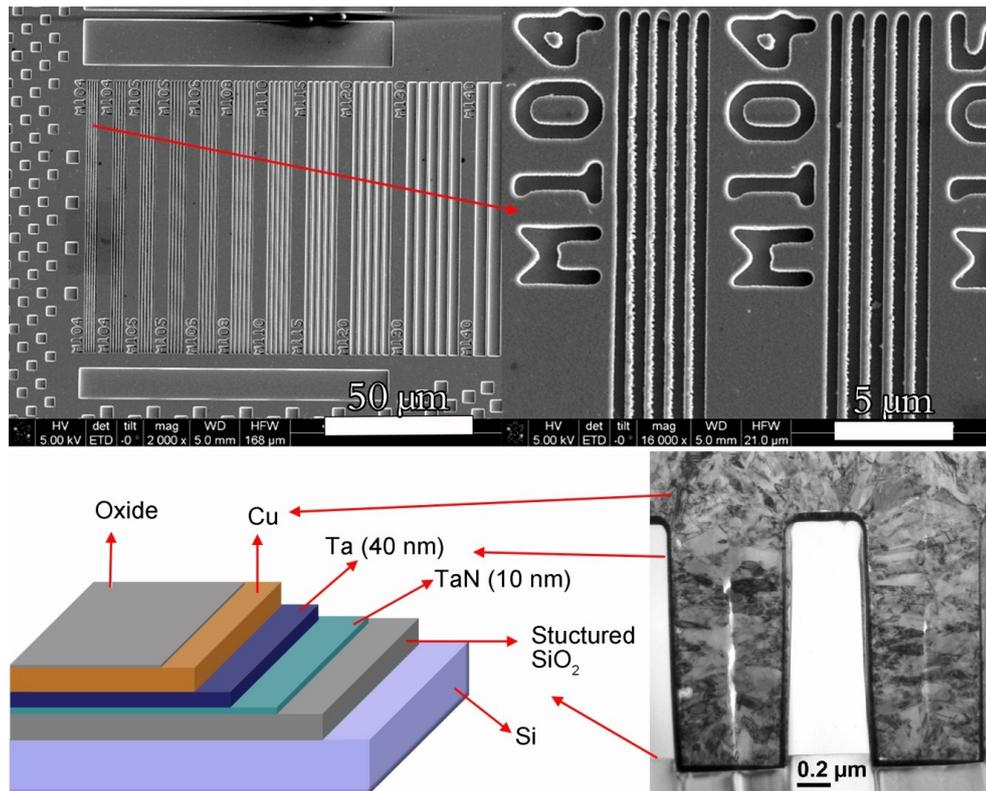
**Figure 3.5:** The aim of the CopPeR project was to make the copper seed layer redundant. Taken from [62].

The deposition of copper plating direct on tantalum is accomplished using a state-of-the-art electroplating cells, where the wafer, acting as a cathode, is placed above the anode facing down as seen in figure 3.6 [64] [65]. The wafer is circulating to assure even current density distribution. A uniform liquid flow is continuously applied from bottom to up and a bulk copper can be used as a copper source as an anode. The used liquid was non-aqueous ammonia with some additives like suppressors, accelerators, levelers and brighteners [62]. One goal of the project was to achieve an ideal filling of copper into the trenches and vias, known as superconformal or superfilling process, where no voids are formed within the copper resulting in a defect-free trench with low resistivity. This can be achieved by adding levelers to the bath acting to accelerate the deposition rate at the bottom of the trench. Another goal was to avoid an oxygen layer between the tantalum and copper interface, which was a result of too low vacuum conditions in the chamber. This was accomplished by removing the water by heating the ionic solution to 150 °C degrees and the oxygen gas was removed by adding an oxygen scavenger like hydroquinone [66].



**Figure 3.6:** Copper electroplating cell as used in the University of Leuven. Taken from [62].

On figure 3.7 a SEM image of the electroplated wafer is shown with vertical lines having different thicknesses. The investigated interconnect lines M104 are shown in the zoomed in image, where a clear distinction can be seen on the edge roughness between the left and right side of the lines. This was also evident from the tomographic investigations. Two of the five trenches are seen in the BF TEM image showing polycrystalline copper in different diffraction conditions. The line showing dark contrast around the copper is the Ta/Tan barrier layer as depicted in the lower left image of the elemental structure and layout of the sample. The vertical voids in the copper trenches implicates that superfilling was not achieved in the deposition process with this sample.



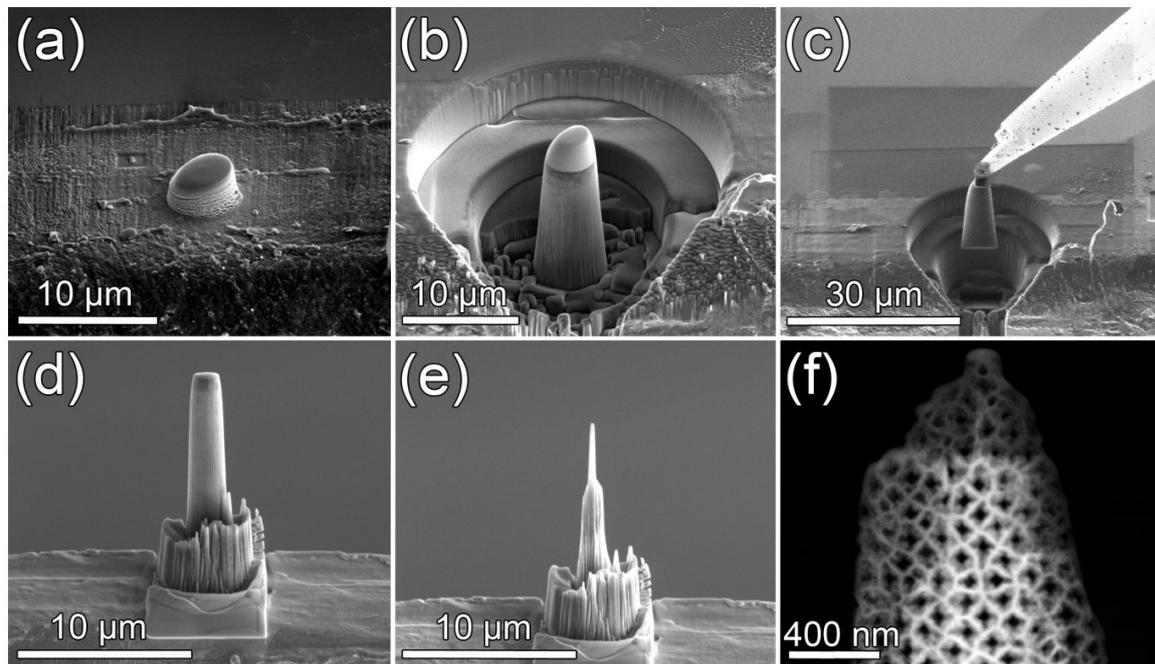
**Figure 3.7:** On top a SEM image of the chip is shown with zoomed in image on the right showing the 5 interconnect lines of interest. Below the structure of the sample is given and a BF TEM image of two trenches.

The metrology within the integrated circuit industry is vast and diverse and TEM is the mostly used characterization technique within the atomic scale. However with decreasing dimensions the small overlaps of materials within the projection dimension cannot be fully distinguished using cTEM. For reliable and complete analysis the three dimensional structure and morphology of the material becomes crucial and can be acquired using ET. To maximize the investigated volume with increasing the sample thickness one has to keep in mind the limitations of the imaging modes with respect to the requirements of a tomographic reconstruction, that is... The projection requirement.

### 3.4 TEM Sample Preparation

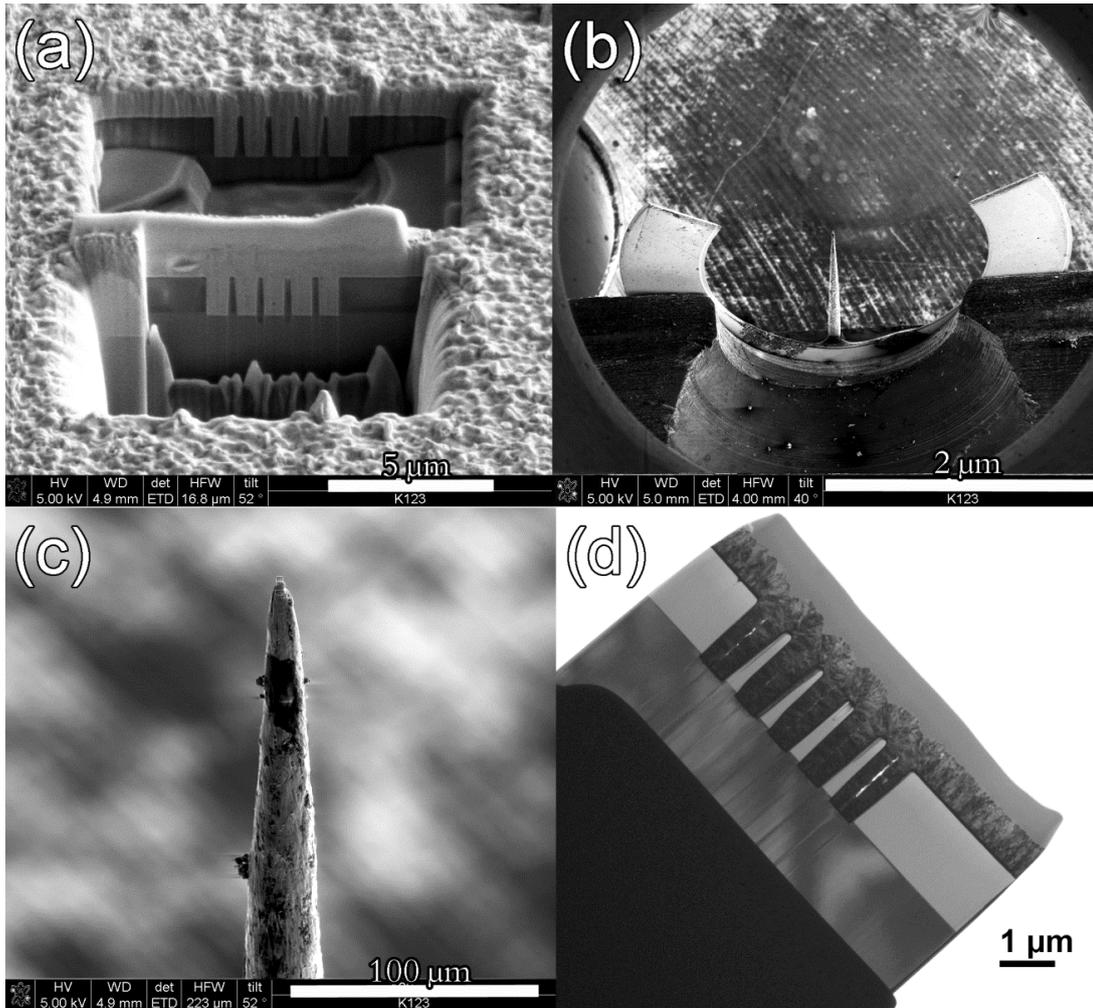
FEI Nova 200 Nanolab Dual-Beam FIB-SEM was used to prepare the pSi samples infiltrated with magnetite nanoparticles and the copper interconnects for TEM investigations. For the former, a pillar shaped sample was milled in room and cryo temperature and transferred to a TEM grid as seen in figure 3.8. Such a morphology is preferred in tomography since there is no self-shadowing from the sample, the thickness during tilting remains constant and it allows very high maximum tilt angles to minimize the missing wedge effect (see chapter 4.4.1). Before the TEM investigations, the samples were plasma cleaned using a Fischione Plasma Cleaner model 1020 for 2 minutes in order to minimize the carbon contamination during the acquisition of the tilt series. This procedure

effectively also removes the oleic acid coating on the magnetite nanoparticles; therefore within this work the radius of the nanoparticle means the radius without this surface layer.



**Figure 3.8:** (a) Coating of a protective platinum layer. (b) Formation of a pillar shaped sample with milling. (c) The pre-cut was transferred on top of a TEM grid (d) using a micromanipulator and (e) thinned further on the edge applying low ion currents. (f) STEM image showing the resulting structure of the sample. [2]

Best results for tomographic reconstructions were obtained such that the pores were parallel to the optical axis of the microscope. Therefore a cross-section of the sample was used to deposit a coating of a protective platinum layer using EBID, followed by IBID in the dual-beam platform (figure 3.8(a)). The platinum layer protects the underlying sample from the timely milling process. A pillar shaped sample was then formed by means of ion milling using annular patterns with high beam currents (figure 3.8(b)). With sufficient height of the pillar, a micromanipulator was then positioned at the top of the sample and glued together using IBID as shown in figure 3.8(c) followed by cutting the pillar from below. The pre-cut was transferred and glued on top of a Molybdenum semi-grid (figure 3.8(d)). Final milling and polishing was applied using low ion currents (1 pA). Figure 3.8(f) depicts a STEM image showing the resulting structure of the sample as used in the tomographic tilt series acquisition. Only the top triangle shaped structure was reconstructed.



**Figure 3.9:** FIB preparation of the copper interconnect sample. (a) Two trenches were milled below and above the position of interest. Similar as above the lamella was cut and transferred to custom made tomography grid (b). (c) The lamella was attached to the tip by using IBAD and milled further with low ion currents to produce the final structure as seen in BF TEM image (d). (a,b and c image courtesy: Martina Dienstleder)

The copper interconnect sample was prepared as a slab shaped lamella with a thickness about 150 nm. Best results for tomographic reconstructions were accomplished when the long sides of the trenches were in parallel with the tilt axis. The preparation steps were similar to above except here two trenches were milled above and below the sample region as seen in figure 3.9(a). The TEM grid (figure 3.9(b)) was custom made to maximize the tilt angles with no self-shadowing from the grid. The sample was moved with a micromanipulator on top of the TEM grid needle and glued together with IBID (figure 3.9(c)). The final structure as used in the tomographic characterization can be seen on figure 3.9(d), which is a BF TEM image of the specimen.

What follows is a list including all samples investigated in this thesis with a short description of the sample preparation steps and tomographic acquisition parameters. The cobalt iron oxide and carbon soot samples are used later to highlight some features of ET.

## **Magnetite nanoparticles embedded in pSi**

The tilt series was acquired with the TF20 using mentioned in chapter 2.1. A tilt range from  $-79^\circ$  to  $78^\circ$  was used with  $1^\circ$  tilt step and was acquired with HAADF mode. The magnification was 160 kX with a pixel size of 0.76 nm (1024x1024). The acquisition was started at  $0^\circ$  tilt angle continuing to minimum tilt angle and then again from  $0^\circ$  to maximum tilt angle. A camera length of 200 mm was used and a spot size 9 having a FWHM of 0.295 nm. The two pSi tilt series prepared in room temperature and cryo conditions were acquired using TF20 with tilt angles ranging from  $-78^\circ$  to  $78^\circ$  with a  $2^\circ$  tilt increment.

## **Copper interconnect**

The tilt series was acquired with TF20. For both tilt series taken with IBF and HAADF mode, a tilt range from  $-72^\circ$  to  $74^\circ$  was used with  $2^\circ$  tilt step. The magnification was 28.5 kX with a pixel size of 2.18 nm (2048x2048). The acquisition was started at minimum tilt angle and then continuing to maximum tilt angle.

## **Cobalt iron oxide (CoFe<sub>2</sub>O<sub>4</sub>)**

### **Nanocubes**

The fabrication of the nanocubes was performed by Mykhailo Sytnyk and described in [67]. The tilt series was acquired with TF20 using HAADF mode. The nanocubes were dispersed in hexane and drop casted to a regular TEM grid. A tilt range from  $-74^\circ$  to  $74^\circ$  was used with  $2^\circ$  tilt step. The magnification was 225 kX with a pixel size of 0.55 nm (1024x1024). The acquisition was started at  $0^\circ$  tilt angle continuing to minimum tilt angle and then again from  $0^\circ$  to maximum tilt angle.

### **Nanospheres**

The fabrication of the nanocubes was performed by Mykhailo Sytnyk and described in [67]. The tilt series was acquired with T12 using 120 kV beam energy and BF mode. The nanospheres were dispersed in hexane and drop casted to a regular holey carbon TEM grid. A tilt range from  $-74^\circ$  to  $74^\circ$  was used with  $2^\circ$  tilt step. The magnification was 67 kX with a pixel size of 1.52 nm (1024x1024). The acquisition was started at minimum tilt angle and then continuing to maximum tilt angle.

### **Carbon soot nanoparticles**

The tilt series was acquired with T12 using 120 kV beam energy and BF mode. A tilt range from  $-72^\circ$  to  $64^\circ$  was used with  $2^\circ$  tilt step. The magnification was 350 kX with a pixel size of 0.29 nm (1024x1024). The acquisition was started at minimum tilt angle and then continuing to maximum tilt angle. Spherical gold nanoparticles were drop casted into the grid serving as fiducial markers for alignment purposes. The alignment was done using 4 colloidal gold nanoparticles as fiducial

markers and was tracked using IMOD [68] giving a residual error of 0.527 (distance in pixels between the measured and predicted position of a fiducial) with a standard deviation of 0.302, meaning good alignment quality.

## 4 Electron Tomography

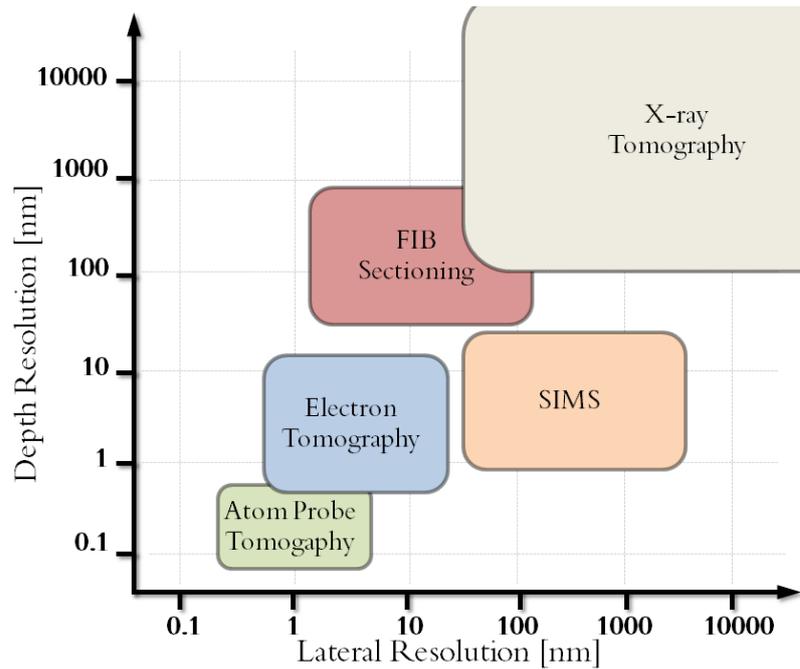
Electron tomography (ET) [69] [70] [71] [72] [73] is a method of recovering the lost projection dimension in the TEM by tilting the sample and subsequently taking images at different projection angles. One can recover the original structure by specific reconstruction algorithms that are based on the inverse of the Radon transform (RT) [74] [75]. The mathematical procedure of reconstruction from projections belongs to the field of inverse problems. The projected intensities from the acquired images are back-projected taking account the projection angles to form a reconstructed image of the object by accumulation of intensities.

Historically the main field studied by ET has been the life sciences in particular molecules, bacteria and cells [76] [77] and is still widely used [78]. For material sciences it became a routine application only after the introduction of HAADF mode [79] [80], which allow reconstructions of crystalline materials since the projection requirement is fulfilled. Since then, ET has been used for wide variety of different materials and structures. Much attention has also been given to other modes of acquisition like EFTEM [81], EDS and EELS [82] [83] [84] to gain insight into the elemental distribution within the investigated samples.

A tomographic characterization consists of 4 different steps: acquisition of the tilt series, the alignment, reconstruction and segmentation. On a well calibrated microscope, the acquisition is usually automatic after the initial alignment procedures. Also for the reconstruction, well established algorithms exist and can be readily used. Most problematic are the precise alignment of the tilt series and the segmentation, where in this thesis, more emphasis has been given. In this chapter a short overview of other 3D metrologies is given followed by a practical description of the necessary alignment procedures needed before the acquisition of a tilt series. Common methods for alignment are explained and different reconstruction algorithms are introduced. Additionally the resolution within tomograms is explored and finally methods to segment and visualize the data in 3D are introduced.

### 4.1 Three Dimensional Metrology

ET is not the only method to characterize materials in three dimensions and consideration of its use should be based on the spatial resolution and field of view one needs to achieve. In the following sections three additional imaging methods are briefly introduced. Figure 4.1 shows different metrologies for 3D characterization given in terms of achievable spatial resolution. Here methods are limited to only those that are used within the material science field and in the micro- or nanoscale. Figure 4.1 clarifies that when considering the resolution needed to quantitatively characterize the magnetite nanoparticles; and simultaneously achieve the field of view necessary for the mesoporous network -for this work the only metrology available is ET.



**Figure 4.1:** Resolution limits for different 3D metrologies.

### 4.1.1 X-Ray Tomography

Used in diagnostic medicine and material science, X-ray tomography [85] [86] or computer assisted tomography provide micrometer resolution and large fields of view. Similarly to ET it is based on recording the transmitted signal from the object to a CCD detector. However the shape of the beam can be parallel, cone or fan shaped, which must be considered in the reconstruction process. X-ray tomography is by far the most used method for three dimensional imaging widely used in archeological, material science, biological and medical research. The X-ray source can be a synchrotron beam, a laboratory microfocus tube or a focused electron beam on a thin foil generating X-rays. The highest resolution is achieved with a synchrotron since it delivers 10 orders of magnitude more brightness, is monochromatic and the distance to the object can be large giving parallel beams.

By using different lens systems, higher resolution of the acquired images, also known as radiographs can be achieved. Depending on the high-energy (hard) or low energy (soft) X-rays different setups are available for high resolution imaging. For latter Fresnel zone plates can be used to condense the beam onto the sample and then magnify the image. For the former asymmetrically cut Bragg crystal magnifiers or elliptical shaped Kirkpatrick-Baez mirrors can be utilized for improved resolution. [87]

The most used contrast mechanism in X-ray imaging is attenuation, where the recorded intensity is related to the absorption coefficient, which is linked to density, atomic number and the energy of the incident beam. Different phase contrast methods are additionally used specifically for low Z materials, which gives low attenuation contrast. Here a series of images is taken with different

focal distances to the sample at every rotation step and the difference between attenuation and phase can be separated.

#### 4.1.2 FIB/SEM/SIMS Serial Sectioning

To create a three dimensional representation of the sample, serial sectioning does not rely on the radon transform. Instead slices of the sample are either sputtered or cut away in thin sections, imaged and then combined layer after layer into a 3D stack. In secondary ion mass spectrometry (SIMS) depth profiling the surface is done by sputtering away one layer at a time and at every point a time of flight spectra is taken, which can be converted to mass spectra since the ions time of flight is merely proportional to its charge/mass ratio. In FIB serial sectioning, the ion beam is similarly used to sputter away thin sections with low ion currents and then SEM is commonly used to image the new surface.

In addition to these ion sputtering schemes, ultramicrotomy can be harnessed by using a thin blade to cut away sections and then image the surface with SEM. In all of these methods the spatial resolution is limited to the devices capabilities of delivering images (SEM > 1 nm, SIMS > 50 nm [88]) and the depth resolution depends on the used ion beam energy or how thin sections the ultramicrotomy blade can cut, for the latter usually limited to 50 nm and 1 nm for the former. Beam and mechanical damage to the material is always present in these methods. [89] [90] [91]

#### 4.1.3 Atom Probe Tomography

In atom probe tomography the morphology of the investigated sample is limited to a needle shaped pillar and the tip is placed in an ultra-high vacuum close to an electrode. A pulsed high voltage ionizes the surface atoms of the tip causing field evaporation. These ions are then accelerated towards a crossed delay line, a position sensitive detector. The time between the pulsed signal and detection gives the time of flight, from which the charge/mass ratio and subsequently, the elemental composition can be acquired. Whereas the original lateral position in the sample tip can be approximated by the point of impact of the ion hitting the detector. This leads to a three dimensional chemical map of the sample as more and more monolayers of the tip are evaporated. This setup works well with metals and other good conductive materials but for insulators and semi-conductors a laser-assisted atom probe has to be used, where the laser beam introduces thermal evaporation. [92] [93]

Atomic resolution can be in principle achieved, there are however uncertainties if the field evaporation is not constant like for example complex materials having many different elements. Also the original position of the ion in the sample tip is not always straightforward to calculate from the ions impact position to the detector. Not to forget that the detector itself has a considerable dead time between two successive detection events.

## 4.2 Electron Tomography Calibrations

For the automatic acquisition of the tilt series, several pre-calibrations have to be performed including image shift, stage shift, dynamic focus and sample holder calibrations. Within the FEI software Xplore 3D [94], which was used in this work, these calibrations are automated when appropriate calibration samples and a calibrated microscope are available. [95]

### **Image and stage shift calibration**

To accurately correct the sample movement after tilting of the stage, the image and stage shift calibrations has to be correct. This is simply done by acquiring a reference image, then shifting the sample and taking another image. The amount of shift is measured using cross-correlation (CC, see chapter 4.3) on the two images. Simultaneously the scan rotation is determined such that the scanning direction is normal to the microscope's tilt axis to ensure the correct behavior of the dynamic focusing and for easier tilt axis alignment. For the TF20 in Felmi-ZFE, this was found to be  $88.7^\circ$  degrees.

### **Dynamic focusing calibration**

Dynamic focusing [96] [15] ensures that the image remains in focus throughout the whole image even at high tilt angles. In TEM this is not possible since with high tilt angles, the depth of focus is limited to the area near the tilt axis only, while the edges of the image remain unfocused. In STEM mode one can change the condenser lens excitation value dynamically while rastering through the image such that the whole image remains focused. To accomplish this, a calibration has to be performed, where the sample is shifted away from eucentric height and then refocused. The automatic focusing is done by taking a series of images changing the defocus while using auto-correlation to determine the quality of focus. This calibration gives a value of how much the defocus has to be changed for certain height difference.

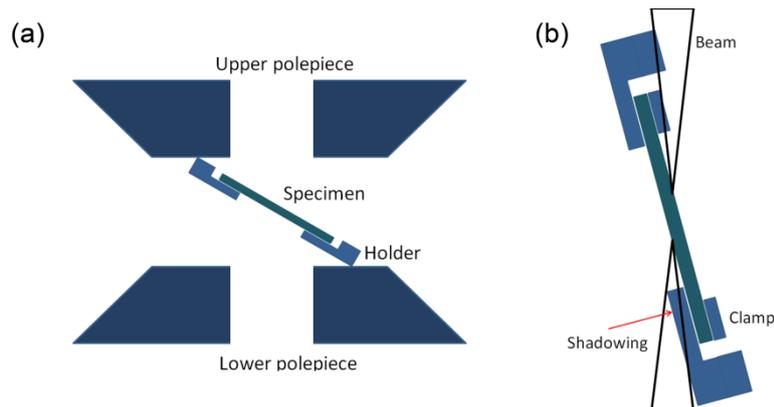
### **Holder calibrations**

Sample holders have unique responses to tilting and shifting due to the holder's absolute and center of mass as well as the position of the grid. Therefore each holder used in tomography has to be calibrated separately and the calibrations can be saved and reloaded if several holders are used. A Fischione 2040 tomography (figure 4.2) holder features a tip width of only 3 mm (figure 4.2(b)). It fits between the objective lens upper and lower pole pieces such that no pole touch takes place as seen in figure 4.3(a). This holder allows a maximum tilt angle of about  $72^\circ$  degrees before self-shadowing (Figure 4.3(b)). It has a possibility to manually rotate in-plane the specimen  $360^\circ$  degrees for dual-axis tomography, where two reconstructions are averaged taken with a  $90^\circ$  degree rotation difference. Additionally a Fischione 2020 tomography holder was used in this work, which allows  $80^\circ$  degrees maximum tilt, the maximum rotation of the goniometer.



**Figure 4.2:** (a) A Fischione 2040 tomography holder with manual in-plane rotation possibility for multiple axis tomography. (b) A detail of the sample holder. The sample is secured to the holder by the use of flexi-clamps.

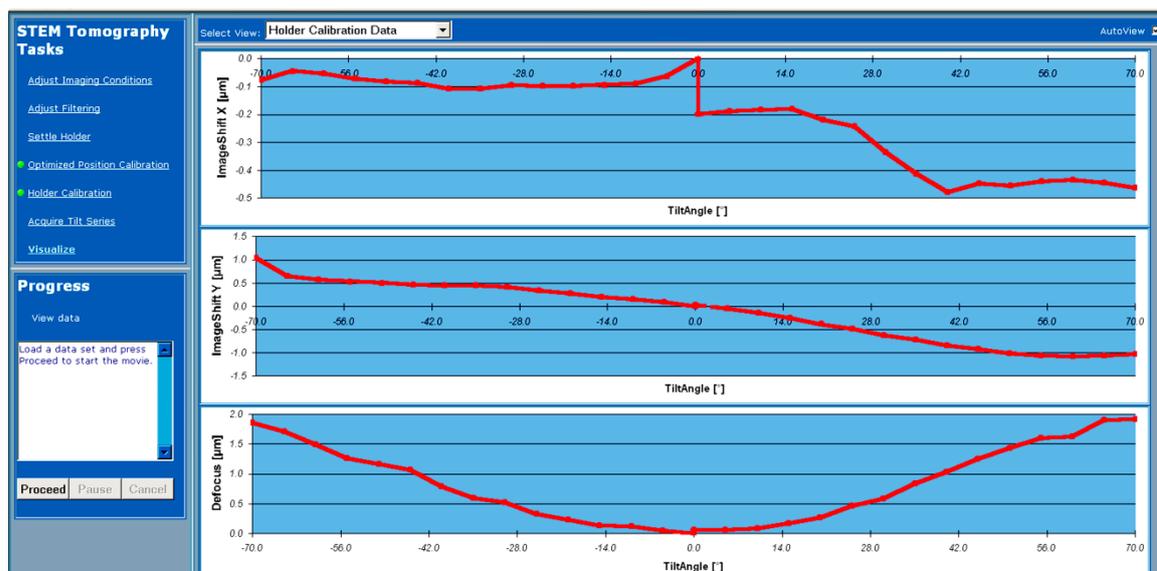
A tilt series has to be acquired to pre-calibrate the holder for shifts and defocus changes. [97] These calibrations are then used to predict the behavior of the holder in an actual tilt series acquisition at higher magnifications. Figure 4.4 shows these calibrations for the Fischione 2040 holder acquired with the TF20. The upper and middle curves shows the measured x and y shifts respectively and the lower curve gives the change in defocus. Additionally an optimized position calibration for the holder needs to be measured, which ensures that the beam is translated to the holder's center. This calibration shifts the beam many times, then tilts the sample and determines the defocus. User might have to retune the astigmatism and rotation center for large shifts.



**Figure 4.3:** (a) Image of the sample holder in between the upper and lower objective lens pole pieces. The holder must have as small enough dimensions to prevent pole piece touch. This limits the maximum tilt axis rotation for the holder. (b) Self-shadowing of the holder.

### Routines before the acquisition

For the actual tilt series acquisition after the calibrations one always has to perform the calibration steps as detailed in the STEM calibrations in chapter 2.4. Additionally the maximum and minimum tilt angles have to be checked by rotating the stage and following the sample using stage shift. Proper image acquisition times need to be verified along with the filtering parameters such that the CC procedure is optimized for tracking. For the auto focusing part the defocus scale and step size are important as well as the location of the area for the multiple exposures during the acquisition. If possible, one should use an uncritical area situated below or above the object of interest within the tilt axis to minimize the beam damage and carbon contamination. [18] [20] [94]



**Figure 4.4:** Calibration data from the holder Fischione 2040 measured with the TF20.

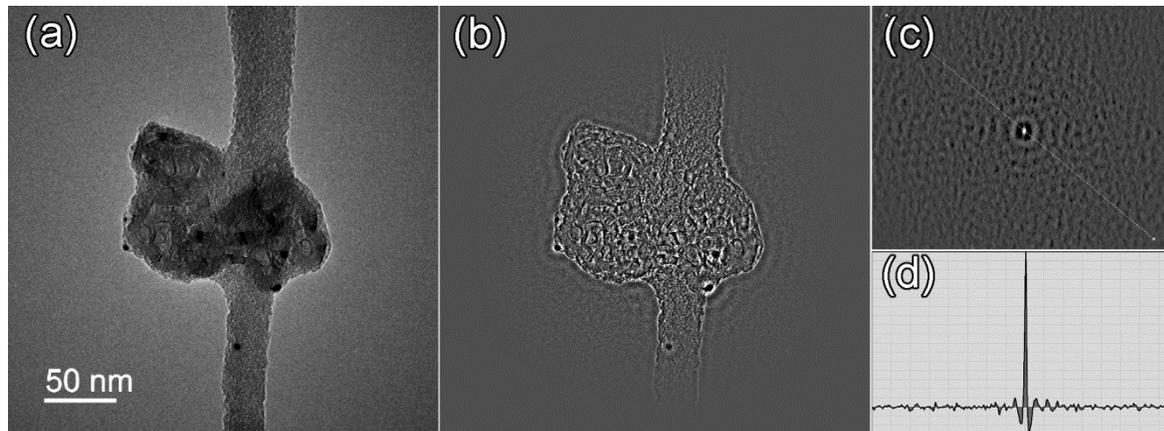
### 4.3 Alignment Methods

Cross-correlation (CC) forms the basis for the tracking in the acquisition process and for the subsequent rough alignment of the tilt series. CC is defined as a convolution of the two images. Provided that similar images are translated with respect to each other, then the CC gives a peak at a location, which shows the applied translation. The cross-correlation theorem states, that multiplying the Fourier transforms (FT) of the two images, gives equivalent results as a convolution and this is the common way of performing CC [98]. However the comparison of images at different projection angles resembles more of a deformable registration problem than a rigid one and CC is not suitable for accurate final alignment. A rigid transformation implies there is a point to point correspondence between the reference image and the moving image to be aligned, where a mere translation and rotation will suffice. But this is not the case with subsequent tilt images since they are projection images of a rigid 3D body and the structures are seen as deformed. [99]

The tilt series is usually aligned starting from zero degree tilt angle and comparing subsequent images marching to both directions of maximum and minimum tilt angle. Small errors are therefore accumulated throughout the series and the error between zero and maximum tilt angle could be quite large. For this reason, CC is often not a suitable alignment method for high quality reconstructions. A region of interest (ROI) close to the tilt axis and narrower in the direction normal to the tilt angle coupled with filtering is a necessary step for CC. Band pass filtering and usage of hanning windows are common choices. In figure 4.5(a) one sees a TEM BF zero degree tilt image of a carbon soot particle on a holey carbon grid and the filtered version of it 4.5(b), where band pass filtering and hanning window filter were used. The CC with the next tilt image (not shown here) then gives the zoomed in CC peak as seen in figure 4.5(c) and the diagonal line profile of it 4.5(d). Without filtering this peak would be quite broad.

The most precise and best alignment method (from low to intermediate magnifications) for the tomographic tilt series is the fiducial marker tracking. This obviously means that one has to deposit

golden colloidal markers onto the sample, which could be sometimes difficult or impossible to achieve. For example with the pSi sample infiltrated with magnetite nanoparticles, the purpose was to characterize the volume and spatial distribution of the particles. The spherical gold particles acting as markers would then be difficult to separate, thus interfering with the distribution producing false results. Therefore new alignment methods were written and existing ones reviewed.



**Figure 4.5:** (a) TEM BF image of a carbon soot particle with 6 nm gold colloidal particles as fiducial markers. (b) Result of a band pass (low frequencies are cut off) and hanning window filter gives the zoomed in CC peak at (c) and the diagonal line profile of it at (d).

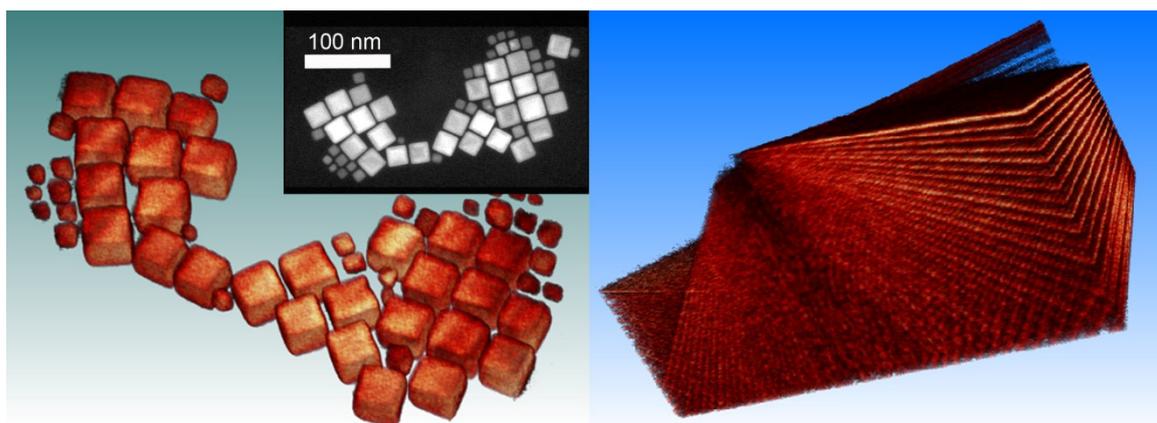
Colloidal gold can be purchased commercially by different vendors and the particles can be mixed with water or hexane. The agglomerated particles in the solution can be separated with ultrasound treatment and drop casted on top of the sample using a micropipette. The protocol to get enough and not too many markers onto the sample has to be experimentally verified. The drawback is that one often does not have enough (8-20) markers around the object of interest in the sample. The best open source software for tracking the markers is IMOD [68], where the particles are tracked separately and locally using CC. For good quality projections these markers are found automatically but often one has to manually correct these positions especially at high tilt angles. The tracked positions are then used as an input to the projection matrix equation that uses variable metric minimization routine to minimize the error between the tracked positions and the predicted ones [68]. This method is quite powerful since it can find the global rotation and tilt axis shift as well as translation, in-plane rotation, stretch, scaling, skewing, thinning and the non-perpendicularity of the electron beam with respect to the tilt axis. There are some restrictions however of what can be simultaneously solved. Additionally it is a unique alignment method with respect to the quality of the alignment as it gives a residual error of the fit: a measure for the quality of the alignment such that two different reconstructions could be quantitatively compared.

The global tilt axis rotation and shift has to be corrected within the tilt series and a useful manual and visual way to do this, first used in [100] and as implemented in Digital Micrograph™ [101], is to reconstruct three planes from the tilt series situated high, middle and low within the image. The middle one is used to correct the shift, which shows parabola shaped errors on high contrast objects if the tilt axis is shifted. Then the high and low ones are used for the global tilt axis rotation, which gives similar errors if the rotation is wrong. This can also be done automatically using exhaustive search with a maximum variance as the cost function.

New and improved automatic and marker-less alignment techniques for tomography are being published at a steady rate. An alignment and reconstruction program called Tomato [102] considers the quality of the reconstruction to obtain a valid alignment. Projections are translated individually within a loop of an exhaustive search and the variance of the 2D reconstruction is calculated. The peak of the variance then gives the correct alignment. Software called Raptor [103] can track low contrast fiducial markers within the tilt series and has been implemented within IMOD. There are also frameworks to track arbitrary objects within the tilt series [104]. Center of mass –alignment method [105] was proposed, where the projection images were first projected in x and y direction and with those 1D images then the center of gravity was calculated and cross-correlated.

## 4.4 Reconstruction Methods

To understand why it is possible to reconstruct a 3D object from its 2D projections, one can turn to the Fourier slice theorem (FST). It states that the Fourier transform (FT) of a projection image, is a 2D plane in the 3D FT of the reconstruction tilted by the projection angle used [106]. This can be seen in figure 4.6, where a volume rendered visualization of the STEM-HAADF reconstruction is shown from a cobalt iron oxide ( $\text{CoFe}_2\text{O}_4$ ) nanocube assembly, lying on a holey carbon film. On the right, as an example the power spectrum (modulus of the FT) of the reconstruction shows the FT planes of the projection images. By placing the FTs of the acquired images into a 3D complex image according to their tilt angles, it is then possible to reconstruct the object in real space by taking the inverse 3D transform. This is known as the direct Fourier reconstruction method [107]. It is not used often because until recently, the computers could not hold such a large data set in the RAM and it is severely hampered with interpolation problems between the Cartesian and polar coordinates. There exist promising new methods however, which can bypass this problem by using non-uniform FTs [108].

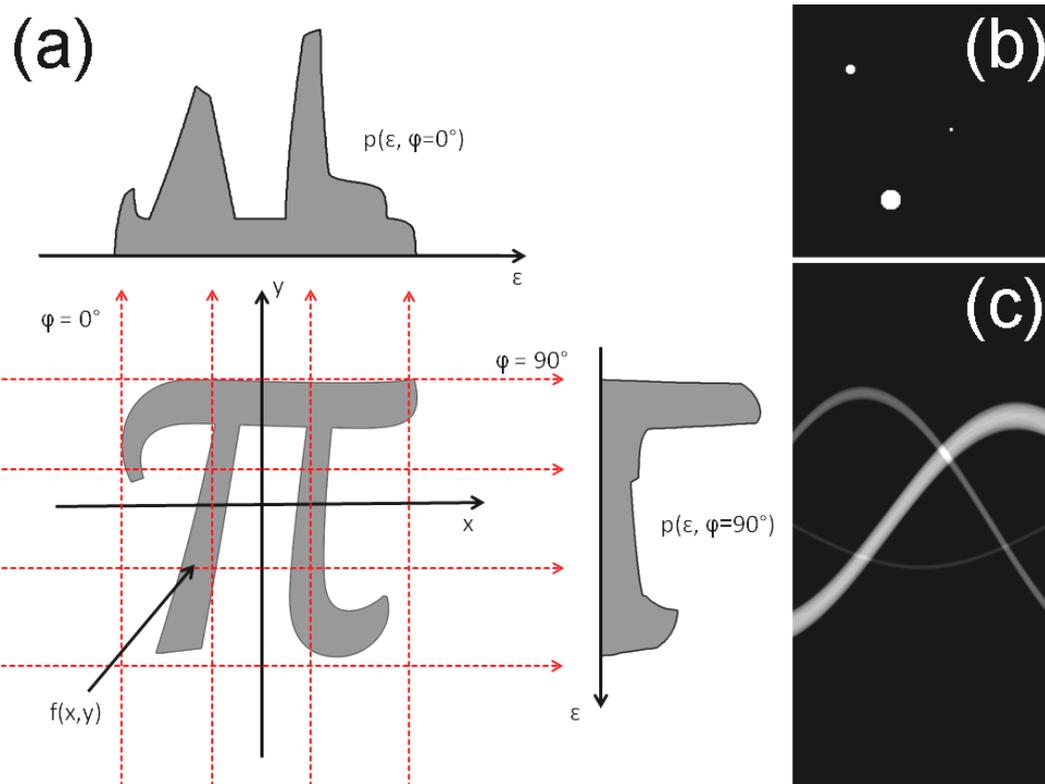


**Figure 4.6:** Left: volume rendered visualization of the HAADF WBP reconstruction of a  $\text{CoFe}_2\text{O}_4$  nanocubes assembly. The size of the bigger nanocubes was in average 25 nm as seen in the included zero degree tilt image. Right: Half of the 3D power spectrum of the reconstruction showing the planes arranged according to the tilt angles used, serving as a visual confirmation of the Fourier slice theorem.

Today, the most used methods for reconstruction are based on back-projection, wherein the mathematical basis to reconstruct an object from projections, is the Radon transform (RT). It transforms the 2D image into 1D projection at a specific projection angle, where the intensity is integrated along the projected ray [74] [109]. Every image taken with the TEM using various contrast mechanisms can be thought of as a RT of the specimen. A 3D reconstruction is achieved through the inverse of the RT. It is defined as

$$p(\varepsilon, \varphi) = \int f(x, y) \delta(x \cos \varphi + y \sin \varphi - \varepsilon) dx dy, \quad (19)$$

where  $\varepsilon$  is the distance from the origin normal to the projection angle  $\varphi$  (see figure 4.7(a)). Function  $f(x, y)$  is the object and  $\delta(\dots)$  is the Dirac delta function specifying an individual line across the object giving the intensity  $p$  at that ray. Integration ensures that all points within that ray are considered. This formulation for the line inside the delta function avoids the infinity problem at zero angle in the Cartesian formulation. The function  $p(\varepsilon, \varphi)$  is called the sinogram (figure 4.7(c)) of the object and has been used for the alignment of the tilt series [110]. Solving the inverse RT can be difficult in practice because of the limited number of tilt angles available in ET.



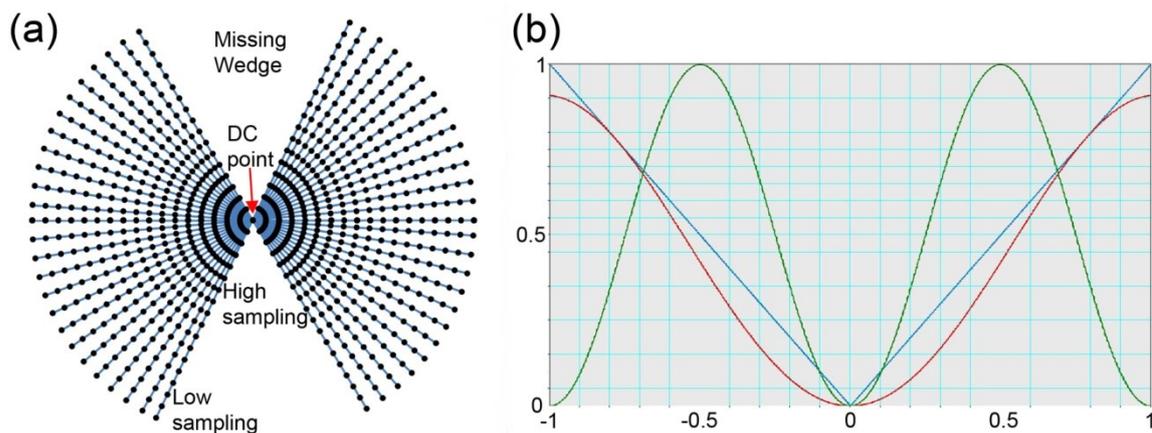
**Figure 4.7:** (a) Radon transforms of the symbol  $\pi$  ( $=f(x, y)$ ) at  $\varphi = 0^\circ$  and  $90^\circ$  while  $\varepsilon$  is continuous. (b) Image with three spheres having different radius. Applying radon transform with tilt angles ranging from  $-89^\circ$  to  $90^\circ$  with a  $1^\circ$  degree tilt step, giving the sinogram (c) after combining the 1D projections into a single 2D image.

The reconstruction algorithms based on back-projection imposes restrictions on the projected signal. The projection requirement states that the intensity/signal acquired on a microscope under different contrast mechanisms has to be a monotonic function of some physical property of the sample. Although STEM HAADF mode minimizes the diffraction contrast, it is not monotonic with thick samples as seen in figure 2.6. Hence within this thesis, IBF mode was used for the copper interconnect samples. For biological samples the contrast reversal is hardly a problem, as successful tomograms of over 1  $\mu\text{m}$  thick samples has been reported [111]. But within the materials science field, the thickness and density of the sample has to be always considered to avoid false interpretations. For polymers, an IBF-STEM based tomography for even 8  $\mu\text{m}$  thickness samples was suggested possible by Monte Carlo simulations. [19] [112]

#### 4.4.1 Weighted Back Projection

Weighted back projection (WBP) is the most common strategy for the reconstruction of an object from its projections. In back projection as a first approximation, the 1D projections are simply smeared through the 2D image on different angles. The complete 3D reconstruction is usually performed for one 2D plane at a time since they are independent of each other, subsequently stacked into a 3D volume.

Back projection however produces images with very low contrast and resolution, which can be understood from the FST. When the sampling of the image is a constant, then in Fourier space the low frequencies near the DC-point are over sampled and the high frequencies (small details in the image) are under-sampled by just geometrical considerations as seen in figure 4.8(a). Increasing the sampling for higher frequencies by taking more projection images will give reconstructions with better resolution, but the ratio of the over/under sampling will remain constant. The missing data in fourier space has a wedge shape and is hence called the missing wedge problem.

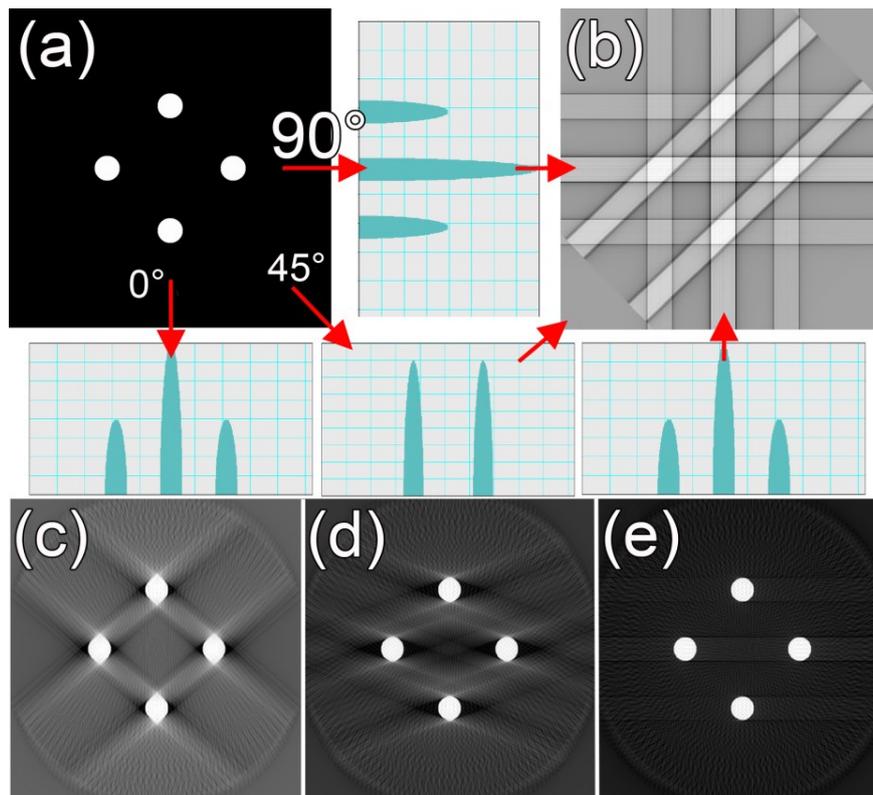


**Figure 4.8:** (a) Presentation of the inadequate sampling in the Fourier domain at high frequencies and the absence of data known as the missing wedge caused by the limited maximum tilt angles in the acquisition process of the tilt series. (b) Window functions for filtering in the Fourier domain as used in the WBP reconstruction method. Hanning filter (green), Shepp-Logan filter (red) and Ram-Lak filter (blue). These functions extend from the DC point to the 1D FT image edge.

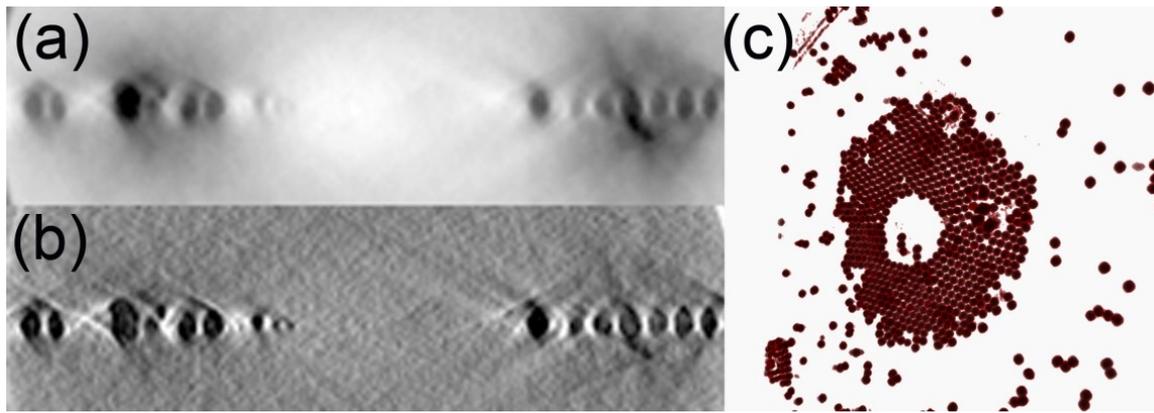
WBP tries to overcome this issue by enhancing the high frequency intensities through multiplication of the FT's with a weighting window function. Common functions are Ram-Lak filter, Shepp-Logan filter, cosine filter and hanning window (see figure 4.8(b)) [113]. The slopes of these curves are based on the tilt series sampling and the most important difference between them is the tail of the curve at high frequencies. The 1D projections' FT's are first filtered and the inverse FT's are then back projected through the image as illustrated in figure 4.9(a,b). [114]

In figure 4.9(c-e) the effect of the missing wedge is demonstrated on four spheres and the elongation of the spheres with limited tilt angles is severe. The purpose of projections from certain tilt angle is to confine the intensities of objects in normal direction to that angle. The elongation can be qualitatively understood, because there are no projections to confine the intensities of the objects in the z-direction at high tilt angles.

On figure 4.10(a) one sees a central plane of the reconstruction of  $\text{CoFe}_2\text{O}_4$  spherical nanoparticle assembly on a holey carbon film formed by back projection without any filtering. The low contrast and resolution are immanent as compared to figure 4.10(b), which is a WBP reconstruction with the 1D FT's filtered using a Ram-Lak window. Some elongation is clearly seen on the spherical nanoparticles in the vertical direction caused by the missing wedge. In figure 4.10(c), a volume rendered visualization of the whole reconstruction is shown. A process of self-assembly is seen as the nanoparticles are formed as a donut shape lattice.



**Figure 4.9:** From the original test image (a) consisting of four spheres, three re-projections were generated to angles  $0^\circ$ ,  $45^\circ$  and  $90^\circ$  as shown in figure. These re-projections were again back projected to form the image (b). The complete reconstruction is then formed by back projecting using all tilt angles as shown in (c-e), where the maximum tilt angle was  $50^\circ$ ,  $70^\circ$  and  $90^\circ$  respectively.

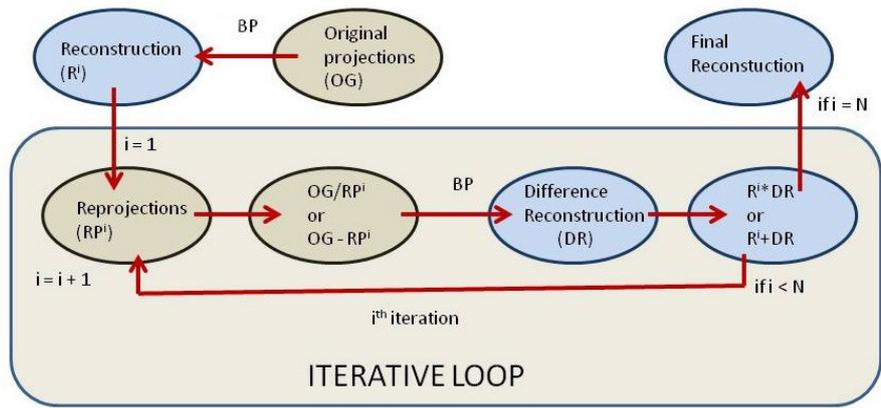


**Figure 4.10:** (a) Back projection reconstruction of spherical  $\text{CoFe}_2\text{O}_4$  nanoparticles ( $xz$ -plane) showing low contrast and resolution as compared to (b), which is a weighted back projection (WBP) filtered with a Ram-Lak window. The whole  $\text{CoFe}_2\text{O}_4$  nanoparticles assembly can be seen in (c), which is a volume rendered visualization of the reconstruction. A donut shaped lattice of nanoparticles has been formed by self-assembly on top of a holey carbon film (transparent).

#### 4.4.2 Simultaneous Iterative Reconstruction Technique

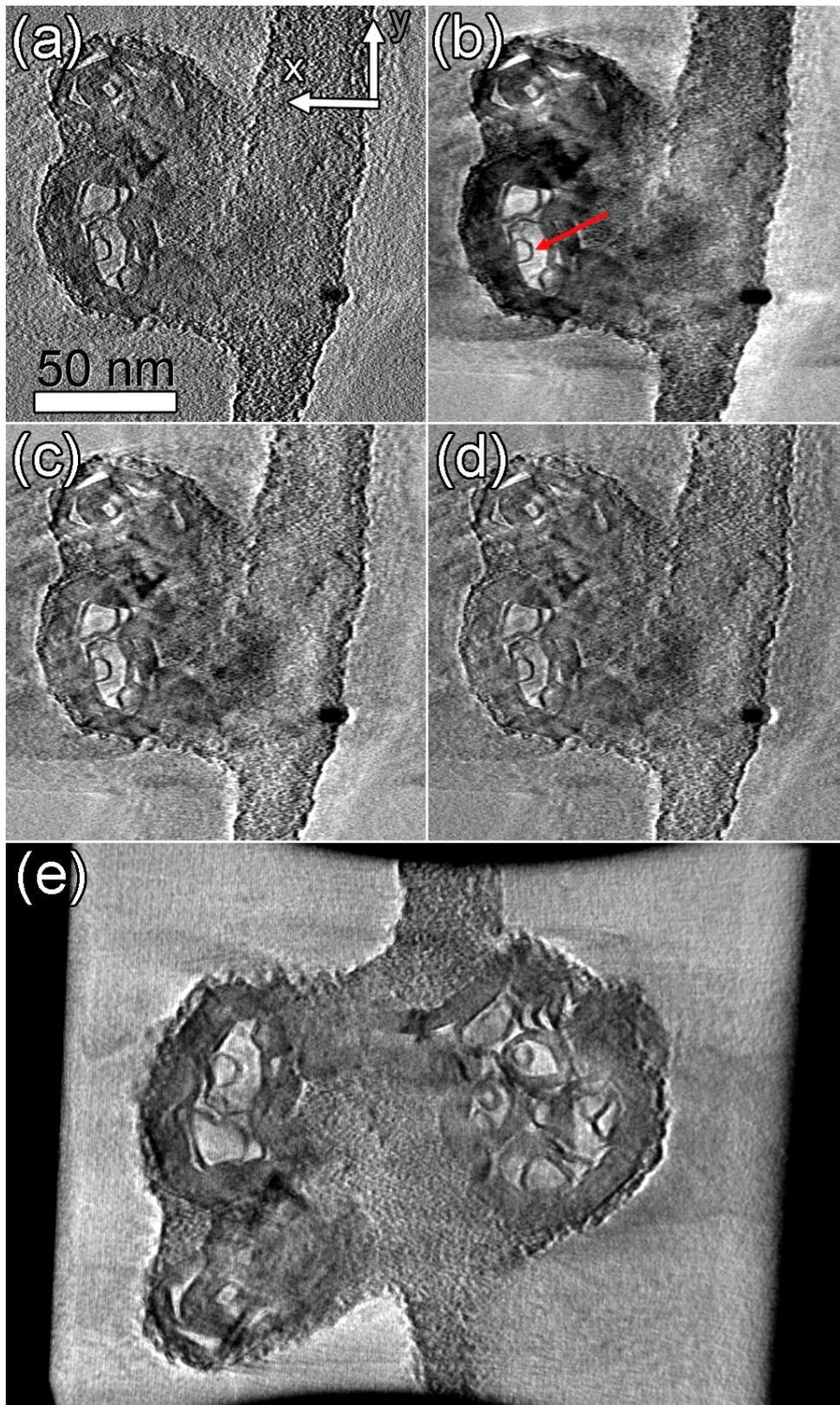
Another method for reconstruction is called simultaneous iterative reconstruction technique (SIRT). It is related to the re-projections of the initial reconstruction, which is done using back-projection [115]. An iterative loop is generated, where a reconstruction is calculated from the subtraction/division between the original tilt series and the re-projections, then added/multiplied with the current BP reconstruction to form a new reconstruction for comparison with the next iteration. The flow of this technique is illustrated in figure 4.11 and comprises successive estimates that are being compared to the acquired projections. This is a classic example of a constrained reconstruction method.

The fidelity of the reconstruction improves with iterations; however there is no common means of stopping criteria for the number of iterations in the loop to be used [116]. Additionally all possible misalignments and imaging artifacts of the tilt series will be inherently included and enhanced in the reconstruction. There is a positivity constraint in SIRT meaning no negative values are allowed and that the intensity should be ascending with thickness, which is the case in HAADF imaging.



**Figure 4.11:** Illustration of the SIRT framework. Reconstructions are colored as blue and tilt series as light brown.  $N$  is the total number of iterations and  $i$  is the current iteration.

The central plane of the reconstructed carbon soot particle is presented in figure 4.12, where (a) WBP, (b) SIRT (10 iterations), (c) SIRT (30 iterations) and (d) SIRT (50 iterations) were used. The purpose here is to present an example where it is difficult to choose the correct number of iterations to be used in the SIRT algorithm. Related to the resolution of the tomogram, one should always consider the quality of contrast differences between the object of interest and background. That is, optimize the contrast for segmentation. Figure 4.12(e) shows a volume rendered visualization of the reconstruction after smoothing with anisotropic filter. If the graphene layers are bound together by van der Waals forces, then the layer distance is 0.335 nm [117]. The layer thickness as measured at location marked to figure 4.12(b) by a red arrow was  $1.47 \pm 0.15$  nm giving 4-5 graphene layers, where the mistake is half the pixel size. The data cube was cut by a cylinder placed on a corner to highlight the graphene layers inside. Both the WBP and SIRT using different iterations should be always used and compared to obtain a high quality reconstruction with good contrast.



**Figure 4.12:** Central planes of the carbon soot reconstruction using (a) WBP, SIRT with (b) 10 (c) 30 (d) 50 iterations. (e) Volume rendered visualization of the reconstruction. The data cube was partially cut cylindrically to highlight features inside.

### 4.4.3 Other Reconstruction Techniques

Resolution measurements from test images indicate that the simultaneous algebraic reconstruction technique (SART) can give even better results than SIRT [118]. In SART, a linear system of equations is solved as  $Ax = b$ , where  $x$  is the object,  $b$  is the projections and  $A$  is a weighting matrix. SART starts often with a WBP reconstruction initially and iterates until convergence. SART is fairly vulnerable to noise in the images, and hence has not been used much. [119] [120]

In the cases when there are detailed knowledge of the sample for example in pure porous materials, where it is known that the sample is homogenous and consists of one material only, discrete tomography is often used. For these kinds of binary materials the SART equation can be modified and constrained such that the  $x$  variable consists of zeros and ones only. Discrete tomography (DART) has the huge advantage, that the volume is segmented automatically. The equations can be generalized to consist of any discrete number of materials but has the same disadvantages as SART does. Much attention has also been given to compressed sensing reconstruction technique, which is based on the assumption that the object  $x$  or its derivative is sparse consisting mainly of zeros. The algorithm tries to minimize the linear system of equations coupled to the sparse objective function also known as total variation minimization. The advantage is that no discrete assumptions of the material have to be made since the algorithm also supports linear density variations within the object  $x$ . [84] [112] [121] [122]

## 4.5 Resolution

The spatial resolution of tomograms is anisotropic in Cartesian coordinates. In the direction parallel to the tilt axis, the tomogram resolution  $r_y$  is the same as in the original projection images. However, for the direction normal to it, the resolution is defined via Crowther's criterion

$$r_x = \frac{\pi D}{N}, \quad (20)$$

with  $D$  being the diameter of the reconstructed volume, and  $N$  is the number of projections used and a maximum tilt range of  $\pm 90^\circ$  is assumed. A spherical reconstruction volume is considered and (20) hence gives a conservative estimation of the resolution if a thin section is reconstructed. Often in ET an extended slab geometry is used with thickness  $T$ , comprising as thin section. For this geometry,  $T = D \cos \alpha$ , where  $\alpha$  is the maximum tilt angle – can be inserted into (20) thus improving the resolution. [72]

For the  $z$ -axis, the resolution is further affected by the elongation factor

$$e = \sqrt{\frac{\alpha + \sin \alpha \cos \alpha}{\alpha - \sin \alpha \cos \alpha}} ; \quad r_z = r_x e, \quad (21)$$

where  $\alpha$  is the maximum tilt angle [123]. The elongation factor causes elongation in the  $z$ -direction and is a function of the maximum tilt angle  $\alpha$  only. The elongation is due to what is known as the missing wedge problem, since in the Fourier space, the limited tilt angles cause a wedge shaped

volume of missing information (figure 4.8(a)) and can be understood by the FST. The elongation is caused by the convolution of the point spread function of the reconstruction. In dual axis tomography, two tilt series are acquired in sequence by rotating the sample in-plane 90° degrees and the reconstructions are later averaged, reducing the problem to a missing pyramid. Most effective sampling in Fourier space is achieved with a conical tilting geometry, where the sample is first tilted and then rotated 360° degrees in-plane, while acquiring images at constant rotation steps [124]. This procedure necessitates a special sample holder. Since the essence of tomography is to recover the information in the projection dimension, the elongation factor has to be minimized by maximizing the tilt range. Within this thesis mainly FIB prepared samples were used. The pillar and lamella shaped structures attached to the TEM grids were optimized to maximize the tilt range for single axis tomography. [125]

#### 4.5.1 Fourier Shell Correlation

Fourier shell correlation (FSC) [126] and spectral signal to noise ratio (SSNR) [127] are commonly used techniques to describe the attainable resolution in single particle cryo tomography of viruses. Although rarely used in ET, here we apply them to measure resolution of the tomograms.

In FSC, the tilt series is divided in two parts with odd and even tilt angles. These two tilt series are reconstructed separately and the FT's of the reconstructions are related as

$$FSC(r) = \frac{\sum_{r_i \in r} F_e(r_i) \cdot F_o(r_i)^*}{\sqrt{\sum_{r_i \in r} |F_e(r_i)|^2 \cdot \sum_{r_i \in r} |F_o(r_i)|^2}} \quad (22)$$

where  $F_e$  is the FT of the reconstruction from the even tilt series and  $F_o$  from the odd tilt series. The FSC is a measure of correlation between the 3D shells in Fourier space of the two reconstructions with constant radius, where  $r_i$  is the radial step size within the shell  $r$ . This equation takes two 3D complex volumes and transforms them into real 1D normalized cross-correlation coefficients as a function of radius/frequency away from the DC point. The FSC value of 0.5 is a common agreement for the actual resolution within the reconstruction. [12] [126]

#### 4.5.2 Spectral Signal to Noise Ratio

The SSNR [127] takes account for the reconstruction process as well and the noise generated therein. Prior to the resolution assessment, a reconstruction has to be made and a re-projected tilt series is then generated from this reconstruction. The input SSNR (ISSNR) is defined as

$$ISSNR = \frac{\sum_{j=1}^J \sum_{r_i \in r} |X_r^j|^2}{\sum_{j=1}^J \sum_{r_i \in r} |X_r^j - X_r'^j|^2}, \quad (23)$$

where  $J$  is the total number of projection images in the tilt series,  $X^j$  is the FT of the projection image  $j$  and  $X^{j'}$  is the FT of the re-projection image  $j$ . Here the  $r_j$  is the 2D radial step size within the ring  $r$  of each FT's and the ISSNR is summed over all tilt angles.

To include the effect of the reconstruction process, a new image has to be created consisting of white noise only, having the same dimensions as the tilt series. Proceeding with similar steps as above the noise reduction factor is defined as

$$\alpha = \frac{\sum_{j=1}^J \sum_{r_i \in r} |N_r^j|^2}{\sum_{j=1}^J \sum_{r_i \in r} |N_r^{j'}|^2}, \quad (24)$$

where  $N^j$  is the FT of the noise image  $j$  and  $N^{j'}$  is the FT of the re-projection noise image  $j$ . Then the SSNR becomes

$$SSNR = \max \left\{ 0, \frac{ISSNR}{\alpha} - 1 \right\}. \quad (25)$$

## 4.6 Segmentation and Visualization

For any quantitative analysis, the reconstruction has to be segmented. That is, to separate the objects of interests from the rest of the data based on chemical or structural similarity. Segmentation is usually an underestimated process in the 3D analysis pipe line yet its impact to the results is overwhelming. The data coming from ET is not clear enough for simple global threshold segmentation, mainly because of the missing wedge effect, low contrast and noise of the acquired images. Manual segmentation by outlining the objects of interest is slow and highly subjective but commonly done and justified with the educated guess of the scientist. The need for objective and automated segmentation routines for the many individual instances is a great one. The implementation of the ITK routines within DM served this purpose as well.

Filtering the tomograms prior to segmentation with ET data is a necessity. General smoothing operations like Gaussian smoothing are not applicable since it also smooths the already fuzzy edge intensities making segmentation ambiguous. To smooth out the small intensity differences in the data originating from noise and low signal to noise ratio usually anisotropic edge preserving smoothing or non-local means filter are used. In the former, the strength of the smoothing parameter, i.e. the diffusion coefficient depends on the local gradient direction of the data and diffusion takes place on voxels within a similar region rather than across boundaries, thus smoothing the data while preserving the edges [128]. The latter assumes self-similarity in the image and uses a search window to correlate the target voxel's similarity to all voxels in the image, which is then used as a weighting function to be multiplied with the target voxel [129].

The segmentation routines can be roughly separated to global and local ones. The simplest one is the global threshold as was used in both of the CoFe<sub>2</sub>O<sub>4</sub> samples, where only the intensities below (cTEM) or above (STEM HAADF) a user given threshold are retained. More robust methods have been developed that couple the global and local analysis, for example connected threshold segmentation where affinity between voxels is exploited e.g. proximity and grey level similarity [130] and in conjunction with eigenvector analysis [131]. Often used method to separate agglomerated objects is the watershed transform that treats intensities as a topographic map and

connects objects into volumes above a certain level set [132]. In combination with the distance map filter, that calculates the Euclidean distance to an edge, this procedure can separate agglomerated objects or particles.

The 3D visualization of data can be divided into three distinctive methods: 2D slicing, voxel projection and surface rendering. The first one, as seen in figure 4.12, consists of cutting a 2D slice through the 3D data stack to highlight some special feature in it. The contrast is however a result from the density distribution contained only within that plane and not from accumulated projection intensities. The voxel projection (figure 4.7) follows the same principles as observing a transparent object in a light microscope. The volume rendering is commonly a summed projection of the intensities within a ray, but also maximum voxel values can be used. The projected rays can be parallel to each other or emerging from a single point depending how the perspective is chosen. The amount of transparency, color or luminosity can be set to the voxels with various dependencies on the intensity levels. This makes it possible to visualize the data in full 3D and easily rotate and zoom in on various details. It is particularly useful to detect subtle intensity changes within the data. In surface or iso-surface rendering a single threshold intensity value is chosen as a topographic map to create a triangulated surface containing the lower intensity voxels. For raw ET data this is not useful however since of the noise and low contrast; additionally any intensity variations are impossible to detect. However for readily segmented data consisting of integer values assigned to different materials or objects, this method is quite powerful in speed and robustness as well as in visualization clarity as depicted in figure 5.5 in the next chapter. Additionally, triangulation opens up possibilities for straightforward analysis of any surface or topology related investigations e.g. curvature. [133]

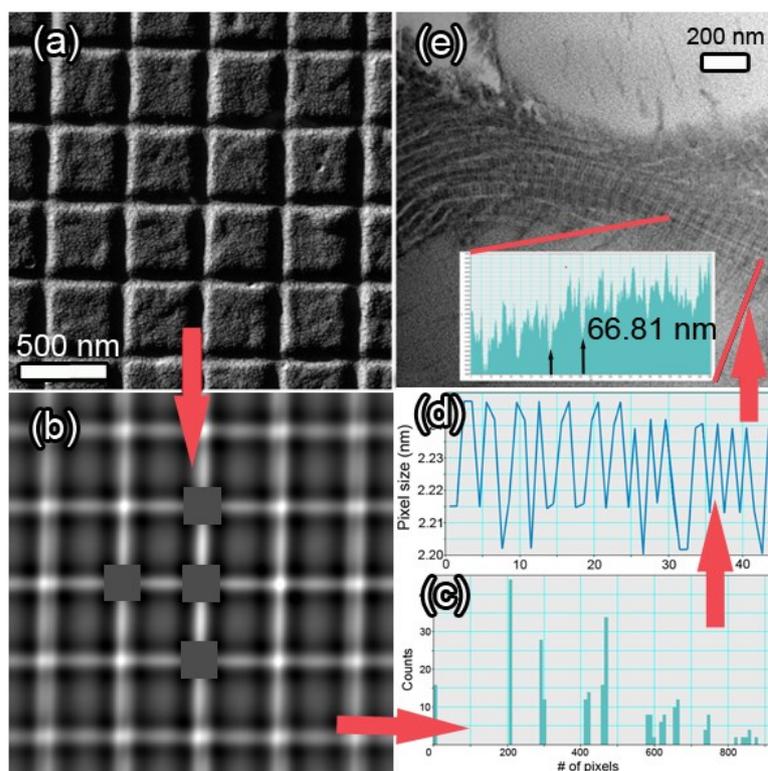
## 5 Results

### 5.1 Magnification Calibration Using Auto-Correlation

The magnification calibration of the microscope was done using the common cross grating sample (figure 5.1(a)): shadow casted carbon replicas, where 2160 lines per mm are ruled 90° degrees to one another forming squares with a  $L = 462.9$  nm edge length. Rather than performing a subjective and manual line to line measurement, a script was written in DM environment to measure the average value of all the edge distances of the cross grating image. A power law curve was then fitted to the mean values measured with different magnifications.

The script first filters the image with a weak hanning window to smooth the intensities to zero in the edge: a common cause for problems in the Fourier domain. An auto-correlation of the image is calculated and the script then finds the maxima of the image within a sub-pixel resolution, stores the position and makes the point and the surroundings zero (figure 5.1(b)) to search for the next maxima. Every point's distance to all other points are calculated and stored in a matrix that becomes symmetric and diagonally zero. This is because every point's distance is calculated twice and the latter is the distance to the point itself. The histogram (figure 5.1(c)) of this matrix gives peaks to corresponding distances  $L$ ,  $L\sqrt{2}$ ,  $2L$ , etc. A threshold is applied near the first peak and the user has a chance to cut out distances that are too large to take into the calculation of the average value. This calculation is then repeated to gross grating images taken with different magnifications and brought to Matlab [134] environment, where a power law curve is fitted to these pixel size points with different magnifications.

As a proof of concept, the d-spacing of collagen fibrils, taken from a high pressure frozen sample of a mouse aorta is presented in figure 5.1(e). The d-spacing is always found to be 67 nm [135] and in the sub-image of (e), it is found to be 66.8 nm instead of 65.2 nm with the old calibrated pixel size of the microscope at this magnification.



**Figure 5.1:** (a) STEM HAADF image of the cross grating sample at 56 kX magnification. (b) Auto-correlation image of (a), where some peak surroundings are cleared. The point locations are stored and all distances are then calculated to yield the histogram image (c). The points in the first peak after zero are then extracted and the final pixel sizes are presented at (d), which are averaged to get the final pixel size. (e) TEM image of collagen fibrils. As a proof of concept, the 67 nm d-spacing of the fibrils is presented.

## 5.2 New Alignment Methods

### 5.2.1 Manual Projection Based Alignment

A custom alignment program was written within the IDL environment [136], where three different planes are reconstructed as seen in figure 5.2. Within each reconstruction, the line projections of the image under alignment are enhanced by multiplication and filtering such that the user can translate, rotate and change the tilt angle of the image to guide the enhanced projections to the objects in the reconstruction where they originate from. Each image in the tilt series has to be aligned manually and iteratively until a visually acceptable reconstruction quality is achieved. The copper interconnect samples characterized in this thesis were aligned using this procedure.



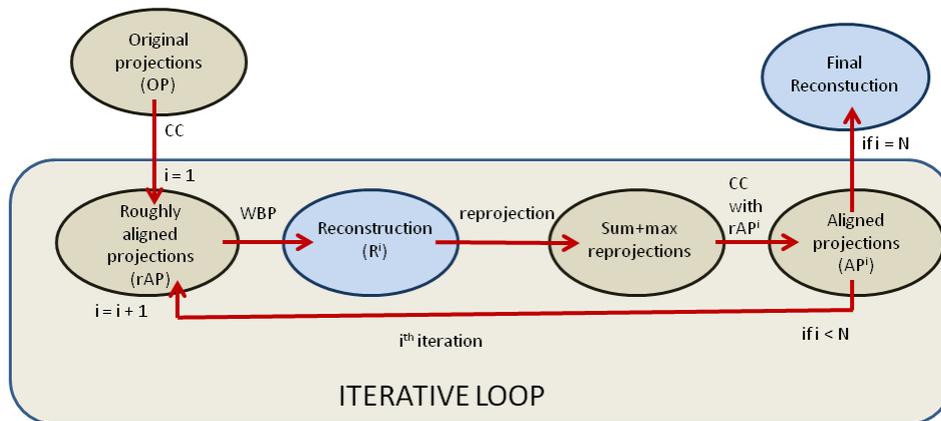
**Figure 5.2:** Screenshot of the alignment program written in IDL environment. On top left, three different 2D reconstructions are made and the user can translate the enhanced line projections of the tilt image under alignment. Here one sees clearly the black vertical lines originating from the tantalum barriers. On top right there is the image under alignment, where the red lines show the position of the reconstructed planes. On right below is the difference image between the subsequent tilt images for guidance.

## 5.2.2 Projection Re-projection Comparison Alignment

This section is partly reproduced from the publication [2] and written by the author of the thesis. A new kind of marker-less, iterative and semi-automatic alignment method for the tilt series was designed. The idea is to compare the original, roughly aligned tilt series to a re-projected one, obtained from an initial reconstruction. However, this would not have been possible without the integration of an open source public library of image manipulation algorithms called Insight ToolKit (ITK) [137]. Various algorithms in the field of image manipulation, filters, segmentation, registration and deconvolution were implemented from the library to DM environment. More information can be found in appendix 7.3.

For the projection re-projection comparison (PRC) method one first roughly aligns the original tilt series, by CC and/or using the center of mass –alignment, however both lacking precision for a high quality reconstruction. After the rough alignment with CC, a WBP reconstruction is made (it is worthwhile to mention, that SIRT should not be used here, because of the nature of the algorithm, the possible misalignments are inherently included and enhanced in the reconstruction such that all re-projections only produce the same image as in the tilt series). Then, another tilt series is made

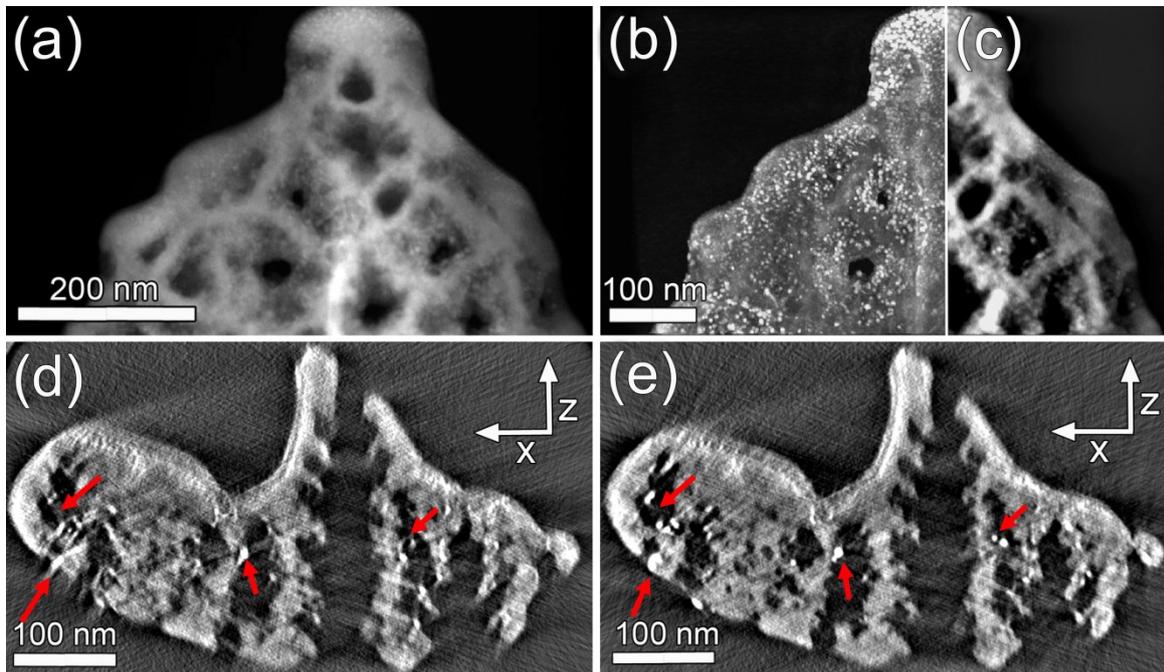
by re-projection from the initial reconstruction. The re-projection should not be a regular integral projection; it was crucial to implement also a maximum projection, where the intensity of the projected ray is not accumulated with the pixels intensities on that ray, but merely the maximum pixel intensity within the ray. For example, the accumulated projected intensity from pixels having values 3, 7 and 4 is 14 and the maximum projected intensity is 7. This makes the high contrast/ high intensity objects, in the reconstruction acquired using the STEM-HAADF mode, clearly visible as illustrated in figure 5.4(b-c). Also minimum projection is available for cTEM BF tilt series. The re-projected tilt series is a weighted sum of the integral re-projections and the maximum re-projections, which has to be set accordingly. In figure 5.4  $I_{tot} = 30I_m + I_a$  was used, where  $I_m$  is the maximum projection image and  $I_a$  the accumulated one. The original tilt series is then compared again to the re-projected one using CC or by finding the maximum mutual information by exhaustive search on some small interval; that is, creating a loop with a small step size going through minimum and maximum displacement both in x and y directions. After finding the misalignments, the tilt series is again translated and the tilt axis shift and rotation is visually checked. A new reconstruction is then made for the next iteration. The complete alignment cycle is illustrated in figure 5.3.



**Figure 5.3:** Illustration of the PRC alignment framework. CC can also be accomplished by maximum mutual information exhaustive search. Reconstructions are colored as blue and tilt series as light brown. N is the total number of iterations and i is the current iteration.

In figure 5.4(a) one sees the zero tilt image from the acquired tilt series using the pSi sample infiltrated with magnetite nanoparticles. In 5.4(b) and 5.4(c) are the maximum re-projection and integral re-projection respectively. The latter image picks up the general shape of the structure and the former the high intensity details necessary for a good comparison. A weighted sum (30:1) of these images was used in the CC procedure with the image (a). After two iterations of CC, followed by a center of mass –alignment, the reconstruction (xz-plane) in 5.4(d) was done using SIRT with 20 iterations. Obvious misalignment errors are visible as compared to 5.4(e), which was reconstructed after three iterations with the PRC method using the same parameters. The misalignments during each iteration were saved and summed to be used to the original tilt series such that the subpixel translations were used only once before the final reconstruction. Currently only translation is included but by using the ITK commands it is possible to implement rotation and scaling for example by using the optimization based registration and affine transform [137]. Alternatively, when cross-correlating the polar transforms and log-polar transforms of the images,

the CC peak location gives the difference between rotation and scaling between the images respectively [138]. [2]



**Figure 5.4:** (a) Zero tilt image from the original tilt series and re-projections from a WBP reconstruction, using (b) maximum projection and (c) integral projection. Weighted sum (30:1) of these images was used in the CC with the image (a). (d) Reconstructed xz-plane (SIRT, 20 iterations) after two iterations of the CC and one center of mass –alignment procedure, showing obvious misalignment errors (red arrows) compared to (e), which is reconstructed (SIRT, 20 iterations) after three iterations of the PRC method on the same plane. In the middle, a main pore is shown going through the sample accompanied with some diagonal dendritic ones. [2]

The method described here is based on statistics in a way that the individual tilt images' projections are averaged in the reconstruction and the correct alignment is an asymptotic process through the iterations. It is a semi-automatic process as it requires a visual check of the tilt axis rotation and shift. (This is however possible to as is done in the TOMATO alignment program [139].) Reconstructing and re-projecting data through numerous iterations is time consuming (1 iteration  $\approx$  1 hour). Using only part of the reconstruction is one way to improve this. Alternatively the code can be optimized such that the heavy calculations are performed by the graphics card in parallel mode (GPU), which has dramatically accelerated the computing power within the image analysis field [140].

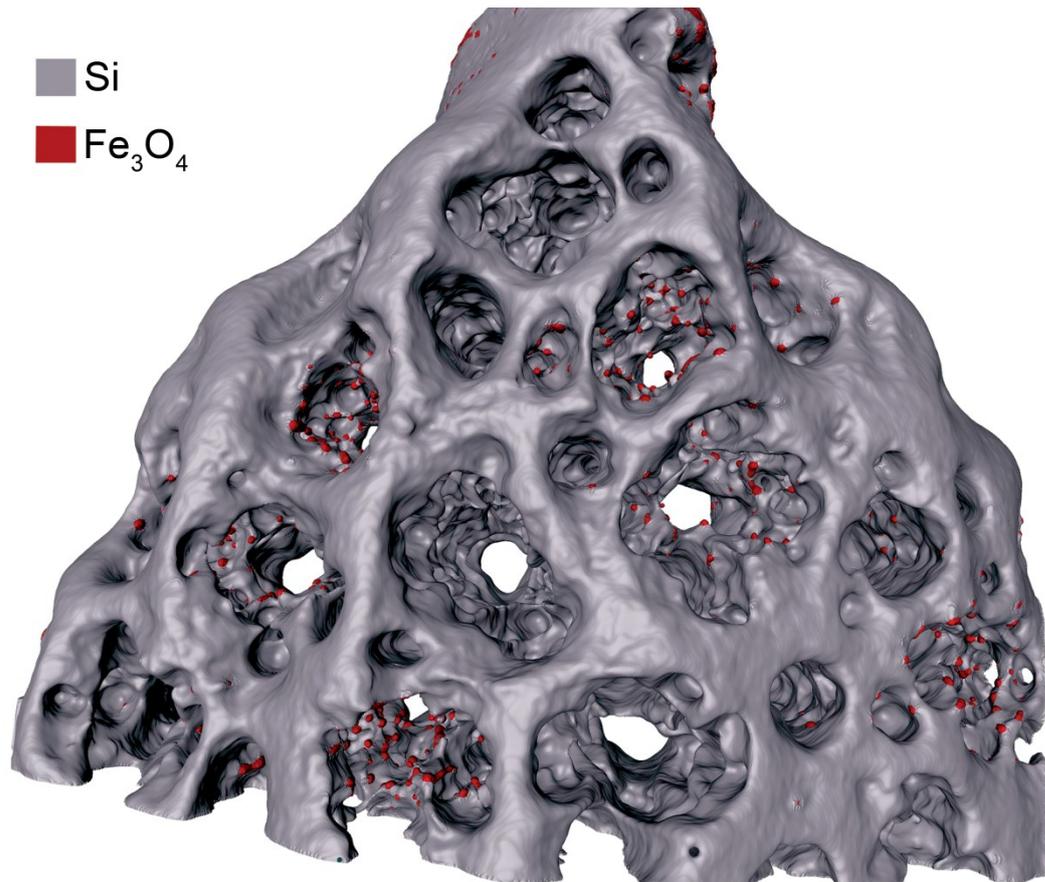
### 5.3 ITK Segmentation

The segmentation routines used within this thesis are introduced using the pSi sample infiltrated with magnetite nanoparticles. After the PRC alignment of the tilt series, a reconstruction was performed using SIRT with 20 iterations. To reduce noise, the reconstruction was filtered with a modified curvature anisotropic diffusion filter [141] and a non-local means filter.

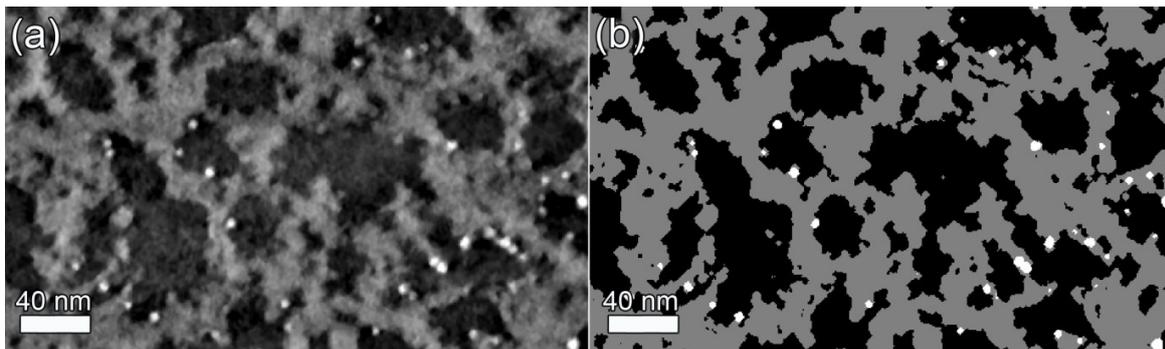
The segmentation of the particles and the template was done separately and with different methods to be combined later. For the segmentation of the magnetite particles, the volume was further filtered using a top hat filter, which takes account for the area of the objects [142]. The filter enhances the relative intensity of the particles such that global segmentation becomes safe. The segmented particles were then processed by the common distance transform and watershed algorithm [143] to label the particles and to separate the few agglomerated ones. The segmented particles still had to pass the restrictions explained in chapter 5.6.1.

A connected threshold algorithm [144] was used in the segmentation of the template, which is a region growing method that includes those voxels that are connected to a seed point and whose intensities fall within the user given upper and lower threshold values. A custom script was written (with the help of Dr. Bernhard Schaffer, Gatan Inc.) in DM to easily depict seed points into the volume and save the point locations as tags in the image to be used by the segmentation algorithm.

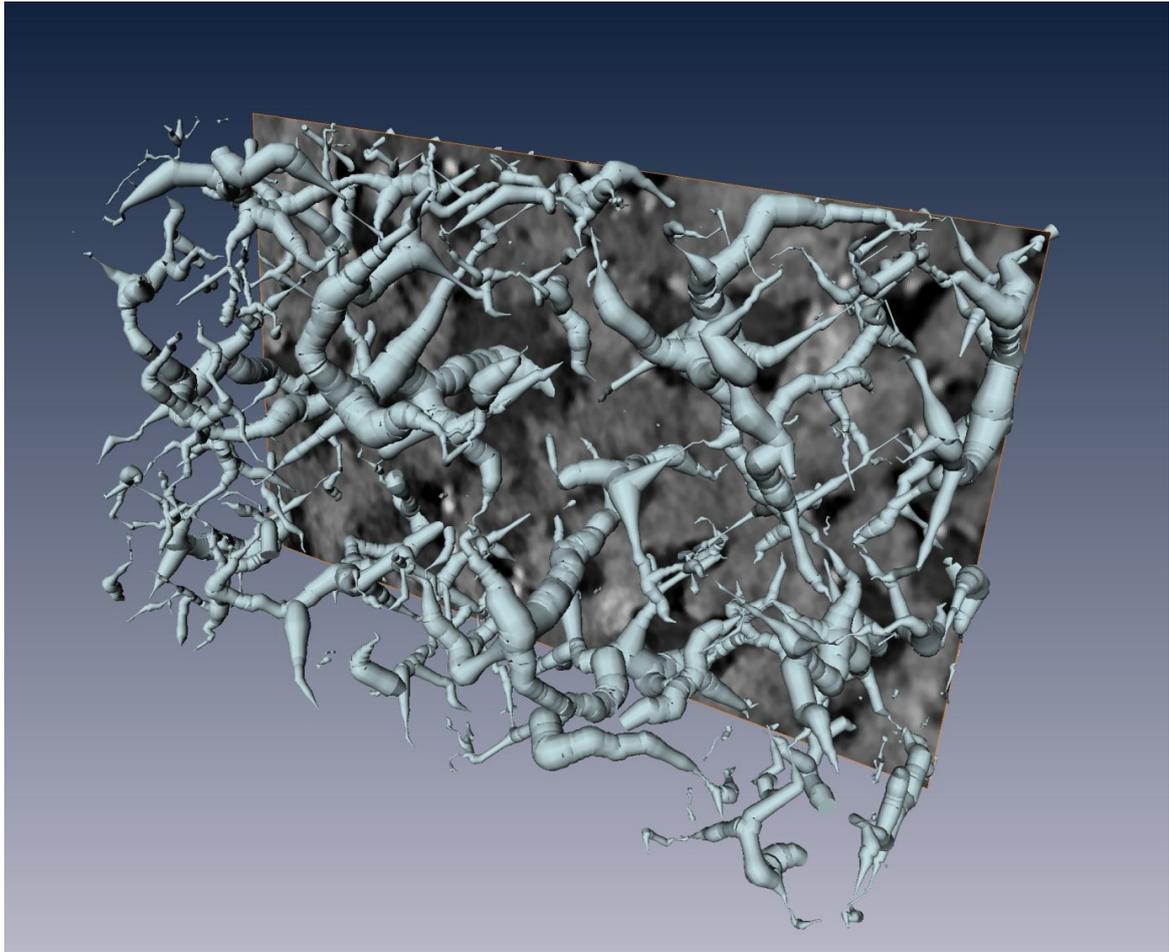
In figure 5.5, one sees the generated surface from the whole segmentation. Only the inner structure was used in the local curvature and Monte Carlo analysis. Figure 5.6(a) is a detail from the central slice (xy-plane) of the reconstruction after non-local mean and modified curvature anisotropic diffusion filtering and 5.7(b) is the segmentation of it. The dendritic growth of the pores in the  $\langle 111 \rangle$  direction of the silicon wafer creates a highly complex network of the pores, making it difficult to separate out the main pores quantitatively. A skeletonization of the data produces a so called centerline tree from the segmentation of the pores only. The Euclidian distance was calculated to the nearest boundary and stored in the graph as a thickness label as seen in figure 5.7. A visual investigation of this graph showed that the main pores investigated here were found to be separated as claimed in [58]. On two pores, there were no magnetite nanoparticles found, which might indicate, that the pores were not filled continuously along the entire pore in the infiltration process.



**Figure 5.5:** Triangulated surface of the whole pSi reconstruction. The outer part of this structure was not used in the calculations. [2]



**Figure 5.6:** (a) A detail of the central slice (xy-plane) of the reconstruction after filtering and (b) the segmentation of it. [2]



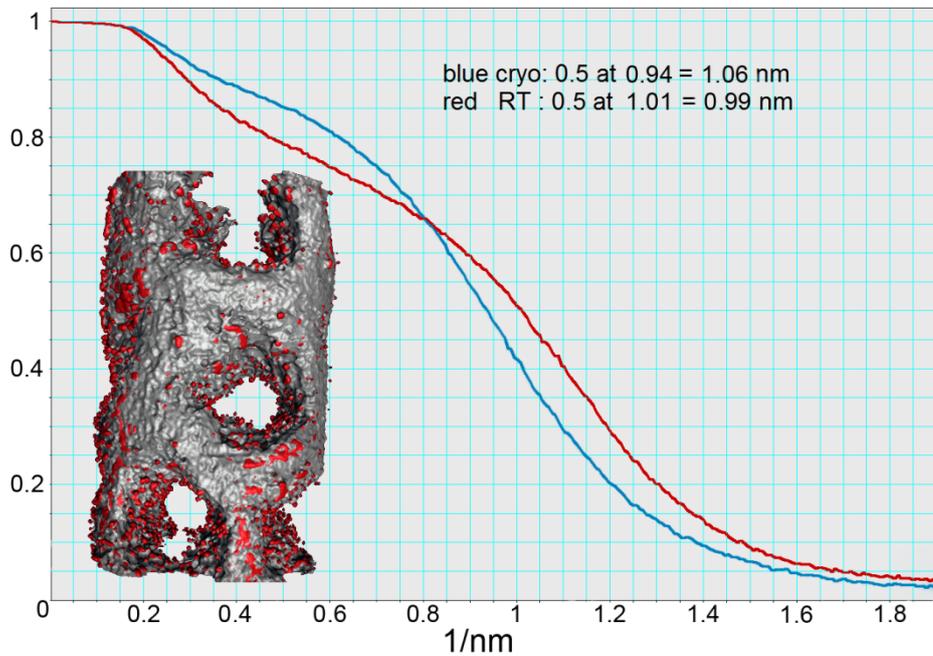
**Figure 5.7:** Centerline tree graph of the pores in the reconstruction. The thick branches show the path of the main pores and the thinner ones the dendritic pores. The main pores were found to be separated from each other. [2]

## 5.4 Resolution Assessment

The FSC and SSNR were implemented as a script in DM environment to measure the resolution of the reconstructions. For the pSi sample, the resolution was compared between samples prepared with a FIB under room temperature and cryo (-160 °C measured at the stage) conditions. For the copper interconnects sample, the resolution was compared between tilt series acquired using HAADF and IBF mode.

The FSC of the reconstructions of samples prepared in room temperature and cryo conditions can be seen in figure 5.8 together with a rendered visualization of the cryo sample with three pores. From a visual inspection of the room temperature reconstruction it was found, that there are numerous particles embedded fully inside the silicon material, whereas in the cryo sample this was not the case. This might indicate that the ion beam in the FIB is re-depositing silicon during the sample preparation process such that the particles are covered within the silicon, since the particles should not be able to infiltrate the crystalline silicon otherwise. Cooling of the sample in cryo

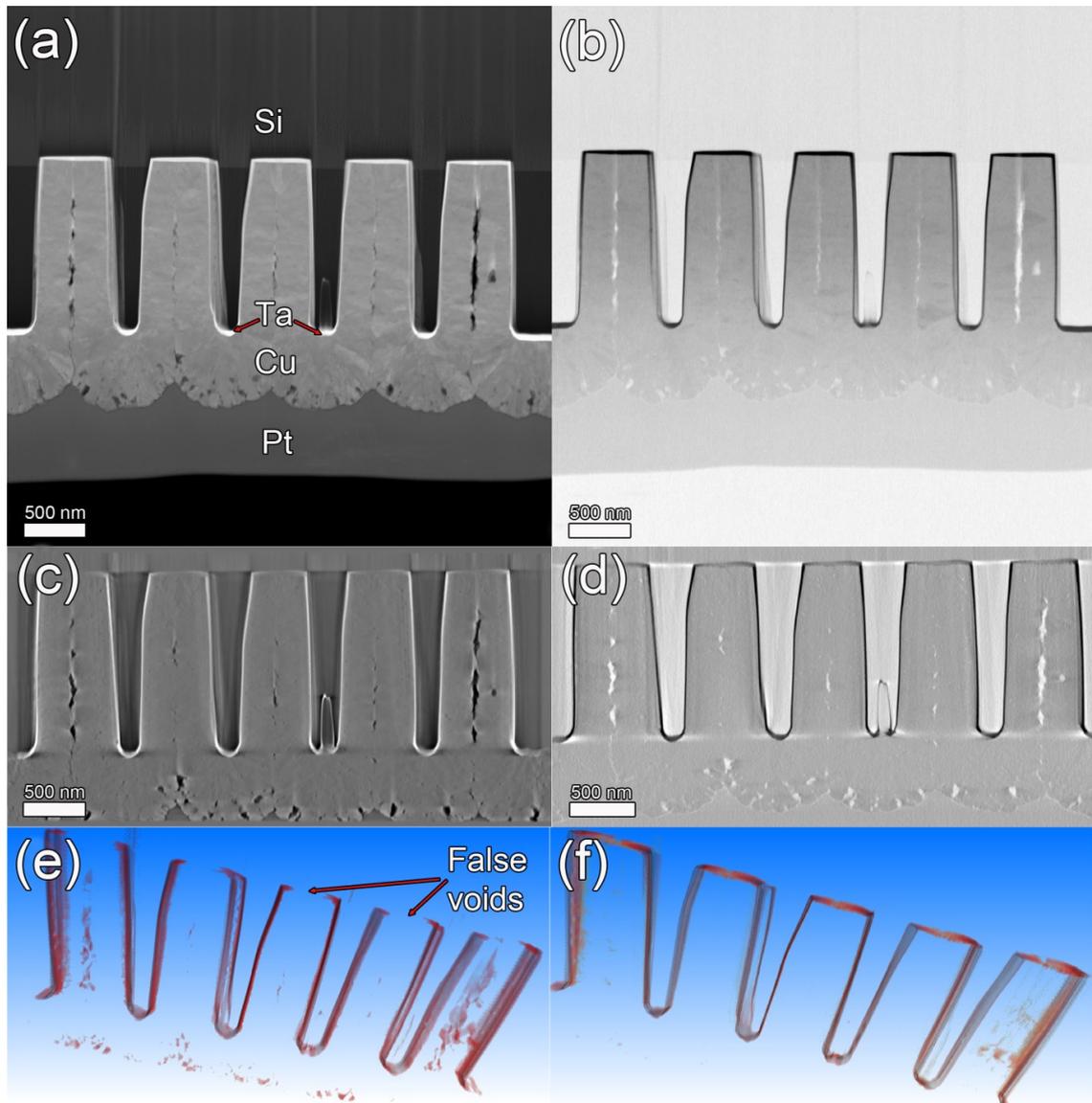
conditions during preparation seems to prevent this. On the other hand, FSC is better in the room temperature sample. The FSC was found to be 1.06 nm and 0.99 nm for cryo and room temperature samples respectively. However, the tilt series were aligned using the manual alignment in IDL environment, which does not offer quantitative value for the alignment quality, which in turn affects the reconstruction fidelity. The areas imaged were additionally different, and the small differences in the FSC curve are not necessarily statistically significant.



**Figure 5.8:** FSC curve for cryo and room temperature prepared samples with a surface rendered visualization of the cryo prepared pSi sample.

The effect of the contrast reversal on the reconstruction was demonstrated using a thick copper interconnects sample consisting of trenches filled with copper and a tantalum diffusion barrier. A visual check on the zero tilt images acquired before and after the tilt series showed no beam damage effects. The tilt series were aligned using the IDL alignment script and the reconstructions were done using SIRT with 30 iterations.

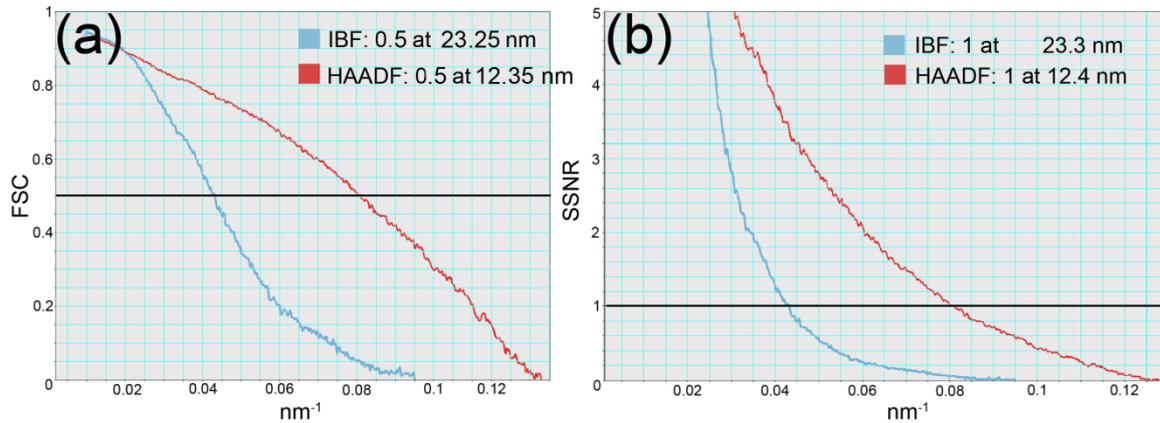
In figure 5.9(a) and (b) one sees the HAADF and IBF zero tilt images of the sample respectively. Starting from below, first layer is a platinum protection layer deposited in the FIB sample preparation process. Above the platinum are the copper trenches showing clear polycrystalline grains and some voids within the lines, which are defects in the filling process. To stop the diffusion of copper within the top silicon material, a protective layer of tantalum was deposited between the copper filling and bulk silicon. The vertical lines in the silicon material are a result from a so called curtain effect coming from the FIB milling. From figure 5.9(c), which is a central plane (xy-plane) of the HAADF reconstruction, one can see the false voids in the horizontal tantalum layer as opposed to 5.9(d), where the contrast is reserved. There are two simultaneous effects occurring: first the contrast reversal at high tilt angles and secondly a reconstruction defect within the SIRT algorithm.



**Figure 5.9:** (a) STEM-HAADF and (b) STEM-IBF image of the copper interconnects sample. (c) Central plane (xy-plane) of the HAADF and (d) IBF reconstruction. (e) A volume rendered visualization of the HAADF and (f) IBF reconstruction, where a threshold has been applied to visualize the tantalum lines. In the latter the contrast has been inverted and the horizontal tantalum lines are conserved as opposed to the HAADF reconstruction, where there are false voids.

The ratio of the tantalum layer and the background in the WBP and SIRT reconstructions computed in the DM environment was 75 and 8 respectively. This was calculated from the line profile across the tantalum layer within the same location and clearly, the contrast in the latter is severely diminished. It is unclear if this is also an effect of the reversed contrast at high tilt angles or this is an inherent problem in the SIRT algorithm. Nevertheless both IBF and HAADF reconstructions were calculated with the same parameters so this does not affect the comparison. Performing a global threshold and a volume rendered visualization to bring forth the tantalum layer

in both reconstructions clearly shows the false voids in figure 5.9(e), which is a HAADF reconstruction as opposed to 5.9(f), an IBF reconstruction.

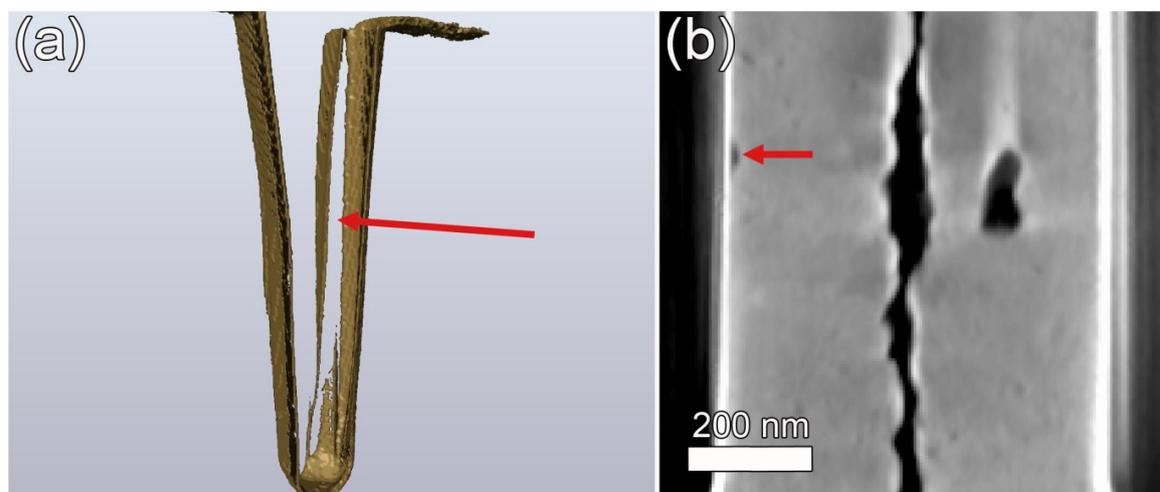


**Figure 5.10:** (a) FSC and (b) SSNR of the IBF and HAADF reconstructions from the copper interconnect sample. The data was 2 times down sampled.

The resolution between the IBF and HAADF mode was determined using FSC and SSNR. As seen from figure 5.10 the resolution within the HAADF reconstruction is almost twice as better than of the former. Because of memory restrictions within the DM, the reconstructions were 2 times down sampled before the SSNR and FSC assessment. From the HAADF projection image, the grain structure (Bragg reflections) of copper crystals can be resolved more clearly, which gives better SSNR and FSC at higher frequencies. These resolution measures do not necessarily give an objective resolution value for two different modalities of imaging, since the amount of features may differ at different frequencies. The SSNR in the IBF tilt series is inherently lower as it is collecting the central beam, that gives large noise component but yet it is monotonic at high tilt angles and with thick materials giving directly interpretable reconstructions. As investigated in [17], the IBF-STEM mode requires high quality pre-amplifier for the HAADF detector and the authors had bought a custom one to adequately detect the high bandwidth signal related to the IBF mode in order to differentiate the contrast between materials. Within this thesis, a pre-amplifier that came with the HAADF detector model 3000 was used, which could in part explain the low SSNR and FSC of the IBF mode. Monte Carlo simulations were used by the authors to define a cutoff thickness for materials where the IBF mode becomes more advantageous as compared to HAADF mode. For copper it was found to be 305 nm and for tantalum 90 nm (200 keV). A thickness of 156 nm was measured from the HAADF reconstruction; way beyond the cutoff thickness of tantalum, which fits well with the results since contrast reversal in HAADF reconstruction for tantalum occurred but not for copper. However for example for tilt angle of 70° degrees, the thickness of the sample in projection dimension is 456 nm, which far exceeds the cutoff thickness for copper, so one would expect some contrast reversal occurring for copper also. Apparently the high intensities at lower angles will average this out in the reconstruction yet lowering the contrast.

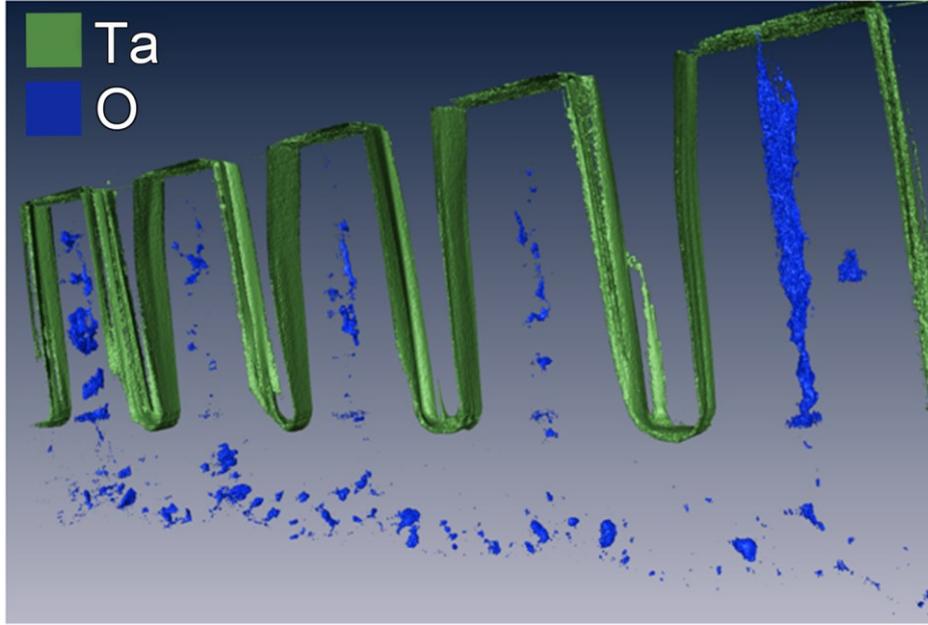
A factor of 2 is a huge difference between the measured resolutions of the IBF and HAADF mode, although as explained above, this might stem from the polycrystalline structure of copper. In the SSNR the acquired images are used directly by comparing them to the re-projected ones and so the polycrystals are more clearly visible and influential to the result. But in contrast, the FSC uses the reconstructions only, where the quasi-random Bragg reflections should be, at least to some

extent, averaged out by the reconstruction process. Nevertheless they both give equal results. The quandary remains, what is the important information to be acquired from the tomogram, direct interpretation or resolution? The choice between IBF and HAADF should be based on this question.



**Figure 5.11:** (a) Surface rendered visualization of one trench of the tantalum barrier liner showing an abrupt discontinuity (red arrow) in the liner. (b) Xy-plane of the HAADF reconstruction showing adhesion flaw (red arrow) between the copper and the tantalum. A vertical void from the preparation process is also shown.

The volume of the materials was calculated from the trenches only using the IBF reconstruction because the tantalum contrast was preserved. The volume of the voids, tantalum and copper was calculated to be  $0.009 \mu\text{m}^3$ ,  $0.063 \mu\text{m}^3$  and  $0.789 \mu\text{m}^3$  respectively. The unevenness in the tantalum diffusion barrier is seen in figure 5.11(a), where there is an abrupt discontinuity and a hole in the tantalum liner. The adhesion of copper with tantalum was found to be good, except a few minor voids as seen from figure 5.11(b), which is a detail from the xy-plane image from the HAADF reconstruction. Figure 5.12 shows the surface rendered tantalum (green) and the oxygen voids (blue). The void formation in the preparation process is clearly more severe at the edge trenches as compared to those in the middle; so the process of superfilling is not taking place. Additionally the surface roughness was always smaller on one side of the trenches (left side in figure 5.12).



**Figure 5.12:** Surface rendered visualization of the tantalum barrier (green) and the oxygen voids (blue) from the IBF reconstruction. Copper is transparent.

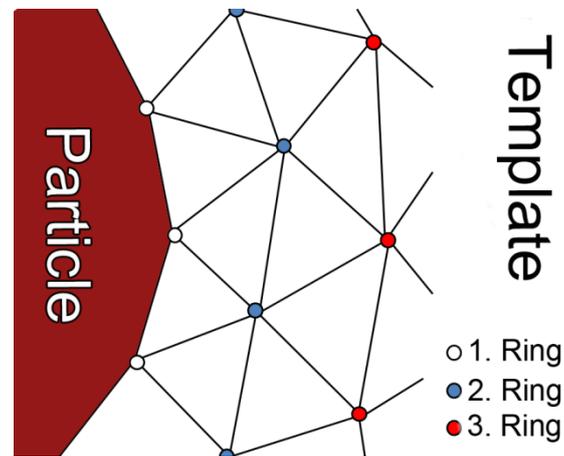
## 5.5 Local Curvature Distribution

This section is partly reproduced from the publication [2] and written by the author of the thesis. To optimize the design of any catalytic material with a supporting template and adsorbed particles e.g. in terms of reactivity, the knowledge of the preferred adsorption sites of the particles are hence an advantage. The most straight-forward method to acquire the local curvature, is to extract the principal curvatures  $\kappa_1$  and  $\kappa_2$  from a triangulated surface generated from a segmentation of a three dimensional reconstruction. There are several free and commercial programs to achieve this, and within this thesis, Amira software [145] environment was used. The principal curvatures  $\kappa_1$  and  $\kappa_2$  are the maximum and minimum value of the normal curvature respectively and are related to the mean and Gaussian curvatures of the surface, which in turn are a functions of the gradient of the surface. The methods and insights provided here are mainly restricted within the digital image manipulation problematics, and are therefore applicable to any composite material consisting of objects lying on a surface [2]. For clear and intuitive interpretation of the curvature, a single valued shape index [146], defined as

$$S = \frac{2}{\pi} \arctan \frac{\kappa_2 + \kappa_1}{\kappa_2 - \kappa_1} \quad (\kappa_1 \geq \kappa_2), (\kappa_1 = 0 \text{ only if } \kappa_2 \neq 0), \quad (26)$$

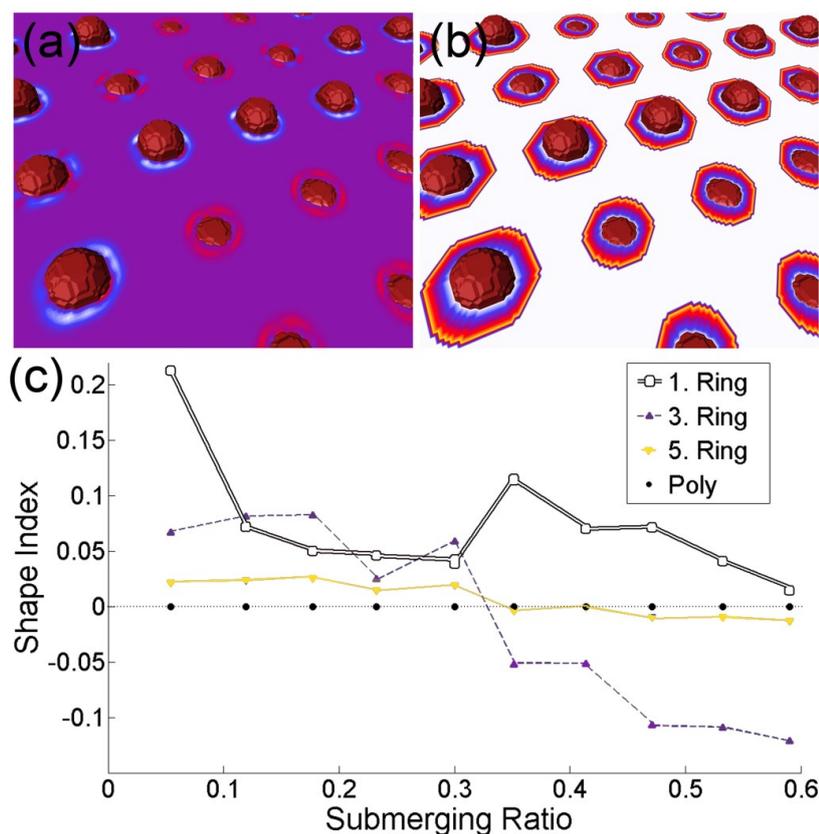
was used to represent the curvature as it omits values  $-1 \leq S \leq 1$ . Here a positive value indicates a concave or a cup like surface and a negative value depicts a convex or cap like surface. Whereas saddle or a near flat surface has a shape index close to zero. The visualization can be performed using a colormap within the Amira program. [2]

Two methods to acquire the local curvature distribution (LCD) are proposed and discussed here. A custom module was written using Visual Studio 2008 [147] and the developer option in the Amira program for both of these methods. The first is called the ring method and consists of rings of points defined around the particle within the template's 3D point coordinates. The LC was calculated as an averaged value of the curvature of the points within each ring and separately for each ring. Within the code, patch indexes, that is the material identification numbers and triangles' neighboring connectivity information was used to define and separate the rings as illustrated in figure 5.13. A similar approach was reported in [148], where a sphere was defined close to the particle and every point's curvature value was averaged that fell within that sphere. However this poses difficulties if there is an adjacent surface close by. Nevertheless the low quality of the triangulation process near sharp edges and in the vicinity of two different materials (template and particle) is a common problem for both of these methods. To show this, a test image (1024x1024x50) was created, where 1000 particles were deposited on a flat template with different submerging ratios  $r_s = A_o / (A_o + A_i)$ , where  $A_o$  is the area of the particle outside the template and  $A_i$  the area inside – summed from the individual triangle areas within the mesh. [2]



**Figure 5.13:** Schematics of the triangulation scheme for the ring method. [2]

In figure 5.14(a) one sees the curvature values within the flat template and in 5.14(b) the visualization of the rings colored according to the ring value. In the former the curvature around the particles is not zero and is seen to depend on the submerging ratio. If  $r_s$  is close to 1, then the curvature is negative (=blue, white) and positive (= red) for low  $r_s$  instead of zero (= purple). A flat surface was chosen here for simplicity even though the shape index is not explicitly defined. The exact number of points depends on the chosen mesh quality, on the size of the particles themselves and the submerging ratio, and differed consequently within the sample. The mesh quality was determined by a minimum edge length value, which was chosen to be non-limiting. In the test image (and in the reconstruction), the rings were separated approximately one pixel apart. The distance of the rings to the particle surface was hence directly correlated with the voxel size of the volume such that, within the pSi reconstruction, the first ring was touching the particle (branching points), the second ring was one pixel apart, etc., amounting to 0.76 nm each. [2]



**Figure 5.14:** (a) Particles with different submerging ratios, showing negative (blue, white) or positive (red) local curvature instead of zero (purple). (b) Visualization of the rings colored according to the ring index. (c) Shape index of the test image with respect to the submerging ratio. The first ring was always positive with the ring method (white squares). Then positive or negative depending on the submerging ratio. Not until the 5<sup>th</sup> ring, the curvature was zero within 0.05 resolution (yellow triangles). The second method using polynomial fitting was always zero (black circles, see text below). [2]

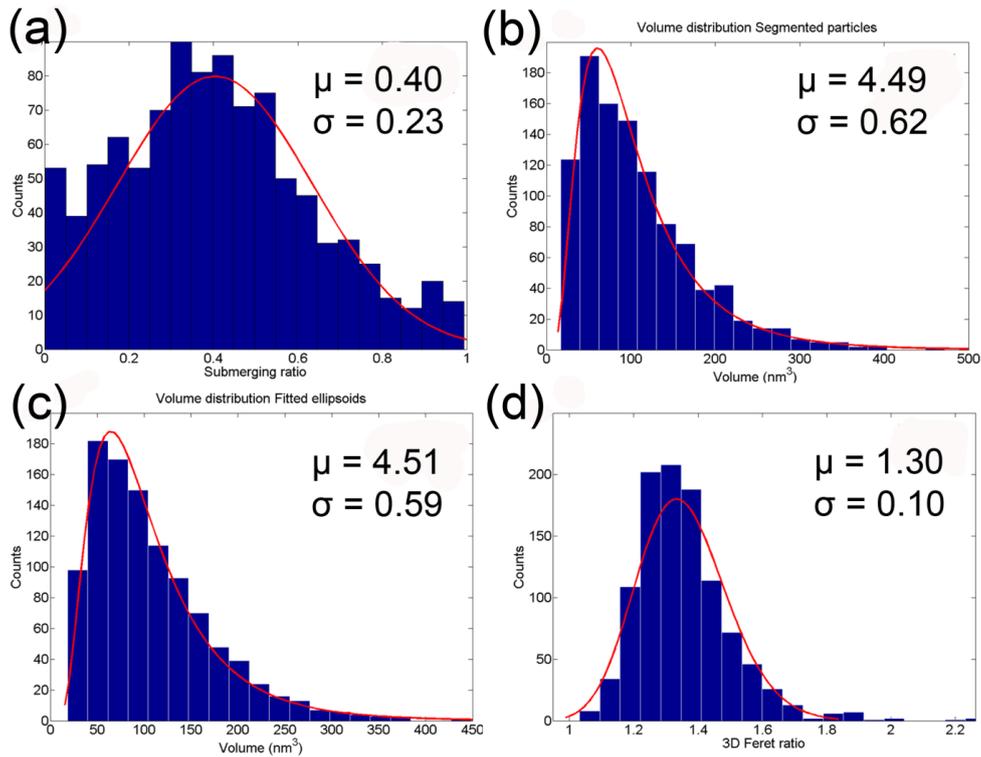
The mean value for the number of points in rings 1, 3 and 5 can be exactly given and was 38.5, 50.5 and 55 respectively. Figure 5.14(c) shows the relation of the shape index to the  $r_s$  for different ring numbers. The shape index flips from negative to positive around the  $r_s$  value of 0.33. Additionally the shape index of the first ring (white squares) around the particle was always found to be positive regardless of the real curvature and not until the 5<sup>th</sup> ring (yellow triangles) the shape index was zero within 0.05 resolution. For this reason the curvature values were calculated separately for each ring so that the first false rings could be excluded and the real curvature could be extracted further away; naturally however with the expense of locality. [2]

The histogram of the submerging ratio of the magnetite particles within the pSi template is shown in figure 5.15(a). Potential particles that were fully submerged or fully separated from the template as a result of segmentation errors were not included in the plot. The Gaussian fit gives a mean value of 0.4 with a standard deviation of 0.23. There exist two possibilities for such a low value: First a segmentation error, which is caused by the intensity spreading around the magnetite particles causing difficulties to fully separate the particles from the template because of similar intensity values. Secondly silicon re-deposition could submerge the particles within the silicon material during the sample preparation process in the FIB. Both of these possibilities will affect the

end results for LCD. The re-deposition of silicon or any material in the FIB sample preparation process is a difficult problem to solve and calls for another thesis in the field of the FIB technique coupled with analytical high-resolution TEM investigations. As discussed earlier, cooling of the FIB sample stage to cryo conditions will help to minimize this effect. [2]

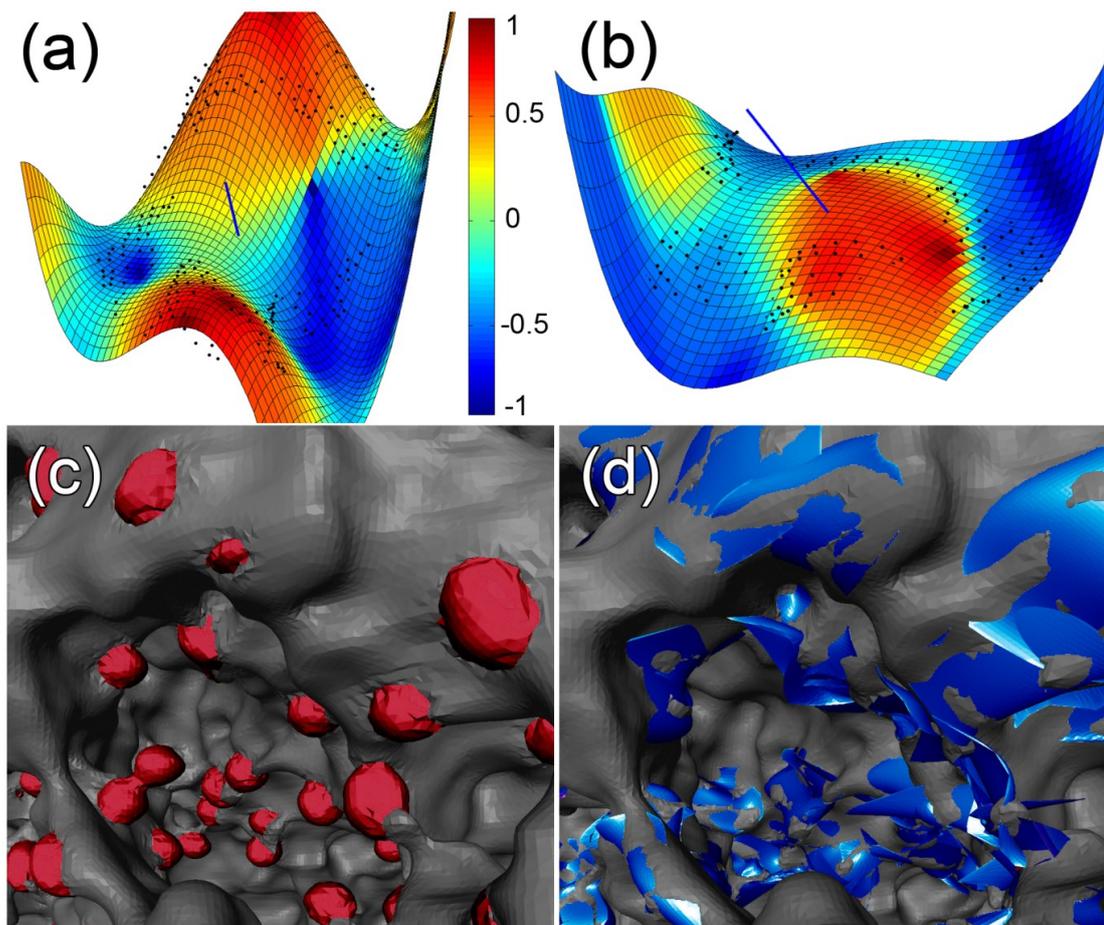
The volume distribution calculated from the segmented particles and from the fitted ellipsoids (see next chapter) are given in figure 5.15(b) and (c) respectively. The volume in the former was calculated using a 3D convex hull of the point set and in the latter, the analytical equation for ellipsoid volume was used. The mean volume from the former plot using the log-normal fit was found to be  $108 \text{ nm}^3$  and if assuming spherical particles the mean diameter is then 5.9 nm. However a useful histogram for the eccentricity of the particles is presented in figure 5.15(d), which is a 3D Feret ratio histogram. This was acquired by dividing the longest semi axis of the fitted ellipsoids with the shortest one. The mean value of 1.3 from the Gaussian fit suggests that the particles are not in fact spherical. The elongation from the missing wedge problem could be the cause of this and the problem is discussed in more detail in the following chapters. [2]

The second method is based on polynomial regression [149] to fit an arbitrary degree polynomial surface to the points within the template around the particles, known as the polyfit method. Within this module the 3D point coordinates and normal directions were extracted to a spreadsheet and then transferred to Matlab [134] environment. This method assumes that the template's surface would continue smoothly according to the surrounding surface around the particle, as if the particle were not there at all. The mean of the normal directions was used to ensure a correct sign for the curvature, meaning on which side the particle was. To acquire a reliable fit, it was crucial to implement a minimization routine for an arbitrary 3D rotation to the point sets such that in the Matlab script, the actual polynomial regression was inside a least squares minimization function for an arbitrary rotation. Since the optimization routine did not support integer values, the best polynomial degree was determined independently in a loop, where the degree itself was limited to 4. Higher degrees would naturally give better mean squared errors (MSE) for the distances of the original points to the fitted surface, but may produce unphysical zero crossings in the area of the particle. MSE values of 1.1 and 0.25 were used as threshold values to exclude inaccurate fits and to accept the current fit with the lowest polynomial degree respectively. [2]



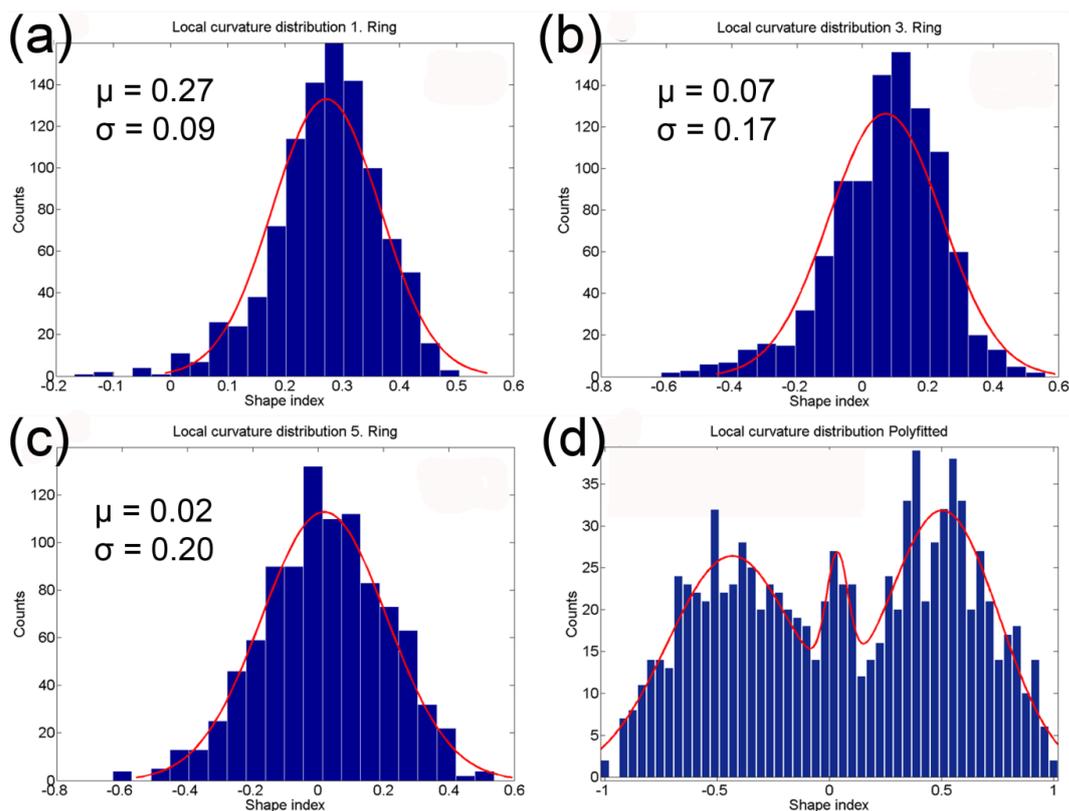
**Figure 5.15:** (a) Submerging ratio of the segmented magnetite particles and the Gaussian fit. (b) Volume distribution obtained from the segmentation using 3D convex hull with a log-normal fit. (c) Volume distribution of the particles obtained using the fitted ellipsoids with a log-normal fit. (d) 3D Ferret ratio of the particles obtained from the fitted ellipsoids by simply dividing the biggest semi radii with the smallest one and the Gaussian fit. [2]

Most of the failed fitting processes took place near the edges of the analyzed volume and was caused by the rough cutting of the outer surface of the template. The obtained symbolic equation for the surface was then used to calculate the shape index from the analytical principal curvatures and the local curvature for each particle was taken as an averaged value of a small patch around the center point of the first ring in the fitted surface. As seen in figure 5.14(c), this method yields zero curvature for each submerging ratio for the test sample. For this calculation, the points in the first three rings were excluded. [2]



**Figure 5.16:** A polynomial surface was fitted to the point sets to predict the local curvature on the position of the particles. Examples of the process are shown in (a - saddle) and (b - concave), where the original points from the pSi template are shown (particles absent) and the fitted surface with coloring indicating the shape index. The blue line is the mean direction normal of the points. In (c) one sees the triangulated mesh of the magnetite particles in one pore, and in (d) the visualization of the results of the polynomial regression. [2]

A visualization of the fitting process is presented in figure 5.16(a) and 5.16(b), where one observes the fitted surface to the original black points. The blue line depicts the mean normal direction of the points and the coloring of the surface indicates the analytical shape index. To ensure the correctness of the polyfit method, the fitted surface points were again brought back to Amira environment, where a module was written to generate a surface from these points on top of the template surface. In figure 5.16(c) is a triangulated surface of one pore only and 5.16(d) is the polyfitted representation of it. On some locations the fitted surface came abruptly out of the mesh, but this happened always far outside the particle area and did not contribute to the LCD results. [2]

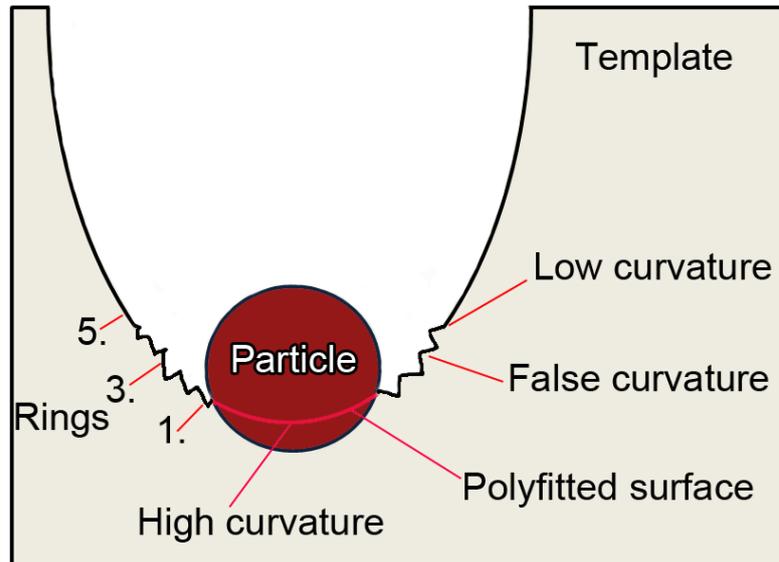


**Figure 5.17:** (a) LCD of the magnetite nanoparticles obtained with the ring method, averaged from the first ring. The vast majority of the values are positive as expected from the test results. A Gaussian distribution curve was fitted yielding a mean value of 0.27 and a standard deviation of 0.09. (b) LCD of the third ring. The mean value of the distribution moved linearly towards the zero value with respect to the ring index as seen in (c), which shows the LCD of the fifth ring. (d) LCD obtained with the polynomial regression showing not one, but two distinct main peaks centered at -0.43 and 0.51, as well as a smaller one at 0.07 according to the Gaussian fits (see text below). [2]

To compare the outcome of these two methods, the LCD for the ring method was extracted separately for each ring. As seen in figure 5.17(a), the fitted Gaussian curve consists of mainly positive values for the first ring, which was expected from the test sample, giving a mean value of 0.27 and a standard deviation of 0.09. Within the subsequent rings (figure 5.17(b)) the mean value of the Gaussian tends to migrate towards zero until the 5<sup>th</sup> ring (figure 5.17(c)) gives a mean value of 0.02 and a standard deviation of 0.2. This suggests that the magnetite nanoparticles embedded within the pSi prefer a near flat or saddle -like surfaces for adsorption and is in accordance to the results as offered in [148]. [2]

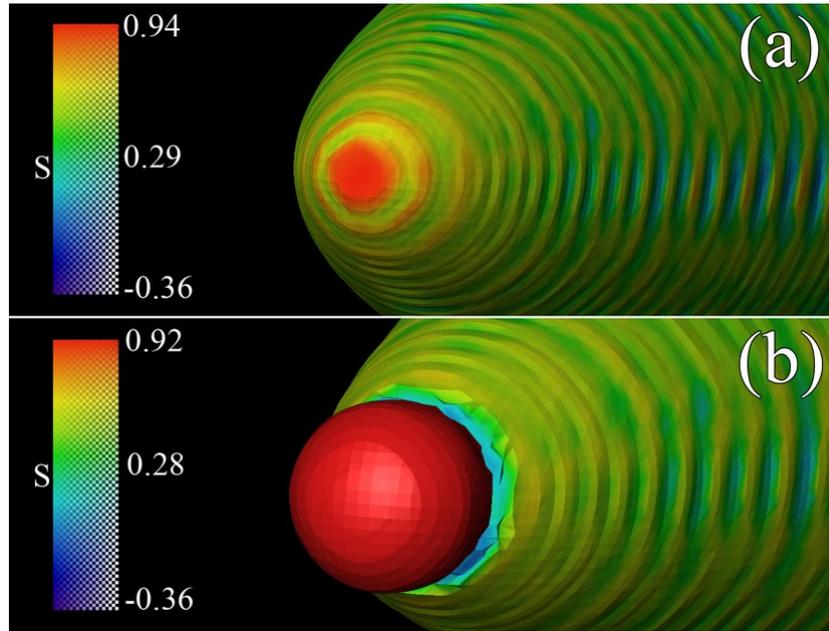
However, the Polyfit method paints a completely different picture of the LCD, where two main peaks at -0.43 and 0.51 are accompanied with a smaller one at 0.07 as seen in figure 5.17(d), suggesting that the particles prefer concave and convex sites for adsorption. And this conforms better to the visual interpretation of the particles' docking sites in average. An explanation to this clear incongruity stems from the way the data is extracted. If the particle's radius is comparable to the radius of curvature of the surface, then the polyfit method picks up the high curvature point where the particle is docked while the ring method due to its' loss of locality picks up a point further away with a curvature value closer to zero as illustrated in figure 5.18. This is because of the

extension of the particle and even more, the distance to the rings used. So the two main peaks in the polyfitted LCD histogram both migrate towards zero to form the Gaussian curve as in figure 5.17(c) simply because of the geometrical data extraction procedures. [2]



**Figure 5.18:** Illustration of the geometrical difference on the local curvature extraction between the polyfit and the ring method. The polyfit method analyzes the surface at the high curvature location. The ring method gives false results at low ring values until in ring number 5, the curvature values are lower. [2]

In order to prove that the polyfit method can accurately fit high curvature locations, a test image was created as seen in figure 5.19. There, a particle was situated on a location with a shape index of 0.94 (Amira). With the ring method, strongly deviating results of 0.35, 0.4 and 0.5 were obtained on the first, third and fifth ring respectively. In contrast to that, the polyfit method yielded a value of 0.99 using the mean coordinate of the fitted points. Alternatively, when taking the mean shape index of a small patch of points (as was done in the pSi sample), i.e. points that are within a Euclidian distance of 0.5 to the mean value of the fitted point coordinates, a value of 0.96 was achieved. It is then reasonable to conclude that the polyfit method gives more intuitive and truthful results than the ring method. Polynomial regression however cannot fit arbitrary surfaces that occur in real situations, hence MSE threshold values should be used carefully and the results should be always visually certified. Possible zero crossings occurring with higher polynomial degrees could be crosschecked for example using the gradient of the surface inside the first ring. The ring method might suffice if the particle radius is much smaller than the average radius of curvature of the surface. [2]

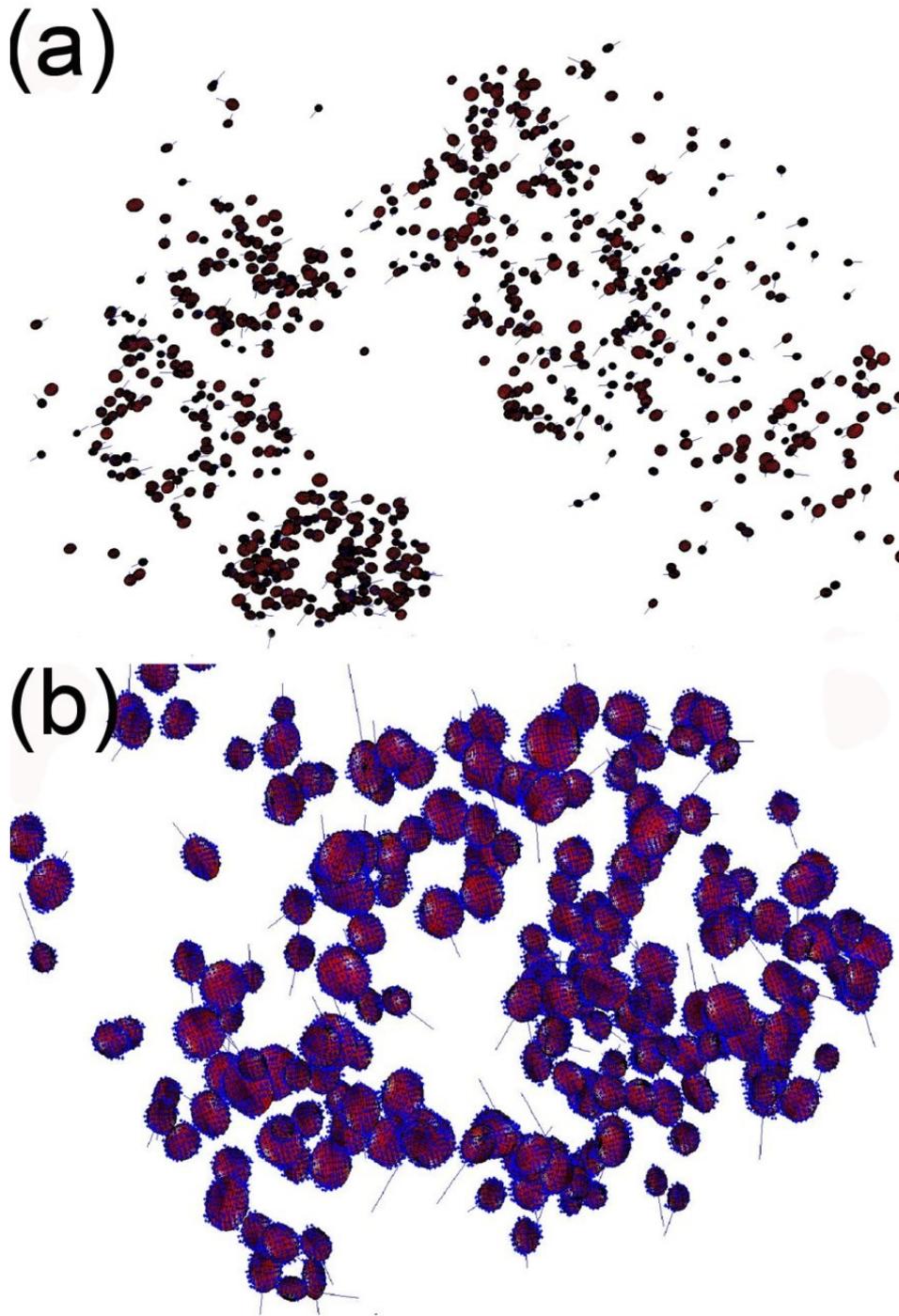


**Figure 5.19:** (a) A test image with a high curvature location. (b) An adsorbed particle. The color bars represent the shape index value. [2]

## 5.6 Monte Carlo Simulations

### 5.6.1 Demagnetizing Factors

The demagnetizing field generated within a particle as a response to the external magnetic field acts to reduce the magnetic moment of the particle. This gives rise to shape anisotropy and is determined by the demagnetizing factors (DF). From the particles' segmentation of the reconstructed volume, the DFs were acquired using linear least squares fitting procedure to fit a general ellipsoid to the segmented particles. First within the Amira environment, a surface was generated without smoothing from the segmentation and a module was written to bring the 3D point coordinates to Matlab environment. Smoothing the segmentation before the triangulation attaches nearby particles together giving false results. A linear least squares method [150] was used to fit an ellipsoid to the point sets, which gives the semi radii and their directions. Using equations 13 to 15, the DFs were then calculated. At least nine points had to be defined to fit an arbitrary ellipsoid and no singular matrices within the fitting equations were allowed. Additionally threshold values were used such that the surface area and volume of the particles were similar to the fitted ones and MSE's had to be smaller than 1. The volume of the particles was calculated using the 3D convex hull of the point set and the surface area was the sum of the triangle areas within the surface mesh calculated in Amira. These were compared to analytical area and volume of the fitted ellipsoids with given semi radii. [2]



**Figure 5.20:** (a) Fitted ellipsoids (red) from the whole sample and (b) from one pore only. Blue points are from the mesh of the generated surface and the lines are the directions of the shape anisotropy easy axis. The length of the line is the inverse of DF multiplied by three for better visualization. [2]

In figure 5.20(a) are the magnetite particles of the whole sample and in 5.20(b) one pore only. The red surface is the fitted ellipsoid to the original blue points from Amira environment. The blue line gives the direction of the longest semi axis and the magnitude of the line is the inverse of the DF multiplied by three for a better visualization. The demagnetization energy is minimized into the

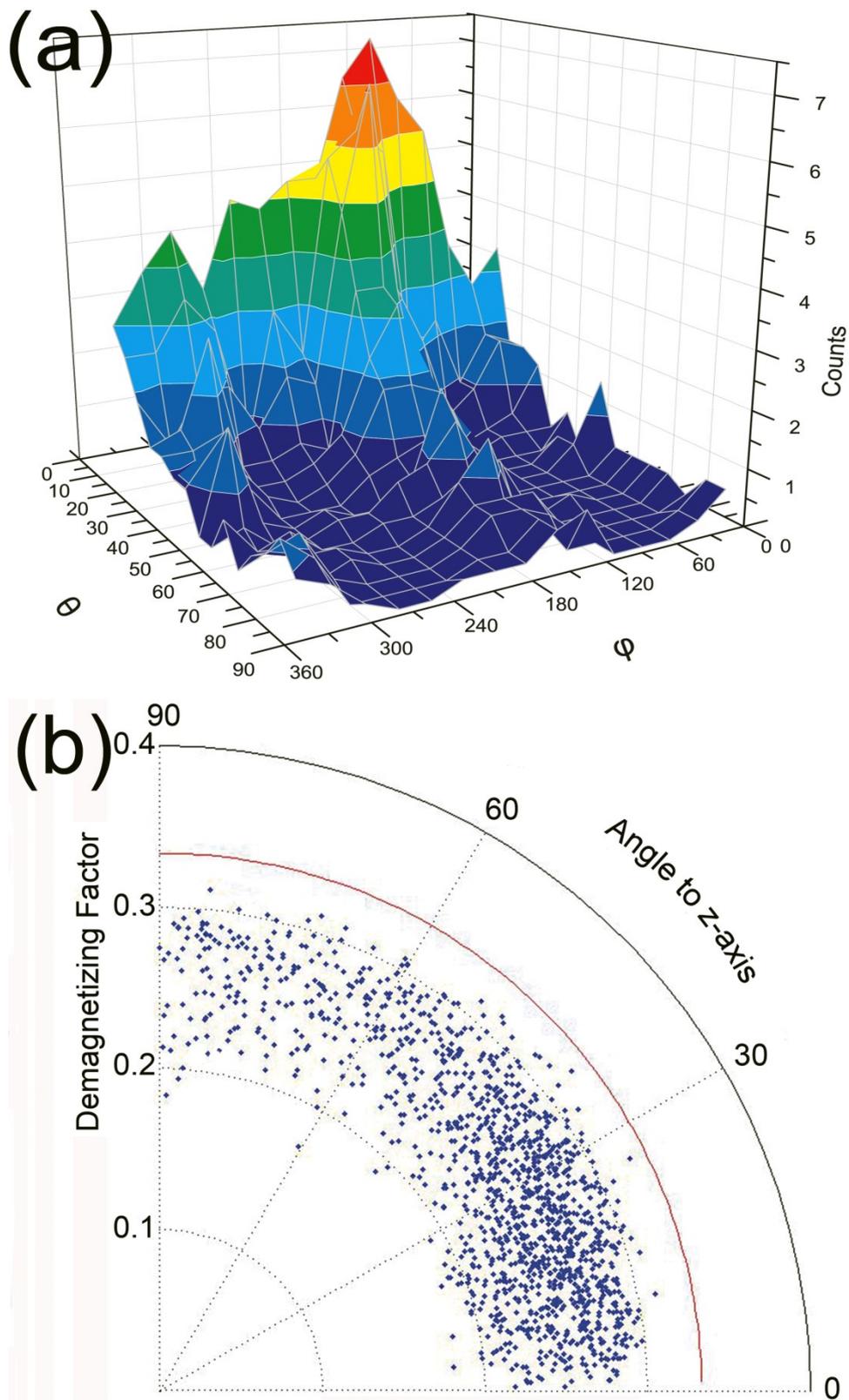
direction of the longest semi axis, therefore one can interpret this axis to be the shape anisotropy easy axis. The directions of the shape anisotropy easy axes  $\hat{a}$  of every particle can then be mapped out and plotted in spherical coordinates in the upper half of a unit sphere. The major semi-axis direction is given as  $\hat{a} = (a_x, a_y, a_z)$  for each particle and from

$$a_x = \cos \varphi \sin \theta, \quad a_y = \sin \varphi \sin \theta, \quad a_z = \cos \theta, \quad (27)$$

it can be expressed in spherical coordinates. The interval  $[0,360^\circ] \times [0,90^\circ]$  of the variables  $(\varphi, \theta)$  was divided in  $10 \times 10$  areas and in the histogram of figure 5.21(a), the number of particles whose the direction of the easy axis is in the solid angle  $\Omega_{ij} = [\varphi_i, \varphi_{i+1}] \times [\theta_j, \theta_{j+1}]$ , is plotted. For the solid angles to be equal, the azimuth range  $[0,360^\circ]$  was divided in equal intervals, while the polar range  $[0,90^\circ]$  was divided so as the range  $[0,1]$  of the parameter  $u = \cos \theta$  was in equal intervals. As it is seen, the histogram has a maximum at  $(\varphi, \theta) \approx (100^\circ, 0^\circ)$  (red and orange areas) so the  $a$ -axis is oriented around this preferential direction.

In figure 5.21(b) the DFs of the shape anisotropy easy axis is plotted with its' direction to the z-axis and one sees there is an accumulation of counts near the zero angle. The z-axis was the direction of the main pores and the direction of the optical axis of the microscope. As will be important later, it is worthwhile to mention that this axis is the direction, where the missing wedge creates elongation. The red annular line in the figure gives the value of  $1/3$ , which stands for perfectly spherical particles. The absence of particles on that line is surprising, but care should be taken when interpreting this plot. The particle closest to the red line had semi radii of 2.6 nm, 2.6 nm and 2.5 nm, which is more or less a perfect sphere for all practical purposes. Also when the pixel size in the reconstruction was 0.76 nm, the 0.1 nm difference becomes problematic. The maximum difference of the minor diameter from the major diameter was 4.4 nm, while the average was 1.6 nm and minimum of 0.1 nm. Both the average and the maximum values are well above the error coming from the voxel size. [2]

The volume and spatial distribution of the nanoparticle assembly can be accessed directly and quantitatively using ET. Along with the knowledge of the DF distribution, ET proves to be a unique and powerful tool to share light to the Dormann's unknowns and is therefore a great asset to the magnetic simulation community. The third unknown could also be resolved using electron tomography holography [151] [152], which can map the individual magnetic moments of the nanoparticles. With the knowledge of the shape anisotropy easy axis and the net direction of the magnetic moment, the magnetocrystalline easy axis could be deduced, although the current achievable resolution might force one to use larger, yet still single domain nanoparticles.

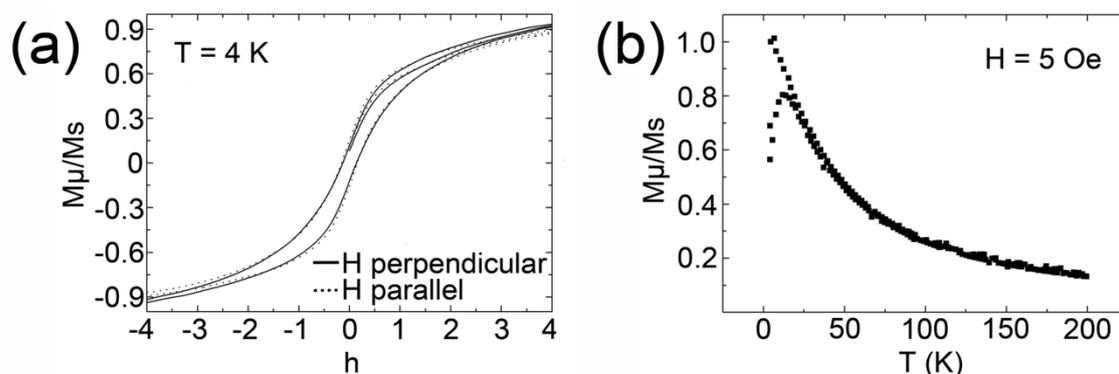


**Figure 5.21:** (a) Direction distribution of the shape anisotropy easy axis in spherical coordinates. (b) The angular axis is the angle of the shape anisotropy easy axis with respect to the z-axis vs. the DF on the radial axis. The red annular line gives a DF value of  $1/3$ . [2]

## 5.6.2 Comparison and Discussion

Magnetic simulations and their analysis as discussed here were performed by Dr. G. Margaris and Dr. K. Trohidou and is partly reproduced here from the publication [2]. The MC simulation, using the standard Metropolis algorithm was employed to obtain the magnetic configuration under an applied field  $H$  and finite temperature  $T$  [153]. At a given temperature and applied field, the system was allowed to relax towards equilibrium for the first  $10^3$  MC steps per spin, and thermal averages were calculated over the subsequent  $10^4$  steps. The measurements were averaged over 10 different initial conditions (random configurations of the anisotropy easy axis and initial spin orientations). The error bars were very small and were not included in the figure 5.23. Each particle was modeled according to the Hamiltonian in Eq. (18).

The used magnetite nanoparticles (about 5 nm in size) are superparamagnetic but due to magnetostatic coupling the blocking temperature  $T_B$  is shifted above the value of non-interacting particles. Magnetization measurements (see figure 5.22) have been carried out by Petra Granitzer (KF University, Graz) with a SQUID (Cryogenics) and a VSM (Quantum Design), respectively. To achieve field dependent magnetization curves a magnetic field between  $\pm 6$  T was applied parallel and perpendicular to the pores. Temperature dependent magnetization was measured between 4 and 300 K. [56]. It was estimated that the metal volume fraction of the assembly is  $x_V \sim 0.04$  (concentration  $\sim 20$  mg/ml) so one can assume that the particles do not touch each other and there is no direct exchange interaction between them. The nanoparticles interact only via long-range dipolar forces.

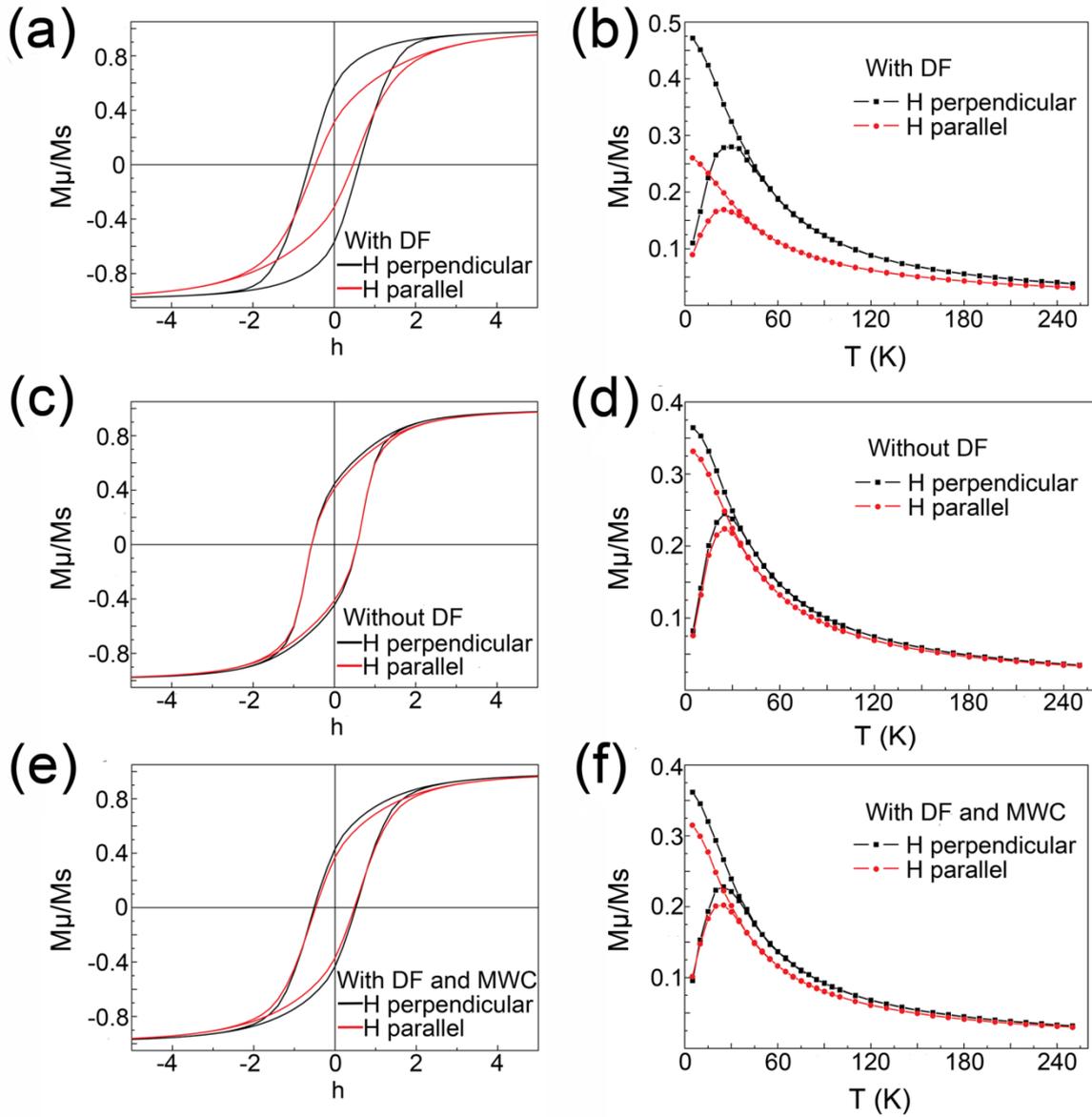


**Figure 5.22:** (a) Measured hysteresis loops with magnetic field applied perpendicular (z-axis) and parallel (xy-plane) to the surface. (b) Measured ZFC/FC magnetization vs.  $T$  curves showing a blocking temperature of 12 K. (Image courtesy: P. Granitzer)

The magnetic MC simulations were performed parallel to the surface (xy-plane) and perpendicular to it (z-axis) and with including or omitting the DFs. Parallel simulations were averaged over all in-plane directions by performing multiple simulations with the magnetic field rotated in steps of  $30^\circ$  degrees. The measured magnetic hysteresis curve and ZFC/FC magnetization vs.  $T$  curve can be seen in figure 5.22(a) and 5.22(b) respectively. In the former, a coercive field of 60 Oe was measured and a slight difference on the saturation magnetization between parallel and perpendicular field to the surface can be distinguished. This is due to the difference in the spatial

distribution of the particles in the parallel and perpendicular direction to the sample surface. The dipolar coupling between the particles is less in the former direction because of the pore walls that separates the pores and the particles therein. This behavior becomes stronger as the particle radius grows. The blocking temperature in the latter figure was found to be 12 K, which is a bit higher than in non-interacting assemblies of magnetite nanoparticles and indicates the existence of weak dipolar interactions within the sample. The blocking temperature for non-interacting case can be calculated with the equation  $K_{\mu}V = 25k_B T_B$  and by using the mean diameter of 5.9 nm from the fitted log-normal curve in figure 5.17(b) assuming spherical particles, the blocking temperature is found to be  $T_B = 4.05$  K. [2]

As it can be seen in figure 5.23(a), where the DFs were included, there is a clear difference between the hysteresis loops taken from the two different directions of the applied field. When the field is along the z-axis, the hysteresis loop presents a square-like shape while when it is on the plane, its shape is thinner and more inclined. Consequently, the coercive field is much larger when the field is perpendicular to the surface  $H_{c,perp} = 0.611$  M<sub>μ</sub>/Ms compared to the case with the field parallel to the surface  $H_{c,par} = 0.462$  M<sub>μ</sub>/Ms. The comparison of these results to the measured ones suggests a too strong magnetic anisotropy between the two directions and fits better to the measured curves acquired from samples, where 8 nm particles were used as seen in [56]. Figure 5.23(b) presents the simulated ZFC/FC magnetization vs. T curves, performed at cooling field  $H_{cool} = 50$  Oe. It can be seen that the blocking temperature is bigger in the perpendicular applied field (along the z-axis) case ( $T_{B,perp} = 29$ K) than in the case that the applied field is in the xy-plane ( $T_{B,par} = 24.5$ K) and both values are bigger than the measured one. The difference between the two directions can be explained by looking at figure 5.21(b), where a clear accumulation of counts are found to be close to the zero angle to the z-axis. This means that the demagnetization energy in the Hamiltonian equation (27) is stronger in z-direction. [2]



**Figure 5.23:** (a) Simulated hysteresis loops with magnetic field applied perpendicular (z-axis) and parallel (xy-plane) to the surface including the shape anisotropy (DFs). (b) Simulated ZFC/FC magnetization vs.  $T$  curves including the shape anisotropy. (c) Simulated hysteresis loops with magnetic field applied perpendicular (z-axis) and parallel (xy-plane) to the surface omitting the shape anisotropy. (d) Simulated ZFC/FC magnetization vs.  $T$  curves omitting the shape anisotropy. The magnetic hysteresis simulations were performed at  $T = 4.2$  K and the ZFC/FC simulations were performed at cooling field  $H_{cool} = 50$  Oe. (e) Simulated hysteresis loops with missing wedge correction (see text below) and including the shape anisotropy. (f) Simulated ZFC/FC magnetization vs.  $T$  curves with missing wedge correction and including the shape anisotropy. [2]

Next, by omitting the demagnetising (shape) energy term in the Hamiltonian, the hysteresis (figure 5.23(c)) and ZFC/FC (figure 5.23(d)) curves were calculated. Spherical shape of the particles was assumed and the radius is calculated from the volume distribution. The differences between the curves calculated with the two directions of the applied become smaller. The coercive field is  $H_{c,no\_demagn} = 0.557 \text{ M}_\mu/\text{Ms}$  and the blocking temperature is  $T_{B,no\_demagn} = 25.5\text{K}$  for both directions of the applied field. So the difference between the two hysteresis loops in figure 5.23(a) and (c), in this assembly model, can be attributed to the shape anisotropy. As it is known, the demagnetizing energy is equivalent to a bi-directional anisotropy (shape anisotropy) with anisotropy constants  $K_{shape1} = 1/2 \mu_0(N_a - N_c)M_S^2$ ,  $K_{shape2} = 1/2 \mu_0(N_b - N_c)M_S^2$  and ( $K_{shape1} > K_{shape2}$ ) [154] [30] [39], where the direction of the first easy axis is along the major axis  $\hat{a}$  and the second easy axis is along the axis  $\hat{b}$ . Each particle in the assembly, apart from the uniaxial crystallographic anisotropy (which is randomly oriented in the assembly) has also shape anisotropy which has easy axis with a narrow, non-uniform distribution around the z-axis. So if by adding the demagnetizing energy term in the Hamiltonian one actually includes another anisotropy term (shape anisotropy) approximately directed along the z-axis. The average shape anisotropy density divided with the crystallographic anisotropy is  $K_{shape1}/K_1 = 1/2 \mu_0 \langle(N_a - N_c)V\rangle M_S^2 / K_1 \langle V \rangle = 1.26$ , so one sees that the shape anisotropy plays an important role. As more particles have shape anisotropy easy axis along the z-axis, this accounts for the bigger value of the coercive field in this direction and the difference at the shape of the hysteresis loop. Also, it explains the increase of the blocking temperature when the cooling field is perpendicular to the surface. [2]

When the demagnetizing energy is not included, the only anisotropy of the assembly is the random anisotropy and the shape of the hysteresis loops is typical of a random anisotropy assembly. The small differences between the loops along the two directions are due to the dipolar interactions between the particles in the non-symmetric shaped assembly. Considering the importance of the demagnetizing energy term in the Hamiltonian as explained above, and the better qualitative agreement of the simulated data compared to the measured ones, when omitting this term suggest that the particles are indeed spherical.

The remanence of a uniaxial particle assembly with easy axes randomly oriented in space is usually of the order of  $0.5 \text{ M}_\mu/\text{Ms}$  [39], as was the case with simulated curves, yet clearly in the measured hysteresis curve 5.22(a), this was found to be close to  $0.15 \text{ M}_\mu/\text{Ms}$ . This would suggest that one cannot model the sample with a Hamiltonian used within this study. The concentration of particles in the pores is different in the vicinity of the surface and the bulk silicon and here we studied just a small fraction of the pores approximately in the middle of the sample (figure 3.4(a)). Higher concentrations close to the surface will bring stronger dipole forces and fanning effects. The shape of the measured FC curve (figure 5.22(b)) at low temperatures is descending linearly, which implies weak dipolar interactions and the measured hysteresis curve does not saturate, which could be explained by a non-random assembly of ellipsoidal particles with the hard axis perpendicular to the field. Exchange interaction between the particles appears when there is physical contact between them. The exchange interaction is expected to play an important role in samples with concentration close and above the percolation threshold. As the nanoparticle concentration increases, the interparticle interactions modify the distribution of the effective energy barrier. This results in more complex phenomena, such as superspin glass behavior in low-enough temperatures for intermediate concentration systems [155] and superferromagnetic order for very dense systems [33]. The characteristics of the hysteresis loop (remanence and coercivity) and the blocking temperature have been shown to vary with nanoparticle concentration in granular metals and frozen ferrofluids [156].

One obvious explanation for this discrepancy might come from the missing wedge problem, which causes elongation of the particles in the z-direction making them ellipsoidal. Another cause might stem from the low sampling of the particles, which comes from the low magnification used. Higher magnifications means on the other hand low statistics to MC simulations since fewer particles are detected. A way out of this is to use montage tomography with high enough magnification for high sampling of the particles. The missing wedge comes less problematic with dual axis tomography or with special 180° degrees tilting sample holders. The elongation factor in the z-direction arising from the missing wedge from equation (23) is 1.15 for an averaged maximum tilt angle of 78.5° degrees. If the z-coordinates of the particles are divided by the elongation factor comprising of a missing wedge correction (MWC) and proceed with the simulations again, then we get the hysteresis loops as seen in figure 5.23(e) and ZFC/FC vs. T curves in figure 5.23(f). Here the qualitative agreement is better because of the inclination than when omitting the DFs. The directions of the shape anisotropy easy axis of the particles are more randomly oriented and therefore the ZFC/FC vs. T curves taken from different directions to the surface are also more similar.

## 6 Summary and Outlook

The application of electron tomography for the quantitative characterization of materials at the nanoscale and in all three dimensions offers vast possibilities and new insights for the field of materials science. For qualitative analysis ET is a straightforward method and easy to use technique (assuming the use of fiducial markers for alignment) but for a detailed quantitative characterization the acquisition parameters and alignment quality must be as good as possible to obtain a high quality reconstruction for reliable and objective segmentation. Various three dimensional image manipulation methods and algorithms are available (Amira, Avizo, IMOD, DM, IDL, MATLAB), yet for new characterization methods one has to possess good programming and coding skills to manipulate data in three dimensions.

Within this thesis a new precise and automatic method for the magnification calibration of the microscope was introduced using the auto-correlation function. A plug-in was written to integrate the vast public library ITK into DM environment thus bringing various useful algorithms for the electron microscopy community. Without these commands this thesis would not have been possible to accomplish. Using the ITK library, a new semi-automatic alignment method called projection re-projection comparison (PRC) was written for marker-less alignment of the tilt series and the effectiveness of it was shown. Additionally ITK offered new helpful filters and methods for the segmentation. The resolution in IBF and HAADF modes was discussed along with the analysis of copper interconnects. The FIB sample preparation in cryo and RT conditions was investigated in terms of resolution and submerging ratio using a pSi sample infiltrated with magnetite nanoparticles. Two different methods were introduced and discussed to analyze the local curvature distribution of the magnetite nanoparticles within the pSi sample surface. A program to fit an arbitrary ellipsoid to the same nanoparticles was written and the distribution of the directions and magnitudes of the demagnetizing factors of a nanoparticle assembly was characterized for the first time. This distribution was additionally used along with the knowledge of the volume and spatial distribution of the nanoparticle assembly in magnetic Monte Carlo simulations to resolve the magnetic response of the assembly in terms of magnetic hysteresis curves and zero field cooling / field cooling curves. The simulations were compared to the measured ones with good qualitative agreement.

For the field of materials science, ET can be extremely useful as it gives nanoscale three dimensional information of real materials, which can be further used to any kind of simulations, not just limited to magnetic ones. For example the optical response of porous materials or electron transport in solar cells. The models used in these kinds of simulations for the material's morphology are extremely basic and non-physical, compared to the real acquired three dimensional data produced by ET. Therefore as a general outlook, it is the author's view that it would be highly interesting and rewarding to use ET in combination with physical simulations within various fields to produce more accurate predictions and results with fewer assumptions on the materials. The metrology used so far for the characterization of porous materials in three dimensions is restricted to FIB tomography [157] and ET. The achievable spatial resolution for the former is sufficient to resolve the main pores but only the latter is capable to fully explore the dendritic nanopores. For a

complete study and understanding of any porous material however, a combination of these two methods could be meaningful. It could harness both, the huge sampling volume in the FIB and the higher resolution in ET. This information could then be further used for example for simulations of the optical response of porous materials. For magnetic simulations, it would be an interesting experiment as an outlook to map out the net magnetic moments of a few interacting particles using 3D electron holography [151], simulate the behavior using Magpar [158] program and retrieve the magnetocrystalline easy axes of the particles. One could then compare these simulated relative easy axis directions ( $\langle 111 \rangle$  for spinel type structures) to the measured crystallographic zone axes obtained for example using the Kikuchi lines as a proof of concept.

To analyze the reconstruction image in terms of a correct alignment is a difficult task. Usually for focus measurements a sharpness function is used; several methods are available: derivatives, Fourier transforms/power spectrums, autocorrelation, variance and histograms [159]. However in practice, starting with incorrect alignment, the reconstruction image is severely flawed and noisy; the exhaustive search to optimize the sharpness function in the multi-dimensional parameter space is often futile. In addition, even when enhancing the projections under alignment in the reconstruction, it is not given that the sharpness function is maximized at correct alignment. As an outlook two different methods to align the tilt series are proposed.

The first one consists of an exhaustive search on the alignment parameter space while projecting the enhanced images under alignment in the reconstruction and calculating the gray level co-occurrence matrix (GLCM) at each step [160]. GLCM of an image indicates co-occurrences of intensity differences with a certain offset. The results depend on the direction of the offset, which can be varied e.g.  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$  and so on. The GLCM is implemented in the DM environment through the ITK library and provides 2<sup>nd</sup> order statistics to be calculated from the reconstruction. It is commonly used to find the texture within the image for segmentation but it is also possible to calculate the energy, entropy, correlation, difference moment, inertia... etc. as defined in [161] to be used as sharpness functions. Additional benefit of the function is that one can choose the in-plane direction for the offset pixels to be approximately in normal direction to the projected rays since this is the direction where the intensities are confined within objects in the reconstruction. GLCM for 16 bit images means a matrix consisting over 429 million cells, so conversion should be used and to limit the calculation to small ROI's placed on high contrast objects. Rather than exhaustive search, an optimization routine could be employed using the sharpness function as a cost function. Additionally several other sharpness functions could be explored for example using the package "fmeasure" as provided in the Matlab exchange website, which offers 28 different focus measurement implementations [162].

The second one involves a method called demons registration [99], which is commonly used in medical images e.g. to align three dimensional CT reconstructions to MRI, or to the tracing of tumor growth. The name comes from allusion to Maxwell's demon. Within the algorithm, the image to be registered to the reference image is iteratively deformed using the optical flow equation and Gaussian smoothing for regularization. The output is a vector field of the individual pixel translations. Considering the tracking of the fiducial markers within the tilt series, the trace of the center pixel of the colloidal gold particles is obtained and used in a minimization process to predict the three dimensional voxel coordinates of those particles' center. Instead of acquiring only one voxel location in the reconstruction, using deformable registration to the subsequent tilt images, it could be possible to obtain all voxel locations (at least the ones with contrast) in the reconstruction. Hence with demons registration one can not only align the tilt series, but also do a reconstruction. The deformation characteristics should be rewritten for projection images such that two directions for the diffusion are possible. This extra complication is however compensated by considering the rotation of a centered 3D rigid object such that these two directions are always parallel and in

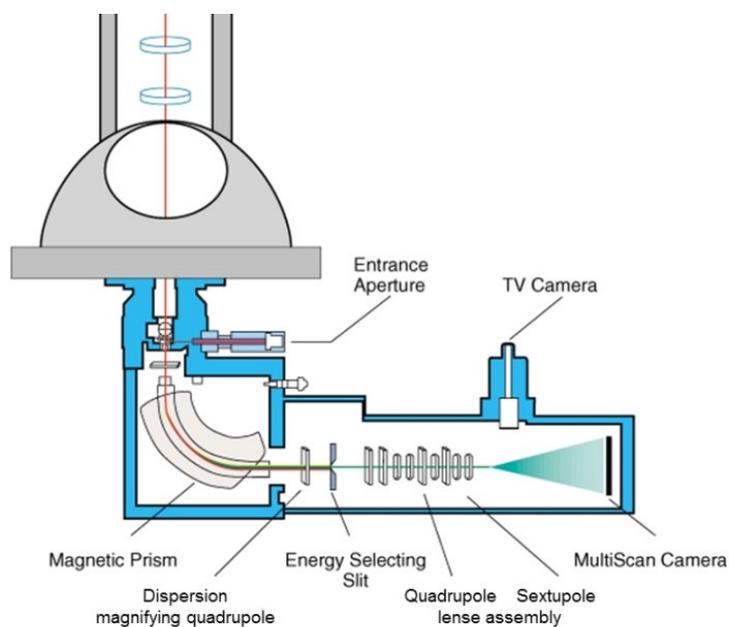
aligned tilt series in normal direction to the tilt axis. Two versions of the demons registration algorithm are implemented in DM through the ITK library.

## 7 Appendices

### 7.1 EFTEM Tomography

After a short introduction to energy filtering in the TEM, a quantitative analysis of the elemental distribution of a flash memory device is given. With element specific contrast it is possible to fully ascertain the quality of the deposited layers of the device.

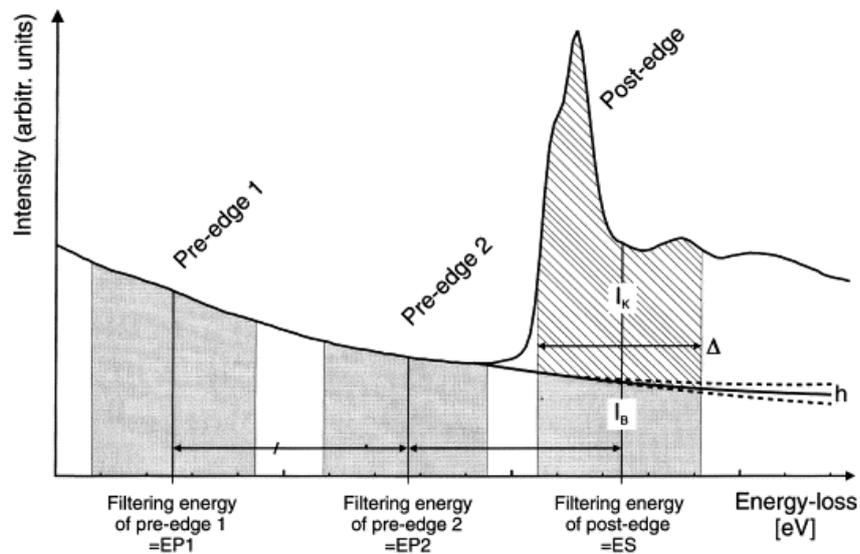
A post-column energy filter [163] [164] [165] can produce 2D images by a complex system consisting of multipole electromagnetic lenses. In figure 7.1 the schematics of a filter is shown. The transmitted electrons are separated in energy by a special shaped magnetic prism before they enter in energy selecting slit. The energy dispersion is highest at the plane of the slit for simplified energy selection, while the spatial information is reduced. The purpose of the lens system is to recover energy uniform spatial information for imaging with a CCD camera. Selection of particular spectral component and forming an image thereof leads to a technique called energy-filtering TEM (EFTEM) [166].



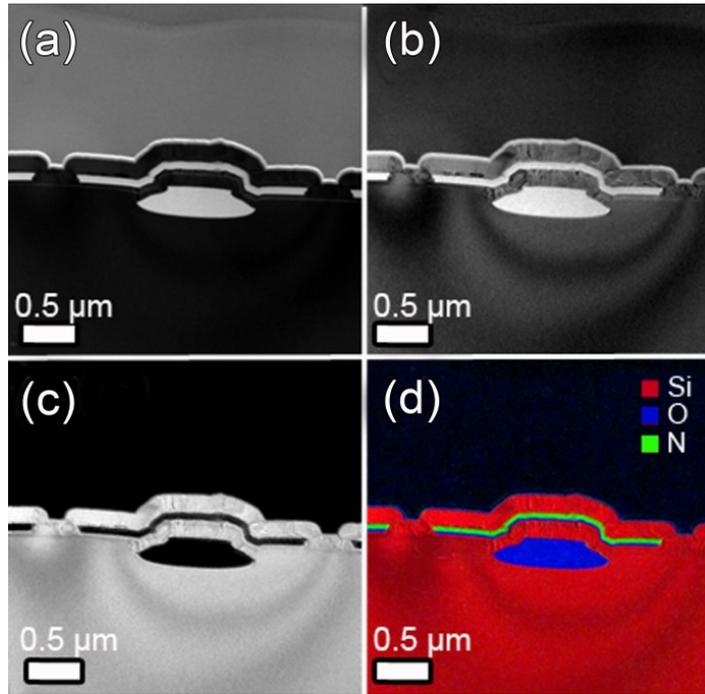
**Figure 7.1:** Gatan GIF post-column energy filter below a TEM. Taken from [167].

One prominent application of EFTEM is elemental mapping, where the image contrast comes from the chemical composition within the sample and the imaging of element specific edges. By selecting the ZLP for imaging the contrast and resolution of images can be enhanced by excluding the inelastic contribution, which blurs the image due to chromatic aberrations [168].

For quantitative chemical analysis using EFTEM, usually a technique called “three- windows- imaging” is used [169], where three different images are taken with a specific energy window before and after the ionization edge as seen in figure 7.2. The two images before the edge are used to estimate the background under the ionization edge, which can then be extrapolated and subtracted. Usually an inverse power law  $I = A\Delta E^{-r}$  is used, where A and r are constants and can be acquired by curve fitting from the two pre-edge images [3]. Figure 7.3 illustrates this technique for a semiconductor device. Figure 7.3(a) is a pre-edge image taken before the ionization edge (@69 eV, 10 eV slit width) and (b) is the post-edge image (@104 eV, 10 eV slit width). (The second pre-edge image (@84 eV, 10 eV slit width) is not shown here.) The three images then allow to generate the elemental map of silicon (c) using the  $L_{2,3}$  edge located at 99 eV. Combining the silicon map with oxygen and nitrogen maps yields the RGB image (d), giving an overview of the chemical composition of the sample. Oxygen (@ 467 eV, 507 eV and 552 eV) and nitrogen (@ 343 eV, 372 eV and 413 eV) images were recorded with 40 eV and 24 eV slit width respectively. Because of the long acquisition times usually needed for EFTEM images, sample drift can occur, necessitating image alignments using cross-correlation and filtering techniques.



**Figure 7.2:** The three-windows–method. The two pre-edge images are used to estimate the background under the post-edge image using a specific energy window  $\Delta$ . From [170]



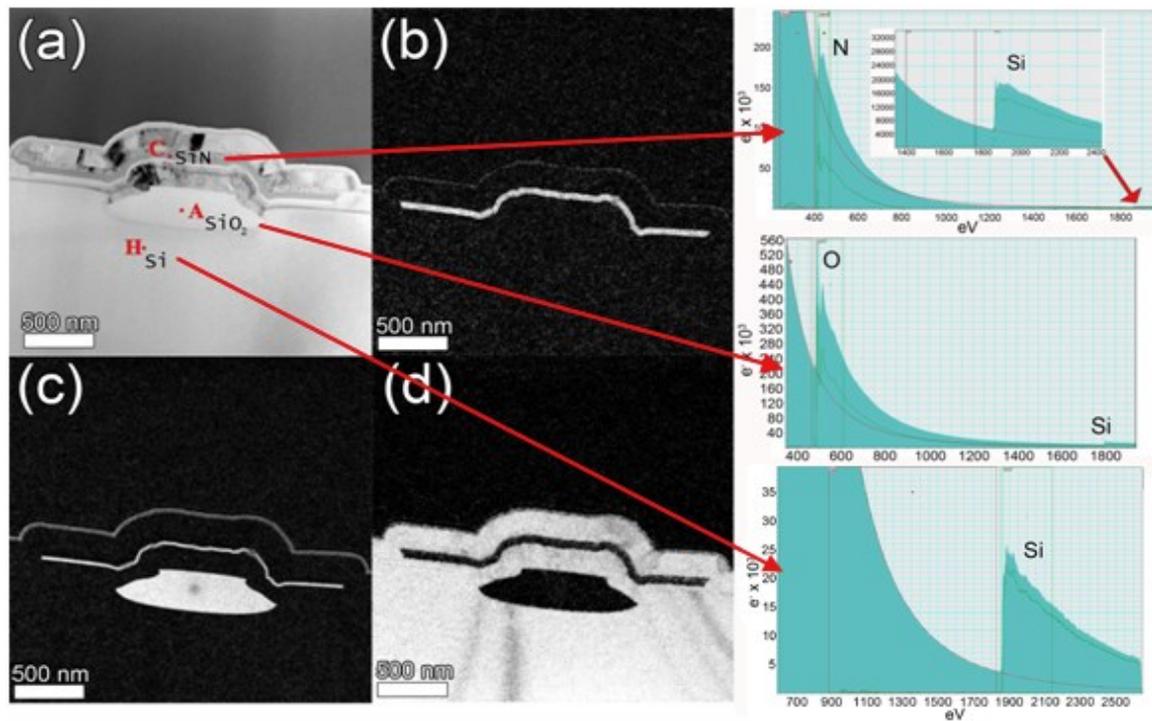
**Figure 7.3:** (a) Si  $L_{2,3}$  pre-edge image of a flash memory cell. (b) A post-edge image and (c) a silicon elemental map showing only the silicon content. (d) Multimap with combined silicon (red), oxygen (blue) and nitrogen (green) contributions. The second pre-edge image is not shown here.

Here EFTEM is extended to three dimensions known as EFTEM tomography or core-loss tomography to extract the three dimensional layering of silicon, oxygen and nitrogen from a flash memory device sample. To fulfill the projection requirement the sample thickness had to be less than the inelastic mean free path of the electrons within the sample at every tilt angle, ensuring that the edge intensity increases linearly with the mass-thickness of the associated element and plural scattering is minimized. In principle, the same information can be achieved in STEM mode and spectrum imaging, where an EEL spectrum is acquired for every pixel in the image thus building a three dimensional dataset (x and y: spatial, z: energy). [83] [171] [172]

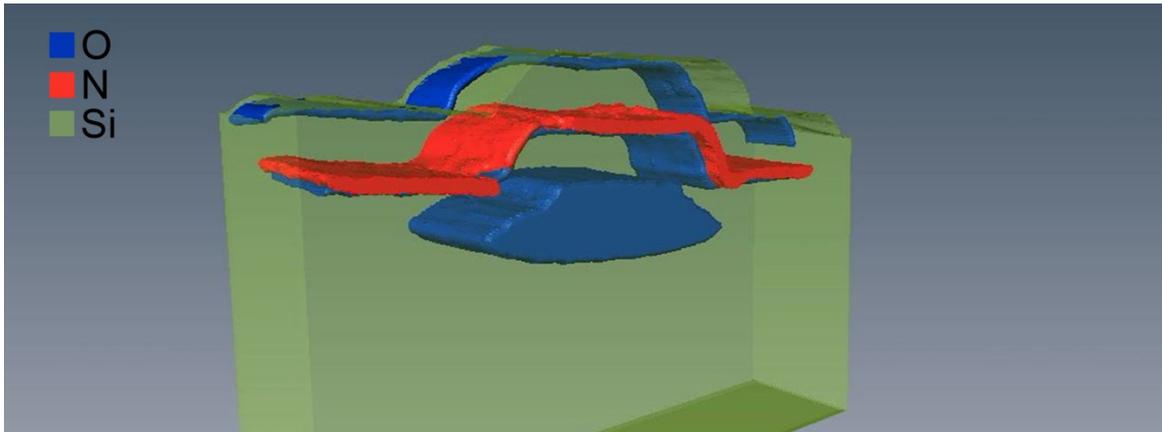
EFTEM tomography was applied to a flash memory cell sample. Three different elemental maps were acquired within a 17 hour experiment using the TF20 microscope. The sample was first plasma cleaned to avoid carbon contamination for 2 minutes and zero tilt images acquired before and after the tilt series showed no carbon contamination and beam damage effects. Silicon elemental maps were acquired using an energy slit of 10 eV centered at 99 eV (L-edge), oxygen an energy slit of 40 eV at 532 eV (K-edge) and nitrogen an energy slit of 24 eV at 401 eV (K-edge) using a Gatan GIF Quantum ERS. A tilt series was acquired with angles ranging from  $-58^\circ$  to  $58^\circ$  degrees with  $2^\circ$  degree tilt step for each element, unfiltered and the ZL. In figure 7.4(a) one sees the ZL image of the sample and in 7.4(b-d) the elemental maps of nitrogen, oxygen and silicon respectively. In the ZL image below is the silicon substrate and on location A is the field isolation oxide. The crystal grains of the two layers of poly-silicon are seen. Commonly in between the poly-silicon is the  $\text{SiO}_2$ - $\text{SiN}$ - $\text{SiO}_2$  dielectric layer known as the SONOS charge trap, but as seen in figure 7.4(c), the upper oxide layer is missing. The EELS point spectra are in the right side on figure 7.4 clearly shows the presence of silicon in locations A and C at electron loss energy of 1839 eV indicating the presence of SiN and  $\text{SiO}_2$  respectively. This was also seen in the silicon map in figure 7.4(d) but since the

peak at energy loss of 99 eV was used, the signal was too weak in comparison to the pure silicon in the substrate and in the poly-silicon layers and appears to be missing in the silicon map. The 99 eV peak was used here since the peak at 1839 eV would have been impractical in the acquisition of the tilt series in terms of measurement time.

The alignment of the pre- and post edge images was done using CC and the alignment of the tilt series was done using the routines in IDL environment as explained in previous chapters. The reconstructions were filtered using anisotropic diffusion filter and segmented by global threshold. The three reconstructions were combined as seen in figure 7.5 to produce a surface rendered elemental multimap representation of the sample. The z-coordinates in the image are stretched to better visualize the projection dimension. From the three dimensional qualitative analysis it is immediately seen, that the upper oxygen layer is only half covered within the sample and there is a crack in the nitrogen layer on the left trench. The upper oxygen layer of the SONOS stack is also missing. This type of information is difficult or often impossible to acquire using cTEM 2D micrographs.



**Figure 7.4:** Flash memory cell with (a) ZL image, (b) nitrogen map, (c) oxygen map and (d) silicon map. On the right one sees the EELS point spectra at indicated locations showing the presence of SiN at location C, SiO<sub>2</sub> at location A and single crystalline Si at location H.



**Figure 7.5:** Surface visualization of the three combined elemental tomograms of silicon (green), oxygen (blue) and nitrogen (red) of the flash memory device sample. The reconstruction is stretched along the z direction.

## 7.2 Digital Micrograph Scripts

The following scripts are necessary to perform the PRC alignment as described in chapter 5.2.2. To prealign the tiltseries, it is often useful to only cross-correlate the center area of the image with a ROI, which is as wide as possible parallel to the tiltaxis and narrow perpendicular to the tiltaxis. One can then use the following script:

PreAlignment.s

```
// $BACKGROUND$

// sub-pixel search function from D. Mitchell
Number ImageRefineExtrema(Image img, number &px, number &py)
{
  number tx1, tx2, ty1, ty2
  number numa, numb, numc
  number valX, valY

  // do a parabolic fit in direction x
  numa = img.GetPixel(px-1, py)
  numb = img.GetPixel(px, py)
  numc = img.GetPixel(px+1, py)
  tx2=(numa-numc)/2.0
  tx1=(numa+numc)/2.0-numb
  valX=(2*numa*numc-numa**2-numc**2)/8/(numa+numc-2*numb)+numb

  // do a parabolic fit in direction y
  numa = img.GetPixel(px, py-1)
  numb = img.GetPixel(px, py)
  numc = img.GetPixel(px, py+1)
  ty2=(numa-numc)/2.0
  ty1=(numa+numc)/2.0-numb
  valY=(2*numa*numc-numa**2-numc**2)/8/(numa+numc-2*numb)+numb

  // update pixel position
  px += tx2/(2.0*tx1)
  py += ty2/(2.0*ty1)

  // check to see whether to return minimum or maximum
```

```

If(numa>=numb && numc>=numb) return min(min(valX, valY), numb)
If(numa<=numb && numc<=numb) return max(max(valX, valY), numb)

Return numb
}

Image CrossCorrelateWithSubPixelPrecision(image src, image ref, number &pX, \
number &pY, number &qval)
{
try
{
Number sizeX,sizeY
src.GetSize(sizeX,sizeY)

ReallImage xcorr = src.CrossCorrelate(ref)

// find the maximum and its coordinates in correlation image
Number spotX, spotY
qval = max(xcorr, spotX, spotY)

qval = xcorr.ImageRefineExtrema(spotX, spotY)

pX = spotX
pY = spotY

return xcorr
}
catch break
}

image front := getfrontimage()
number x,y,z,l,r,b,t,i,xx,yy
get3dsizesize(front, xx,yy,z)
imagedisplay frontdisp = imagegetimagedisplay(front,0)
number nroi = imagedisplaycountrois(frontdisp)
string frontname = GetName( front )

image roifront
image shifts := exprsize(z,2,0)

if(nroi!=1) throw("put n^2 roi in the image")

roi fRoi = imagedisplaygetroi(frontdisp,0)
ROIGetRectangle( froi, t, l,b, r )
if(mod(r-l,2)!=0) r+=1
number xn = (r - l)/2
roifront = front[xx/2 - xn,t,0,xx/2+ xn,b,z]
get3dsizesize(roifront,x,y,z)

image ref := exprsize(x,y,0)
image mov := exprsize(x,y,0)
number zero = ceil(z/2), cnt = 0,px,py,q

for(i=zero ; i<z-1 ; i++)
{
cnt+=1
OpenAndSetProgressWindow( "Aligning", "Plane", ""+(cnt)+"/"+z+" " )

ref = roifront[0,0,i,x,y,i+1]
mov = roifront[0,0,i+1,x,y,i+2]

ifmapplyfilterinplace(ref,"Combined Filter 0")
ifmapplyfilterinplace(mov,"Combined Filter 0")

CrossCorrelateWithSubPixelPrecision(ref,mov, px, py, q)

setpixel(shifts,i+1,0,(x/2 -px))
setpixel(shifts,i+1,1,(y/2 -py))
}

for(i=zero ; i>=2 ; i--)
{

```

```

cnt+=1
OpenAndSetProgressWindow( "Aligning", "Plane", ""+(cnt)+"/"+z+" ")

ref = roifront[0,0,i,x,y,i+1]
mov = roifront[0,0,i-1,x,y,i]

ifmapplyfilterinplace(ref,"Combined Filter 0")
ifmapplyfilterinplace(mov,"Combined Filter 0")

CrossCorrelateWithSubPixelPrecision(ref,mov, px, py, q)

setpixel(shifts,i-1,0,(x/2 -px))
setpixel(shifts,i-1,1,(y/2 -py))
}
OpenAndSetProgressWindow( "", "", "" )

showimage(shifts)
ImageSetName( shifts, "shifts of " + frontname)

SetDisplayType( shifts, 4 )
imagedisplay shdisp = imagegetimagedisplay(shifts,0)
LinePlotImageDisplaySetSliceDrawingStyle( shdisp, 0, 1 )
LinePlotImageDisplaySetSliceDrawingStyle( shdisp, 1, 1 )

```

To shift the tiltseries after the coarse alignment, the following script can be used:

### ShiftTiltseries.s

```

// $BACKGROUND$

image stack,shiftsx,shiftsy
number x,y,z,i
getthreeimages("tiltseries and shiftsx, shifty",stack,shiftsx,shiftsy)
get3dsizes(stack,x,y,z)
string stackname = GetName( stack)

image plane := exprsize(x,y,0)
image plane1 := exprsize(x,y,0)
image newstack := stack.imageclone()

for(i=0 ; i<=z-1 ; i++)
{
OpenAndSetProgressWindow( "Shifting", "Plane", ""+(i)+"/"+z+" ")

plane1 = stack[0,0,i,x,y,i+1]
plane = warp(plane1, icol + getpixel(shiftsx,i,0),irow + getpixel(shiftsy,i,1))

newstack[0,0,i,x,y,i+1] = plane

}

OpenAndSetProgressWindow( "", "", "" )

showimage(newstack)
ImageSetName( newstack, "aligned " + stackname )
ImageCopyCalibrationFrom( newstack, stack)

```

As mentioned, the global shift and rotation has to be done manually. One can use the DM Reconstruction window to add a tiltaxis to the tiltseries and check whether the stack needs to be shifted and rotated. After finding the correct shift and rotation, the following script can be used to correct them. The shift and rotation variables have to be set accordingly.

## ShiftandRotateTiltseries.s

```
// $BACKGROUND$

number shift = 9
number rotation = 1.5

image front:= getfrontimage()
number x,y,z,i
get3dszsize(front,x,y,z)
image front1 := front.imageclone()

for(i=0;i<z;i++)
{
image plane = front[0,0,i,x,y,i+1]
image plane1 := plane.imageclone()
plane1 = warp(plane,icol + shift,irow)
front1[0,0,i,x,y,i+1] = plane1
}
front1 = Rotate3D( front1,0,0, rotation )
ImageCopyCalibrationFrom( front1, front )
showimage(front1)
```

To acquire re-projections from the original tiltseries, a ROI must be defined to limit the area for faster computing. The following script gives two stacks, the ROI projections and the re-projections, which can then be cross-correlated using CCTiltseriesandProjections.s. This gives the shifts to be used to the original tiltseries.

The aligned tiltseries can then be used for the next iteration. The minimum tiltangle and tiltstep variables have to be changed accordingly. The weight parameter multiplies the maximum projection images, since their intensity is far less than the sum projections. If the bright objects are not seen clearly in the re-projections, this value can be raised.

## MakeProjectionsFromTiltseries.s

```
// $BACKGROUND$

number mintiltangle = -70
number tiltstepsize = 5
number weight = 20

void SetUsePlane( image img, number plane, number use_it )
{
TagGroup tags = img.ImageGetTagGroup()
TagGroup toUseList
tags.TagGroupGetTagAsTagGroup( "Tomography:Tomography setup:Tilt angles:Custom Tilt Angles", toUseList )
toUseList.TagGroupSetIndexedTagAsBoolean( plane, use_it )
}

////////////////////////////////////

void settomotags(image tiltseries, number start, number tilt)
{
string str = "min -> max"
string st = "equal"

number x,y,z,i,a

get3dszsize(tiltseries,x,y,z)
number end = (z-1)*tilt + start
image tiltang = exprsize(z,1,0)

tiltang = icol*tilt
tiltang+=start
```

```

tiltseries.setNumberNote("Tomography:Tomography setup:Tilt angles:Maximum tilt angle (deg)",end)
tiltseries.setNumberNote("Tomography:Tomography setup:Tilt angles:Minimum tilt angle (deg)",start)
tiltseries.setNumberNote("Tomography:Tomography setup:Tilt angles:Starting tilt angle (deg)",start)
tiltseries.setStringNote("Tomography:Tomography setup:Tilt angles:Exposure time increasing model","constant")
tiltseries.setNumberNote("Tomography:Tomography setup:Tilt angles:Time factor for 60 degree projection",2.0)

```

```

tiltseries.setNumberNote("Tomography:Tomography setup:Tilt angles:Tilt angle step (deg)",tilt)
tiltseries.setStringNote("Tomography:Tomography setup:Tilt angles:Tilt direction",str)
tiltseries.setStringNote("Tomography:Tomography setup:Tilt angles:Tilt model",st)

```

```

TagGroup tgRoot = NewTaglist();
TagGroup tgRoot1 = NewTaglist();
TagGroup tgRoot2 = NewTaglist();

```

```

TagGroup ba = NewTaglist();
TagGroup ba1 = NewTaglist();
TagGroup ba2 = NewTaglist();

```

```

TagGroup tags = tiltseries.ImageGetTaggroup()
TagGroupsetTagAsTagGroup(tags,"Tomography:Tracking data:Online tracking data",ba)
TagGroupsetTagAsTagGroup(tags,"Tomography:Tracking data:Used precalibration data",ba2)
TagGroupsetTagAsTagGroup(tags,"Tomography:Tomography setup:Tilt angles:Custom Tilt Angles",ba1)

```

```

TagGroup tgItem = NewTagGroup()
TagGroup tgItem1 = NewTagGroup()
TagGroup tgItem2 = NewTagGroup()

```

```

for( number idx=0; idx<z; idx++ )
{
a = getpixel(tiltang,idx,0)

```

```

troot.TagGroupAddTagGroupatbeginning(tgItem)
tgitem.TaggroupSetTagAsNumber("stage angle (degree)", a)
TagGroupAddTagGroupatend( ba,tgitem )
tiltseries.setNumberNote("Tomography:Tracking data:Online tracking data:["+idx+"]:Index in series",idx)

```

```

tgroot2.TagGroupAddTagGroupAtEnd(tgItem2)
tgitem2.TaggroupSetTagAsNumber("stage angle (degree)", a)
TagGroupAddTagGroupAtEnd( ba2,tgItem2)

```

```

tgroot1.TagGroupAddTagGroupAtEnd(tgItem1)
TagGroupAddTagGroupAtEnd( ba1,tgItem1 )
tiltseries.setNumberNote("Meta Data:Dimension info:2:Data:"+idx+"",a)

```

```

SetUsePlane(tiltseries, idx, 1)
}

```

```

}

```

```

////////////////////////////////////

```

```

Object objListener
String messagemap
Number ListenerID

```

```

Class MyListenerClass
{

```

```

number count
String event_desc
MyListenerClass(Object self) yield()
~MyListenerClass(Object self) yield()

```

```

Number GetCount( object self ) { return count; }

```

```

Void SetCount( object self, number c) { count = c; }

```

```

void MyImageAction(object self, number e_fl, Image Img)
{
count++
ImageGetEventMap().deconstructEventFlags( e_fl, event_desc )
objListener.SetCount(count)
}

```

```

}

```

// the reconstruction algorithm in dm does not tell when it is finished. Therefore an image listener was added  
// suggested by Bernhard Schaffer to ensure that the algorithm is ready before the script continues.

```

image dorec(image tiltseries, number plane, image useangles)
{
    number x,y,z
    get3dszsize(tiltseries,x,y,z)

        for( number idx=0; idx<z; idx++ )
        {
            SetUsePlane(tiltseries, idx, getpixel(useangles,idx,0) )
        }

    image p := slice2(tiltseries,0,plane,0,0,x,1,2,z,1)
    image rec := exprsize(x,y,0)

    Launch2DReconstruction("2D WBP Algorithm",rec,p,1)

    messagemap = "data_changed,data_size_changed,data_type_changed,"
    messagemap += "calibration_changed,data_value_changed,slices_changed"
    messagemap += ".MyImageAction"
    objListener = Alloc(MyListenerClass)
    ListenerID = rec.ImageAddEventListener( objListener, messagemap)

    While(objListener.GetCount() != z*2 + 1) 1==2
    rec.ImageRemoveEventListener(ListenerID)

    return rec
}

image front := getfrontimage()
number x,y,z,l,r,b,t,i
get3dszsize(front, x,y,z)
image doplanes := exprsize(z,1,1)
image roifront
imagedisplay frontdisp = imagegetimagedisplay(front,0)
number nroi = imagedisplaycountrois(frontdisp)
if(nroi==1)
{
    roi fRoi = imagedisplaygetroi(frontdisp,0)
    ROIGetRectangle( froi, t, l,b, r )
    if(mod(r-l,2)!=0) r+=1
    number xn = (r - l)/2
    roifront = front[x/2 - xn,t,0,x/2+ xn,b,z]
    //showimage(roifront)
}

settomotags(roifront, mintiltangle, tiltstepsize)
get3dszsize(roifront, x,y,z)
image reca := exprsize(x,y,0)
image rec := exprsize(x,y,y,0)
for(i=0;i<y;i++)
{
    OpenAndSetProgressWindow( "Reconstructing", "Plane", ""+(i)+"/"+y+" ")
    reca = dorec(roifront,i,doplanes)
    rec[0,0,i,x,y,i+1] = reca
}
OpenAndSetProgressWindow( "", "", "" )
showimage(roifront)
//showimage(rec)
TagGroup tags = NewTagGroup()

image newfront := roifront.imageclone()

for(i=0;i<z;i++)
{
    OpenAndSetProgressWindow( "Projecting", "angle", ""+(i)+"/"+z+" ")
    image rot = Rotate3D(rec,0,0,mintiltangle + i*tiltstepsize )
    image sumPr = Project(rot,1,8)
    tags = ImageGetTagGroup(rec)
    TagGroupSetTagAsString( tags,"Projection:Label" , "MaximumProjection")
    image maxPr = Project(rot,1,8)
    newfront[0,0,i,x,y,i+1] = sumPr + weight*maxPr
}

```

```

TagGroupDeleteAllTags( tags)
}
OpenAndSetProgressWindow( "", "", "" )
showimage(newfront)

```

## CCTiltseriesandProjections.s

```

// $BACKGROUND$

// sub-pixel search function from D. Mitchell
Number ImageRefineExtrema(Image img, number &px, number &py)
{
    number tx1, tx2, ty1, ty2
    number numa, numb, numc
    number valX, valY

    // do a parabolic fit in direction x
    numa = img.GetPixel(px-1, py)
    numb = img.GetPixel(px, py)
    numc = img.GetPixel(px+1, py)
    tx2=(numa-numc)/2.0
    tx1=(numa+numc)/2.0-numb
    valX=(2*numa*numc-numa**2-numc**2)/8/(numa+numc-2*numb)+numb

    // do a parabolic fit in direction y
    numa = img.GetPixel(px, py-1)
    numb = img.GetPixel(px, py)
    numc = img.GetPixel(px, py+1)
    ty2=(numa-numc)/2.0
    ty1=(numa+numc)/2.0-numb
    valY=(2*numa*numc-numa**2-numc**2)/8/(numa+numc-2*numb)+numb

    // update pixel position
    px += tx2/(2.0*tx1)
    py += ty2/(2.0*ty1)

    // check to see whether to return minimum or maximum
    If(numa>=numb && numc>=numb) return min(min(valX, valY), numb)
    If(numa<=numb && numc<=numb) return max(max(valX, valY), numb)

    Return numb
}

Image CrossCorrelateWithSubPixelPrecision(image src, image ref, number &pX, \
number &pY, number &qval)
{
    try
    {
        Number sizeX,sizeY
        src.GetSize(sizeX,sizeY)

        ReallImage xcorr = src.CrossCorrelate(ref)

        // find the maximum and its coordinates in correlation image
        Number spotX, spotY
        qval = max(xcorr, spotX, spotY)
        qval = xcorr.ImageRefineExtrema(spotX, spotY)

        pX = spotX
        pY = spotY

        return xcorr
    }
    catch break
}

image refS,projS
gettwoimages("reference and projected", refS,projS)
number x,y,z,i,px,py,q

```

```

get3dsize(refS, x,y,z)

image shifts := exprsize(z,2,0)
image ref := exprsize(x,y,0)
image proj := exprsize(x,y,0)

for(i=0 ; i<z-1 ; i++)
{

OpenAndSetProgressWindow( "Aligning", "Plane", ""+(i)"/"+z+" ")

ref = refS[0,0,i,x,y,i+1]
proj = projS[0,0,i,x,y,i+1]

ifmapplyfilterinplace(ref,"Combined Filter 0")
ifmapplyfilterinplace(proj,"Combined Filter 0")

CrossCorrelateWithSubPixelPrecision(proj,ref ,px, py, q)

setpixel(shifts,i+1,0,(x/2 -px))
setpixel(shifts,i+1,1,(y/2 -py))
}

OpenAndSetProgressWindow( "", "", "")

showimage(shifts)

SetDisplayType( shifts, 4 )
imagedisplay shdisp = imagegetimagedisplay(shifts,0)
LinePlotImageDisplaySetSliceDrawingStyle( shdisp, 0, 1 )
LinePlotImageDisplaySetSliceDrawingStyle( shdisp, 1, 1 )

```

## 7.3 Insight Toolkit Implementation

Insight Toolkit (ITK) [137] originates back to 1999 as a project from US National Library of Medicine of the National Institutes of Health to build an open source segmentation and registration toolkit. It is a cross-platform toolkit and uses the build environment CMake [173] to manage the configuration process. Here 34 different algorithms were integrated into DM environment as a plug-in, configuring a 32-bit version for DM 1.x and a 64-bit version for DM 2.x. [174] The ITK library was configured to be built within the Visual basic 2006 environment using DM version 1.x and within Visual Basic 2008 environment using DM version 2.x.

All algorithms in the library could be used as they were by just including the header and source files provided that they were transformed to functions. The pointer to the image data had to be converted from DM to a form that ITK understands. This was implemented as a separate object to be included in every ITK algorithm. Because in the ITK environment, all variables have to be resolved within build-time, only a pixel type of float was implemented except a few cases where a pixel type of char was necessary. Various algorithms in the field of image manipulation, filters, segmentation, registration and deconvolution were implemented as listed below. A dynamic library was built to be copied to the DM plug-ins folder. These ITK algorithms are then available as simple commands to be used within the DM scripting language. The scripting environment within DM is easy to use high level language with multiple ready-made commands and open source user scripts library [175]. Some of the algorithms require meta-information from the image header, also known as tags in DM. The following algorithms were implemented:

```

Image DiscreteGaussian(Image src_img, float GaussianVariance, long Kernelseize)

Image Project(Image src_img, ulong DimProj, ulong NofThreads)

////////////////////////////////////
Example:

Image ref := getfrontimage()
TagGroup tags = NewTagGroup()
tags = ImageGetTagGroup(ref)

// Use any of the following. If no tags are found Sumprojection is used
//TagGroupSetTagAsString( tags,"Projection:Label" , "MeanProjection" )
//TagGroupSetTagAsString( tags,"Projection:Label" , "MedianProjection" )
//TagGroupSetTagAsString( tags,"Projection:Label" , "StandardDeviationProjection" )

//TagGroupSetTagAsString( tags,"Projection:Label" , "BinaryProjection")
//TagGroupSetTagAsFloat( tags, "Projection:backgroundvalue", 100 ) //for binary
//TagGroupGetTagAsFloat( tags, "Projection:foregroundvalue", 0 ) //for binary

TagGroupSetTagAsString( tags,"Projection:Label" , "MaximumProjection")
//TagGroupSetTagAsString( tags,"Projection:Label" , "MinimumProjection" )

Image mov = Project(ref,2,8)
showimage(mov)
setname(mov,"Z projected")

////////////////////////////////////

Image Rotate3D(Image src_img, float X, float Y, float Z)

Image ObliqueSlice3Dto2D(Image src_img, Image point_center_up)

Image CurvatureAnisotropicDiffusion(Image src_img, float Timestep, float Conductance, ulong Iter)

Image MorphologicalWatershed(Image src_img, float tre, ulong rad, float level,bool fullyconnected)

Image WhiteTopHatByArea(Image src_img, ulong radius)

Image GradientMagnitudeRecursiveGaussian(Image src_img, float Sigma)

Image OtsuMultipleThresholdsCalculator(Image src_img, ulong NumberofHistogramBins, ulong NumberofThresholds)

Image MinMaxCurvatureFlow(Image src_img, ulong Iter, float TimeStep, float Radius)

Image SFDemonsRegistration(Image ref_img, Image mov_img, ulong Iterations, float StandardDeviation, double IntTreshold)

Image SFDemonsRegistration1D(Image ref_img, Image mov_img, ulong Iterations, float StandardDeviation, double IntTreshold, Image
shiftx)

Image SFDemonsRegistration2D(Image ref_img, Image mov_img, ulong Iterations, float StandardDeviation, double IntTreshold, Image
shiftx, Image shifty)

Example

////////////////////////////////////
Image sx,sy,ref,mov,dmov
gettwoimages("r and m",ref,mov)
getsize(mov,x,y)
sx = exprsize(x,y,0)
sy = exprsize(x,y,0)
dmov = SFDemonsRegistration2D(ref, mov, 200, 5.2,sx, sy)
showimage(dmov)
setname(dmov,"Deformed")
showimage(sx)
Setname(sx,"Xcomponent of the vectorfield")
showimage(sy)
Setname(sy,"Ycomponent of the vectorfield")

////////////////////////////////////

```

Image SFDemonsRegistration3D(Image ref\_img, Image mov\_img, ulong Iterations, float StandardDeviation, double IntTreshold, Image shiftx, Image shifty, Image shiftz)

Image SFDemonsRegistration4D(Image ref\_img, Image mov\_img, ulong Iterations, float StandardDeviation, double IntTreshold, Image shiftx, Image shifty, Image shiftz, Image shiftE)

Image CurvatureFlow(Image src\_img, ulong Iterations, float TimeStep)

Image ConnectedThreshold(Image src\_img, float LowerThreshold, float UpperThreshold)

Example:

```
////////////////////////////////////
Image ref := getfrontimage()
TagGroup optitags = NewTagGroup()
optitags = ImageGetTagGroup(ref)

// this filter requires seed points, put them in the tags like this:
TagGroupSetTagAsLong( optitags, "ConnectedThreshold:Seedx1", 57 )
TagGroupSetTagAsLong( optitags, "ConnectedThreshold:Seedy1", 253 )
TagGroupSetTagAsLong( optitags, "ConnectedThreshold:Seedx2", 97 )
TagGroupSetTagAsLong( optitags, "ConnectedThreshold:Seedy2", 453 )

// 3d: Seedz1
Image mov = ConnectedThreshold(ref,0,1000)
Showimage(mov)
```

```
////////////////////////////////////
Image Optimization(Image ref_img, Image mov_img)
```

Example

// In the Optimization function many different ITK optimizing algorithms were gathered under the same function call. It registers a moving // image to a reference image. It requires a metric, a transform, an interpolator and an optimizer specified in the tags.

```
////////////////////////////////////

image ref,mov,dmov
gettwoimages("r and m",ref,mov)

TagGroup optitags = NewTagGroup()

TagGroup metriclist = NewTaglist( )
TagGroup Transformlist = NewTaglist( )
TagGroup Interpolatorlist = NewTaglist( )
TagGroup Optimizerlist = NewTaglist( )

optitags = ImageGetTagGroup(mov)

metriclist = TagGroupGetOrCreateTaggroup( optitags, "Metric" )
Transformlist = TagGroupGetOrCreateTaggroup( optitags, "Transform" )
Interpolatorlist = TagGroupGetOrCreateTaggroup( optitags, "Interpolator" )
Optimizerlist = TagGroupGetOrCreateTaggroup( optitags, "Optimizer" )

/*
TagGroupSetTagAsString( optitags,"Optimizer:Label", "AmoebaOptimizer" )
//TagGroupSetTagAsFloat( optitags, "Optimizer:SimplexSize",0.01 ) TagGroupSetTagAsFloat( optitags, "Optimizer:ParTol", 1e-3 )
TagGroupSetTagAsFloat( optitags, "Optimizer:FuncTol", 1e-3 )
TagGroupSetTagAsLong( optitags, "Optimizer:Iter", 500 )
TagGroupSetTagAsboolean( optitags, "Optimizer:AutomaticInitialSimplex", 1 )
*/

/*
TagGroupSetTagAsString( optitags,"Optimizer:Label", "LBFGBOptimizer" )
TagGroupSetTagAsLong( optitags, "Optimizer:gridSizeOnImage", 24 )
TagGroupSetTagAsLong( optitags, "Optimizer:gridBorderSize", 3 )
TagGroupSetTagAsdouble( optitags, "Optimizer:boundSelect", 0 )
TagGroupSetTagAsdouble( optitags, "Optimizer:upperBound", 0 )
TagGroupSetTagAsdouble( optitags, "Optimizer:lowerBound", 0 )
TagGroupSetTagAsdouble( optitags, "Optimizer:SetCostFunctionConvergenceFactor", 1e7 )
TagGroupSetTagAsdouble( optitags, "Optimizer:SetProjectedGradientTolerance", 1e-4 )
TagGroupSetTagAsLong( optitags, "Optimizer:Iter", 500 )
TagGroupSetTagAsLong( optitags, "Optimizer:Eval", 500 )
*/
```

```

TagGroupSetTagAsLong( optitags, "Optimizer:Correct", 24 )
*/

/*
TagGroupSetTagAsString( optitags,"Optimizer:Label", "PowellOptimizer" )
TagGroupSetTagAsDouble( optitags, "Optimizer:ValTol", 1e-51 )
TagGroupSetTagAsDouble( optitags, "Optimizer:StepTol", 0.00002 )
TagGroupSetTagAsDouble( optitags, "Optimizer:StepLength",0.001 ) //make smaller for samples outside
TagGroupSetTagAsLong( optitags, "Optimizer:Iter", 600 )
TagGroupSetTagAsLong( optitags, "Optimizer:MaxLineIter", 50 )
TagGroupSetTagAsboolean( optitags, "Optimizer:SetMaximize", 1)

*/

/*
TagGroupSetTagAsString( optitags,"Optimizer:Label", "OnePlusOneEvolutionaryOptimizer" )
TagGroupSetTagAsdouble( optitags, "Optimizer:Epsilon", 1 )
TagGroupSetTagAsLong( optitags, "Optimizer:Iter", 4000 )
TagGroupSetTagAsLong( optitags, "Optimizer:Initialize", 10 )
TagGroupSetTagAsboolean( optitags, "Optimizer:SetMaximize",1)

*/

/*
TagGroupSetTagAsString( optitags,"Optimizer:Label", "LevenbergMarquardtOptimizer" )
TagGroupSetTagAsdouble( optitags, "Optimizer:gradientTolerance", 1e-5 )
TagGroupSetTagAsdouble( optitags, "Optimizer:valueTolerance", 1e-5 )
TagGroupSetTagAsdouble( optitags, "Optimizer:epsilonFunction", 1e-6 )
TagGroupSetTagAsdouble( optitags, "Optimizer:scale", 0.01 )
TagGroupSetTagAsLong( optitags, "Optimizer:Iter", 100 )

*/

/*
TagGroupSetTagAsString( optitags,"Optimizer:Label", "GradientDescentOptimizer" )
TagGroupSetTagAsfloat( optitags, "Optimizer:MaximumStepLength",0.5)
TagGroupSetTagAsLong( optitags, "Optimizer:Iter", 400 )
TagGroupSetTagAsfloat( optitags, "Optimizer:MinimumStepLength",0.001)

*/

/*
TagGroupSetTagAsString( optitags,"Optimizer:Label", "LBFGSOptimizer" )
TagGroupSetTagAsdouble( optitags, "Optimizer:SetGradientConvergenceTolerance", 0.05 )
TagGroupSetTagAsdouble( optitags, "Optimizer:SetLineSearchAccuracy", 0.9 )
TagGroupSetTagAsdouble( optitags, "Optimizer:SetDefaultStepLength", 1.5)
TagGroupSetTagAsLong( optitags, "Optimizer:Eval", 1000 )
TagGroupSetTagAsboolean( optitags, "Optimizer:SetTrace", 1)

*/

/*
TagGroupSetTagAsString( optitags,"Transform:Label", "TranslationTransform" )
TagGroupSetTagAsdouble( optitags, "Transform:InitParX", 0 )
TagGroupSetTagAsdouble( optitags, "Transform:InitParY", 0 )
*/

/*
TagGroupSetTagAsString( optitags,"Transform:Label", "AffineTransform" )
TagGroupSetTagAsdouble( optitags, "Transform:InitParX", 0.001 )
TagGroupSetTagAsdouble( optitags, "Transform:InitParY", 0.001 )
TagGroupSetTagAsdouble( optitags, "Transform:R00", 1000 )
TagGroupSetTagAsdouble( optitags, "Transform:R01",1000)
TagGroupSetTagAsdouble( optitags, "Transform:R10", 1000)
TagGroupSetTagAsdouble( optitags, "Transform:R11", 1000)
*/

/*
TagGroupSetTagAsString( optitags,"Transform:Label", "CenteredRigid2DTransform" )
TagGroupSetTagAsdouble( optitags, "Transform:0", 0.001 )
TagGroupSetTagAsdouble( optitags, "Transform:1", 0.001 )
TagGroupSetTagAsdouble( optitags, "Transform:2", 0.001 )
TagGroupSetTagAsdouble( optitags, "Transform:3",0.001)
TagGroupSetTagAsdouble( optitags, "Transform:4", 0.001 )

*/

```

```

/*
TagGroupSetTagAsString( optitags,"Metric:Label", "MatchCardinalityImageToImageMetric" )
TagGroupSetTagAsBoolean( optitags, "Metric:MeasureMatches", 0)
*/

/* maximize 0
TagGroupSetTagAsString( optitags,"Metric:Label", "MattesMutualInformationImageToImageMetric" )
TagGroupSetTagAsLong( optitags, "Metric:numberOfBins", 24)
TagGroupSetTagAsLong( optitags, "Metric:numberOfSamples", 1000)
TagGroupSetTagAsBoolean( optitags, "Metric:UseAllPixelOn", 0)
*/

/* maximize 1
TagGroupSetTagAsString( optitags,"Metric:Label", "NormalizedMutualInformationHistogramImageToImageMetric" )
TagGroupSetTagAsLong( optitags, "Metric:numberOfBins", 240)
*/

/*
TagGroupSetTagAsString( optitags,"Metric:Label", "MeanReciprocalSquareDifferenceImageToImageMetric" )
TagGroupSetTagAsDouble( optitags, "Metric:SetLambda",0.5 )

*/

TagGroupSetTagAsString( optitags,"Interpolator:Label", "NearestNeighborInterpolateImageFunction" )

//TagGroupSetTagAsString( optitags,"Interpolator:Label", "LinearInterpolateImageFunction" )
//TagGroupSetTagAsString( optitags,"Interpolator:Label", "BSplineInterpolateImageFunction" )**

dmov = Optimization( ref, mov)
showimage(dmov)

////////////////////////////////////
Image CannyEdgeDetection(Image src_img, float variance, float UpThreshold, float LowThreshold)

Image HoughTransform2DCircles(Image src_img, ulong NofCirc, float MinRadius, float MaxRadius)

Image HoughTransform2DLines(Image src_img, ulong NofLines)

Image Sigmoid(Image src_img, float MinOut,float MaxOut, float Alpha, float Beta)

Image FastMarching(Image src_img, float TimeThreshold, float StoppingTime)

Example:

////////////////////////////////////
Image ref := getfrontimage()

TagGroup optitags = NewTagGroup()
optitags = ImageGetTagGroup(mov)

TagGroupSetTagAsLong( optitags, "FastMarching:seedPositionx1", 57 )
TagGroupSetTagAsLong( optitags,"FastMarching:seedPositiony1", 253 )

Image mov = FastMarching( mov,335.1,283.5)
showimage(mov)
////////////////////////////////////

Image NeighborhoodConnected( Image src_img, float LowerThreshold, float UpperThreshold)

Example:

////////////////////////////////////
Image ref := getfrontimage()
TagGroup optitags = NewTagGroup()
optitags = ImageGetTagGroup(ref)

TagGroupSetTagAsLong( optitags, "NeighborhoodConnected:Seedx1", 57 )
TagGroupSetTagAsLong( optitags, "NeighborhoodConnected:Seedy1", 253 )

Image mov = NeighborhoodConnected(ref,0,1500)
showimage(mov)

////////////////////////////////////

```

Image ConfidenceConnected(Image src\_img, float Multiplier, ulong radius, ulong lter)

Example:

```
////////////////////////////////////
Image ref := getfrontimage()
TagGroup optitags = NewTagGroup()
optitags = ImageGetTagGroup(ref)

TagGroupSetTagAsLong( optitags, "ConfidenceConnected:Seedx1", 146 )
TagGroupSetTagAsLong( optitags, "ConfidenceConnected:Seedy1", 225 )

Image mov = ConfidenceConnected(ref,2.3,2,4)
Showimage(mov)
```

float MutuallInformation(Image ref\_img, Image mov\_img, float Variance, float MetVariance, float constant, float Learningrate, ulong iteration)

Example:

```
////////////////////////////////////
number mutinf, normmutinf
image ref,mov
gettwoimages("r and m",ref,mov)

number numberentref = mutualinformation(ref,mov,10,512,1)
number entrmov = mutualinformation(ref,mov,10,1,512)
number mutual = mutualinformation(ref,mov,10,512,512)

mutinf = entref + entrmov - mutual
normmutinf = (2*mutual)/(entref+entrmov)

result("ref entropy: "+entref+" mov entropy: "+entrmov+" Mutual information: "+mutinf+" normalized mutual information:
"+normmutinf+"\n")
```

Image DemonsRegistrationFunction2D(Image ref\_img, Image mov\_img, Image Iterations, Image Shiftx, Image Shifty)

Example

```
////////////////////////////////////7
Image ref, mov
Number x,y
gettwoimages("r and m",ref,mov)
getsize(mov,x,y)
image sx = exprsize(x,y,0)
image sy = exprsize(x,y,0)

TagGroup optitags = NewTagGroup()
optitags = ImageGetTagGroup(mov)

TagGroupSetTagAsLong( optitags, "useHistogramMatching", 0 )
TagGroupSetTagAsLong( optitags,"updateRule", 0 )
TagGroupSetTagAsDouble( optitags, "maxStepLength" , 2 )
TagGroupSetTagAsLong( optitags, "gradientType", 0 )
TagGroupSetTagAsDouble( optitags,"sigmaDef" , 3 )
TagGroupSetTagAsDouble( optitags, "sigmaUp" , 0 )

image iterations = exprsize(3,1,0)
setpixel(iterations,0,0,5)
setpixel(iterations,1,0,5)
setpixel(iterations,2,0,5)

image dmov = DemonsRegistrationFunction2D(ref,mov,iterations,sx,sy)
showimage(dmov)
setname(dmov,"Deformed")
showimage(sx)
Setname(sx,"X component of the vectorfield")
showimage(sy)
Setname(sy,"Y component of the vectorfield")
```

```

void ImageMoments(Image src_img)
// see image tags after this function

Image Houghsphericalvoting2D(Image src_img, ulong NofSpheres , float minRad, float maxRad)

Image Houghsphericalvoting3D(Image src_img, ulong NofSpheres , float minRad, float maxRad)

Image plane(Image src_img, ulong layer)

Image GPUDemons(Image src_img, Image Xflow, Image Yflow, ulong lter)

```

In the Visual Studio 2008 build environment the following simple example is shown to build ITK functions as dynamic libraries into DM. All the proper DM and ITK dependencies has to be properly linked. The following is built with ITK version 3.2 and later versions are not supported. The DM SDK x64 can only be acquired by a request at Gatan.

First the ITK 3.2 has to be built and checked that the Hello World example is working. Here is the CmakeLists.txt for Cmake, the path to the DMSDK folder must be set accordingly.

```

CMAKE_MINIMUM_REQUIRED(VERSION 2.8)

PROJECT(DMITK2)
SET(DMITK2 CMAKE_CXX_FLAGS_RELEASE)
INCLUDE_DIRECTORIES( C:/ProgramData/Gatan/DMSDK/lib/x64 )

find_package(ITK REQUIRED)
include(${ITK_USE_FILE})

ADD_LIBRARY(DMITK2 SHARED

ITKMain.cpp
ITKMain.h

DiscreteGaussianFilter.cpp
DiscreteGaussianFilter.h

dmtoitk1.cpp
dmtoitk1.h
dmtoitk2.cpp
dmtoitk2.h
dmtoitk3.cpp
dmtoitk3.h
dmtoitk4.cpp
dmtoitk4.h
)

TARGET_LINK_LIBRARIES(DMITK2 ${ITK_LIBRARIES} DMPlugInBasic.lib Foundation.lib)

```

Now DMITK2.sln can be opened in the build folder in VS 2008. Here is an example of dmtoitk2.h and dmtoitk2.cpp. The number refers to the image dimensions. New files for all other dimensions has to be created, see comments in code. It is also useful to change the output path directly to the DM plugins folder. In

Projects->DMITK2 Properties go to Configuration Properties->linker->General and change the output to C:\Program Files\Gatan\Plugins\DMITK2.dll .

## dmtoitk2.h

```
#ifndef GATANPLUGIN_USES_LIBRARY_VERSION
#define _GATANPLUGIN_USES_LIBRARY_VERSION 2
#endif

#ifdef _DMPluginBasic_h_INCLUDED__
#include "DMPluginBasic.h"
#endif

#ifdef _itkImage_h_INCLUDED__
#include <itkImage.h>
#endif

#include "itkImportImageFilter.h"
using namespace Gatan;
typedef itk::Image< float, 2 > ItkImage2; // <float, 3> for 3D
ItkImage2::Pointer dmtoitk2(DM::Image );
```

## dmtoitk2.cpp

```
#if defined(_MSC_VER)
#pragma warning ( disable : 4786 )
#endif

typedef float PixelType;
#include "dmtoitk2.h"

ItkImage2::Pointer dmtoitk2(DM::Image input )
{
    PLUG_IN_ENTRY

    typedef itk::ImportImageFilter< PixelType, 2> ImportFilterType; // <PixelType,3> for 3D images!
    ImportFilterType::Pointer importFilter = ImportFilterType::New();

    ImportFilterType::SizeType size;

    size[0] = input.GetDimensionSize( 0 );
    size[1] = input.GetDimensionSize( 1 );
    //size[2] for 3D

    ImportFilterType::IndexType start;
    start.Fill( 0 );

    ImportFilterType::RegionType region;
    region.SetIndex( start );
    region.SetSize( size );

    importFilter->SetRegion( region );

    double origin[ 2 ];
    origin[0] = 0.0; // X coordinate
    origin[1] = 0.0; // Y coordinate
    // origin[2] for 3D

    importFilter->SetOrigin( origin );

    double spacing[ 2 ];
    spacing[0] = 1.0; // along X direction
    spacing[1] = 1.0; // along Y direction
    // spacing[2] for 3D

    importFilter->SetSpacing( spacing );

    const unsigned int numberOfPixels = size[0]*size[1]; // *size[2] for 3D
    const bool importImageFilterWillOwnTheBuffer = false;

    Gatan::Plugin::ImageDataLocker inputL( input,Plugin::ImageDataLocker::lock_data_CONTIGUOUS );

    PixelType *input_data ;

    input_data = (PixelType *) ( inputL.get_image_data().get_data() );

    importFilter->SetImportPointer(input_data, numberOfPixels,
    importImageFilterWillOwnTheBuffer );
```

```

importFilter->Update();

ItkImage2::Pointer output = importFilter->GetOutput();

output->DisconnectPipeline();

return output;
}
PLUG_IN_EXIT
}

```

## DiscreteGaussianFilter.h

```

#ifndef _GATAN_USE_STL_STRING
#define _GATAN_USE_STL_STRING
#endif

#ifndef GATANPLUGIN_USES_LIBRARY_VERSION
#define GATANPLUGIN_USES_LIBRARY_VERSION 2
#endif

#ifndef DMPlugInBasic_h_INCLUDED__
#include "DMPlugInBasic.h"
#endif

#include "dmtoitk1.h"
#include "dmtoitk2.h"
#include "dmtoitk3.h"
#include "dmtoitk4.h"

DM_ImageToken_1Ref Discretegaussian( DM_ImageToken,float ,ulong );

```

## DiscreteGaussianFilter.cpp

```

#ifdef _MSC_VER
#pragma warning ( disable : 4786 )
#endif
#include "itkDiscreteGaussianImageFilter.h"

#include "DiscreteGaussianFilter.h"

DM_ImageToken_1Ref Discretegaussian( DM_ImageToken src_img_inn,float gaussianVariance ,ulong maxKernelWidth )
{
    DM::Image l_deriv_img_out;
    DM::Image src_img_in = src_img_inn;

    ulong dim = src_img_in.GetNumDimensions();
    ulong x,y,z,E;

    bool del=FALSE;
    if ( src_img_in.GetDataTypes() != ImageData::REAL4_DATA )
    {
        DM::Image src_img_in1 = DM::ImageClone(src_img_in);
        DM::ConvertToFloat(src_img_in1);
        src_img_in =src_img_in1;
        DM::DeleteImage( src_img_in1 );
        del=TRUE;
    }

    switch( dim ) {
    case 1 :
        {
            x = DM::ImageGetDimensionSize( src_img_in, 0 );
            l_deriv_img_out = DM::NewImage( ( std::string("gaussian of ") + src_img_in.GetName(), ImageData::REAL4_DATA, x);

            break;
        }
    }
}

```

```

case 2 :
{
x = DM::ImageGetDimensionSize( src_img_in, 0 );
y = DM::ImageGetDimensionSize( src_img_in, 1 );
l_deriv_img_out = DM::NewImage( ( std::string("gaussian of ") ).c_str(), ImageData::REAL4_DATA, x,y);

break;
}
case 3 :
{
x = DM::ImageGetDimensionSize( src_img_in, 0 );
y = DM::ImageGetDimensionSize( src_img_in, 1 );
z = DM::ImageGetDimensionSize( src_img_in, 2 );
l_deriv_img_out = DM::NewImage( ( std::string("gaussian of ") ).c_str(), ImageData::REAL4_DATA, x,y,z);

break;
}
case 4 :
{
x = DM::ImageGetDimensionSize( src_img_in, 0 );
y = DM::ImageGetDimensionSize( src_img_in, 1 );
z = DM::ImageGetDimensionSize( src_img_in, 2 );
E = DM::ImageGetDimensionSize( src_img_in, 3 );
l_deriv_img_out = DM::NewImage( ( std::string("gaussian of ") ).c_str(), ImageData::REAL4_DATA, x,y,z,E);

break;
}
default :
{
DM::OkDialog( ( std::string("Well that's not fair...") ).c_str() );
//goto stop;
}
}

```

```

PlugIn::ImageDataLocker l_deriv_img_outl( l_deriv_img_out, PlugIn::ImageDataLocker::lock_data_CONTIGUOUS );
float *l_deriv_img_out_data = (float*)(l_deriv_img_outl.get_image_data().get_data() );

```

PLUG\_IN\_ENTRY

```

switch( dim ) {
case 1 :
{
typedef itk::Image<float, 1> ImageType;
typedef itk::DiscreteGaussianImageFilter<ImageType, ImageType > FilterType;
FilterType::Pointer filter = FilterType::New();

filter->SetInput( dmtoitk1(src_img_in) );

filter->SetMaximumKernelWidth( maxKernelWidth );

filter->GetOutput() ->GetPixelContainer()->SetImportPointer( l_deriv_img_out_data, x, false );
filter->GetOutput()->Allocate();
filter->Update();

break;
}
case 2 :
{
typedef itk::Image<float, 2> ImageType;
typedef itk::DiscreteGaussianImageFilter<ImageType, ImageType > FilterType;
FilterType::Pointer filter = FilterType::New();

filter->SetInput( dmtoitk2(src_img_in) );

filter->SetVariance( gaussianVariance );
filter->SetMaximumKernelWidth( maxKernelWidth );

filter->GetOutput() ->GetPixelContainer()->SetImportPointer( l_deriv_img_out_data, x*y, false );
filter->GetOutput()->Allocate();
filter->Update();

break;
}
case 3 :
{
typedef itk::Image<float, 3> ImageType;
typedef itk::DiscreteGaussianImageFilter<ImageType, ImageType > FilterType;

```

```

        FilterType::Pointer filter = FilterType::New();

        filter->SetInput( dmtoitk3(src_img_in) );

        filter->SetVariance( gaussianVariance );
        filter->SetMaximumKernelWidth( maxKernelWidth );

        filter->GetOutput() ->GetPixelContainer()->SetImportPointer( l_deriv_img_out_data, x*y*z, false );
        filter->GetOutput()->Allocate();
        filter->Update();

        break;
    }
case 4 :
    {
        typedef itk::Image<float, 4> ImageType;
        typedef itk::DiscreteGaussianImageFilter<ImageType, ImageType > FilterType;
        FilterType::Pointer filter = FilterType::New();

        filter->SetInput( dmtoitk4(src_img_in) );

        filter->SetVariance( gaussianVariance );
        filter->SetMaximumKernelWidth( maxKernelWidth );

        filter->GetOutput() ->GetPixelContainer()->SetImportPointer( l_deriv_img_out_data, x*y*z*E, false );
        filter->GetOutput()->Allocate();
        filter->Update();

        break;
    }
default :
    {
        DM::OkDialog( ( std::string("Well that's not fair...") ).c_str() );
        //goto stop;
    }
}

PLUG_IN_EXIT

if(del) DM::DeleteImage( src_img_in );
l_deriv_img_outl.MarkDataChanged();
return l_deriv_img_out.release();
}

```

## ITKMain.h

```

#ifndef _GATAN_USE_STL_STRING
#define _GATAN_USE_STL_STRING
#endif

#ifndef GATANPLUGIN_USES_LIBRARY_VERSION
#define _GATANPLUGIN_USES_LIBRARY_VERSION 2
#endif

#ifndef _DMPlugInBasic_h_INCLUDED__
#include "DMPlugInBasic.h"
#endif

#ifndef _GATANPLUGIN_USE_CLASS_PLUGINMAIN
#define _GATANPLUGIN_USE_CLASS_PLUGINMAIN
#endif

#ifndef _DMPlugInMain_h_INCLUDED__
#include "DMPlugInMain.h"
#endif

#ifndef _itkImage_h_INCLUDED__
#include <itkImage.h>
#endif

```

## ITKMain.cpp

```
#define _AFXDLL
#ifdef VC_EXTRALEAN
#define VC_EXTRALEAN // Exclude rarely-used stuff from Windows headers
#endif

#ifdef WINVER // Allow use of features specific to Windows 95 and Windows NT 4 or later.
#define WINVER 0x0501 // Change this to the appropriate value to target Windows 98 and Windows 2000 or later.
#endif

#ifdef _WIN32_WINNT // Allow use of features specific to Windows NT 4 or later.
#define _WIN32_WINNT 0x0501 // Change this to the appropriate value to target WindowsXP or later.
#endif

#ifdef _WIN32_WINDOWS // Allow use of features specific to Windows 98 or later.
#define _WIN32_WINDOWS 0x0410 // Change this to the appropriate value to target Windows Me or later.
#endif

#ifdef _WIN32_IE // Allow use of features specific to IE 4.0 or later.
#define _WIN32_IE 0x0700 // Change this to the appropriate value to target IE 5.0 or later.
#endif

#define _ATL_CSTRING_EXPLICIT_CONSTRUCTORS // some CString constructors will be explicit

#ifdef _BIND_TO_CURRENT_CRT_VERSION
#define _BIND_TO_CURRENT_CRT_VERSION 1
#endif

#ifdef _BIND_TO_CURRENT_VCLIBS_VERSION // Force the CRT/MFC version to be put into the manifest
#define _BIND_TO_CURRENT_VCLIBS_VERSION 1
#endif

#define _ATL_CSTRING_EXPLICIT_CONSTRUCTORS // some CString constructors will be explicit

#include <afxwin.h> // MFC core and standard components
#include <afxext.h> // MFC extensions

#define _GATAN_USE_MFC
#define _GATANPLUGIN_USES_LIBRARY_VERSION 2
#define _GATAN_USE_STL_STRING

#ifdef DMPluginBasic_h_INCLUDED__
#include "DMPluginBasic.h"
#endif

#define _GATAN_USE_MFC
#define _GATANPLUGIN_USE_CLASS_PLUGINMAIN
#include "DMPluginMain.h"

#ifdef itkImage_h_INCLUDED__
#include <itkImage.h>
#endif

using namespace Gatan;

extern AFX_EXTENSION_MODULE gPluginExtensionModule = { NULL, NULL };

#include "DiscreteGaussianFilter.h"

class ITKMain : public GatanPlugin::PluginMain
{
    virtual void Start();
    virtual void Run();
    virtual void Cleanup();
    virtual void End();
    virtual void AddDMFunctions( void );
    virtual void RemoveDMFunctions( void );
    typedef std::vector<Gatan::DM::Function> FunctionList;

    FunctionList fFunctionList;
};

void ITKMain::Start()
{
    this->AddDMFunctions();
}

void ITKMain::Run()
{
}

void ITKMain::Cleanup()
```

```

{
this->RemoveDMFunctions();
}

void ITKMain::End()
{
}

void ITKMain::AddDMFunctions( void )
{
    fFunctionList.push_back( AddFunction("BasicImage Discretegaussian( BasicImage src_img,float GaussianVariance,ulong
KernelSize)",(void *) Discretegaussian ) );
}

void ITKMain::RemoveDMFunctions( void )
{
    FunctionList::const_iterator functionPtr;

    for(      functionPtr = fFunctionList.begin();
            functionPtr != fFunctionList.end();
            ++functionPtr )
    {
        RemoveFunction( *functionPtr );
    }
}

ITKMain gITKMain;

```

## 7.4 Amira Modules

The Amira modules were built in the Visual Studio environment. A very useful feature of Amira in terms of new analysis methods is the developer module for custom made code. It gives unrestricted access to the data in various forms (stacked coordinates, triangulated coordinates) as well as readymade functions to manipulate them. The following code shows how to get the local curvature distribution of particles within a template. The names of the files do not have any meaning. In Amira a segmentation of the template with particles has to be made. For the template the first material should have intensity value of 1 and for particles as the second material, intensity should be 2. A surface from the label field can be generated with Compute->SurfaceGen. There should be 3 patches in the .surf file. The .surf file should be saved at this point to avoid some problems within the code later. Compute->GetCurvature can be used to calculate the curvature values. The Method-> On Vertices should be used and within this work, Shape Index was used as output. After compiling the below code a Local->DisplayVertices3 option should be available to the .surf file. A new module will appear, where the user can choose the number of inner rings, that is, how many rings one wants to exclude from the calculation. Zero means branching points that are touching the particles. For the polyfit method 2 or 3 is recommended to exclude the first rings. A number of outer rings should be set as well, that is, how many rings one wants to include in the calculation, for example 8. After pressing find and then calculate, a spreadsheet for local curvatures appears, these are the values for the ring method as explained in this thesis. The point coordinates spreadsheet is for Matlab for further analysis with the polyfit method.

Appropriate libraries should be included. The Package file in the amira installation folder /src should look like this:

```

set PACKAGE {mypackage}

set LIBS {
  hxplot hxtime hxsurface hxcolor hxfield
  Amira amiramesh mclib oiv tcl hxitk itk
  hxstatistics hxsurftools hxcluster hxgridgen
}

set SHARE {
  share/resources/mypackage.rc
}

```

## MyDisplayVertices3.h

```

////////////////////////////////////
//
// Example of a display module (version 3)
//
////////////////////////////////////
#ifndef MY_DISPLAY_VERTICES3_H
#define MY_DISPLAY_VERTICES3_H

#include <Amira/HxCompModule.h> // include declaration of base class
#include <mypackage/mypackageAPI.h> // storage-class specification
#include <Amira/HxPortButtonList.h>
#include <hxstatistics/HxSpreadSheet.h>
#include <Amira/HxPortFloatTextN.h> // provides float text input

class MYPACKAGE_API MyDisplayVertices3 : public HxCompModule
{
  HX_HEADER(MyDisplayVertices3);

public:
  // Constructor.
  MyDisplayVertices3();

  // Destructor.
  ~MyDisplayVertices3();

  HxPortButtonList portShowSpreadSheet1;
  HxPortButtonList portShowSpreadSheet;

  // A port providing float text input fields.

  HxPortFloatTextN portRange1;
  HxPortFloatTextN portRange;

  virtual void compute();
  virtual void update();

protected:
  HxSpreadSheet *mySpreadSheet;
  HxSpreadSheet *mySpreadSheet1;
};

#endif

```

## MyDisplayVertices3.cpp

```

////////////////////////////////////
//
// Example of a compute module (version 1)
//
////////////////////////////////////

#include <mclib/McSmallArray.h>

```

```

#include <mclib/McDArray.h>
#include <Amira/HxMessage.h>
#include <hxsurface/HxSurface.h> // class representing a surface
#include <mypackage/MyDisplayVertices3.h> // header of this class
#include <hxsurface/HxSurfaceField.h>
#include <hxsurface/HxSurfaceScalarField.h>
#include <hxsurface/Surface.h>
#include <Amira/HxWorkArea.h>

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

HX_INIT_CLASS(MyDisplayVertices3,HxCompModule)

MyDisplayVertices3::MyDisplayVertices3() :
    HxCompModule(HxSurface::getClassTypeId(),
        portShowSpreadSheet(this,"Local Curvature",1),
        portShowSpreadSheet1(this,"Find Contours",1),
        portRange1(this,"nOuterRings",1),
        portRange(this,"nInnerRings",1)
    )
{
    portRange1.setValue(0,3);
    portRange.setValue(0,0);
    mySpreadSheet = 0;
    mySpreadSheet1 = 0;

    portShowSpreadSheet.setLabel(0,"Calculate");
    portShowSpreadSheet1.setLabel(0,"Find");
}

MyDisplayVertices3::~MyDisplayVertices3()
{
}

void MyDisplayVertices3::update()
{
    if (portShowSpreadSheet1.wasHit())
    {
        HxSurface* surface1 = (HxSurface*) portData.source();
        if (!surface1) { return; }

        surface1->patches[0]->boundaryId = 1;
        surface1->patches[1]->boundaryId = 2;
        surface1->patches[2]->boundaryId = 3;
        surface1->recompute();

        surface1->recomputeNoReNumberPoints();
        int da = surface1 ->removeUnusedPoints();
        surface1 ->reNumberPoints();
        theMsg->printf("Removed %d points",da);
        resultChanged(0);

    }//if (portShowSpreadSheet1.wasHit()) {
}

void MyDisplayVertices3::compute()
{
    if (portShowSpreadSheet.wasHit()) {

        int i,nValues=0;
        int nRings = portRange1.getValue(0);
        int nInnerRings = portRange.getValue(0);
        if(nRings<nInnerRings){ theMsg->printf("nOuterRing must be >= nInnerRings!");return;}
        // Access input object (portData is inherited from HxModule):

        HxSurface* surface = (HxSurface*) portData.source();
        if(!surface) return;
    }
}

```

```

if(surface->contours.size() == 0){ theMsg->printf("Do Find first!");return;}

int nPoints = surface -> getNumPoints();

// Search for surfacescalarfields connectected to surface

    for (i=0; i<surface->downStreamConnections.size(); i++)
    {
        HxSurfaceScalarField* field = (HxSurfaceScalarField*) portData.source() -> downStreamConnections[i] -> object();

            if (field->isOfType(HxSurfaceScalarField::getClassTypeId()))
            {
                nValues = field -> nValues();
                if (nValues == nPoints) { theMsg->printf("Found the curvature field!"); break; }
            }
        }
    if(nValues==0){ theMsg->printf("No connected fields found!"); return;}

theWorkArea->busy();

HxSurfaceScalarField* field = (HxSurfaceScalarField*) portData.source() -> downStreamConnections[i] -> object();

// Get the curvature values
float* v = field -> dataPtr();

mySpreadSheet = new HxSpreadSheet();
mySpreadSheet->setNumRows(0);
mySpreadSheet->addColumn("AreaRatio", HxSpreadSheet::Column::FLOAT);
mySpreadSheet->addColumn("0", HxSpreadSheet::Column::FLOAT);

mySpreadSheet1 = new HxSpreadSheet();
mySpreadSheet1->setNumRows(0);
mySpreadSheet1->addColumn("", HxSpreadSheet::Column::FLOAT);
mySpreadSheet1->addColumn("", HxSpreadSheet::Column::FLOAT);
mySpreadSheet1->addColumn("", HxSpreadSheet::Column::FLOAT);
mySpreadSheet1->addColumn("", HxSpreadSheet::Column::INT);
mySpreadSheet1->addColumn("", HxSpreadSheet::Column::FLOAT);
int rowcount=0;

surface -> computeTrianglesPerPoint();
surface -> computeNormalsPerVertexIndexed();
//////////////////////////////////////AREARATIO//////////////////////////////////////
McDArray<int> matpatches;
McDArray<int> pats;

//For every contour, that is for every particle
for(int j=0;j< surface->contours.size();j++)
{
    //theMsg->printf("No %d",surface->contours[j]->patches.size());
    int patsize = surface->contours[j]->patches.size();
    int patok1=0;
    int patok2=0;
    int patok3=1;
    int output=0;
    int input=0;

    for(int g=0;g< patsize;g++)
    {
        if(surface->patches[surface->contours[j]->patches[g]]->boundaryId == 2){ patok1 = 1;
        if(surface->patches[surface->contours[j]->patches[g]]->boundaryId == 3){ patok2 = 1; input=g;}
    }

    for(int u=0;u< pats.size();u++)
    {
        if(surface->contours[j]->patches[output] == pats[u]){ patok3 = 0; }
        if(surface->contours[j]->patches[input] == pats[u]){ patok3 = 0; }
    }
}
theMsg->printf("No %d %d %d %d",j,patok1,patok2,patok3);
// particles' contours always should have 3 patches

```

```

//if(surface->contours[j]->patches.size() == 3)
if(patok1 == 1 && patok2 == 1 && patok3 ==1)
{
pats.insert(pats.size(),surface->contours[j]->patches[outpat])      ;
pats.insert(pats.size(),surface->contours[j]->patches[inpat]);

double outarea = surface->getPatchArea(surface->contours[j]->patches[outpat]);
double inarea = surface->getPatchArea(surface->contours[j]->patches[inpat]);

// set the AreaRatio to column number 1, particles fully outside = 1, particles fully inside = 0
mySpreadSheet->setNumRows(rowcount+1);
mySpreadSheet->columns[0].setValue(rowcount,(float) outarea/(outarea + inarea));
rowcount++;

//For visualization
// for(i=0;i< surface->contours[j]->points.size();i++) v[surface->contours[j]->points[i]] = (j+1)*1000;
}

//With real data, matrix can be broken, collect here the index of the matrix patches
if(surface->contours[j]->patches.size() == 1)
{
matpatches.insert(matpatches.size(),surface->contours[j]->patches[0]);
}

} //for(int j=0;j< surface->contours.size();j++)

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

if(matpatches.size()==0) matpatches.insert(matpatches.size(),0);
int nParticles = rowcount;
theMsg->printf("Number of particles: %d", nParticles );

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////Mean Curvature RINGS////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

//Loop for the number of rings to calculate the mean curvature of every ring.
// Here 1.ring is the contourpoints, then neighboring points are searched that belong only to the patch 0, matrix

McDArray< McDArray<float> > curvatures(nRings+1);
McDArray< McDArray<int> > ringpointindexes(nParticles );
McDArray< McDArray<float> > center(3);
McDArray<int> pats1;
int cnt=0;

// mean curvatures for contourpoints
curvatures[0].resize(nParticles );
curvatures[0].fill(0);

center[0].resize(nParticles );
center[1].resize(nParticles );
center[2].resize(nParticles );

McDArray<McVec3f> normalf(nParticles);

// First the contour points for every particle
for(int j=0;j< surface->contours.size();j++)
{
float mean =0;
float xcenter=0,ycenter=0,zcenter=0;
int patsize = surface->contours[j]->patches.size();
int patok1=0;
int patok2=0;
int patok3=1;
int outpat=0;
int inpat=0;

```

```

        for(int g=0;g< patsize;g++)
        {
            if(surface->patches[surface->contours[j]]->patches[g]->boundaryId == 2){ patok1 =1;
            if(surface->patches[surface->contours[j]]->patches[g]->boundaryId == 3){ patok2 =1; inpat=g;}
        }

        for(int u=0;u< pats1.size();u++)
        {
            if(surface->contours[j]->patches[outpat] == pats1[u]){ patok3 =0; }
            if(surface->contours[j]->patches[inpat] == pats1[u]){ patok3 =0; }
        }

        // contour of the matrix edge not included
        theMsg->printf("No %d %d %d %d",j,patok1,patok2,patok3);
// if(surface->contours[j]->patches.size() == 3)
if(patok1 == 1 && patok2 == 1 && patok3 ==1)
{

        pats1.insert(pats1.size(),surface->contours[j]->patches[outpat]) ;
        pats1.insert(pats1.size(),surface->contours[j]->patches[inpat]);

ringpointindexes[cnt] = surface->contours[j]->points ;

        //Get the mean curvature of the contourpoints
        for(i=0;i<surface->contours[j]->points.size() ;i++)
        {
            mean = mean + v[surface->contours[j]->points[i]];

        //Get the center coordinates

        McVec3f p;
        p = surface ->points[surface->contours[j]->points[i]];
        xcenter = xcenter + p[0];
        ycenter = ycenter + p[1];
        zcenter = zcenter + p[2];

        }

        //Get the contourpoints mean curvatures for every particle
        curvatures[0][cnt] = mean/(surface->contours[j]->points.size());

//Get the center coordinates for every particles
        center[0][cnt] = xcenter/(surface->contours[j]->points.size());
        center[1][cnt] = ycenter/(surface->contours[j]->points.size());
        center[2][cnt] = zcenter/(surface->contours[j]->points.size());

        // fill the spreadsheet for contourpoint curvature means
        mySpreadSheet->columns[1].setValue(cnt,(float) curvatures[0][cnt]);
        cnt++;

    }

}

// Here allPoints array is all the processed points and ringpointindexes is a dynamic array for every ring
McDArray< McDArray<int> > allPoints = ringpointindexes;
McDArray< McDArray<int> > allPoints1(nParticles );
if(nInnerRings==0 ) allPoints1 = ringpointindexes;

//Initialize the ringnumber array for weighting purposes in matlab fitting
McDArray< McDArray<int> > allCs(nParticles );
if(nInnerRings==0 ) allCs = ringpointindexes;
for(int l=0; l<nParticles;l++) allCs[l].fill(0);

//Get the triangles per point array

```



```

//Get the direction normal vectors
normal = normal + surface -> getVertexNormal(tri.points[0]);

//if(r==nInnerRings-1) v[tri.points[0]]=1000;
// if(r==nRings-1) v[tri.points[2]]=1000;

// v[tri.points[0]] = (r+1)*1000;
}
if(ok2)
{
allPoints[j].insert(allPoints[j].size(),tri.points[1]);
if( r>=nInnerRings-1 )

allPoints1[j].insert(allPoints1[j].size(),tri.points[1]);

allCs[j].insert(allCs[j].size(),r+1);
ringpointindexes[j].insert(ringpointindexes[j].size(),tri.points[1]);
curvatures[r+1][j] = curvatures[r+1][j] + v[tri.points[1]];

//Get the direction normal vectors
normal = normal + surface -> getVertexNormal(tri.points[1]);
// if(r==nInnerRings-1) v[tri.points[1]]=1000;
// if(r==nRings-1) v[tri.points[2]]=1000;
// v[tri.points[1]]=v[tri.points[1]] + 1000;
// v[tri.points[1]] = (r+1)*1000;
}
if(ok3)
{
allPoints[j].insert(allPoints[j].size(),tri.points[2]);
if( r>=nInnerRings-1 )

allPoints1[j].insert(allPoints1[j].size(),tri.points[2]);

allCs[j].insert(allCs[j].size(),r+1);
ringpointindexes[j].insert(ringpointindexes[j].size(),tri.points[2]);
curvatures[r+1][j] = curvatures[r+1][j] + v[tri.points[2]];

//Get the direction normal vectors
normal = normal + surface -> getVertexNormal(tri.points[2]);

// if(r==nInnerRings-1) v[tri.points[2]]=1000;
//if(r==nRings-1) v[tri.points[2]]=1000;
// v[tri.points[2]] = v[tri.points[2]] + 1000;
// v[tri.points[2]] = (r+1)*1000;
}

ok1=1;ok2=1;ok3=1;ok4=0;

}if(tri.patch ==0 )
}for(int k=0;k<neighs.size();k++)

}for(i=0;i<newringpointindexes[j].size();i++)

if(ringpointindexes[j].size()==0)
{
curvatures[r+1][j] = -2;
mySpreadSheet->columns[r+2].setValue(cnt,(float) curvatures[r+1][j]);
//Get the mean value of the normal vector of the contourpoints
//normalf[j] = normalz;
cnt++;
}
else
{
curvatures[r+1][j] = curvatures[r+1][j] / ringpointindexes[j].size();
mySpreadSheet->columns[r+2].setValue(cnt,(float) curvatures[r+1][j]);
//Get the mean value of the normal vector of the contourpoints
normalf[j] = normalf[j] + normal/sqrt(normal[0]*normal[0] + normal[1]*normal[1] + normal[2]*normal[2]);
//theMsg->printf("siz %d",allCs[j].size());
cnt++;
}

}for(int j=0;j< surface->contours.size()-1;j++)

}for(int r=0;r<nRings;r++)

```

```
////////////////////////////////////////////////////////////////7
```

```
////////////////////////////////////////////////////////////////AND STILL THE POINT COORDINATES FOR MATLAB////////////////////////////////////////////////////////////////
```

```
cnt=0;

// For every particle
for(int j=0;j< nParticles;j++)
{
float curv=0;
  for(int r=0;r<=nRings;r++)
  {
      curv = curv + curvatures[r][j];
  }
  curv = curv/(nRings+1);

  McDArray<int> temp = allPoints1[j];
  //McDArray<int> temp1 = allCs[j];

      mySpreadSheet1->setNumRows(cnt+1);
      mySpreadSheet1->columns[0].setValue(cnt,(float) (normalf[j][0] / nRings));
      mySpreadSheet1->columns[1].setValue(cnt,(float) (normalf[j][1] / nRings) );
      mySpreadSheet1->columns[2].setValue(cnt,(float) (normalf[j][2] / nRings));
      mySpreadSheet1->columns[3].setValue(cnt,(int) -2 );
      mySpreadSheet1->columns[4].setValue(cnt,(float) -2.0 );
      cnt++;

      mySpreadSheet1->setNumRows(cnt+1);
      mySpreadSheet1->columns[0].setValue(cnt,(float) center[0][j] );
      mySpreadSheet1->columns[1].setValue(cnt,(float) center[1][j] );
      mySpreadSheet1->columns[2].setValue(cnt,(float) center[2][j] );
      mySpreadSheet1->columns[3].setValue(cnt,(int) -2 );
      mySpreadSheet1->columns[4].setValue(cnt,(float) -2.0 );
      cnt++;

  for(i=0;i< temp.size();i++)
  {

      //Get the point coordinates and save them in spreadsheet
      McVec3f p;
      int ok=0;
      p = surface ->points[temp[i]];
      mySpreadSheet1->setNumRows(cnt+1);
      mySpreadSheet1->columns[0].setValue(cnt,(float) (p[0] - center[0][j] ) );
      mySpreadSheet1->columns[1].setValue(cnt,(float) (p[1] - center[1][j] ) );
      mySpreadSheet1->columns[2].setValue(cnt,(float) (p[2] - center[2][j] ) );
      mySpreadSheet1->columns[3].setValue(cnt,(int) allCs[j][i] );
      mySpreadSheet1->columns[4].setValue(cnt,(float) curv);

      // put 1 if point is a innerRing point
      for(int k=0;k<allCs[j].size();k++)
      {
          if(temp[i] == allCs[j][k]) ok=1;
      }

      if(ok) mySpreadSheet1->columns[3].setValue(cnt, 1 );
      else mySpreadSheet1->columns[3].setValue(cnt, -1 );

      cnt++;

  }
  // Fill in 999 as a mark to matlab script to separate the particles
  mySpreadSheet1->setNumRows(cnt+1);
  mySpreadSheet1->columns[0].setValue(cnt,(float) 999.0 );
  mySpreadSheet1->columns[1].setValue(cnt,(float) 999.0 );
  mySpreadSheet1->columns[2].setValue(cnt,(float) 999.0 );
  mySpreadSheet1->columns[3].setValue(cnt,(int) 999);
}
```

```

mySpreadSheet1->columns[4].setValue(cnt,(float) 999.0 );
cnt++;

}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
delete points////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
McDArray<int> pats2;
//Delete the particle points

//For every contour, that is for every particle
for(int j=0;j< surface->contours.size();j++)
{
    int patok1=0;
    int patok2=0;
    int patok3=1;
    int outpat=0;
    int inpat=0;

    int patsize = surface->contours[j]->patches.size();

    for(int g=0;g< patsize;g++)
    {
        if(surface->patches[surface->contours[j]->patches[g]->boundaryId == 2){ patok1 =1;
        if(surface->patches[surface->contours[j]->patches[g]->boundaryId == 3){ patok2 =1; inpat=g;}
        }

        for(int u=0;u< pats2.size();u++)
        {
            if(surface->contours[j]->patches[outpat] == pats2[u]){ patok3 =0; }
            if(surface->contours[j]->patches[inpat] == pats2[u]){ patok3 =0; }
        }

        // particles' contours always should have 3 patches
        if(patok1 == 1 && patok2 == 1 && patok3 ==1)
        {
            surface->patches[surface->contours[j]->patches[outpat]->triangles.clear();
            surface->patches[surface->contours[j]->patches[inpat]->triangles.clear();
        }
    }
} //for(int j=0;j< surface->contours.size();j++)

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

theWorkArea->notBusy();

mySpreadSheet->setLabel("LocalCurvatures");
setResult(mySpreadSheet);

mySpreadSheet1->setLabel("PointCoordinates");
setResult(mySpreadSheet1);

} // if (portShowSpreadSheet.wasHit()) {
}

```

To acquire the demagnetizing distribution, the point coordinates of the particles have to be extracted. A label field must be generated with only the segmented particles and then a surface can be generated. Smoothing is not recommended if the particles are close to each other, this will agglomerate the adjacent particles. Local->MyDisplayVertices1 should be available by right clicking the .surf file. A spreadsheet ParticlePoints will appear, where the point coordinates and

areas are shown. The particles are separated by a value -999. All the values can be copied to create a new variable named “unnamed” in Matlab and paste the values in that variable for further Matlab analysis. If necessary, the values can be pasted first to notepad to delete the first string line “X Y Z AREA”

## MyDisplayVertices1.h

```

////////////////////////////////////
//
// Example of a display module
//
////////////////////////////////////
#ifndef MY_DISPLAY_VERTICES1_H
#define MY_DISPLAY_VERTICES1_H

#include <mclib/McHandle.h> // smart pointer template class
#include <Amira/HxCompModule.h> // include declaration of base class
#include <hxstatistics/HxSpreadSheet.h>
#include <mypackage/mypackageAPI.h> // storage-class specification

class MYPACKAGE_API MyDisplayVertices1 : public HxCompModule
{
    HX_HEADER(MyDisplayVertices1);

public:
    // Constructor.
    MyDisplayVertices1();

    // Destructor.
    ~MyDisplayVertices1();

    // This is called when an input port changes.
    virtual void compute();

protected:
    HxSpreadSheet *mySpreadSheet;
};

#endif

```

## MyDisplayVertices1.cpp

```

////////////////////////////////////
//
// Example of a compute module (version 1)
//
////////////////////////////////////

#include <Amira/HxMessage.h> // for output in Amira console
#include <hxsurface/HxSurface.h> // class representing a surface
#include <mypackage/MyDisplayVertices1.h> // header of this class
#include <mclib/McDArray.h>
#include <hxsurface/Surface.h>

HX_INIT_CLASS(MyDisplayVertices1,HxCompModule)

MyDisplayVertices1::MyDisplayVertices1() :
    HxCompModule(HxSurface::getClassTypeId())
{
}

```

```

MyDisplayVertices1::~MyDisplayVertices1()
{
}

void MyDisplayVertices1::compute()
{
    int i,j;

    // Access input object (portData is inherited from HxModule):
    HxSurface* surface = (HxSurface*) portData.source();
    surface->recompute();

    int nVertices = surface->points.size();
    int nTriangles = surface->triangles.size();

    mySpreadSheet = new HxSpreadSheet();

    mySpreadSheet->addColumn("X", HxSpreadSheet::Column::FLOAT);
    mySpreadSheet->addColumn("Y", HxSpreadSheet::Column::FLOAT);
    mySpreadSheet->addColumn("Z", HxSpreadSheet::Column::FLOAT);
    mySpreadSheet->addColumn("area", HxSpreadSheet::Column::FLOAT);

    int Nparticles = surface->patches.size();
    int cnt=0;
    for (i=0; i<Nparticles; i++)
    {
        double area = surface->getPatchArea(i);

        nTriangles = surface->patches[i]->triangles.size();
        McDArray<int> allPoints;

        for (j=0; j<nTriangles; j++)
        {
            Surface::Triangle& tri = surface->triangles[surface->patches[i]->triangles[j]];
            int ok1 = 1, ok2 = 1, ok3 = 1;

            for(int l=0;l<allPoints.size();l++)
            {
                if(tri.points[0] == allPoints[l]) ok1 = 0;
                if(tri.points[1] == allPoints[l]) ok2 = 0;
                if(tri.points[2] == allPoints[l]) ok3 = 0;
            }

            McVec3f p;
            if(ok1) allPoints.insert(allPoints.size(),tri.points[0]);
            if(ok2) allPoints.insert(allPoints.size(),tri.points[1]);
            if(ok3) allPoints.insert(allPoints.size(),tri.points[2]);
        }

        for (j=0; j<allPoints.size(); j++)
        {
            McVec3f p;
            p = surface ->points[allPoints[j]];

            mySpreadSheet->setNumRows(cnt+1);
            mySpreadSheet->columns[0].setValue(cnt,(float) p[0]);
            mySpreadSheet->columns[1].setValue(cnt,(float) p[1]);
            mySpreadSheet->columns[2].setValue(cnt,(float) p[2]);
            mySpreadSheet->columns[3].setValue(cnt,(float) area);
            cnt++;
        }

        mySpreadSheet->setNumRows(cnt+1);
        mySpreadSheet->columns[0].setValue(cnt,(float) -999.0 );
        mySpreadSheet->columns[1].setValue(cnt,(float) -999.0 );
        mySpreadSheet->columns[2].setValue(cnt,(float) -999.0 );
        mySpreadSheet->columns[3].setValue(cnt,(float) -999.0 );
        cnt++;
    }
    mySpreadSheet->setLabel("ParticlePoints");
}

```

```

    setResult(mySpreadSheet);
}

```

## 7.5 Matlab Code

To acquire the demagnetization distribution, a variable named “unnamed” should be available as explained before in the Amira section. These scripts are presented as they are with hardly any comments or explanations.

These files should be on the Matlab path:

<http://www.mathworks.com/matlabcentral/fileexchange/24693-ellipsoid-fit>

<http://www.mathworks.com/matlabcentral/fileexchange/8805-elliptic-integrals-and-functions/content/elliptic12.m>

Also these functions are necessary:

The Solveellipse function is taken from

[http://www.mathworks.com/matlabcentral/newsreader/view\\_thread/159497](http://www.mathworks.com/matlabcentral/newsreader/view_thread/159497)

### Solveellipse.m

```

function [H X0 W err] = solveellipse(X)
% function [H X0 W] = solveellipse(X);
%
% INPUT:
% X : (m x n) : m data points in R^n, they are supposed to belong
% to an ellipsoide (or more generally a second-order
% implicate hyper-surface)
%
% OUTPUT:
% H : Matrix of the Bilinear form associated with the ellipsoide
% Ellipsoide = { X in R^n : (X-X0)' * H * (X-X0) = 1 }
% X0 : (n x 1), center of the ellipsoide
% W : (n x n) square matrix where each column is the axis of ellipsoide
%
[ndata ndim]=size(X);

if ndim<2
    error('solveellipse: dimension number must be greater than 1');
end

% This vector will be used in few places for reshapping purpose
uno = ones(1,ndim);

%
% Generate all combinations of polynomial-order <=2 for ndim variables
%
order=cell(1,ndim);
order(:)={0:2};
ORDER=cell(1,ndim); % 1 x ndim cell of vectors (0:2)
[ORDER{:}] = ndgrid(order{:}); % Set {(0:2)}^ndim
ORDER = reshape(cat(ndim+1,ORDER{:}),[],ndim);
ORDER = ORDER(sum(ORDER,2)<=2,:); % second order only

%
% Remove constant term
%
ORDER = ORDER(2:end,:); % similar to ORDER(~any(ORDER,2),:)=[];
npo1 = size(ORDER,1); % number of basis 2nd order-polynomials

```

```

if ndata<npol
    error('solveellipse: Not enough data');
end

function loc=findloc(order) % nested function
%
% Look for the location of the 'order' in the set of{basis
% 2nd-order polynomials}
%
[dummy loc]=ismember(reshape(order,1,[]),ORDER,'rows');
end

%
% Both of these are 3d array of (ndata x ndim x npol)
%
BIGX= repmat(X,[1 1 npol]);
BIGORDER= repmat(reshape(ORDER',[1 ndim npol]),[ndata 1 1]);
M=squeeze(prod(BIGX.^BIGORDER,2)); % product of power in every dimensions
clear BIGX BIGORDER

%
% Solve for polynomial coefficients that lead to 1 on inputpoints
%
rhs=ones(ndata,1);
P=pinv(M)*rhs;
clear M rhs;

%
% Extract the Hessian from the solution
%
I= repmat(eye(ndim),ndim,1);
I= reshape(I,ndim*[1 1 1]);
J= permute(I,[2 1 3]); % swap the first two dimensions
K= squeeze(mat2cell(I+J,uno,uno,ndim));
P2_loc= cellfun(@findloc,K);
P2=P(P2_loc); % second order term
A=(tril(P2)+triu(P2))/2; % divide by 2, accept diagonal terms

%
% Extract the gradient
%
I= mat2cell(eye(ndim),uno,ndim);
P1_loc= cellfun(@findloc,I);
P1=-0.5*P(P1_loc); % -0.5 * first order term

X0=pinv(A)*P1;
lambda=1/(1+X0*P1);
H=lambda*A;

if nargout>=3 % Compute main-axis of ellipsoide
    if(isnan(trace(H))~=1 && isinf(trace(H))~=1)
        [V D]=eig(H);

        d=diag(D);
        if any(d<0)
            warning('solveellipse: npdH', ...
                'solveellipse: non positive Hessian matrix');
        end
        W = V.*repmat(1./sqrt(d'),ndim,1);
        % W = repmat(1./sqrt(d'),ndim,1);
        err = 1;
    else

        W = zeros(3,3);
        err = 0;
    end
end
end
end

```

## ellipsoidarea.m

```
function area = ellipsoidarea(a,b,c)

if(a==b && b==c)
    area = 4*pi*a^2;
else

phi = acos(c/a);

k = sqrt(((b^2 - c^2)*a^2)/((a^2 - c^2)*b^2));

[Fi,Ei] = elliptic12(phi,k);

area = 2*pi*c^2 + ((2*pi*a*b)/(sin(phi)))*(Ei*sin(phi)^2 + Fi*cos(phi)^2);
end

end
```

## getdemagfactors.m

```
function [LL,flag] = getdemagfactors(A)
%[LL,MM,NN,IX,flag]
flag = 1;
%A = [Az,Ax,Ay];
[B IX] = sort(A);

v = acos((B(1)/B(3)));
u = acos((B(2)/B(3)));
k = sin(u)/sin(v);

if (v < 0 | u < 0 | k < 0 | v > pi/2 | u > pi/2 | k >= 1)
%warning('out of bounds');
flag=0;
end

if(isreal(v) && isreal(u))
[Fi,Ei] = elliptic12(v,k^2);

L = ((cos(v)*cos(u))/((sin(v)^3)*k^2));
LLL = L*(Fi -Ei);

M = ((cos(u)*cos(v))/((sin(v)^3)*(sin(asin(k))^2)*(cos(asin(k))^2)));
MM = M*(Ei - (cos(asin(k))^2)*Fi - ((sin(asin(k))^2)*sin(v)*cos(v))/cos(u) );

N = (cos(v)*cos(u))/((sin(v)^3)*(cos(asin(k))^2));
NN = N*(((sin(v)*cos(u))/cos(v)) -Ei);

SUM = LLL + MM + NN;

if (SUM > 1.01 | SUM < 0.99)
%warning('Sum of factors is incorrect');
flag=0;
end

pLL = [NN MM LLL];
LL(IX) = pLL;
%isequal(LL, pLL);

else
LL = [0 0 0];
end
end
```

The main script starts here. This will create an info array where the columns are as follows:

Info(i , 1:3) X, Y and Z coordinates and i is the number of the particle

Info(i , 4:6) Radius in x,y,z

Info(i , 7:9) normalized direction of the longest axis: easy axis

Info(i , 10:12) normalized direction of the middle axis

Info(i , 13:15) normalized direction of the shortest axis: hard axis

Info(i , 16) Volume from fitted radii

Info(i , 17:19) Demagnetization factors from longest to shortest axis

Info(i , 20) Angle to x axis

Info(i , 21) Angle to y axis

Info(i , 22) Angle to z axis

Info(i , 23) OK flag =1

Info(i , 24) Area from amira

Info(i , 25) Area from fitted radii

Info(i , 26) Volume from amira

Info(i , 27) mean distance of the points distance to the fitted surface -> goodness of the fit

### getellipsoids\_amira.m

```
alpha = 90; %put 90 for e=1 elongation factor
scale = 0.76; %pixelsize !! this is not implemented see below !!

%do rounding
doround = 1;

alpha = alpha*(pi/180);
e = sqrt((alpha + sin(alpha)*cos(alpha))/(alpha - sin(alpha)*cos(alpha)));

siz = size(unnamed,1);
count = 0;
lim=1;
particles = cell(1,4000);

% Separate the points into cells
for i = 1:siz

    val = unnamed(i,1);
    if(val==999 || val==999)
        count=count+1;
        particles{1,count} = unnamed(lim:i-1,:);
        lim = i+1;
    end
end

info = zeros(count,28);

for i=1:count

sizok=0;
na=0;
points = [particles{i}(:,1) particles{i}(:,2) particles{i}(:,3)]*scale;
points(:,3) = points(:,3)*(1/e);
xE= points(:,1);
yE= points(:,2);
zE= points(:,3);
xE = xE - mean(xE);
yE = yE - mean(yE);
zE = zE - mean(zE);
siz = size(points);
```

```

if(siz(1,1)>9)
    area = (particles{i}(1,4))*(scale*scale);
sizok=1;

i

[center, radii, evecs, pars,err] = ellipsoid_fit( points,0 );

[H X0 W err] = solveellipse(points);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if(isreal(W) ==1 & isnan(W) ~=1 )

v1 = roundn(normc(W(:,1)),-2);
v2 = roundn(normc(W(:,2)),-2);
v3 = roundn(normc(W(:,3)),-2);
v1 = normc(v1);
v2 = normc(v2);
v3 = normc(v3);

zang1 = vrrotvec(v1,[0 0 1]);
xang1 = vrrotvec(v1,[1 0 0]);
yang1 = vrrotvec(v1,[0 1 0]);

zang = zang1(1,4)*(180/pi);
xang = xang1(1,4)*(180/pi);
yang = yang1(1,4)*(180/pi);

if(zang>=90)
    zang = 180 - zang ;
end
if(xang>=90)
    xang = 180 - xang;
end
if(yang>=90)
    yang = 180 - yang ;
end

else
zang = -1000;
xang = -1000;
yang = -1000;
v1 =[2 2 2];
v2 =[2 2 2];
v3 =[2 2 2];
err = -5;
end % if(isreal(W) ==1 & isnan(W) ~=1 )

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if(isreal(sqrt(radii)))
count = count+1;
info(i,1) = roundn(center(1),-1);%*scale; %ymax -y coord in DM y in amira
info(i,2) = roundn(center(2),-1);%*scale; %X coord in DM x in amira
info(i,3) = roundn(center(3),-1);%*scale;

if(doround)
info(i,4) = roundn(radii(1)*2,0)/2;%*scale;
info(i,5) = roundn(radii(2)*2,0)/2;%*scale;
info(i,6) = roundn(radii(3)*2,0)/2;%*scale;
else
info(i,4) = roundn(radii(1),-1);%*scale;
info(i,5) = roundn(radii(2),-1);%*scale;
info(i,6) = roundn(radii(3),-1);%*scale;
end
info(i,7:9) = v1;%evecs(1,:);
info(i,10:12) = v2;%evecs(2,:);
info(i,13:15) = v3;%evecs(3,:);
info(i,16) = roundn((4/3)*pi*info(i,4)*info(i,5)*info(i,6),-1);

```

```

info(i,23) = err;
info(i,20) = xang;
info(i,21) = yang;
info(i,22) = zang;      info(i,24) = area;
info(i,25) = ellipsoidarea(info(i,4),info(i,5),info(i,6));
info(i,26) = (4/3)*pi*info(i,4)*info(i,5)*info(i,6);
[K, vol] = convhulln(points);
info(i,27) = vol;%*scale^3;

Rz = eye(4);
Rz(1:3,1) = [info(i,7) info(i,8) info(i,9)];
Rz(1:3,2) = [info(i,10) info(i,11) info(i,12)];
Rz(1:3,3) = [info(i,13) info(i,14) info(i,15)];

Rerr=0;
for j = 1:size(points,1)
    min=1000;
    kmin=0;
    xp = EllPrj([xE(j,1) yE(j,1) zE(j,1)], info(i,4:6)', Rz(1:3,1:3)');
    for k = 1:size(xp,2)
        D = pdist2([xE(j,1) yE(j,1) zE(j,1)],xp(1:3,k)','euclidean');
        if(D<min)
            min = D;
            kmin = k;
        end
    end

    % D = pdist2([x(j,1) y(j,1) z(j,1)],xp(1:3,kmin)','euclidean');
    Rerr = Rerr + min^2;
end

info(i,28) = Rerr/size(points,1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if(info(i,4) == info(i,5) && info(i,4) == info(i,6))
    LL = [0.3333 0.3333 0.3333];
    flag = 1;

elseif(info(i,4) == info(i,5))
[LL,flag] = getdemagfactors(info(i,4:6));
LL(1) = (1 -LL(3))/2;
LL(2) = (1 -LL(3))/2;
flag = 1;
elseif(info(i,4) == info(i,6))
[LL,flag] = getdemagfactors(info(i,4:6));
LL(1) = (1 -LL(2))/2;
LL(3) = (1 -LL(2))/2;
flag = 1;
elseif(info(i,5) == info(i,6))
[LL,flag] = getdemagfactors(info(i,4:6));
LL(2) = (1 -LL(1))/2;
LL(3) = (1 -LL(1))/2;
flag = 1;
else
[LL,flag] = getdemagfactors(info(i,4:6));
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if (flag==0)
    info(i,23) = -1;
end

info(i,17:19) = LL;
else
    info(i,23) =-2;
end

else
    info(i,23) =-3;
end

end

```

```

clear B H LL W X0 center count err evecs flagi im na pars
clear radii scale siz sizok v1 v2 v3 xang yang zang flag i
clear e j k kmin lim min particles point val doround area
clear alpha Rz Rerr K D zang1 zE yang1 xp yE vol points xE xang1

```

For the polynomial fitting a variable “unnamed” has to be available in Matlab, copied from the PointCoordinates spreadsheet acquired using the MyDisplayVertices3.cpp module. The .surf file should be saved before applying the GetCurvature module.

### Fitsurface1.m

```

function F = fitsurface1(X, data,n)

x = data(:, 1);
y = data(:, 2);
z = data(:, 3);
W = data(:, 4);
X = real(X);
FF = [x y z]*vrotvec2mat([X(1) X(2) X(3) X(4)]);

p = polyfitn(FF(:,1:2),FF(:,3),W,n);
newz = polyvaln(p,FF(:,1:2));

FA = [FF(:, 1:2) newz]/(vrotvec2mat([X(1) X(2) X(3) X(4)]));
F = FA(:,3);
end

```

### Fitsurface2.m

```

function F = fitsurface2(X, data,n)

x = data(:, 1);
y = data(:, 2);
z = data(:, 3);
W = data(:, 4);
X = real(X);
FF = [x y z]*vrotvec2mat([X(1) X(2) X(3) X(4)]);

p = polyfitn(FF(:,1:2),FF(:,3),W,n);

F = p.R2;

end

```

### DoPolyFitting3.m

```

%output

%outval{1,num} = inr; fitted point coordinates inside the first ring
%outval{2,num} = inv(mb); rotation matrix
%outval{3,num} = center; center coordinates

%curvatu values, first is the point at centre. last is the mean value of
% shapeindex nearby points

%outval{4,num} = [shapeindexatcenter gaussianat meanat mean(mean(VVV))];

%values from optimization and used polynomial order
%outval{5,num} = [a jminx];

%outval{6,num} = VVV; near points shapeindex values

```

```

method = 1; % use 1 for lsqcurvefit. use 2 for fminsearchbnd
plotone = 1; % use 1 if you want to plot, see line 29 also
jmax = 3; % maximum polynomial order
MAXPART=2000; % maximum number of particles

lb = [-1 -1 -1 -pi/2]; % lower bounds of rotation matrix
ub = [1 1 1 pi/2]; % upper bounds of rotation matrix
a0 = [0.1 0.1 0.1 0.1]; % starting values of rotation matrix
options = optimset('TolFun', 1e-8,'TolX',0.0001);

siz = size(unnamed,1);
count = 0;
lim=1;
particles = cell(1,4000);

% Separate the points into cells
for i = 1:siz

    val = unnamed(i,1);
    if(val==-999 || val==999)
        count=count+1;
        particles{1,count} = unnamed(lim:i-1,:);
        lim = i+1;
    end
end

mincount=1;
%count=1;
%mincount=count; % for plotting a surface number count

outval = cell(6,MAXPART);

%For every point set
for num = mincount:count

    siz = size(particles{num},1);

    if(siz>10)

        Vxyz = [particles{num}(1,1) particles{num}(1,2) particles{num}(1,3)];
        center = [particles{num}(2,1) particles{num}(2,2) particles{num}(2,3)];

        xyz = [particles{num}(3:end,1) particles{num}(3:end,2) particles{num}(3:end,3)];
        da = particles{num}(3:end,4);
        W = 1./(da);
        WW = ones(size(W));
        W=WW;
        if(size(W(W==1),1)>5)
            num

        data = [xyz W];
        jfin = zeros(jmax,1);

        for j = 1:jmax

            if(method==1)
                a = lsqcurvefit(@(x,data) fitsurface1(x,data,j),a0,data,xyz(:,3),lb,ub,options);
            end
            if(method==2)
                [a, resnorm] = fminsearchbnd(@(x) fitsurface2(x,data,j),a0,lb,ub,options);
            end
            mb = vrrotvec2mat([a(1) a(2) a(3) a(4)]);
            nxyz = xyz*mb;

            p = polyfitn(nxyz(:,1:2),nxyz(:,3),W,j);
            jfin(j,1) = p.RMSE ; %AdjustedR2;%mean(D(:))^2;

        end
    end
end

```

```

mb = vrrotvec2mat([a(1) a(2) a(3) a(4)]);
nxyz = xyz*mb;
[jmin,jminx] = min(jfin);

```

```

p = polyfitn(nxyz(:,1:2),nxyz(:,3),W,jminx);

```

```

mixx = min(min(nxyz(:,1)));
maxx = max(max(nxyz(:,1)));
miy = min(min(nxyz(:,2)));
may = max(max(nxyz(:,2)));
miz = min(min(nxyz(:,3)));
maz = max(max(nxyz(:,3)));

```

```

mi = min(min(mixx,miy),miz);
ma = max(max(maxx,may),maz);

```

```

[xg,yg]=meshgrid(mi:.5:ma);

```

```

newz = polyvaln(p,[xg(:),yg(:)]);

```

```

zg = reshape(newz,size(xg));
[FF,V]=mesh2tri(xg,yg,zg,'F');

```

```

Nx = nxyz(:,1);
Ny = nxyz(:,2);
Nz = nxyz(:,3);

```

```

Cx = Nx(W>0.6);
Cy = Ny(W>0.6);
Cz = Nz(W>0.6);

```

```

[A,b] = vert2con([Cx,Cy]);
xxg = xg(:);
yyg = yg(:);
cnt =0;

```

```

inx = zeros(size(V,1),1);
iny = zeros(size(V,1),1);
inz = zeros(size(V,1),1);

```

```

yesno = zeros(size(V,1),1);

```

```

for k = 1:size(V,1)

```

```

    BB = A*[V(k,1), V(k,2)]' - b ;
    yesno(k,1) = 0;
    inx(k,1) = -888;
    iny(k,1) = -888;
    inz(k,1) = -888;
    if( size(BB(BB>=0),1)==0)
        cnt = cnt +1 ;
        inx(k,1) = V(k,1);
        iny(k,1) = V(k,2);
        inz(k,1) = V(k,3);
        yesno(k,1) = 1;
    end
end

```

```

end

```

```

inx = inx(inx~-888);
iny = iny(iny~-888);
inz = inz(inz~-888);

```

```

inr = zeros(size(inx,1),3);
inr = [inx, iny, inz];

```

```

if(plotone==1)

```

```

    plot3(inx,iny,inz,')

```

```

    plot3(mean(inx),mean(iny),mean(inz),'.')
    hold on
    plot3(nxyz(:,1),nxyz(:,2),nxyz(:,3),'.')
    plot3(Cx,Cy,Cz,'.')
    surf(xg,yg,zg)
    end

sp = polyn2sym( p);
syms X1 X2

hx = diff(sp, X1);
hy = diff(sp,X2);
hxx = diff(hx,X1);
hyy = diff(hy,X2);
hxy = diff(hx,X2);

E = 1 + hx^2;
F = hx*hy;
G = 1 + hy^2;
L = hxx/sqrt(1 + hx^2 + hy^2);
M = hxy/sqrt(1 + hx^2 + hy^2);
N = hyy/sqrt(1 + hx^2 + hy^2);

K = (L*N - M^2)/(E*G - F^2);
H = (E*N + G*L - 2*F*M)/(2*(E*G-F^2));

kmax = H + sqrt(H^2 -K);
kmin = H - sqrt(H^2 -K);

VVV = zeros(51,count);

for h = -25:25
    kmaxatxcenter = subs(subs(kmax,X1,(mean(inx) - h*0.05),X2,(mean(iny)- h*0.05)));
    kminatxcenter = subs(subs(kmin,X1,(mean(inx)- h*0.05)),X2,(mean(iny)- h*0.05));
    shapeindexatcenter = -(2/pi)*atan((kmaxatxcenter + kminatxcenter)/(kminatxcenter-kmaxatxcenter));
    gaussianat = kminatxcenter*kmaxatxcenter;
    meanat = (kminatxcenter+kmaxatxcenter)/2;
    VVV(h+26,count) = shapeindexatcenter;
end

outval{1,num} = inr;
outval{2,num} = inv(mb);
outval{3,num} = center;
outval{4,num} = [shapeindexatcenter gaussianat meanat mean(mean(VVV))];
outval{5,num} = [a jminx];
outval{6,num} = VVV;
end %W
    end %if(siz~=0)

end %for num = 1:count

clear xg ma mi yyg x y z par count lim siz SI cnt count Nxyz
clear num i j val xy pid x xcenter ycenter zcenter IDX D mea
clear curv output msiz tri siz1 en1 en2 k1 k2 curv out jmax
clear parcount hx hy hxx hyy hxy K L M E G H N dt kmax kmin maxx
clear may maz mixx miy miz xg yg xx yy zg zz zzg F X1 X2
clear A BB Cx Cy Cz E F FF G H K L M MAXPART N Nx Ny Nz RMSE
clear particles sp jfin meapre Vxyz m001 m010 m100 rot xyz hfin h m
clear N001 N010 N100 k meafin BB A Cx Cy Cz Nx Ny Nz W b inx
clear iny inz poutf pout ans da R2plus RMSEplus hj jplus mb
clear minRMSE p center V d s F1 F11 F2 F22 F3 F33 FF jhfin mincount
clear allpoints as flag fnt gnt hgg inr yesno RMSE a0 a newz skip resnorm
clear lb nxyz ub V VVV Vxyz W WW X1 X2 a a0 b center cnt count da dg
clear en1 gaussianat h hx hxx hxy hy hyy i inr inx iny inz k kmax
clear kmaxatcenter kmin kminatcenter lb lim ma maxsize maxx may maz
clear mb meanat mi mincount mixx miy miz mrcout newz num nxyz
clear p particles resnorm siz skip sp ub method
clear val xg xgz xyz yesno zg yyg zg kminatxcenter kmaxatxcenter
clear gaussianat data f g jmin jminx options plotone shapeindexatcenter

```

## 8 References

- [1] J. L. Dormann, D. Fiorani and E. J. Tronc, "On the models for interparticle interactions in nanoparticle assemblies: comparison with experimental results," *J. Magn. Mater.*, vol. 202, pp. 251-267, 1999.
- [2] T. Uusimäki, G. Margaris, K. Trohidou, P. Granitzer, K. Rumpf, M. Sezen and G. Kothleitner, "Three dimensional quantitative characterization of magnetite nanoparticles embedded in mesoporous silicon: local curvature, demagnetizing factors and magnetic Monte Carlo simulations," *Nanoscale*, vol. 5, no. 23, pp. 11944-53, 2013.
- [3] D. B. Williams and C. B. Carter, *Transmission Electron Microscopy*, New York: Springer Science + Business Media, 2009.
- [4] U. Kaiser, J. Biskupek, J. C. Meyer, J. Leschner, L. Lechner, H. Rose, M. Stöger-Pollach, A. N. Khlobystov, P. Hartel, H. Müller, M. Haider, S. Eyhusen and G. Benner, "Transmission electron microscopy at 20 kV for imaging and spectroscopy," *Ultramicroscopy*, vol. 111, no. 8, pp. 1239-1246, 2011.
- [5] D. C. Bell and A. J. Garratt-Reed, *Energy Dispersive X-ray Analysis in the Electron Microscope*, Oxford: BIOS Scientific Publishers Limited, 2003.
- [6] J. Goldstein, D. E. Newbury, D. C. Joy, C. E. Lyman, P. Echlin, E. Lifshin, L. Sawyer and J. R. Michael, *Scanning Electron Microscopy and X-ray Microanalysis*, 3 ed., New York: Springer Science + Business Media, Inc., 2003.
- [7] R. F. Egerton, "Electron energy-loss spectroscopy in the TEM," *Reports on Progress in Physics*, vol. 72, no. 1, 2009.
- [8] R. F. Egerton and M. Malac, "EELS in the TEM," *Journal of Electron Spectroscopy and Related Phenomena*, vol. 142, no. 2-3, pp. 43-50, 2005.
- [9] R. Egerton, *Electron Energy-Loss Spectroscopy in the Electron Microscope*, Plenum Press, 1996.
- [10] O. L. Krivanek, A. J. Gubbens and N. Dellby, "Developments in EELS instrumentation for spectroscopy and imaging," *Microsc. Microanal. Microstruct.*, vol. 2, no. 2-3, pp. 315-332, 1991.
- [11] A. Gubbens, M. Barfels, C. Trevor, R. Twesten, P. Mooney, P. Thomas, N. Menon, B. Kraus, C. Mao and B. McGinn, "The GIF Quantum, a next generation post-column imaging energy filter," *Ultramicroscopy*, vol. 110, no. 8, pp. 962-970, 2010.
- [12] S. J. Pennycook and P. D. Nellist, *Scanning Transmission Electron Microscopy: Imaging and Analysis*, S. J. Pennycook and P. D. Nellist, Eds., New York: Springer Science + Business Media, 2011.

- [13] B. E. Batson, "Electron microscopy: Hydrogen brightens up," *Nature Materials*, vol. 10, pp. 270-271, 2011.
- [14] S. J. Pennycook, "Z-contrast stem for materials science," *Ultramicroscopy*, vol. 30, no. 1-2, pp. 58-69, 1989.
- [15] S. Benlekber, "PhD Thesis, STEM-HAADF nanotomography: application to nanomaterials," L'institut national des sciences appliquées de Lyon, Lyon, 2009.
- [16] P. A. Midgley, M. Weyland, J. M. Thomas and B. F. G. Johnson, "Z-Contrast tomography: a technique in three-dimensional nanostructural analysis based on Rutherford scattering," *Chem. Commun.*, pp. 907-908, 2001.
- [17] P. Ercius, M. Weyland, D. A. Muller and L. M. Gignac, "Three-dimensional imaging of nanovoids in copper interconnects using incoherent bright field tomography," *Appl. Phys. Lett.*, vol. 88, pp. 243116-243116-3, 2006.
- [18] P. Ercius, "PhD Thesis, Three-Dimensional Electron Tomography of Integrated Circuit Devices," Cornell University, Cornell, 2009.
- [19] Z. H. Levine, "Theory of bright-field scanning transmission electron microscopy for tomography," *Journal of Applied Physics*, vol. 97, no. 033101, pp. 033101 - 033101-12, 2005.
- [20] C. Kübel and A. Voigt, "Manual, Xplore 3D - STEM Tomography," FEI, 2009.
- [21] J. M. Rodenburg and E. G. Macab, "Optimising the Resolution of TEM/STEM with the Electron Ronchigram," *Microscopy and Analysis*, vol. 90, pp. 5-7, 2002.
- [22] N. Delpy, O. L. Krivanek, P. D. Nellist, P. E. Batson and A. R. Lupini, "Progress in aberration-corrected scanning transmission electron microscopy," *Microscopy and Microanalysis*, vol. 50, pp. 177-185, 2001.
- [23] J. Goldstein, D. E. Newbury, D. C. Joy, C. E. Lyman, P. Echlin, E. Lifshin, L. Sawyer and J. R. Michael, *Scanning Electron Microscopy and X-ray Microanalysis*, 3 ed., New York: Springer Science + Business Media, Inc., 2003.
- [24] L. Reimer, *Scanning Electron Microscopy: Physics of Image Formation and Microanalysis*, Berlin Heidelberg New York: Springer-Verlag, 1998.
- [25] M. Sezen, "PhD Thesis, Nanostructuring and Modification of Conjugated Polymer Based Optoelectronic Device Structures by Focused Ion Beam," Graz University of Technology, Graz, 2009.
- [26] L. A. Giannuzzi and F. A. Stevie, *Introduction to Focused Ion Beams: Instrumentation, Theory, Techniques and Practice*, New York: Springer Science + Business Media, Inc., 2010.
- [27] J. Orloff, M. Utlaut and L. Swanson, *High resolution focused ion beams: FIB and its applications*, New York: Kluwer Academic/Plenum Publishers, 2003.
- [28] R. Krahne, G. Morello, A. Figuerola, C. George, S. Deka and L. Manna, "Physical properties of elongated inorganic nanoparticles," *Physics Reports*, no. 501, pp. 75-221, 2011.
- [29] P. Weiss, "L'hypothèse du champ moléculaire et la propriété ferromagnétique," *J. Phys. Theor. Appl.*, vol. 6, no. 1, pp. 661-690, 1907.
- [30] B. D. Cullity and C. Graham, *Introduction to Magnetic Materials (2nd ed.)*, New Jersey: IEEE Press & Wiley, 2009.
- [31] R. C. O'Handley, *Modern Magnetic Materials*, New York: John Wiley & Sons, 2000.

- [32] S. Bedanta, PhD Thesis, Supermagnetism in magnetic nanoparticle systems, Duisburg-Essen: Universität Duisburg-Essen, 2006.
- [33] S. Bedanta and W. Kleemann, "Supermagnetism," *J. Phys. D: Appl. Phys.*, vol. 42, no. 013001, 2009.
- [34] E. -. L. Salabas, "PhD Thesis, Structural and Magnetic Investigations of Magnetic Nanoparticles and Core-Shell Colloids," Universität Duisburg-Essen, Duisburg, 2004.
- [35] E. C. Stoner and E. P. Wohlfarth, "A Mechanism of Magnetic Hysteresis in Heterogeneous Alloys," *Philos. Trans. R. Soc. London A*, vol. 240, pp. 599-642, 1948.
- [36] D. Kechrakos, "Magnetic Nanoparticle Assemblies," *arXiv:0907.4417 [cond-mat.mes-hall]*, 2010.
- [37] B. Young and S.-Y. Kwak, "Assembly of magnetite nanocrystals into spherical mesoporous aggregates with a 3-D wormhole-like pore structure," *J. Mater. Chem*, vol. 20, pp. 8320-8328, 2010.
- [38] O. Petracic, "Superparamagnetic nanoparticle ensembles," *arXiv*, vol. 0911.4031, 2009.
- [39] J. M. D. Coey, *Magnetism and Magnetic Materials*, New York: Cambridge University Press, 2010.
- [40] M. Darbandi, F. Stromberg, J. Landers, N. Reckers, B. Sanyal, W. Keune and H. Wende, "Nanoscale size effect on surface spin canting in iron oxide nanoparticles synthesized by the microemulsion method," *J. Phys. D: Appl. Phys.*, vol. 45, p. 195001, 2012.
- [41] J. A. Osborn, "Demagnetizing factors of the general ellipsoid," *Phys. rev.*, vol. 67, no. 11 and 12, pp. 351-357, 1945.
- [42] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller and E. Teller, "Equation of State Calculations by Fast Computing Machines," *J. Chem. Phys.*, vol. 21, pp. 1087-1092, 1953.
- [43] K. Trohidou, Vasilakaki and S. M. (Ed.), "Monte Carlo Studies of Magnetic Nanoparticles," in *Applications of Monte Carlo Method in Science and Engineering*, InTech, 2011, pp. 513-538.
- [44] J. Smit and H. Wijn, *Ferrites: Physical Properties of Ferri-magnetic Oxides in Relation to Their Technical Applications*, New York: John Wiley, 1959.
- [45] I. Joffe and R. Heuberger, "Hysteresis properties of distributions of cubic single-domain ferromagnetic particles," *Philosophical Magazine*, vol. 314, pp. 1051-1059, 1974.
- [46] M. J. Sailor, *Porous Silicon in Practice: Preparation, Characterization and Applications*, Weinheim: Wiley-VCH, 2011.
- [47] L. T. Canham, "Silicon quantum wire array fabrication by electrochemical and chemical dissolution of wafers," *Appl. Phys. Lett.*, vol. 57, pp. 1046-1048, 1990.
- [48] V. Lehmann and U. Gösele, "Porous silicon formation: A quantum wire effect," *Appl. Phys. Lett.*, vol. 58, p. 856, 1991.
- [49] N. Koshida and H. Koyama, "Visible electroluminescence from porous silicon," *Appl. Phys. Lett.*, vol. 60, p. 347, 1992.
- [50] L. T. Canham, "Bioactive silicon structure fabrication through nanoetching techniques," *Adv. Mater.*, vol. 7, p. 1033, 1995.

- [51] E. Pastor, E. Matveeva, V. Parkhutik, J. Curiel-Esparza and M. C. Milan, "Influence of porous silicon oxidation on its behaviour in simulated body fluid," *Phys. Status Solidi C*, vol. 4, pp. 2136-2140, 2007.
- [52] P. Granitzer, K. Rumpf, M. Albu and P. Pölt, "Double-sided mesoporous silicon with embedded quasi-regular arranged ferromagnetic nanostructures fabricated by electrodeposition," *ECS Transactions*, vol. 25, pp. 139-145, 2010.
- [53] S. Mornet, S. Vasseur, F. Grasset and E. Duguet, "Magnetic nanoparticle design for medical diagnosis and therapy," *Journal of Materials Chemistry*, vol. 14, pp. 2161-2175, 2004.
- [54] J. Mürbe, A. Rechtenbach and J. Töpfer, "Synthesis and physical characterization of magnetite nanoparticles for biomedical applications," *Materials Chemistry and Physics*, vol. 110, no. 2-3, pp. 426-433, 2008.
- [55] L. Harivardhan Reddy, J. L. Arias, J. Nicolas and P. Couvreur, "Magnetic Nanoparticles: Design and Characterization, Toxicity and Biocompatibility, Pharmaceutical and Biomedical Applications," *Chemical Reviews*, vol. 112, no. 11, pp. 5818-5878, 2012.
- [56] P. Granitzer and K. Rumpf, "Magnetic Nanoparticles Embedded in a Silicon Matrix," *Materials*, vol. 4, pp. 908-928, 2011.
- [57] E. Xifré Pérez, "PhD Thesis, Design, fabrication and characterization of porous silicon multilayer optical devices," Universitat Rovira i Virgili, Tarragona, 2007.
- [58] P. Granitzer, K. rumpf, M. Venkatesan, A. G. Roca, L. Cabrera, M. P. Morales, P. Pölt and M. Albu, "Magnetic Study of Fe<sub>3</sub>O<sub>4</sub> Nanoparticles Incorporated within Mesoporous Silicon," *J. Electrochem. Soc.*, vol. 157, no. 7, pp. K145-K151, 2010.
- [59] A. G. Roca, M. P. Morales, K. O'Grady and C. J. Serna, "Structural and magnetic properties of uniform magnetite nanoparticles prepared by high temperature decomposition of organic precursors," *Nanotechnology*, vol. 17, no. 11, p. 2783, 2006.
- [60] M. E. Fleet, "The Structure of Magnetite: Two Annealed Natural Magnetites, Fe<sub>3</sub>.0005O<sub>4</sub> and Fe<sub>2</sub>.96Mg<sub>0.04</sub>O<sub>4</sub>," *Acta Cryst. C*, vol. 40, p. 1491-1493, 1984.
- [61] A. Pratt, "Overview of the use of copper interconnects in the semiconductor industry," Advanced Energy Industries, Inc., Fort Collins, 2004.
- [62] Technikon\_Forschungsgesellschaft, "Copper Interconnects for Advanced Performance and Reliability," Project reference: 216474, 2011. [Online]. Available: <http://cordis.europa.eu/docs/projects/cnect/4/216474/080/reports/002-D68FinalReportFin120111.pdf>.
- [63] CopPeR, "Feature Stories - A groundbreaking, waterless approach to micro-chip making," 2012. [Online]. Available: [http://cordis.europa.eu/fetch?CALLER=OFFR\\_TM\\_EN&ACTION=D&DOC=1&CAT=OFFR&QUERY=0137c253140f:caad:2338e4d3&RCN=8196](http://cordis.europa.eu/fetch?CALLER=OFFR_TM_EN&ACTION=D&DOC=1&CAT=OFFR&QUERY=0137c253140f:caad:2338e4d3&RCN=8196). [Accessed 11 12 2012].
- [64] L. D'Urzo, S. Schaltin, A. Shkurankov, H. Plank, G. Kothleitner, C. Gspan, K. Binnemans and J. Fransaer, "Direct-on-barrier copper electroplating on ruthenium from the ionic liquid 1-ethyl-3-methylimidazolium dicyanamide," *Journal of Material Science: Materials in Electronics*, vol. 23, no. 4, pp. 945-951, 2012.

- [65] S. Schaltin, N. R. Brooks, L. Stappers, L. D'Urzo, H. Plank, G. Kothleitner, C. Gspan, K. Binnemans and J. Fransaer, "Electrodeposition from a Liquid Cationic Cuprous Organic Complex for Seed Layer Deposition," *Journal of Electrochemical Society*, vol. 158, no. 11, pp. D647-D650, 2011.
- [66] S. Schaltin, L. D'Urzo, Q. Zhao, A. Vantomme, H. Plank, G. Kothleitner, C. Gspan, K. Binnemans and J. Fransaer, "Direct electroplating of copper on tantalum from ionic liquids in high vacuum: origin of the tantalum oxide layer," *Physical Chemistry Chemical Physics*, vol. 14, pp. 13624-13629, 2012.
- [67] M. Sytnyk, R. Kirchsclager, R. Bodnachuk, D. Primetzhofer, D. Kriegner, H. Enser, J. Stangl, B. Bauer, M. Voith, A. W. Hassel, F. Krumeich, F. Ludwig, A. Meingast, G. Kothleitner, M. V. Kovalenko and W. Heiss, "Tuning the Magnetic Properties of Metal Oxide Nanocrystal Heterostructures by Cation Exchange," *Nano Letters*, vol. 13, pp. 586-593, 2013.
- [68] D. N. Mastronarde, "Correction for non-perpendicularity of beam and tilt axis in tomographic reconstructions with the IMOD package," *Journal of Microscopy*, vol. 230, no. 2, pp. 212-217, 2007.
- [69] D. J. De Rosier and A. Klug, "Reconstruction of Three Dimensional Structures from Electron Micrographs," *Nature*, vol. 217, pp. 130-134, 1968.
- [70] P. A. Midgley and R. E. Dunin-Borkowski, "Electron tomography and holography in materials science," *Nature Materials*, vol. 8, pp. 271-280, 2009.
- [71] A. J. Koster, U. Ziese, A. J. Verkleij, A. H. Janssen and K. P. de Jong, "Three dimensional electron microscopy: A novel imaging and characterization technique with nanometer scale resolution for materials science," *Journal of Physical Chemistry B*, vol. 104, pp. 9368-9370, 2000.
- [72] J. Frank, *Three-Dimensional Imaging with the Transmission Electron Microscope*, New York: Plenum Press, 1992.
- [73] R. A. Crowther, D. J. DeRosier and A. Klug, "The reconstruction of a three-dimensional structure from projections and its application to electron microscopy," *Proceedings of the Royal Society of London A*, vol. 307(1530), pp. 319-340, 1970.
- [74] J. Radon, "Über die bestimmung von funktionen durch ihre integralwerte langs gewisser mannigfaltigkeiten," *Berichte Sachsische Akademie der Wissenschaften*, vol. 69, no. Math. - Phys. Kl., pp. 262-267, 1917.
- [75] S. R. Deans, *The Radon Transform and Some of its Applications*, New York: Wiley, 1983.
- [76] W. Hoppe, J. Gassmann, N. Hunsmann, H. J. Schramm and M. Sturm, "3-dimensional reconstruction of individual negatively stained yeast fatty-acid synthetase molecules from tilt series in electron-microscope," *Hoppe-Seylers Zeitschrift für Physiologische Chemie*, vol. 355, no. 11, pp. 1483-1487, 1974.
- [77] W. Baumeister, R. Grimm and J. Walz, "Electron tomography of molecules and cells," *Trends in Cell Biology*, vol. 9, no. 2, pp. 81-85, 1999.
- [78] S. G. Wolf, L. Houben and M. Elbaum, "Cryo-scanning transmission electron tomography of vitrified cells," *Nature Methods*, vol. 11, pp. 423-428, 2014.
- [79] P. A. Midgley and M. Weyland, "3D electron microscopy in the physical sciences: the development of z-contrast and EFTEM tomography.," *Ultramicroscopy*, vol. 96, no. 3-4, pp. 413-431, 2003.

- [80] P. A. Midgley, M. Weyland, J. M. Thomas and B. F. G. Johnson, "Z-contrast tomography: a technique in three-dimensional nanostructural analysis based on rutherford scattering," *Chemical Communications*, vol. 10, pp. 907-908, 2001.
- [81] I. Florea, O. Ersen, C. Hirlimann, L. Roiban, A. Deneuve, M. Houille, I. Janowska, I. Nguyen, C. Pham and C. Pham-Huu, "Analytical electron tomography mapping of the SiC pore oxidation at the nanoscale," *Nanoscale*, vol. 2, pp. 2668-2678, 2010.
- [82] G. Möbus, R. C. Doole and B. J. Inkson, "Spectroscopic electron tomography," *Ultramicroscopy*, vol. 96, pp. 433-451, 2003.
- [83] K. Jarausch, P. Thomas, D. N. Leonard, R. Twesten and B. C. R., "Four-dimensional STEM-EELS: enabling nano-scale chemical tomography," *Ultramicroscopy*, vol. 109, no. 4, pp. 326-37, 2009.
- [84] G. Haberfehlner, A. Orthacker, M. Albu, J. Li and G. Kothleitner, "Nanoscale voxel spectroscopy by simultaneous EELS and EDS tomography," *Nanoscale*, vol. 6, pp. 14563-14569, 2014.
- [85] L. Salvo, P. Cloetens, E. Maire, S. Zabler, B. J. J., J. Y. Buffière, W. Ludwig, E. Boller, D. Bellet and C. Josserond, "X-ray micro-tomography an attractive characterisation technique in materials science," *Nuclear Instruments and Methods in Physics Research B*, vol. 200, pp. 273-286, 2003.
- [86] E. N. Landis and D. T. Keane, "X-ray microtomography," *Materials Characterization*, vol. 6, no. 12, pp. 1305-1316, 2010.
- [87] P. J. Withers, "X-ray nanotomography," *Materials Today*, vol. 10, no. 12, pp. 26-34, 2007.
- [88] J. A. Whitby, F. Östlund, P. Horvath, M. Gabureac, J. L. Riesterer, I. Utke, M. Hohl, L. sedláček, J. Jiruse, V. Friedli, M. Bechelany and J. Michler, "High Spatial Resolution Time-of-Flight Secondary Ion Mass Spectrometry for the Masses: A Novel Orthogonal ToF FIB-SIMS Instrument with In Situ AFM," *Advances in Material Science and Engineering*, vol. 2012, 2011.
- [89] M. D. Uchic, "Serial Sectioning Methods for Generating 3D Characterization Data of Grain- and Precipitate-Scale Microstructures," in *Computational Methods for Microstructure-Property Relationships*, G. S. and D. Dimiduk, Eds., Springer Science + Business Media, LLC, 2011, pp. 31-52.
- [90] A. J. Patkin and G. H. Morrison, "Secondary ion mass spectrometric image depth profiling for three-dimensional elemental analysis," *Analytical Chemistry*, vol. 54, no. 1, pp. 2-5, 1982.
- [91] I. Utke, S. Moshkalev and P. Russell, *Nanofabrication using focused ion and electron beams: principles and applications*, New York: Oxford University Press, Inc., 2012.
- [92] D. Blavette, A. Bostel, J. M. Sarrau, B. Deconihout and A. Menand, "An atom probe for three-dimensional tomography," *Nature*, vol. 363, no. 6428, pp. 432-435, 1993.
- [93] T. Kelly and M. Miller, "Atom probe tomography," *Review of Scientific Instruments*, vol. 78, no. 3, p. 031101, 2007.
- [94] A. Voigt and C. Kübel, "Manual, Xplore 3D—STEM Calibrations," FEI, 2009.
- [95] K. Dierksen, D. Typke, R. Hegerl, A. J. Koster and W. Baumeister, "Towards automatic electron tomography," *Ultramicroscopy*, vol. 40, no. 1, pp. 71-87, 1992.

- [96] U. Ziese, W. J. C. Geerts, T. P. Van Der Krift, A. J. Verkleij and A. J. Koster, "Correction of Autofocusing errors due to Specimen Tilt for Automated Electron Tomography," *Journal of Microscopy*, vol. 211, no. 2, pp. 179-185, 2003.
- [97] U. Ziese, A. H. Janssen, J.-L. Murk, W. J. C. Geerts, T. Van der Krift, A. J. Verkleij and A. J. Koster, "Automated high-throughput electron tomography by pre-calibration of image shifts," *Journal of Microscopy*, vol. 205, no. 2, pp. 187-200, 2002.
- [98] E. W. Weisstein, "Cross-Correlation Theorem," 2014. [Online]. Available: <http://mathworld.wolfram.com/Cross-CorrelationTheorem.html>. [Accessed 6 11 2014].
- [99] J. P. Thirion, "Image matching as a diffusion process: An analogy with Maxwell's demon," *Medical Image Analysis*, vol. 2, no. 3, pp. 243-260, 1998.
- [100] M. Weyland, "PhD Thesis: Two and three dimensional nanoscale analysis : new techniques and applications," University of Cambridge, Cambridge, 2001.
- [101] D. R. G. Mitchell and B. Schaffer, "Scripting-customised microscopy tools for Digital Micrograph™," *Ultramicroscopy*, vol. 103, no. 4, pp. 319-332, 2005.
- [102] L. Houben and M. Bar Sadan, "Refinement procedure for the image alignment in high-resolution electron tomography," *Ultramicroscopy*, vol. 111, no. 9-10, pp. 1512-20, 2011.
- [103] F. Amat, F. Moussavi, L. R. Comolli, G. Elidan, K. H. Downing and M. Horowitz, "Markov random field based automatic image alignment for electron tomography," *Journal of Structural Biology*, vol. 161, no. 3, pp. 260-275, 2008.
- [104] S. Brandt and J. Heikkonen, "A Fully Automatic Alignment of Electron Tomography Images Without Fiducial Markers," in *MVA2000*, Tokyo, 2000.
- [105] M. C. Scott, C.-C. Chen, M. Mecklenburg, C. Zhu, R. Xu, P. Ercius, U. Dahmen and B. C. Regan, "Electron tomography at 2.4-ångström resolution," *Nature*, vol. 483, p. 444-447, 2012.
- [106] R. N. Bracewell, "Strip Integration in Radio Astronomy," *Australian Journal of Physics*, vol. 9, no. 2, pp. 198-217, 1956.
- [107] R. M. Lewitt, "Reconstruction Algorithms: Transform Methods," *Proceedings of the IEEE*, vol. 71, pp. 390-408, 1983.
- [108] Y. Chen and F. Förster, "Iterative reconstruction of cryo-electron tomograms using nonuniform fast Fourier transforms," *Journal of Structural Biology*, vol. 185, pp. 309-316, 2014.
- [109] S. R. Deans, *The Radon transform and its applications*, New York: John Wiley and Sons, 1983.
- [110] P. Tchelidze, C. Sauvage, M. Bonnet, M. Kilian, A. Beorchia, M. F. O'Donohue, D. Ploton and H. Kaplan, "Electron tomography of amplified nanogold immunolabelling: Improvement of quality based on alignment of projections with sinograms and use of post-reconstruction deconvolution," *Journal of Structural Biology*, vol. 156, no. 3, pp. 421-431, 2006.
- [111] K. Aoyama, T. Takagi, A. Hirase and A. Miyazawa, "STEM tomography for thick biological specimens," *Ultramicroscopy*, vol. 109, pp. 70-80, 2008.
- [112] J. Banhart, *Advanced Tomographic Methods in Materials Research and Engineering*, New York: Oxford University Press, 2008.

- [113] S. W. Lee, C.-L. Lee, H.-M. Cho, H.-S. Park, D.-H. Kim, C. Y.-N. and H.-J. Kim, "Effects of Reconstruction Parameters on Image Noise and Spatial Resolution," *Journal of the Korean Physical Society*, vol. 59, no. 4, pp. 2825-2832, 2011.
- [114] J. Fessler, "Analytical Tomographic Image Reconstruction Methods," 2009. [Online]. Available: <http://web.eecs.umich.edu/~fessler/course/516/l/c-tomo.pdf>. [Accessed 12 12 2012].
- [115] Gilbert, "Iterative methods for the three-dimensional reconstruction of an object for projections," *Journal of Theoretical Biology*, vol. 36(1), pp. 105-117, 1972.
- [116] H. Heidari Mezerji, W. Van den Broek and S. Bals, "A practical method to determine the effective resolution in incoherent experimental electron tomography," *Ultramicroscopy*, vol. 111, pp. 330-336, 2011.
- [117] P. L. de Andres, R. Ramírez and J. A. Vergés, "Strong covalent bonding between two graphene layers," *Phys. Rev. B*, vol. 77, p. 045403, 2008.
- [118] A. H. Andersen and A. C. Kak, "Simultaneous algebraic reconstruction technique (SART): a superior implementation of the ART algorithm," *Ultrasonic imaging*, vol. 6, pp. 81-94, 1984.
- [119] A. Alpers, R. J. Gardner, S. König, R. S. Pennington, C. B. Boothroyd, L. Houben, R. E. Dunin-Borkowski and K. J. Batenburg, "Geometric reconstruction methods for electron tomography," *arXiv:1205.5738*, Vols. [math-ph], 2012.
- [120] B. Goris, T. Roelandts, K. J. Batenburg, H. Heidari Mezerji and S. Bals, "Advanced reconstruction algorithms for electron tomography: From comparison to combination," *Ultramicroscopy*, vol. 127, pp. 40-47, 2013.
- [121] S. Krimmel, J. Baumann, Z. Kiss, A. Kuba, A. Nagy and J. Stephan, "Discrete tomography for reconstruction from limited view angles in non-destructive testing," *Electronic Notes in Discrete Mathematics*, vol. 20, pp. 455-474, 2005.
- [122] G. Haberfehlner, "PhD Thesis, 3D nanoimaging of semiconductor devices and materials by electron tomography," Laboratoire d'Électronique et de Technologies de l'Information (CEA, LETI), Grenoble, 2013.
- [123] M. Radermacher, "Weighted Back-Projection Methods," in *Electron Tomography*, F. J., Ed., New York, Springer US, 1992, pp. 91-115.
- [124] S. Lanzavecchia and P. L. Bellon, "Electron tomography in conical tilt geometry. The accuracy of a direct Fourier method (DFM) and the suppression of non-tomographic noise," *Ultramicroscopy*, vol. 63, pp. 247-261, 1996.
- [125] H. H. Mezerji, W. V. d. Broek and S. Bals, "A practical method to determine the effective resolution in incoherent experimental electron tomography," *Ultramicroscopy*, vol. 111, pp. 330-336, 2011.
- [126] W. O. Saxton and W. Baumeister, "The correlation averaging of a regularly arranged bacterial cell envelope protein," *J. Microsc*, vol. 127, no. 2, pp. 127-138, 1982.
- [127] M. Unser, C. O. S. Sorzano, P. Thévenaz, S. Jonic, C. El-Bez, S. D. Carlo, J. F. Conway and B. L. Trus, "Spectral signal-to-noise ratio and resolution assessment of 3D reconstructions," *J. Struct. Biol.*, vol. 149, pp. 243-255, 2005.
- [128] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, pp. 629-639, 1990.

- [129] B. K. Shreyamsha Kumar, "Image denoising base on non-local means filter and its method noise tresholding," *Signal, Image and Video Processing*, vol. 7, no. 6, pp. 1211-1227, 2013.
- [130] S. Dugas, "Connected Threshold Segmentation of Ventricles in the Brain," *The Insight Journal*, 2007.
- [131] A. S. Frangakis and R. Hegerl, "Segmentation of two- and three-dimensional data from electron microscopy using eigenvector analysis," *Journal of Structural Biology*, vol. 138, no. 1-2, pp. 105-113, 2002.
- [132] N. Volkmann, "A novel three-dimensional variant of the watershed transform for segmentation of electron density maps," *Journal of Structural Biology*, vol. 138, no. 1-2, pp. 123-129, 2002.
- [133] J. C. Russ, *The Image Processing Handbook*, 2 ed., London: CRC Press, 1995.
- [134] Matlab, *Version 7.11.0.584 (R2010b)*, The MathWorks Inc., Natick, MA, USA, 2013.
- [135] C. Baldock, C. J. Gilpin, A. J. Koster, U. Ziese, K. E. Kadler, C. M. Kielty and D. F. Holmes, "Three-dimensional reconstructions of extracellular matrix polymers using automated electron tomography," *Journal of Structural Biology*, vol. 138, pp. 130-136, 2002.
- [136] IDL, "Exelis Visual Information Solutions," EXELIS, 2012. [Online]. Available: <http://www.exelisvis.com/ProductsServices/IDL.aspx>. [Accessed 12 12 2012].
- [137] M. McCormick, X. Liu, J. Jomier, C. Marion and L. Ibanez, "ITK: enabling reproducible research and open science," *Front Neuroinform*, vol. 8, no. 13, 2014.
- [138] G. Wolberg and S. Zokai, "Robust Image Registration Using Log-Polar Transform," *Proceedings 2000 International Conference on Image Processing*, vol. 1, pp. 493 - 496, 2000.
- [139] L. Houben, "multi-platform tomography suite (TOMATO)," Ernst Ruska Centre for Microscopy and Spectroscopy with Electrons, 2012. [Online]. Available: <http://www.er-c.org/methods/software.htm>. [Accessed 11 12 2012].
- [140] J. Nickolls, I. Buck, M. Garland and K. Skadron, "Scalable Parallel Programming with CUDA," *ACM Queue*, vol. 6, no. 2, pp. 40-53, 2013.
- [141] R. T. Whitaker and X. Xue, "Variable-conductance, level-set curvature for image denoising," 2012. [Online]. Available: [http://www.cs.utah.edu/~xwxue/Research/files/curvature\\_paper01.pdf](http://www.cs.utah.edu/~xwxue/Research/files/curvature_paper01.pdf). [Accessed 25 Jun 2012].
- [142] R. Beare, "Grayscale morphological attribute operations," *The Insight Journal*, no. July - december, 2007.
- [143] R. Leary, Z. Saghi, M. Armbrüster, G. Wowsnick, R. Schlögl, J. M. Thomas and P. A. Midgley, "Quantitative High-Angle Annular Dark-Field Scanning Transmission Electron Microscope (HAADF\_STEM) Tomography and High-Resolution Electron Microscopy of Unsupported Intermetallic GaPd<sub>2</sub> Catalysts," *The Journal of Physical Chemistry C*, vol. 116 (24), pp. 13343-13352, 2012.
- [144] S. Dugas, "Connected Threshold Segmentation of Ventricles in the Brain," *The Insight Journal - RPI - Open Source*, 2007.
- [145] D. Stalling, F. Westerhoff and H.-. C. Hege, "Amira: A Highly Interactive System for Visual Data Analysis," in *Visualization Handbook*, C. D. Hansen and C. R. Johnson, Eds., San Diego, USA, Elsevier Inc., 2005, pp. 749-767.

- [146] J. Koenderink and A. van Doorn, "Surface shape and curvature scales," *J. Image and Vision Computing*, vol. 10, pp. 557-565, 1992.
- [147] VS, *Visual Studio 2008 Professional Edition*, Microsoft, Redmont, U.S, 2013.
- [148] E. P. W. Ward, T. J. V. Yates, J.-J. Fernández, D. E. W. Vaughan and P. A. Midgley, "Three-Dimensional Nanoparticle Distribution and Local Curvature of Heterogeneous Catalysts Revealed by Electron Tomography," *J. Phys. Chem. C*, vol. 111, no. 31, pp. 11501-11505, 2007.
- [149] J. D'Errico, "Polyfitn," MATLAB Central File Exchange, 2012. [Online]. Available: <http://www.mathworks.com/matlabcentral/fileexchange/34765-polyfitn>. [Accessed 25 Jan 2012].
- [150] Y. Petrov, "Ellipsoid fit," MATLAB Central File Exchange, 2012. [Online]. Available: <http://www.mathworks.com/matlabcentral/fileexchange/24693-ellipsoid-fit>. [Accessed 25 Jan 2012].
- [151] M. Beleggia, T. Kasama and R. E. Dunin-Borkowski, "The quantitative measurement of magnetic moments from phase images of nanoparticles and nanostructures — I. Fundamentals," *Ultramicroscopy*, vol. 110, pp. 425-432, 2010.
- [152] A. C. Twitchett-Harrison, T. J. V. Yates, R. E. Dunin-Borkowski and P. A. Midgley, "Quantitative electron holographic tomography for the 3D characterisation of semiconductor device structures," *Ultramicroscopy*, vol. 108, pp. 1401-1407, 2008.
- [153] K. Binder and D. W. Heerman, *Monte Carlo Simulation in Statistical Physics*, Berlin: Springer-Verlag, 1992.
- [154] A. Aharony, *Introduction to the Theory of Ferromagnetism*, Oxford: Clarendon Press, 1996.
- [155] S. Sahoo, O. Petravic, W. Kleemann, S. Stappert, G. Dumpich, P. Nordblad, S. Cardoso and P. Freitas, "Cooperative versus superparamagnetic behavior of dense magnetic nanoparticles in Co<sub>80</sub>Fe<sub>20</sub>/Al<sub>2</sub>O<sub>3</sub> multilayers," *Appl. Phys. Lett.*, vol. 82, no. 23, pp. 4116-4118, 2003.
- [156] J. L. Dormann, D. Fiorani and E. ( P. I. a. R. S. A. Tronc, "Magnetic Relaxation in Fine-Particle Systems," in *Advances in Chemical Physics*, vol. 98, New York, John Wiley & Sons, 1997, pp. 283-494.
- [157] F. A. Lasagni, A. F. Lasagni, I. Huertas-Olivares, C. Holzapfel and F. Mücklich, "3D Nano-characterisation of materials by FIB-SEI/EDS tomography," *Materials Science and Engineering*, vol. 7, p. 012016, 2010.
- [158] W. Scholz, "MAGPAR," 2012. [Online]. Available: <http://www.magpar.net/>. [Accessed 12 12 2012].
- [159] M. Rudnaya, "PhD Thesis, Automated focusing and astigmatism correction in electron microscopy," Technische Universiteit Eindhoven, Eindhoven, 2011.
- [160] R. M. Haralick, K. Shanmugam and I. Dinstein, "Textural Features for Image Classification," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC 3, no. 6, pp. 610-621, 1973.
- [161] ITK\_Doxygen, "GreyLevelCooccurrenceMatrixTextureCoefficientsCalculator," ITK, 2013. [Online]. Available: [http://www.itk.org/Doxygen320/html/classitk\\_1\\_1Statistics\\_1\\_1GreyLevelCooccurrenceMatrixTextureCoefficientsCalculator.html](http://www.itk.org/Doxygen320/html/classitk_1_1Statistics_1_1GreyLevelCooccurrenceMatrixTextureCoefficientsCalculator.html). [Accessed 21 1 2013].

- [162] S. Pertuz, "Focus Measure," 2013. [Online]. Available: <http://www.mathworks.com/matlabcentral/fileexchange/27314-focus-measure>. [Accessed 16 5 2013].
- [163] O. L. Krivanek, J. Gubbens and N. Dellby, "Developments in EELS instrumentation for spectroscopy and imaging," *Microscopy Microanalysis Microstructures*, vol. 2, no. 2-3, pp. 315-332, 1991.
- [164] A. J. Gubbens, B. Kraus, O. L. Krivanek and P. E. Mooney, "An imaging filter for high voltage electron microscopy," *Ultramicroscopy*, vol. 59, no. 1-4, pp. 255-265, 1995.
- [165] H. A. Brink, M. M. G. Barfels, R. P. Burgner and B. N. Edwards, "A sub-50 meV spectrometer and energy filter for use in combination with 200 kV monochromated (S)TEMs," *Ultramicroscopy*, vol. 96, no. 3-4, pp. 367-384, 2005.
- [166] L. Reimer, *Energy-Filtering Transmission Electron Microscopy*, Berlin: Springer Series in Optical Sciences Vol. 71, Springer-Verlag, 1995.
- [167] "Gatan," [Online]. Available: <http://www.gatan.com/>. [Accessed 2014].
- [168] R. Grimm, D. Typke and W. Baumeister, "Improving image quality by zero-loss energy filtering: quantitative assessment by means of image cross-correlation," *Journal of Microscopy*, vol. 190, no. 3, pp. 339-349, 1997.
- [169] F. Hofer, P. Warbichler, H. Kronberger and J. Zweck, "Mapping the chemistry in nanostructured materials by energy-filtering transmission electron microscopy (EFTEM)," *Spectrochimica Acta Part A*, vol. 57, pp. 2061-2069, 2001.
- [170] G. Kothleitner and F. Hofer, "Optimization of the Signal to Noise Ratio in EFTEM Elemental Maps with Regard to Different Ionization Edge Types," *Micron*, vol. 29, no. 5, pp. 349-357, 1998.
- [171] L. Yedra, A. Eljarrad, R. Arenal, E. Pellicer, M. Cabo, A. López-Ortega, M. Estrader, J. Sort, M. Dolores Baró, S. Estradé and F. Peiró, "EEL spectroscopic tomography: Towards a new dimension in nanomaterials analysis," *Ultramicroscopy*, vol. 122, pp. 12-18, 2012.
- [172] N. Y. Jin-Phillipp, C. T. KOch and V. A. P. A., "Toward quantitative core-loss EFTEM tomography," *Ultramicroscopy*, vol. 111, no. 8, pp. 1255-1261, 2011.
- [173] K. Martin and B. Hoffman, *Mastering CMake*, 6 ed., Kitware, Inc, 2013.
- [174] D. R. Mitchell and B. Schaffer, "Scripting-customized microscopy tools for Digital Micrograph," *Ultramicroscopy*, vol. 103, no. 4, pp. 319-332, 2005.
- [175] W. Grogger, "DigitalMicrograph™ script database," 2002. [Online]. Available: <http://www.felmi-zfe.tugraz.at/>. [Accessed 21 8 2014].
- [176] M. Weyland, P. A. Midgley and J. M. Thomas, "Electron Tomography of Nanoparticle Catalysts on Porous Supports: A New Technique Based on Rutherford Scattering," *J. Phys Chem. B*, vol. 105, pp. 7882-7886, 2001.
- [177] F. Wang, R. Egerton and M. Malac, "Fourier-ratio deconvolution techniques for electron energy-loss spectroscopy (EELS)," *Ultramicroscopy*, vol. 109, no. 10, pp. 1245-1249, 2009.
- [178] The Ohio State University, "FEI Probe Corrected Titan3™ 80-300 S/TEM," 2014. [Online]. Available: <http://cemas.osu.edu/node/85>. [Accessed 21 5 2014].

- [179] Y. Qiang, A. Sharma, J. Antony, A. McConnaughey, R. Souza and Y. Tian, "Nanomagnetism and magnetic nanoparticles for biomedical application," July 2008. [Online]. Available: [http://www.docstoc.com/docs/document-preview.aspx?doc\\_id=22003155](http://www.docstoc.com/docs/document-preview.aspx?doc_id=22003155). [Accessed 28 5 2013].
- [180] H. Okorn-Schmidt, "Milestone Report for M06.3 "Technical leader report on Copper design development and strategy", " SEZ AG, Villach, 2009.
- [181] P. Nellist and S. J. Pennycook, "Sub-Ångstrom resolution by underfocused incoherent transmission electron microscopy," *Phys. Rev. Lett.*, vol. 81, p. 4156, 1998.
- [182] D. Muller, "Practical STEM: More than Z contrast," 2013. [Online]. Available: <http://people.ccmr.cornell.edu/~davidm/WEELS/summer06/STEM%20tutorialpractical-Muller.pdf>. [Accessed 4 1 2013].
- [183] FEI, "Xplore3D™," FEI, 2012. [Online]. Available: [http://www.fei.com/uploadedfiles/documentsprivate/content/xplore\\_3d\\_web\\_ds.pdf](http://www.fei.com/uploadedfiles/documentsprivate/content/xplore_3d_web_ds.pdf). [Accessed 11 12 2012].
- [184] FEI, "Tecnai™ Transmission Electron Microscope," FEI, 2013. [Online]. Available: <http://www.fei.com/products/transmission-electron-microscopes/tecnai.aspx>. [Accessed 11 1 2013].
- [185] R. Door and D. Gangler, "Multiple least-squares fitting for quantitative electron energy-loss spectroscopy - an experimental investigation using standard specimens," *Ultramicroscopy*, vol. 58, no. 2, pp. 197-210, 1995.
- [186] K. B. Borisenko, G. Moldovan, A. I. Kirkland, D. Van Dyck, H.-Y. Tang and F.-R. Chen, "Toward electron exit wave tomography of amorphous materials at atomic resolution," *Journal of Alloys and Compounds*, vol. 536S, pp. S94-S98, 2012.
- [187] J. Biskupek, J. Leschner, P. Walther and U. Kaiser, "Optimization of STEM tomography acquisition — A comparison of convergent beam and parallel beam STEM tomography," *Ultramicroscopy*, vol. 110, pp. 1231-1237, 2010.
- [188] I. Arslan and E. A. Stach, "Electron tomography: Seeing atoms in three dimensions," *Nature Materials*, vol. 11, pp. 911-912, 2012.
- [189] S. Benlekber, "STEM-HAADF nanotomography: application to nanomaterials," L'institut national des sciences appliquées de Lyon, Lyon, 2009.
- [190] B. Schaffer, "PhD Thesis, High Energy Resolution Spectrum Imaging by Energy Filtering Transmission Electron Microscopy," Technical University of Graz, Graz, 2006.
- [191] W. Rechberger, "PhD Thesis, High resolution scanning transmission electron microscopy in physics and materials science," Technische Universität Graz, Graz, 2007.
- [192] R. C. O'Handley, *Modern Magnetic Materials: Principles and applications*, New York: John Wiley & Sons, Inc., 2000.
- [193] A. C. Kak and M. Slaney, *Principles of Computerized Tomographic Imaging*, New York: Society for Industrial and Applied Mathematics, 1988.

## List of Abbreviations

BSE	Back scattered electrons
CC	Cross-correlation
CCD	Charged-coupled device
CT	Computerized tomography
cTEM	Conventional TEM
DF	Demagnetizing factor
DM	Digital Micrograph (software)
EBID	Electron beam induced deposition
EDX	Energy dispersive X-ray spectroscopy
EELS	Electron energy loss spectroscopy
EFTEM	Energy filtered TEM
ET	Electron tomography
FC	Field cooling
FEG	Field emitter gun
FIB	Focused ion beam
FSC	Fourier shell correlation
FST	Fourier slice theorem
FT	Fourier transform
FTIR	Fourier transform infrared spectroscopy
FWHM	Full width at half maximum
GIF	Gatan imaging filter
GLCM	Gray level co-occurrence matrix
HAADF	High angle annular dark field
IBF	Incoherent bright field
IBID	Ion beam induced deposition
ITK	Insight toolkit
LCD	Local curvature distribution
LMIS	Liquid metal ion source
MC	Monte Carlo
MRI	Magnetic resonance imaging
MSE	Mean squared error
MWC	Missing wedge correction
PRC	Projection re-projection comparison
pSi	Porous silicon
ROI	Region of interest
RT	Radon transform
SART	simultaneous algebraic reconstruction technique
SE	Secondary electrons
SEM	Scanning electron microscopy
SIRT	Simultaneous iterative reconstruction technique
SNNR	Spectral signal to noise ratio
SQUID	Superconducting quantum interference device

STEM	Scanning TEM
SW	Stoner-Wohlfarth
TEM	Transmission electron microscope
VSM	Vibrating sample micrometer
WBP	Weighted back projection
ZFC	Zero field cooling
ZLP	Zero loss peak

## List of Symbols

$\kappa_1, \kappa_2$	Maximum and minimum curvature (principal curvatures)
A	Exchange constant
a	minimum interparticle distance
$A_I$	Surface area inside
$A_O$	Surface area outside
$a_x, a_y, a_z$	Ellipsoid's major axis coordinates
B,H	Magnetic field
$C_C$	Chromatic aberration
$C_S$	Spherical aberration
d	Crystal lattice distance
D	Dipole interaction tensor
dS	Surface element
e	Elongation factor
$\epsilon$	Reduced energy
$E_0$	Energy of the incident electrons (in keV)
$E_D$	Dipole-dipole energy
$E_{\text{exch}}$	Exchange energy
$E_k$	Magnetocrystalline energy
$E_{\text{shape}}$	Shape anisotropy energy
$E_{\text{surf}}$	Surface anisotropy energy
$E_T$	Total energy
$E_Z$	Zeeman energy
$E_\lambda$	Magnetoelastic energy
$H_C$	Remanent field
$J_{ij}$	Exchange integral
$K_1$	Anisotropy energy density
$k_B$	Boltzmann constant
$K_\mu$	Uniaxial anisotropy constant
M	Magnetization
$M_R$	Coercive magnetization
$M_S$	Saturation magnetization
N	Order integer
$N_a, N_b, N_c$	Demagnetizing factors

$r_c$	Critical radius
$R_{ij}$	Center to center distance
$r_s$	Submerging ratio
$r_x, r_y, r_z$	Resolution
$S$	Shape index
$S_i, S_i$	Spin operator
$T$	Temperature
$T_B$	Blocking temperature
$V$	Volume
$Z$	Atomic number
$\theta$	Scattering angle
$\theta_0$	Angle between easy axis and H
$\theta_B$	Bragg angle
$\theta_E$	Angle between easy axis and M
$\theta_S$	Angle between M and surface normal
$\lambda$	Wavelength
$\mu_0$	Permeability constant
$\mu_0$	Vacuum permeability
$\tau_0$	Average time between flips
$\tau_N$	Neél relaxation time

## List of Figures

<b>2.1:</b> Tecnai TF20 located at Felmi-ZFE, TU Graz . . . . .	5
<b>2.2:</b> Parallel illumination for cTEM mode . . . . .	6
<b>2.3:</b> Different signals generated . . . . .	7
<b>2.4:</b> Typical EEL spectrum showing the . . . . .	9
<b>2.5:</b> Schematics of the detectors used in STEM mode . . . . .	11
<b>2.6:</b> On the left a HAADF STEM image . . . . .	12
<b>2.7:</b> The BF, ADF and HAADF detectors . . . . .	13
<b>2.8:</b> Ronchigram with spherical symmetry . . . . .	15
<b>2.9:</b> FEI Nova 200 Nanolab Dual-Beam . . . . .	16
<b>3.1:</b> Typical magnetic hysteresis curve . . . . .	18
<b>3.2:</b> Magnetic nanoparticle with uniaxial anisotropy . . . . .	20
<b>3.3:</b> The formation of pores in the anodization process . . . . .	24
<b>3.4:</b> SEM overview image . . . . .	25
<b>3.5:</b> The aim of the CopPeR project . . . . .	27
<b>3.6:</b> Copper electroplating cell . . . . .	28

<b>3.7:</b> On top a SEM image of the chip . . . . .	29
<b>3.8:</b> Coating of a protective platinum layer . . . . .	30
<b>3.9:</b> FIB preparation of the copper interconnect sample . . . . .	31
<b>4.1:</b> Resolution limits for different 3D metrologies . . . . .	35
<b>4.2:</b> A Fischione 2040 tomography holder . . . . .	38
<b>4.3:</b> Image of the sample holder. . . . .	38
<b>4.4:</b> Calibration data from the holder. . . . .	39
<b>4.5:</b> TEM BF image of a carbon soot particle . . . . .	40
<b>4.6:</b> Left: volume rendered visualization . . . . .	41
<b>4.7:</b> Radon transforms of the symbol $\pi$ . . . . .	42
<b>4.8:</b> Presentation of the inadequate . . . . .	43
<b>4.9:</b> From the original test image . . . . .	44
<b>4.10:</b> Back projection reconstruction . . . . .	45
<b>4.11:</b> Illustration of the SIRT framework. . . . .	46
<b>4.12:</b> Central planes of the carbon . . . . .	47
<b>5.1:</b> STEM HAADF image of the cross grating . . . . .	53
<b>5.2:</b> Screenshot of the alignment program . . . . .	54
<b>5.3:</b> Illustration of the PRC alignment framework . . . . .	55
<b>5.4:</b> Zero tilt image from the original tilt series . . . . .	56
<b>5.5:</b> Triangulated surface of the whole pSi . . . . .	58
<b>5.6:</b> A detail of the central slice . . . . .	58
<b>5.7:</b> Centerline tree graph of the pores . . . . .	59
<b>5.8:</b> FSC curve for cryo and room temperature . . . . .	60
<b>5.9:</b> STEM-HAADF and (b) STEM-IBF image . . . . .	61
<b>5.10:</b> FSC and (b) SSNR of the IBF and HAADF . . . . .	62
<b>5.11:</b> Surface rendered visualization of one trench . . . . .	63
<b>5.12:</b> Surface rendered visualization of the tantalum barrier . . . . .	64
<b>5.13:</b> Schematics of the triangulation scheme . . . . .	65
<b>5.14:</b> Particles with different submerging ratios . . . . .	66
<b>5.15:</b> Submerging ratio of the segmented magnetite particles . . . . .	68
<b>5.16:</b> A polynomial surface was fitted to the point sets . . . . .	69
<b>5.17:</b> LCD of the magnetite nanoparticles . . . . .	70
<b>5.18:</b> Illustration of the geometrical difference . . . . .	71
<b>5.19:</b> A test image with a high curvature location . . . . .	72
<b>5.20:</b> Fitted ellipsoids (red) from the whole sample . . . . .	73
<b>5.21:</b> Direction distribution of the shape anisotropy . . . . .	75

<b>5.22:</b> Measured hysteresis loops with magnetic field . . . . .	76
<b>5.23:</b> Simulated hysteresis loops with magnetic field . . . . .	78
<b>7.1:</b> Gatan GIF post-column energy filter . . . . .	84
<b>7.2:</b> The three-windows-method . . . . .	85
<b>7.3:</b> Si L <sub>2,3</sub> pre-edge image of a flash memory cell . . . . .	86
<b>7.4:</b> Flash memory cell with (a) ZL image . . . . .	87
<b>7.5:</b> Surface visualization of the three combined . . . . .	88