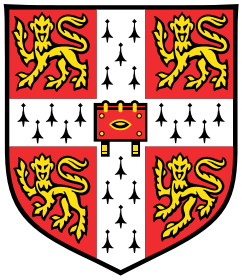
Multi-dimensional Data Analysis in Electron Microscopy



Tomas Ostaševičius

Department of Material Sciences and Metallurgy
University of Cambridge

This dissertation is submitted for the degree of
Doctor of Philosophy

Peterhouse November 2017

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. This dissertation contains fewer than 60 000 words including appendices, bibliography, footnotes, tables and equations and has fewer than 150 figures.

Tomas Ostaševičius
November 2017

Acknowledgements

I would like to thank my supervisor Prof Paul A. Midgley for taking me as a student and his support throughout the years. I greatly appreciate the freedom that I was given, allowing me to shape the degree in ways that none of us would have expected. I would like to acknowledge the support received from the European Union Seventh Framework Program under Grant Agreement 291 522-3DIMAGE.

A huge thank you to Dr. Francisco de la Peña for being my mentor in many areas. He offered invaluable critique, supported ideas that were worth it, and showed me the joy of developing them into state-of-the-art tools. I would like to thank my many collaborators, who trusted me with their data even though it was always new for me. I greatly appreciate all of the Electron Microscopy Group for many great discussions and an exceptionally friendly working environment.

I would like to thank my family for their unwavering support and my friend Mykolas for always offering a much-needed distraction. Finally, I would like to thank Eglė for the unending inspiration, support and care throughout these years, and for making me a better man.

Abstract

This thesis discusses various large multi-dimensional dataset analysis methods and their applications. Particular attention is paid to non-linear optimization analyses and general processing algorithms and frameworks when the datasets are significantly larger than the available computer memory. All new presented algorithms and frameworks were implemented in the HyperSpy analysis toolbox.

A novel Smart Adaptive Multi-dimensional Fitting (SAMFire) algorithm is presented and applied across a range of scanning transmission electron microscope (STEM) experiments. As a result, the Stark effect in quantum disks was mapped in a cathodoluminescence STEM experiment, and fully quantifiable 3D atomic distributions of a complex boron nitride core-shell nanoparticle were reconstructed from an electron energy loss spectrum (EELS) tilt-series. The EELS analysis also led to the development of two new algorithms to extract EELS near-edge structure fingerprints from the original dataset. Both approaches do not rely on standards, are not limited to thin or constant thickness particles and do not require atomic resolution. A combination of the aforementioned fingerprinting techniques and SAMFire allows robust quantifiable EELS analysis of very large regions of interest.

A very large dataset loading and processing framework, "LazySignal", was developed and tested on scanning precession electron diffraction (SPED) data. A combination of SAMFire and LazySignal allowed efficient analysis of large diffraction datasets, successfully mapping strain across an extended (ca. $1\,\mu m \times 1\,\mu m$) region and classifying the strain fields around precipitate needles in an aluminium alloy.

Table of contents

List of figures

Nomenclature

Acronyms / Abbreviations

ADF	Annular Dark Field
BEM	Boundary Elements Method
BF	Bright Field
BN	Boron Nitride
CC	Charge Carrier
CCD	Charge-coupled device
CL	Cathodoluminescence
CS	Compressed Sensing
DAG	Directed Acyclic Graph
DDA	Discrete Dipole Approximation
DP	Diffraction Pattern
EEL	Electron Energy Loss
EELS	Electron Energy Loss Spectrum
ELNES	Energy Loss Near Edge Structure
EM	Electron Microscopy
EQE	External Quantum Efficiency
FWHM	Full Width at Half Maximum

GOF	Goodness Of Fit test
HAADF	High Angle Annular Dark Field
HL	High energy Loss signal (typically $> 100\,\mathrm{eV}$)
ICA	Independent Component Analysis
LED	Light Emitting Diode
LL	Low energy Loss signal (typically $< 50\,\mathrm{eV}$)
LSPR	Localised Surface Plasmon Resonance
ML	Machine Learning
NBED	Nanobeam Electron Diffraction
NMF	Non-negative Matrix Factorization
NW	Nanowire
PCA	Principal Component Analysis
PC	Personal Computer
PES	Photoemission spectroscopy
QCSE	Quantum Confined Stark Effect
QDisk	Quantum Disk
RDD	Resilient Distributed Dataset
SAMFire	Smart Adaptive Multidimensional Fitting algorithm
SI	Spectral Image
SNR	Signal to Noise Ratio
S(P)ED	Scanning (Precession) Electron Diffraction
SPP	Surface Plasmon Polariton
SSD	Single Scattering Distribution
STEM	Scanning Transmission Electron Microscope

SVD	Singular Value Decomposition
TEM	Transmission Electron Microscope
TE	Transverse Electric
TM	Transverse Magnetic
TV	Total Variation
VDF	Virtual Dark Field
ZLP	Zero Loss Peak

Chapter 1

Introduction

Electron microscopes (EMs) have become a large part of many sciences and technologies where the spatial resolution of light microscopy is no longer sufficiently high. The field recently passed an important tipping point leading to accelerated growth. In particular, significant steps have been made from the technical side of EMs [1, 2], allowing previously unprecedented spatial and spectral resolutions. However, there is a caveat that comes with larger and more detailed data than ever before: the analysis often becomes just as important and difficult as the experiment. Previously weak interactions that were blurred and nearly invisible now have to be undone in the analysis stage. On the other hand, the typical size of a dataset nearly doubles every year, requiring even more computational resources. The end result is that old and historically tested data analysis and handling tools cannot keep up with the EM development, even with the growing computer processing power.

The goal of this thesis is to provide new and more advanced tools for data handling and analysis. While the inspiration for the work comes from electron microscopy, I believe to have managed to keep the methods reasonably general. Most experimental examples did not rely on state-of-the-art microscopes, but instead employed new and more powerful data analysis algorithms that offered previously unprecedented results.

Thesis outline

The Scanning Transmission Electron Microscope (STEM) and the electron interactions with the specimen in STEM are introduced in chapter 2. Chapter 3 includes my earliest work, which served as an inspiration to solve the encountered data analysis problems. The rest of the thesis can be grouped into two parts. The first, containing chapters 4 to 6, considers general data analysis. It presents common analysis methods as well as

often encountered problems when applying such methods to real data. Lastly it suggests my solutions to these problems. Even though electron microscopy is used for most examples, the methods are in principle general and can be applied to solve a large array of problems in many fields. The second part contains chapters 7 to 9, which describe various experimental data analysis results that were enabled by the algorithms and frameworks described in the first part. A brief description of each chapter follows.

Chapter 2: A brief introduction to Scanning Transmission Electron Microscopy (STEM)

This chapter gives a brief introduction to STEM. It describes the main working principles of an electron microscope and introduces the two STEM configurations that were used to acquire the data presented later in the thesis. This is followed by succinct descriptions of elastic and inelastic electron interactions with matter that are relevant to the rest of the work.

Chapter 3: Plasmons

Plasmons and, in particular, localised surface plasmon resonances (LSPRs) are introduced. Analytical solution for electron energy loss spectra (EELS) for an LSPR on a sphere as well as the Discrete Dipole Approximation (DDA) LSPR simulation descriptions are given. Finally, EELS response of a particle with smoothly changing shape from cube to sphere is simulated and analysed using the theoretical sphere solution.

Chapter 4: Large Multi-dimensional Data Analysis

Two common data analysis techniques, model fitting and machine learning, are introduced. Strengths, weaknesses and ease of use of both methods are discussed. The two main data analysis and handling issues that will be addressed in the work are presented.

Chapter 5: Smart Adaptive Multi-dimensional Fitting (SAMFire)

This chapter introduces the SAMFire algorithm. It includes a motivating example, explains the two proposed methods of solution, and finally discusses the architecture of the implementation of the algorithm. Three synthetic datasets, two based on real experimental results and one unrealistically complex, are analysed using both conventional methods and SAMFire, and results are compared.

Chapter 6: Big Data

A brief history of large dataset analysis tools is presented, followed by the description of the proposed "LazySignal" framework for data analysis and handling. The chapter also includes examples of operations that would otherwise be impossible to perform on regular computer hardware.

Chapter 7: Monitoring the Stark effect in quantum disks

This chapter considers the analysis and results of Quantum Disks (QDisks) grown in a nanowire. The cathodoluminescence (CL) response of the specimen was measured in a STEM. Quantum Confined Stark (QCSE) and Auger effects and their influences on the QDisk performance are introduced. Data from ten CL maps of the same nanowire are analysed and presented. QDisks are shown to experience efficiency droop, tentatively attributed to the Auger effect.

Chapter 8: Quantifying elemental and bonding maps in 3D in a TEM

This chapter describes the analysis and results of a core-shell BN nanoparticle measured using EELS in a STEM. After underlining the importance of 3D information when describing any system and presenting the sample, two new methods of extracting Energy Loss Near-Edge Structure (ELNES) from the experimental data are presented. Electron tomography is introduced and particular attention is paid to its compressed sensing (CS) implementations. Finally, the experimental data is quantified and reconstructed in 3D using the previously described methods, resulting in the first fully quantitative bonding electron tomography. The measured atomic densities are compared to theoretical values, followed by a discussion.

Chapter 9: Strain mapping in diffraction cartography

Two new ways of mapping strain over large areas of interest using scanning (precession) electron diffraction (S(P)ED) are presented. Strain analysis around the precipitates of an age-hardened aluminium alloy is presented. In parallel, machine learning (ML) is used to decompose that same dataset, allowing the separation of the precipitates into four categories based on their relative crystallographic orientations. ML and strain results are then combined to estimate the strain around the mean precipitate of each class, and later of the whole specimen.

Chapter 10: Conclusions

This gives a summary of the work presented in the thesis. Ideas and suggestions for further work follow.

Chapter 2

A brief introduction to Scanning Transmission Electron Microscopy (STEM)

Historically electron microscopes were developed to overcome the diffraction limit of light, which approximately limited the spatial resolution of an image to (at best) the wavelength of a photon. In contrast, a scanning transmission electron microscope (STEM) uses electrons instead of light to probe the sample: emitted from a gun, collimated and focused onto the specimen, the scattered electrons are collected and focused on a detector. The de Broglie wavelength ($\lambda = h/p$) for an electron is given by

$$\lambda = \frac{h}{\left[2m_0eV\left(1+\frac{eV}{2m_0c^2}\right)\right]^{1/2}}\,, \tag{2.1}$$

where h is Planck's constant and, p is the momentum of the electron expressed using the electron rest mass m_0, charge e, the potential through which it is accelerated V, and the speed of light c. As the accelerating voltages in transmission electron microscopes (TEMs) are usually in the range of 100 kV to 300 kV, the electron velocity is an appreciable fraction of the speed of light and the corresponding wavelength of the order of few picometers, serving the intended purpose.

In STEM the electron beam is raster-scanned over the sample, and the transmitted beam is detected after the sample. In addition, various detectors can be placed above and below the sample plane to collect any signal emitted by the specimen due to the electron excitation or strongly scattered electrons themselves. Of course, the electron beam has to be manipulated at least to the precision of the resolution we want to achieve. However

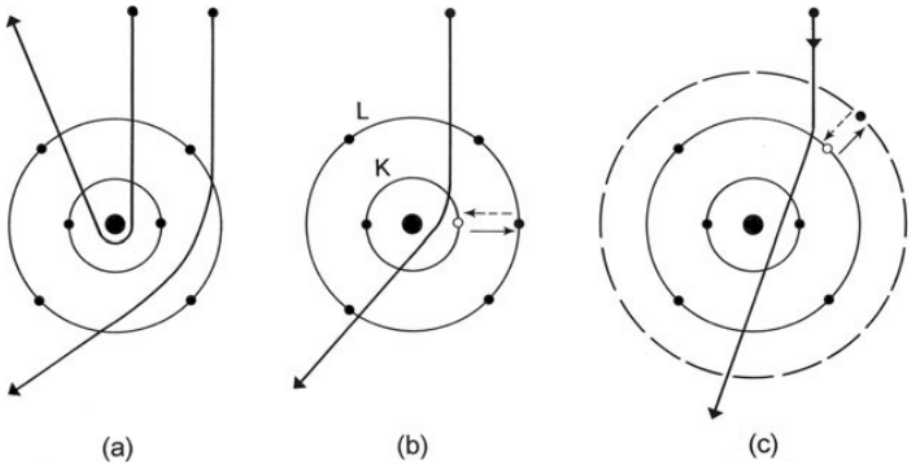


Fig. 2.1 A classical (particle) view of electron scattering by a single atom. (a) Elastic scattering by the nucleus. (b,c) show inelastic scattering by inner- or outer-shell electrons, respectively. Adapted from [4]

the electromagnetic lenses are severely affected by aberrations, limiting the resolution to approximately 150 pm. In specialized microscopes, however, spherical aberration correction has been implemented, enabling 50 pm resolution to be achieved [3].

The swift electron interaction can be separated into two groups: elastic, when no detectable energy is transferred to the sample and the electron interacts mainly with nuclei, and inelastic, when the probing electron interacts with sample electrons, transferring energy (Fig. 2.1). For most samples studied, elastic interactions dominate the contrast seen in electron diffraction, conventional transmission electron microscopy and high resolution electron microscopy. Inelastic scattering is the origin of the spectral signals detectable in a TEM: probing the electron's lost energy, which is measured as an electron energy loss spectrum (EELS), and also energy-dispersive X-ray spectroscopy and Cathodoluminescence (CL) experiments.

To acquire the data used in this work, two different configurations had to be used (on different physical microscopes), with schematic representations shown in Fig. 2.2. The first one is the traditional analytical STEM configuration, including bright and dark field detectors (each measuring the total intensity on the detector per probe position, just one number), as well as spectrometers both below the sample (for EELS) and above it (for CL), both recording a spectrum per probe position. The second one is typical for a scanning electron diffraction (SED) experiment. In this configuration the dark- and bright-field detectors in the back-focal plane are replaced by a pixelated detector, such as charge-coupled device (CCD) camera, able to measure not just a single intensity, but

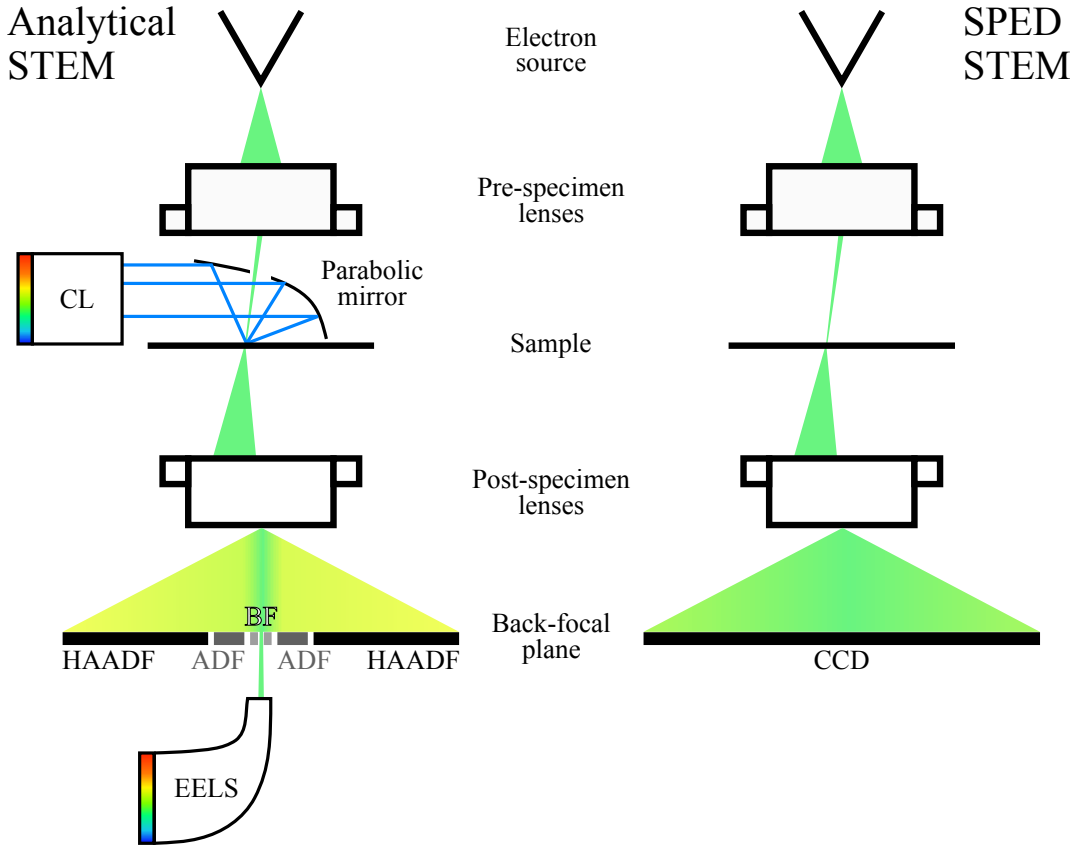


Fig. 2.2 A schematic representation of a STEM in analytical and Scanning Precession Electron Diffraction (SPED) configurations on the left and right respectively. Analytical STEM includes bright field (BF), annular dark field (ADF) and high-angle ADF (HAADF) detectors (each measuring the total intensity on the detector per beam position) and two spectrometers - CL above the sample plane and EELS below. In analytical setup the post-specimen lenses are focused with a large camera length, containing most coherently scattered electrons (shown in green) on the small BF detector, allowing (HA)ADF detectors to measure mostly incoherently scattered electrons. In SPED STEM configuration all back-focal plane detectors are replaced by a CCD detector, and the camera length is picked such that the coherently scattered part of the beam spans the full detector area.

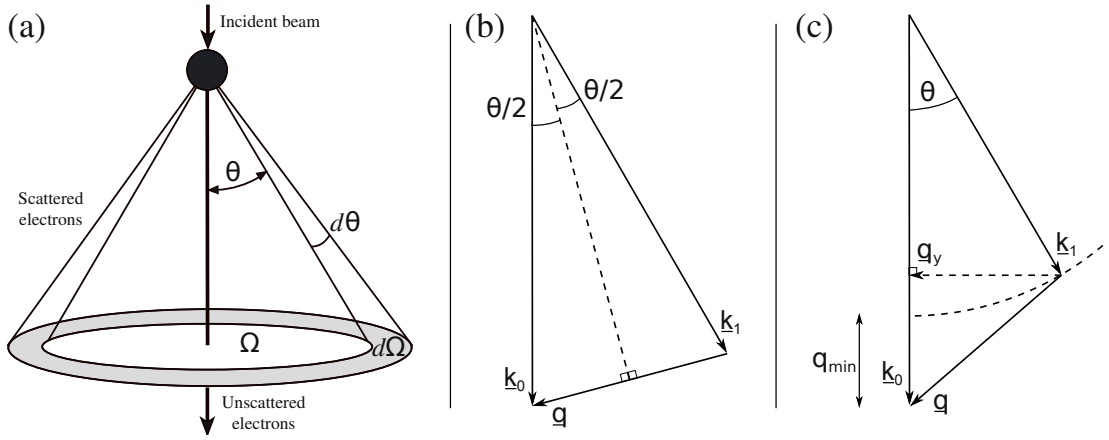


Fig. 2.3 (a) Electron scattering angle notation diagram. (b) Elastic scattering k-vector diagram, where $\boldsymbol{k}_0$ and $\boldsymbol{k}_1$ are for the initial and scattered beams respectively, and $\boldsymbol{q}$ for the energy transfered to the scatterer. (c) Inelastic scattering k-vector diagram, where q_{min} is the smallest lost momentum for the particular energy loss E. q_y is the scattering vector component perpendicular to $\boldsymbol{k}_0$.

the whole diffraction pattern. In addition, for precession experiment, the focused probe is manipulated above the specimen to produce a hollow cone illumination as presented in [5], with the opposite operation occurring before the detector, the net effect being equivalent to precessing the sample about a stationary beam. In such an experiment a diffraction pattern is measured in each probe position, resulting in a four-dimensional dataset.

2.1 Electron interactions

In this section a succinct description of electron-matter interaction is presented, most of it closely following the work of Egerton [4], which is also recommended for a more in-depth review of the topic. The section on cathodoluminescence follows the review article by Kociak and Zagonel [6].

2.1.1 Elastic scattering

A measure of interaction between an incident electron and an atom is the differential cross section $\frac{d\sigma}{d\Omega}$, which describes the effective area of the target in order for the exit trajectory to be in the solid angle $d\Omega$, Fig. 2.3(a). For elastic scattering this can be

written as

$$\frac{d\sigma}{d\Omega} = |f|^2 , \tag{2.2}$$

where f is a complex scattering amplitude, a function of the scattering angle θ or scattering vector $\boldsymbol{q}$. Within the first Born approximation, that is assuming only single scattering within each atom, f is proportional to the three-dimensional Fourier transform of the atomic potential $V(r)$.

Elastically scattered electrons interact with the atom via Coulomb forces. The simplest such interaction model is based on the unscreened electrostatic field of a nucleus, first used by Rutherford [7]. Both classical and quantum theory lead to the same expression, giving

$$\frac{d\sigma}{d\Omega} = \frac{4\gamma^2 Z^2}{a_0^2 q^4} , \tag{2.3}$$

where $\gamma = (1 - v^2/c^2)^{-1/2}$ is the relativistic factor for an electron moving at velocity v, $a_0 = 4\pi\varepsilon_0\hbar^2/m_0e^2 = 0.529 \times 10^{-10}$ m is the Bohr radius, $\varepsilon_0 = 8.854 \times 10^{-12}$ F m^{-1} the vacuum permittivity, Z the atomic number of the scattering atom, and q is the magnitude of the scattering vector, given by $q = 2k_0 \sin(\theta/2)$, where $\hbar\boldsymbol{k}_0 = \gamma m_0 \boldsymbol{v}$ is the momentum of the electron, Fig. 2.3(b).

The nucleus screening can be incorporated through the Yukawa potential with screening radius r_0 [8], leading to Lentz model:

$$\frac{d\sigma}{d\Omega} = \frac{4\gamma^2}{a_0^2}\left(\frac{Z}{q^2 + r_0^{-2}}\right)^2 \approx \frac{4\gamma^2 Z^2}{a_0^2 k_0^4}\frac{1}{\left(\theta^2 + \theta_0^2\right)^2} , \tag{2.4}$$

where $\theta_0 \approx Z^{1/3}/(k_0 a_0)$ is the characteristic angle of elastic scattering. Integrating eq. (2.4) over all scattering angles gives the total elastic cross section:

$$\sigma_e = \int_0^\pi \frac{d\sigma}{d\Omega} 2\pi \sin\theta d\theta = \frac{4\pi\gamma^2}{k_0^2} Z^{4/3} = (1.87 \times 10^{-24}\mathrm{m}^2) Z^{4/3} (v/c)^{-2} . \tag{2.5}$$

While the accuracy of this model decreases for heavy elements, it serves as a useful approximation.

When the electron is scattered over large ($50 - 150$ mrad) angles, the electron passes closer to nucleus and thus the effect of the atomic electrons is small. In this case the differential cross section for elastic scattering is close to the Rutherford value, eq. (2.3), which can be integrated between some smallest considered angle θ_0 and π, resulting in $\sigma_R \propto Z^2$. HAADF detectors are specifically made to image the large scattering angle signals, with their measured intensities calculated as $I_d = NI\sigma_d$, where N is the number

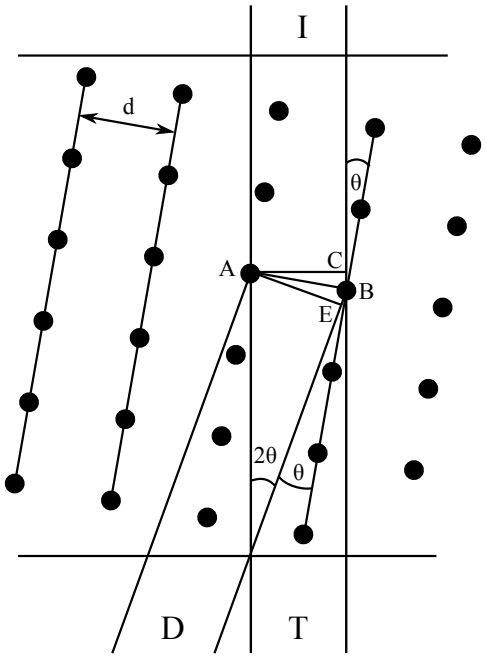


Fig. 2.4 A beam (I) is incident on a crystal with lattice parameter d. The scattered beam (D) is intense if the Bragg conditions are met.

of atoms per unit area, I is the number of electrons per second in the beam, and σ_d the relevant cross section. At these large angles it means that HAADF images show not only thickness ($I_d \propto N$), but also atomic number ($I_d \propto Z^2$) contrast.

If the material is crystalline, the regular arrangement of atom positions requires taking into account the phase difference between scattered beams when calculating the final intensity. This is done by replacing the scattering amplitude f in eq. (2.2) with the structure factor $F(\theta)$, a sum of all atoms j in a unit cell, each with the associated phase $\boldsymbol{q} \cdot \boldsymbol{r}_j$:

$$\begin{aligned} F(\theta) &= \sum_j f_j(\theta) \exp\Big(-i\boldsymbol{q} \cdot \boldsymbol{r}_j\Big) \,, \\ F(\theta) &\propto \int V(r) \exp(-\boldsymbol{q} \cdot \boldsymbol{r}) d\tau \,, \end{aligned} \tag{2.6}$$

where $V(r)$ is the scattering potential, and the integral is over all volume elements $d\tau$ in a unit cell.

Consider the interaction of an electron beam with a very thin slice of a perfect cubic crystal, a cross-section of which is shown in Fig. 2.4. As swift electrons pass through the crystal, some of the atoms, such as those marked A and B, will elastically scatter the beam due to the Coulomb forces. Because the incident beam (I) is coherent and the elastic scattering at small angles does not degrade the coherence, diffracted beams are also coherent. As a result, the scattered electrons interfere. An intense beam (D)

is formed if the path difference for the two shown trajectories is an integer number of electron wavelengths:

$$n\lambda = \mathrm{CB} + \mathrm{BE} = 2d\sin\theta\,. \tag{2.7}$$

This relation is well known as Bragg's law, and is widely applied in many fields. Here n is the diffraction order, and d is the distance between the considered atom planes.

2.1.2 Inelastic scattering

Inelastic electron scattering can be derived in both quantum (Bethe theory [9]) and dielectric frameworks. For brevity, only the latter is presented in this work. Ritchie in 1957 [10] considered the transmitted electron to have a coordinate $\boldsymbol{r}$ and velocity $\boldsymbol{v}$ when moving in the $\hat{z}$ direction. Such an electron can be represented as a point charge $-e\delta(\boldsymbol{r}-\boldsymbol{v}t)$ that generates within the medium a spatially and time dependent potential $\phi(\boldsymbol{r},t)$ which satisfies the Poisson's equation

$$\varepsilon_0\varepsilon(\boldsymbol{q},\omega)\nabla^2\phi(\boldsymbol{r},t) = e\delta(\boldsymbol{r},t)\,, \tag{2.8}$$

where $\varepsilon(\boldsymbol{q},\omega)$ is the dielectric response function of the medium. The stopping power (dE/dz) on the transmitted electron is equal to the force in the $-\hat{z}$ direction, and can be calculated from the potential gradient in the same direction. Using Fourier transforms, Ritchie showed that

$$\frac{dE}{dz} = \frac{2\hbar^2}{\pi a_0 m_0 v^2}\iint \frac{q_y\omega\,\mathrm{Im}[-1/\varepsilon(q,\omega)]}{q_y^2+(\omega/v)^2}\,dq_y\,d\omega\ , \tag{2.9}$$

where $E=\hbar\omega$ and q_y is the scattering vector component perpendicular to $\boldsymbol{v}$ (Fig. 2.3(c)). The imaginary part of $[-1/\varepsilon(q,\omega)]$ is known as the energy-loss function and provides a complete medium response description. The stopping power can be related to the double-differential cross section (per atom) of the inelastic scattering by

$$\frac{dE}{dz} = \iint n_a E\frac{d^2\sigma}{d\Omega dE}\,d\Omega\,dE\,, \tag{2.10}$$

where n_a is the number of atoms per unit volume of the medium. For small scattering angles $dq_y \approx k_0\theta$ and $d\Omega \approx 2\pi\theta d\theta$, giving

$$\frac{d^2\sigma}{d\Omega dE} \approx \frac{\mathrm{Im}[-1/\varepsilon(q,E)]}{\pi^2 a_0 m_0 v^2 n_a}\left(\frac{1}{\theta^2+\theta_E^2}\right)\,. \tag{2.11}$$

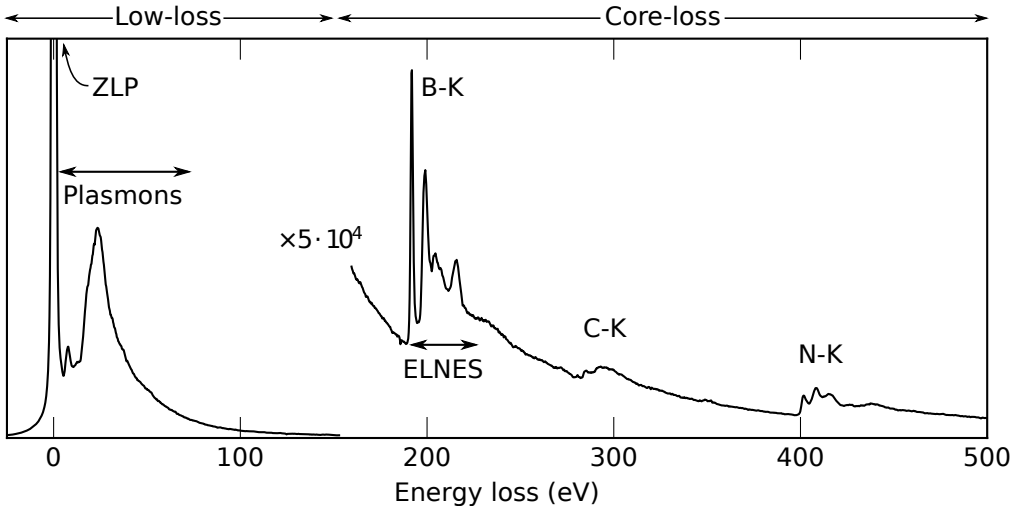


Fig. 2.5 Example EEL spectrum for a BN nanoparticle, further discussed in chapter 8. ZLP at 0 eV dominates the spectrum, with plasmon loses visible from just above 0 eV to around 50 eV, depending on the material. In this example "low-loss" and "core-loss" spectrum regions are clearly separated by the gain change at around 150 eV. B-K, C-K and N-K edges are labeled, with clearly visible B-K ELNES structure around 50 eV after the onset.

Here $\theta_E = E/(\gamma m_0 v^2)$ is the characteristic angle, where the total scattering vector q is approximated as $q^2 \approx 4k_0^2(\theta/2)^2 + q_{\text{min}}^2 = k_0^2(\theta^2 + \theta_E^2)$ [4], see Fig. 2.3(c). Eq. (2.11) allows the calculation of energy loss cross sections for the angles of interest. This full response is usually further divided into the swift electron interaction with outer- and inner-shell electrons of the scattering atom. The former scattering events are significantly more frequent than the latter, and are described next.

Plasmons in EELS

The dominant feature in the low-loss part of the spectrum and in EELS in general is the zero loss peak (ZLP), which represents electrons leaving the sample with negligible energy difference (Fig. 2.5). As the electrons are usually highly relativistic with relatively long mean free paths, the ZLP is often much more intense than other features in the spectrum. The next major contribution for a solid comes from plasmons. Ritchie in 1957 [10] first identified that in addition to a volume plasmon, a resonance arising from the boundary conditions for electric and magnetic fields also contributes to the energy loss of an electron, named the surface plasmon. The description of EELS of surface plasmons has been derived in both quantum [11] and classical dielectric theory, which I will briefly describe here.

A fast electron, moving with constant velocity $\boldsymbol{v}$ along a straight-line trajectory $\boldsymbol{r}_e(t)$, loses energy by doing work against the force due to the scattered electric field $\boldsymbol{E}_{\text{sca}}$ acting back on the electron [10]:

$$\Delta E = \int_{-\infty}^{\infty} dt\bar{e}(\boldsymbol{v} \cdot \boldsymbol{E}_{\text{sca}}[\boldsymbol{r}_e(t), t]) = \int_0^{\infty} d\omega\hbar\omega\Gamma_{\text{EELS}}(\omega)\,, \tag{2.12}$$

where Γ_{EELS} is the probability that electron loses energy. Because $\boldsymbol{E}_{\text{sca}}$ is real, the expression can be simplified by Fourier transforms of the scattered field, resulting in

$$\Gamma_{\text{EELS}}(\omega) = \frac{\bar{e}}{\pi\hbar\omega}\int_{-\infty}^{\infty} dt\,\text{Re}\left[\exp(-i\omega t)(\boldsymbol{v} \cdot \boldsymbol{E}_{\text{sca}}[\boldsymbol{r}_e(t), \omega])\right]\,. \tag{2.13}$$

The problem is then simplified to finding the $\boldsymbol{E}_{\text{sca}}$. The derivation shown in [12] uses the quasi-static approximation, where the speed of light is assumed to be infinite, leading to instantaneous interactions. Using the Greens function solution to express the swift electron potential, the energy loss probability is written as [12]

$$\Gamma_{\text{EELS}} = -\frac{1}{\hbar}\int_{-\infty}^{\infty} dz\,\text{Im}\left\{\rho^*(\boldsymbol{R}_0, z, \omega)\phi_{\text{ind}}(\boldsymbol{R}_0, z, \omega)\right\}\,, \tag{2.14}$$

which can, assuming small angles ($z\omega/v \to 0$), be simplified further to

$$\Gamma_{\text{EELS}} = -\frac{1}{\hbar}\int_{-\infty}^{\infty} dz\cos(z\omega/v)\,\text{Im}\left\{\phi_{\text{ind}}(\boldsymbol{R}_0, z, \omega)\right\} \approx -\frac{1}{\hbar}\int_{-\infty}^{\infty} dz\,\text{Im}\left\{\phi_{\text{ind}}(\boldsymbol{R}_0, z, \omega)\right\}\,, \tag{2.15}$$

where ρ is the swift electron charge density in ($\boldsymbol{R}$, z) spatial coordinates, and ϕ_{ind} is the induced potential. Eq. (2.15) suggests that the EELS probability can be approximately described as the projection of the imaginary part of the induced potential.

The Plasmon EEL probability in the fully relativistic case was solved by García de Abajo [13]. In principle calculating Γ_{EELS} is possible if, in addition to frequency-dependent dielectric function of the material, the screened interaction (quasi-static) or Green's tensor (relativistic) is known for the particular geometry. However, the latter part proved to be rather challenging for arbitrary geometries, with full analytical solutions found only in highly symmetric cases (e.g. a solution for a sphere is given in section 3.1.1). A range of approximate methods have been developed to calculate the probabilities for arbitrary geometries.

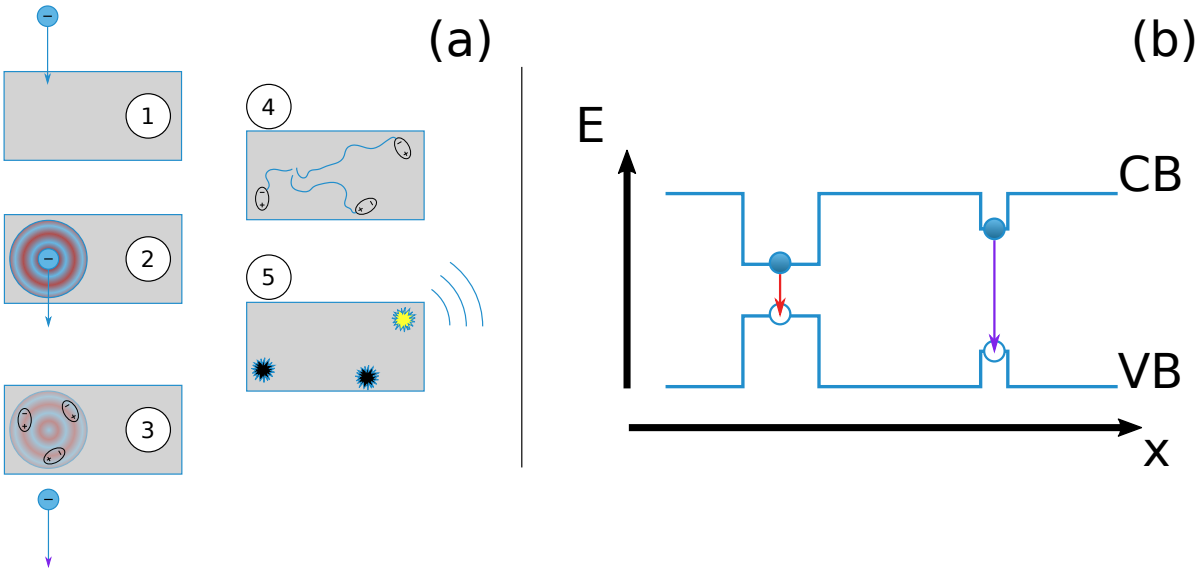


Fig. 2.6 (a) Events, necessary to create incoherent CL emission. A swift electron approaches the material (1) and excites its volume plasmon (2) which rapidly decays into electron-hole pairs (3). Both electrons and holes then diffuse (possibly independently, unlike shown in the figure) to local band gap variations representing energy minima (4), and recombine either radiatively or non-radiatively (5). (b) Local optical transition energy (band gap) variation with arbitrary position coordinate x. Both adapted from [6].

Cathodoluminescence

Cathodoluminescence (CL) is the emission of light from a material upon interaction with an electron. While it has been used in the past as a regular characterisation technique [14], recently the field received a lot of attention from various nanomaterial and nanostructure researchers. The main reason of such resurgence of interest is the ability to make the electron probe effectively arbitrarily small, allowing probing the specimen very locally and precisely.

Following the review by Kociak et. al [6], there are two paths for creating luminescence from a fast electron. The first considers coherent electrostatic waves such as a plasmon or a polariton and their decay into photons. This type of CL experiments led to the first ever electron-based spectroscopic measurement of a plasmon [15] and continued to be an important tool in many other plasmonic nanoparticle experiments [6, 16]. This work, however, does not consider any coherent CL excitations, thus the interested reader is advised to use the previously mentioned review as an excellent reference.

Chapter 7 considers CL experiments where incoherent excitations are used to probe the specimen, Fig. 2.6. We consider the photon emission process as a series of steps, starting with the electron inelastically scattering from the thin specimen. As shown in

Fig. 2.5, the most likely of such scattering events occur by exciting the bulk plasmon of the material, typically in the 20–30 eV range, depending on the material. The bulk plasmons, with a lifetime of a few fs, quickly decay into electron–hole (e-h) pairs [17]. The charge carriers then diffuse within the material and recombine at spots that represent local energy minima. Importantly, these minima can be intentionally engineered by locally changing the composition of the material, or more accidental, such as point defects in nanodiamonds [17]. If the particular location of the minimum allows radiative decay, a photon of the corresponding wavelength is emitted [17] and can be used to measure the relative band gap with $\sim$1 nm spatial resolution [6].

Inner-shell electron excitations

If the swift electrons have sufficient energy, they are able to excite one of the inner-shell electrons of the specimen atom. Usual excitation energies are often significantly higher than those of plasmon interactions and highly depend on the atom species. This gives a way to measure the chemical composition of the sample with the spatial resolution of the focused probe.

Typically, tabulated values of ionization cross sections are used when analysing data. The most basic approximation considers neutral isolated atoms described by hydrogenic wave functions [4]. A more accurate set of cross sections has been calculated [18, 19] using the Hartree-Slater method.

All core-loss edges have certain features that correspond to various energy-transfer methods. In particular, the first $\sim$50 eV after the edge onset are called energy-loss near-edge structure (ELNES), marked for B-K edge in Fig. 2.5. These modulations of the single-scattering intensity can be related to the band structure of the scattering solid. In a one-electron approximation Fermi's Golden Rule [20] says that the transition rate is proportional to the final density of states $\rho(E)$ and the atomic transition matrix $M(E)$:

$$\frac{d\sigma}{dE} \propto |M(E)|^2 \rho(E)\,. \tag{2.16}$$

Intuitively, the transition matrix represents the overall shape of the energy-loss edge, determined by atomic physics, whereas $\rho(E)$ describes the chemical and crystallographic environment of the exited atom. Assuming $M(E)$ to be slowly varying with energy-loss, $\rho(E)$ is a local density of states above the Fermi level, allowing a direct measure of the surrounding environment of the atom in question. As a result, different chemical bonds of an atom can be mapped as measurable ELNES shape differences [21]. Crucially, neither

hydrogenic nor Hartree-Slater cross section calculations take these features into account, and they have to be modelled separately.

The measured core-loss spectrum gets more complicated with increasing specimen thickness. As the sample gets thicker, electrons are more likely to undergo multiple inelastic scattering, "smearing" the single scattering distribution (SSD). As noted by Verbeeck [22], in general the measured spectrum $J(E)$ can be viewed as

$$J(E) = O(E) \otimes P(E) + N(E)\,, \tag{2.17}$$

where $O(E)$ is the SSD, $P(E)$ is the point-spread function describing both multiple scattering and the instrumental broadening, and finally $N(E)$ is the noise term. A number of different deconvolution approaches are present to estimate $O(E)$ from $J(E)$ [4], however their application and results are often subjective and provide few ways to estimate the result quality and validity. Instead, when analysing core-loss EELS in this work we will use the model-fitting approach [22, 23]. It relies on having access to both the high-loss (HL) of interest and the low-loss (LL) spectra at the same time. By using LL as the point-spread function, the $O(E)$ term in eq. (2.17) can be directly modeled and $J(E)$ compared to the measured data. Such approach not only avoids the usual deconvolution problems, but also allows an estimate of the error for the fit results.

Chapter 3

Plasmons

Plasmonics is a rapidly growing field of interest in many scientific communities with many potential applications, pushing our current theoretical understanding of the phenomenon forward. A plasmon is a collective coherent oscillation of electron "cloud" in a material and on the surface. As will be shown in Section 3.1, plasmons are highly dependent on the dielectric surroundings and the shape of the excited particle. These properties make plasmons promising in all applications where sub-wavelength light manipulation is desired. The high dependence on the geometry of the nanoparticle enables potential applications using highly localized and enhanced electric fields (such as waveguides or signal enhancement), whereas the high sensitivity to the surrounding dielectric environment drives research in sensing applications. Here I include a brief (and incomplete) overview of the potential applications of plasmons.

Thermal activators

Metal nanoparticles, due to highly resonant plasmon absorption at certain wavelengths, exhibit light-induced heating that can be used to control chemical reactions with high spatial and temporal resolution [24]. The same property is also used in medical research, enabling killing cancer cells while not affecting its surroundings [25] or delivering drugs in temperature-controlled shells [26].

Sensing

The plasmonic properties of a metal nanoparticle are highly sensitive to its dielectric surroundings. In particular, refractive index variations energy shift extinction and scattering spectral features. The sensitivity enables real-time monitoring of molecular changes [27] and nanoparticle sensors [28].

Molecular spectroscopy

The high-intensity local electric fields near plasmonic nanoparticles at resonant frequencies have been used in several molecular identification techniques, such as enhanced Raman spectroscopy [29] and laser desorption ionization mass spectrometry [30, 31], reducing the required incident radiation multiple times.

Light concentrators

Plasmonic responses of nanorods and metal strips include surface plasmons and surface plasmon polaritons propagating according to Fabry-Pérot resonator laws, resulting in surface plasmon resonances. These, in turn, have the potential to be used as sub-wavelength dielectric waveguides [32, 33]. In order to couple light to the plasmonic waveguides, other plasmon nanostructures have been proposed to act as lenses [34, 35].

Surface plasmons clearly have applications in many diverse fields, however this chapter focuses on the study of fundamental physics of localised surface plasmon resonances at the nanoscale. The measurement and excitation used for the study is Scanning Transmission Electron Microscope (STEM), more specifically electron energy loss spectroscopy (EELS), where the electron acts as a probe for plasmons, enabling direct study of the phenomenon.

3.1 Theoretical background

The theoretical description and derivation closely follows [12] throughout this part of the work, so only additional references are given in the text.

Drude model and volume plasmons

When considering light interactions with matter, the material properties have to be known. The simplest model of electrical conduction of materials is called the Drude model, which works remarkably well for many metals.

The Drude model considers a material to be a collection of stationary ions in a "sea" of free electrons. The electrons are considered to be independent of other electrons, and interact only with the ions (in hard sphere collisions) and external fields. The model also assumes that the average time between subsequent electron collisions is τ, known as the relaxation time of the free electron gas, resulting in a characteristic collision frequency $\gamma = 1/\tau$. At room temperatures typical values of τ are of the order of $1 \times 10^{-14}\,\mathrm{s}$, corresponding to $\gamma = 100\,\mathrm{THz}$. Then for an average electron in the plasma sea subjected to an external electric field $\boldsymbol{E}$ a simple equation of motion (not including the ion cores, because in the model they are of infinite effective mass) can be written:

$$m_0\ddot{\boldsymbol{x}} + m_0\gamma\dot{\boldsymbol{x}} = -\bar{e}\boldsymbol{E}\,. \tag{3.1}$$

where $\bar{e}$ and m_0 is the electric charge and mass of the electron, respectively. Assuming the driving field has a harmonic time dependence $\boldsymbol{E}(t) = \boldsymbol{E}_0 e^{-i\omega t}$, a particular solution of the form $\boldsymbol{x}(t) = \boldsymbol{x}_0 e^{-i\omega t}$ can be shown to be

$$\boldsymbol{x}(t) = \frac{\bar{e}}{m_0(\omega^2 + i\gamma\omega)}\boldsymbol{E}(t)\,. \tag{3.2}$$

The displaced electrons contribute to the polarization $\boldsymbol{P}$:

$$\boldsymbol{P} = -\bar{e}n\boldsymbol{x} = -\frac{n\bar{e}^2}{m_0}\frac{1}{\omega^2 + i\gamma\omega}\boldsymbol{E}\,, \tag{3.3}$$

where n is the number of free electrons per unit volume. Eq. (3.3) and the definition of polarisation can then be used to write [36]

$$\varepsilon(\omega) = 1 - \frac{\omega_p^2}{\omega^2 + i\gamma\omega}\,, \tag{3.4}$$

where $\omega_p^2 = n\bar{e}^2/(\varepsilon_0 m_0)$ is the plasma frequency.

The model has to be extended for noble metals (e.g. Au, Ag, Cu) in the region $\omega > \omega_p$, where the response is dominated by free s electrons. Since the d band of the aforementioned metals is very close to the Fermi energy, the threshold energies for the $d \to s$ interband transitions are very small, lying in the visible or near-ultraviolet regimes. The non-typical absorption at those energies not only results in the distinctive colours of the metals, but also causes a highly polarized environment due to the positive background of the ion cores. It can be taken into account by adding a new term $\boldsymbol{P}_\infty = \varepsilon_0(\varepsilon_\infty - 1)$ the polarisation definition, so that now $\boldsymbol{P}$ represents only the polarization (3.3) due to free electrons, and this residual polarization is described solely by ε_∞ (usually $1 \leq \varepsilon_\infty \leq 10$):

$$\varepsilon(\omega) = \varepsilon_\infty - \frac{\omega_p^2}{\omega^2 + i\gamma\omega} \tag{3.5}$$

By considering eq. (3.5) in transverse electric and transverse magnetic field cases [12], the response is split into two different regimes: for $\omega < \omega_p$ transverse electromagnetic waves do not propagate and decay exponentially in the metal plasma, whereas for $\omega > \omega_p$ the metal is transparent to radiation, with transverse waves travelling with a dispersion relation

$$\omega^2 = \omega_p^2 + k^2 c^2 \tag{3.6}$$

Surface plasmon

The previous plasmon description only considered a homogeneous medium, hence is valid only in the bulk of a conductor. If, however, there exists an interface across which the real part of dielectric function changes sign, it will be able to support Surface Plasmon Polaritons (SPPs)[1]. It can be briefly explained starting with the Helmholtz equation:

$$\nabla^2 \boldsymbol{E} + \frac{\omega^2}{c^2}\varepsilon \boldsymbol{E} = 0\,. \tag{3.7}$$

By taking the interface to be in the $z = 0$ plane and with SPP propagating in the x direction the equation simplifies to

$$\frac{\partial^2 \boldsymbol{E}(z)}{\partial z^2} + \left(\frac{\omega^2}{c^2}\varepsilon - k_x^2\right)\boldsymbol{E} = 0\,. \tag{3.8}$$

[1] A polariton is a quasiparticle, resulting from strong coupling of electromagnetic waves with an electric or magnetic dipole-carrying excitation, in this case a surface plasmon [37].

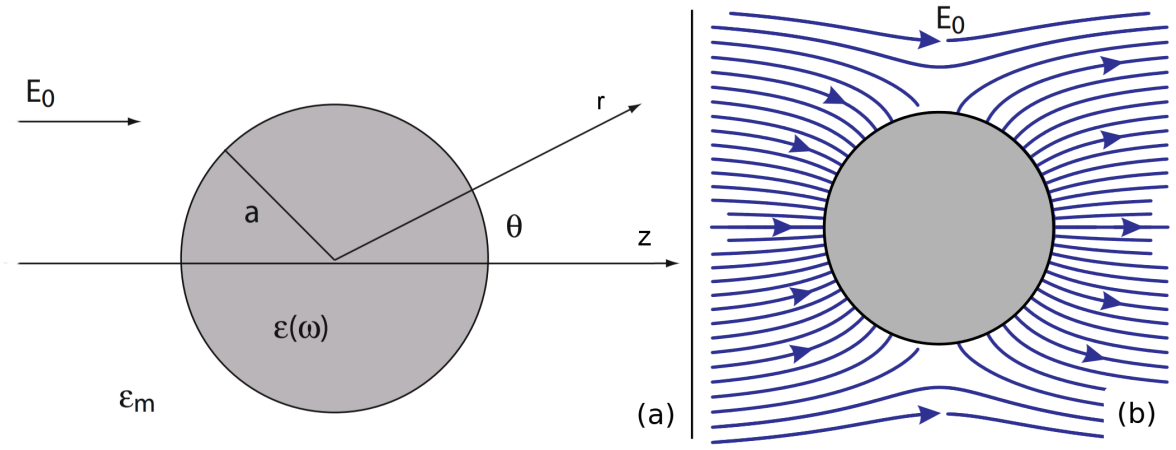


Fig. 3.1 (a) Scheme of a homogeneous isotropic sphere of radius a and dielectric function $\varepsilon(\omega)$ in a dielectric medium of dielectric constant ε_m in a uniform electric field $\boldsymbol{E}_0 = E_0\hat{\boldsymbol{z}}$. (b) Electric field lines, in the case the same sphere is a conductor. Adapted from [12]

Writing out the wave equations for electric and magnetic fields along the interface reveals that the system supports Transverse Magnetic (TM) ($E_y, H_x, H_z = 0$) and Transverse Electric (TE) ($E_x, E_z, H_y = 0$) mode propagation. However, interface continuity requirements for the TE mode are only fulfilled with zero amplitudes, leaving TM as the only allowed SPP mode, for which the dispersion relation is given by

$$k_x = \frac{\omega}{c}\sqrt{\frac{\varepsilon_1\varepsilon_2}{\varepsilon_1 + \varepsilon_2}}, \tag{3.9}$$

where $\varepsilon_{1,2}$ correspond to the dielectric function of the conductor halfspace and a real dielectric constant of the dielectric halfspace, respectively.

Localised surface plasmon resonance

If the considered geometry is finite and confined, for example a nanoparticle, the plasmon excitations instead form non-propagating Localised Surface Plasmon Resonances (LSPRs), with full analytical solutions only available for highly symmetric geometries [13]. As an example, consider a homogeneous isotropic sphere of radius a under the quasi-static approximation in vacuum (or air) (see Fig. 3.1). Due to the symmetry of the particle, it can be described in terms of spherical coordinates. By considering the boundary conditions at the surface of the sphere and at infinity, one can show that for an external electric field E_0 potentials inside and outside the particle can be expressed as Φ_{in} and

Φ_{out}, respectively [36]:

$$\Phi_{\text{in}} = -\frac{3\varepsilon_{\text{m}}}{\varepsilon + 2\varepsilon_{\text{m}}} E_0 r \cos\theta \tag{3.10}$$

and

$$\Phi_{\text{out}} = -E_0 r \cos\theta + \frac{\varepsilon - \varepsilon_{\text{m}}}{\varepsilon + 2\varepsilon_{\text{m}}} E_0 a^3 \frac{\cos\theta}{r^2} \,. \tag{3.11}$$

Here ε_{m} is the embedding medium dielectric constant, ε is the dielectric function of the sphere, and r is the distance from the centre of the particle. Explicitly decomposing the outside potential into a dipole moment $\boldsymbol{p}$ and the external field and using polarizability α, definition of $\boldsymbol{p} = \varepsilon_0 \varepsilon_{\text{m}} \alpha \boldsymbol{E}_0$, this can be written as [36]

$$\Phi_{\text{out}} = -E_0 r \cos\theta + \frac{\boldsymbol{p} \cdot \boldsymbol{r}}{4\pi\varepsilon_0 \varepsilon_{\text{m}} r^3} \,, \tag{3.12}$$

$$\boldsymbol{p} = 4\pi\varepsilon_0 \varepsilon_{\text{m}} a^3 \frac{\varepsilon - \varepsilon_{\text{m}}}{\varepsilon + 2\varepsilon_{\text{m}}} \boldsymbol{E}_0 \,, \tag{3.13}$$

therefore allowing us to write

$$\alpha = 4\pi a^3 \frac{\varepsilon - \varepsilon_{\text{m}}}{\varepsilon + 2\varepsilon_{\text{m}}} \,. \tag{3.14}$$

The polarizability α, a complex quantity, clearly has resonances at minima of $|\varepsilon + 2\varepsilon_{\text{m}}|$, which can be simplified for materials with small imaginary part of ε to

$$\text{Re}\left[\varepsilon(\omega)\right] = -2\varepsilon_{\text{m}} \,. \tag{3.15}$$

For a sphere of Drude metal in air this relation gives a resonant frequency $\omega_0 = \omega_p/\sqrt{3}$.

3.1.1 Analytical EELS solution for a sphere plasmon

García de Abajo showed in 1999 [38] that a fully relativistic EELS probability for a sphere of radius a and impact parameter b can be expressed as

$$\begin{aligned} \Gamma^{\text{EELS}}(b,\omega) = & \frac{1}{\omega} \sum_{l=1}^{\infty} \sum_{m=-l}^{l} \Big[C_{lm}^{\text{EELS},a} \,\text{Im}(i a_l) \\ & + C_{lm}^{\text{EELS},b} \,\text{Im}(i b_l) \Big] \end{aligned} \tag{3.16}$$

where a_l and b_l are electric and magnetic Mie expansion coefficients, given as

$$a_l = \frac{\varepsilon j_l(x_2)[x_1 j_l(x_1)]' - j_l(x_l)[x_2 j_l(x_2)]'}{\varepsilon \left[x_1 h_l^{(1)}(x_1)\right]' j_l(x_2) - h_l^{(1)}(x_1)[x_2 j_l(x_2)]'} \tag{3.17}$$

$$b_l = \frac{j_l(x_2)[x_1 j_l(x_1)]' - j_l(x_l)[x_2 j_l(x_2)]'}{\left[x_1 h_l^{(1)}(x_1)\right]' j_l(x_2) - h_l^{(1)}(x_1)[x_2 j_l(x_2)]'} \tag{3.18}$$

with $x_1 = ka$, $x_2 = ka\sqrt{\varepsilon}$, $k = 2\pi/\lambda$ being the wave number, and j_l and $h_l^{(1)}$ spherical Bessel functions and spherical Hankel functions respectively. Primes denote derivatives with respect to the argument x_1 or x_2. In eq. (3.16) coefficients $C^{\mathrm{EELS},a}$ and $C^{\mathrm{EELS},b}$ are given as

$$C_{lm}^{\mathrm{EELS},a} = K_m^2\left(\frac{\omega b}{v\gamma}\right)\frac{1}{l(l+1)}|2mN_{lm}|^2 \tag{3.19}$$

$$C_{lm}^{\mathrm{EELS},b} = K_m^2\left(\frac{\omega b}{v\gamma}\right)\frac{1}{l(l+1)}\left|\frac{c}{v\gamma}M_{lm}\right|^2 \tag{3.20}$$

where $\gamma = 1/\sqrt{1 - v^2/c^2}$ is the Lorentz factor, K_m is the modified Bessel function of the second kind, and N_{lm} and M_{lm} are given in terms of Gegenbauer polynomials G_n^u:

$$N_{lm} = \sqrt{\frac{(2l+1)}{\pi}\frac{(l-|m|)!}{(l+|m|)!}}\frac{(2|m|-1)!!}{(v\gamma/c)^{|m|}}G_{l-|m|}^{|m|+l/2}\left(\frac{c}{v}\right) \tag{3.21}$$

$$\begin{aligned} M_{lm} =& N_{lm+1}\sqrt{(l+m+1)(l-m)} \\ &+ N_{lm-1}\sqrt{(l-m+1)(l+m)} \end{aligned} \tag{3.22}$$

By examining eq. (3.16) it can be seen that the full EELS response of a sphere can be decomposed into an infinite collection of different order contributions ($l = 1 \to \infty$), each having a spectral shape of a peak. With small sphere radii ($a \leq$40 nm for silver) all higher orders have roughly the same energy and increasingly smaller amplitudes, allowing a truncation of the infinite series while still keeping an accurate spectral response. However as the relativistic retardation effects increase, lower order peaks get broader and redshifted, leading to higher order features becoming visible.

3.2 Simulations

For most particles of experimental interest no analytical LSPR solutions currently exist, hence numerical simulations have to be employed in order to compare experiments and theory. Usually such simulations rely on discretization of space or time, solving either Poisson's equations for quasi-static approximation, or Maxwell's equations, if full relativistic effects are required. Finite difference time domain [39] and discontinuous Galerkin time domain [39] simulations have both been applied to plasmons. Other

methods, operating in the frequency domain, such as Boundary Elements Method (BEM) [40] and Discrete Dipole Approximation (DDA), have also been successfully used in plasmonics [41] and will be described in more detail.

Discrete dipole approximation (DDA)

The Discrete Dipole Approximation (DDA) is one of the ways to numerically solve Maxwell's equations by describing the volume in question as a collection of small dipoles. The idea stems from the fact that every atom can be (to the first order) approximated as a dipole, and hence by increasing the space discretization to sufficiently small subvolumes the real response should be recovered. Draine and Flatau [42] established an empirical limit when the DDA gives reasonable results: $|m|kd < 0.5$, where m is the complex refractive index, k is the wavenumber of radiation, and d is the dipole spacing. For most materials and wavelengths in question the dipole spacing (and the dipole volume) can be appreciably larger than the inter-atomic or inter-molecular spacing, as long as the shape of the particle is described faithfully enough (i.e. increasing the discretization does not change the result).

At the heart of DDA is the Maxwell's equation solution in terms of incident electric field $\boldsymbol{E}$, polarizations of the dipoles $\boldsymbol{P}$ and the polarizability matrix A:

$$\boldsymbol{E} = A\boldsymbol{P}\,. \tag{3.23}$$

The polarizations are induced by the total field at the point of the dipole, which in turn can be thought of as the sum of the incident field plus the field due to all the remaining dipoles:

$$P_j = \alpha_j E_j\,, \tag{3.24}$$

$$E_j = E_{\text{inc}} - \sum_{k\neq j} A_{jk} P_k\,. \tag{3.25}$$

The A_{jk} is the complex relativistic Greens dyad of free space, usually expressed as

$$\begin{aligned} A_{jk} &= \left[\boldsymbol{I}_3 + \frac{1}{k^2}\nabla\nabla\right] g_{jk}\,, \\ g_{jk} &= \frac{1}{4\pi r_{jk}} e^{ikr_{jk}}\,, \end{aligned} \tag{3.26}$$

with ∇ being the gradient with respect to $\boldsymbol{r_j}$, identity matrix $\boldsymbol{I}_3$, and $r_{jk} = |\boldsymbol{r}_j - \boldsymbol{r}_k|$. The trick is then to replace the diagonal elements of A, which are usually zero as the dipole does not feel the field of itself, with α^{-1}, which then allows us to recover eq. (3.25).

This, in turn, can be expressed as a set of $3N$ equations for N dipoles. The two missing pieces are the incident field and polarizability of the material. The former, in the EELS case, has been shown [43] to be

$$\boldsymbol{E}(\boldsymbol{r},\omega) = \frac{2\bar{e}\omega}{v^2\gamma\varepsilon} e^{i\omega z/v} \left[\frac{i}{\gamma} K_0\left(\frac{\omega R}{v\gamma}\right) \hat{\boldsymbol{z}} - K_1\left(\frac{\omega R}{v\gamma}\right) \hat{\boldsymbol{R}} \right] , \tag{3.27}$$

where hats represent unit vectors, ε the dielectric function, $(x, y, z) = (\boldsymbol{R}, z)$ due to the cylindrical symmetry, and under the assumption that the electron does not slow down as it interacts. $K_{0,1}$ here are the modified Bessel functions of the second kind. Polarizabilities, however, have not been expressed generally. The usual starting point is Clausius-Mosotti relationship, giving the polarizability for dipole spacing d as

$$\alpha_j = \frac{3d^3}{4\pi} \frac{\varepsilon - 1}{\varepsilon + 2} . \tag{3.28}$$

Further corrections and the inclusion of the radiative term have led to the lattice dispersion relation [44], which in turn has been corrected as well [45].

While the $3N$ equations can be solved exactly (as far as floating point computation allows) by matrix inversion, it is often more practical to use iterative methods, such as conjugate gradient method, that converge to a solution with a required margin of error much more quickly.

There are many publicly available DDA codes for light scattering, from the original code by Draine and Flatau called "DDSCAT" [45], to openly developed "a-DDA" [46], which is specifically optimized to make use of large computing clusters, hence allowing extremely fine discretization of particles. Electron energy loss simulations using DDA, however, have been sparser, with the Masiello group only in 2012 adapting the DDSCAT v7.1 to simulate a swift electron and calculate energy losses [41]. We improved their original code in our group by adding an arbitrary ambient medium, updating the code base to DDSCAT v7.3 and enabling changes to the direction of the beam, allowing for easier tomographic simulations. All of these codes are implemented with an additional optimization of storing the locations of dipoles as a Fourier transform, hence decreasing the memory required for the calculations to approximately scale linearly with the number of dipoles, as opposed by quadratic scaling if traditional methods are used. The drawback of this approach is that the dipoles have to be arranged on a rectangular lattice, hence increasing the discretization requirement for highly irregular particles. This constraint, however, is not present in Geuquet's DDEELS [47].

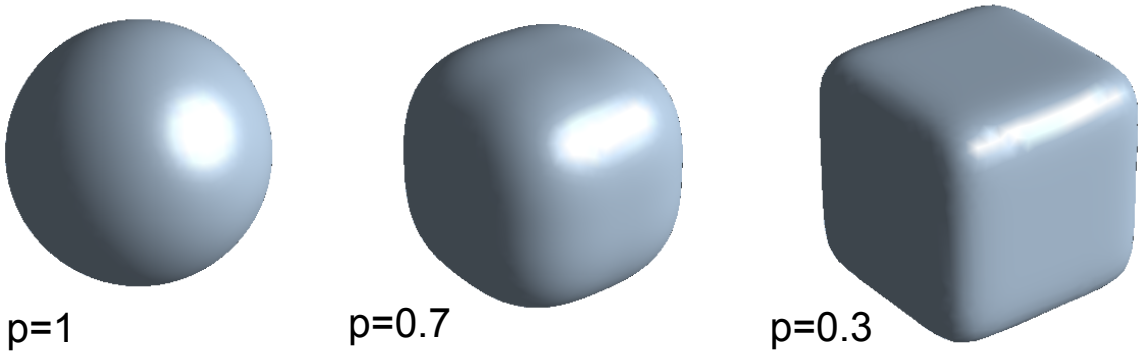


Fig. 3.2 Examples of particles, described using eq. (3.29).

The DDA formulation is advantageous when many dielectric materials have to be simulated or the sample can be accurately described as a collection of dipolar particles (for example dust). DDA also performs well when larger particles are considered (e.g. ca. µm dimensions in EELS). The main drawback of this method is volumetric discretization, leading to a rather high memory requirement, especially if curved geometries are simulated to high accuracy.

3.3 Morphing a cube to a sphere

In Local Surface Plasmon Resonance studies using EELS spectral decomposition is especially important in order to untangle the complex responses. Indeed, many researchers [48–51] study LSPRs not at specific frequencies, but as combinations of eigenspectra, unique to a specific geometry. While machine-learning approaches are able to approximate such decompositions [52, 53], curve-fitting offers a more controlled analysis that directly relates to theory. However, optimization requires a precise mathematical formulation of spectral features in question and even with recent breakthroughs in understanding of LSPRs, some effects are still not fully understood for geometries where no analytical solutions currently exist. In particular, with increasing particle sizes (due to the relativistic effects and the finite speed of light) not all its surface is excited co-instantaneously, resulting in "retardation". As relativistic effects become more prominent, the spectral features become increasingly more asymmetric and higher order resonances in the spectra become visible, hence the usual Lorentzian approximation breaks down. Here I will empirically show that a connection between the spectral features from a rounded cube and a sphere can be used to extract more information from a subset of peaks in a cube spectrum, enabling a more robust and quantitative analysis.

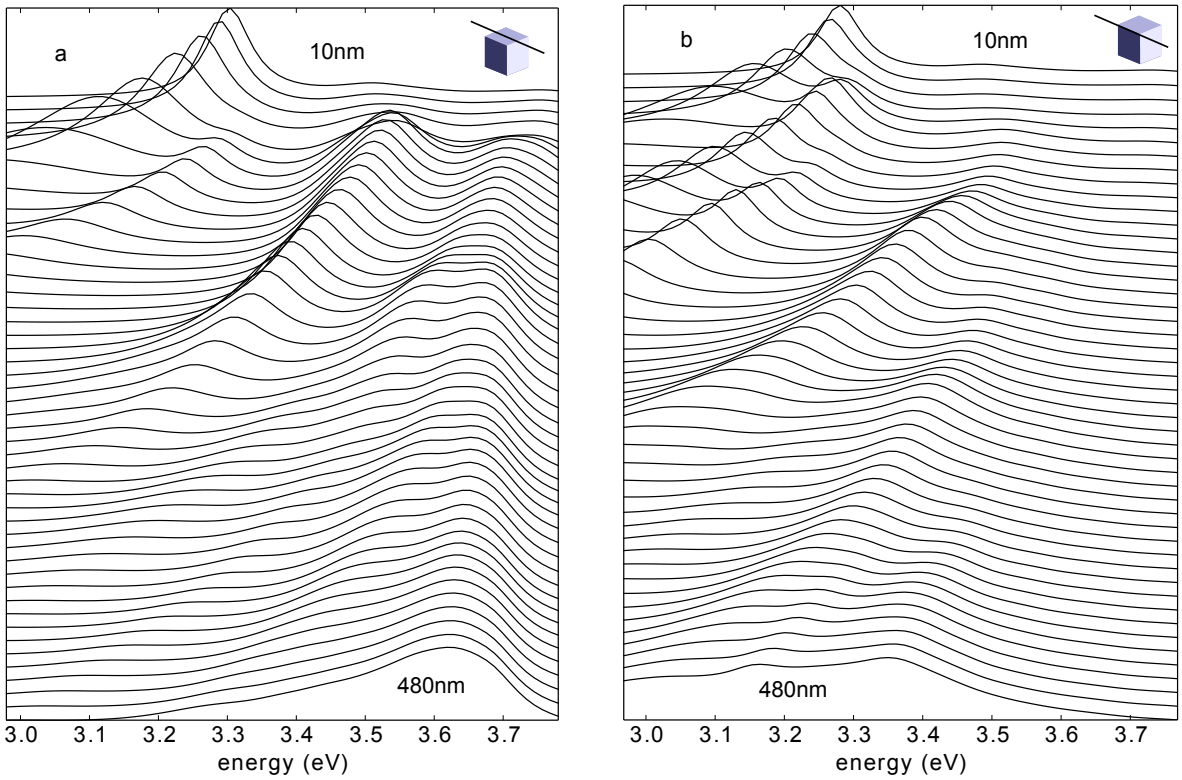


Fig. 3.3 Normalised spectra of rounded cubes with edge lengths ranging from 10 nm (top) to 480 nm (bottom) every 10 nm. Electron passes parallel to the edge of the cube in both figures, with trajectories above the middle of a face and above two corners in (a) and (b) respectively.

The resonances were studied by simulating electron energy loss spectra for rounded silver nanocubes using the discrete dipole approximation code eDDA [41]. The incoming beam energy was 300 keV in all cases. The cubes were modelled using the superellipsoid function:

$$\left|\frac{x}{r}\right|^{2/p} + \left|\frac{y}{r}\right|^{2/p} + \left|\frac{z}{r}\right|^{2/p} = 1\,, \tag{3.29}$$

where rounding parameter values $p = 1$ and $p = 0$ give a sphere and a perfect cube respectively (see Fig. 3.2). For realistic shapes, where the perfect nanocube is extremely difficult if not impossible to manufacture, the smallest value of $p = 0.255$ was used.

To first examine how spectra change with increasing relativistic effects, EELS responses of cubes with edge lengths ranging from 10 nm (top) to 480 nm (bottom), as shown in Fig. 3.3, were calculated. The two shown trajectories are (a) with the electron passing over the middle of a face and parallel to an edge, and (b) parallel and over the edge. In all cases the trajectory was 10 nm above the closest points of the particle. It was shown recently [52] that the strongest cube LSPRs can be considered as three different modes

located at corners, edges and faces respectively, with multiple symmetry-constrained orders in each. According to that interpretation, the lowest energy (∼3.3 eV) features in the top spectra in the figure are the cube corner modes, with higher (up to third) order features becoming visible at similar energies as the cube size (and hence retardation effects) increase down the plot. The third order (octupolar) cube corner mode is forbidden by symmetry of the electron trajectory in Fig. 3.3a and is not visible. Peaks emerging at around 3.5 eV and then redshifting with increasing particle sizes, are associated with cube edge modes. Finally, spectral features just below 3.7 eV and then redshifting were shown to be located on the faces of the cube. All spectra in the figure are normalized, so changing relative strengths of different features are also affected by the time the swift electron spent in the vicinity of the particular plasmon. For example, as the cube dimensions increase (towards the bottom spectra) in Fig. 3.3a, the time for the electron to pass over the perpendicular edge plasmons gets increasingly smaller, when compared to the time required to pass over the face plasmon, hence the "edge" peak gets less pronounced, whilst the "face" mode becomes dominant. It is important to note that for edge and face modes, higher than third orders are present but difficult to distinguish, whereas the corner mode only has the three orders, which are relatively easy to visually identify if relativistic effects are prominent.

Given the parametric form of particle shape and the ability to calculate spectra for arbitrary geometries, further EELS simulations changing the rounding parameter were performed. Fig. 3.4 shows results where the previously considered rounded cube (top) was smoothly changed to to a perfect sphere (bottom) with particles having 100 nm edge or equivalent. With the sphere (bottom) spectra being described by eq. (3.16), it suggests that the dipolar ($l = 1$), quadrupolar ($l = 2$) and octupolar ($l = 3$) orders of the sphere red-shift with decreasing rounding parameter to become dipolar, quadrupolar and octupolar cube corner modes (the dipolar cube peak is out of the energy range of the plot). As the cube corner modes do not have an analytical solution and represent an important part of a nanocube EELS response, the apparent relation with the sphere solution gives a handle to better deal with those spectral features.

To further explore the correspondence between cube and sphere modes, the first order cube corner mode peak shape with increasing particle size was investigated and compared to the evolution of the first order sphere mode EELS peak from eq. (3.16). Both features were analysed using a Lorentzian fit to extract peak positions and FWHM values. The change in FWHM and relative energy shift are plotted in Fig. 3.5. The lowest energy peaks for both a rounded cube and a sphere are shown to be similar in shape up until higher order cube peaks stop overlapping at around 50 nm particle size,

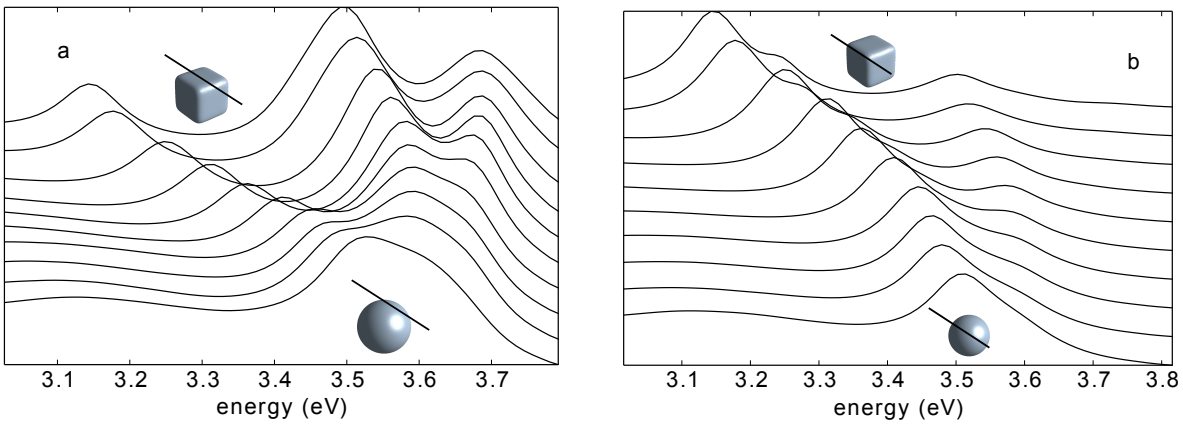


Fig. 3.4 Spectra of particles with rounding parameters changed from $p = 0.255$ (top) to $p = 1$ (bottom) with electron passing (a) over the middle of the face parallel to an edge and (b) over the edge and parallel to it. All particles have the maximum dimension in the electron trajectory direction of 100 nm. The two spectra for perfect spheres are different due to different impact parameter of the beam in each case.

and the Lorentzian fit becomes worse. The procedure also displays the already mentioned fundamental problem - for larger particles the spectral features become increasingly asymmetric and require a different functional form to model them.

3.4 Fitting sphere solutions to cube simulations

The strong similarities between the low order sphere and cube corner modes suggest that it is possible to use the sphere spectral lineshapes (eq. (3.16)) to model the cube corner modes that cannot be accurately described by Lorentzian functions. Fig. 3.6 (a,b) show the results of fitting simulated EEL spectra for two trajectories for 100 nm silver rounded cubes to dipolar, quadrupolar and octupolar sphere modes, calculated using fully retarded Mie theory, plus two Lorentzian function for the edge and face modes. The free parameters for the sphere modes were a global redshift and sphere radius. In addition, each mode has an independent scaling (area) parameter. It is important to note that the sphere lineshape fit is more constrained than just a collection of three Lorentzians, as the first (dipolar) mode has 3 free parameters (just like a Lorentzian would), but every subsequent sphere mode adds only a single free parameter (area). The energy difference and widths of different sphere modes were completely described by the radius of the "effective" sphere. Fig. 3.6 (c) shows an equivalent model for a 10 nm rounded cube. In Fig. 3.6 (d) the estimated sphere diameter parameters as a function of cube edge lengths in the 10 nm to 100 nm size range is plotted.

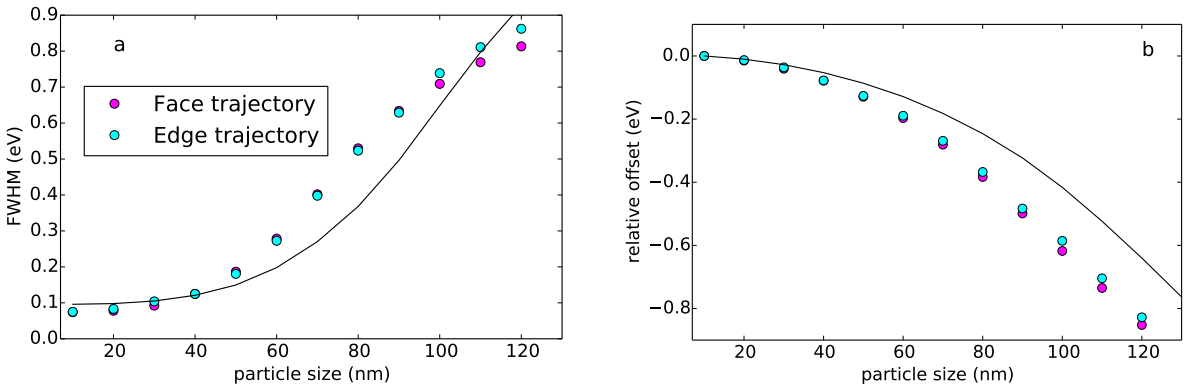


Fig. 3.5 (a) FWHM and (b) change in energy with increasing particle size for the first corner mode of the cube (circles) and the dipole mode of the sphere (line). The values were calculated using a Lorentzian fit. As the retardation effects become increasingly more important, both spectral features become asymmetric and higher order modes stop overlapping, leading to poorer Lorentzian fit.

As can be seen from the Fig. 3.6, the cube corner modes can be fitted rather well using a truncated spectrum of a similar-sized sphere. This enables us to both extract quantitative information about the shape of the cube from its spectrum (i.e. an approximate edge length) and better separate out other, for example edge, cube modes that start to overlap when significantly redshifted and therefore difficult to analyse.

Whilst the presented work is purely empirical, there exists a link between the spherical and cubic plasmon modes from theoretical considerations. García de Abajo showed in 1999 that the EEL response from a sphere can be described using sums of the spherical harmonics [38]. In 2012, Boudarham and Kociak derived the local density of states, a quantity that is closely related to the plasmonic response, and EEL probability descriptions in terms of the "geometric modes" that are described by the shape of the considered particle alone [43]. We speculate that the rounded cube, considered in our work, should have a geometric mode expansion describing the simulated EELS, which is closely related (via the symmetry) to that of a perfect cube. Furthermore, in 1965 Altmann and Cracknell showed that the cubic harmonics can be expressed in terms of the spherical harmonics [54], thus providing us with the link between spherical and rounded cube plasmon modes.

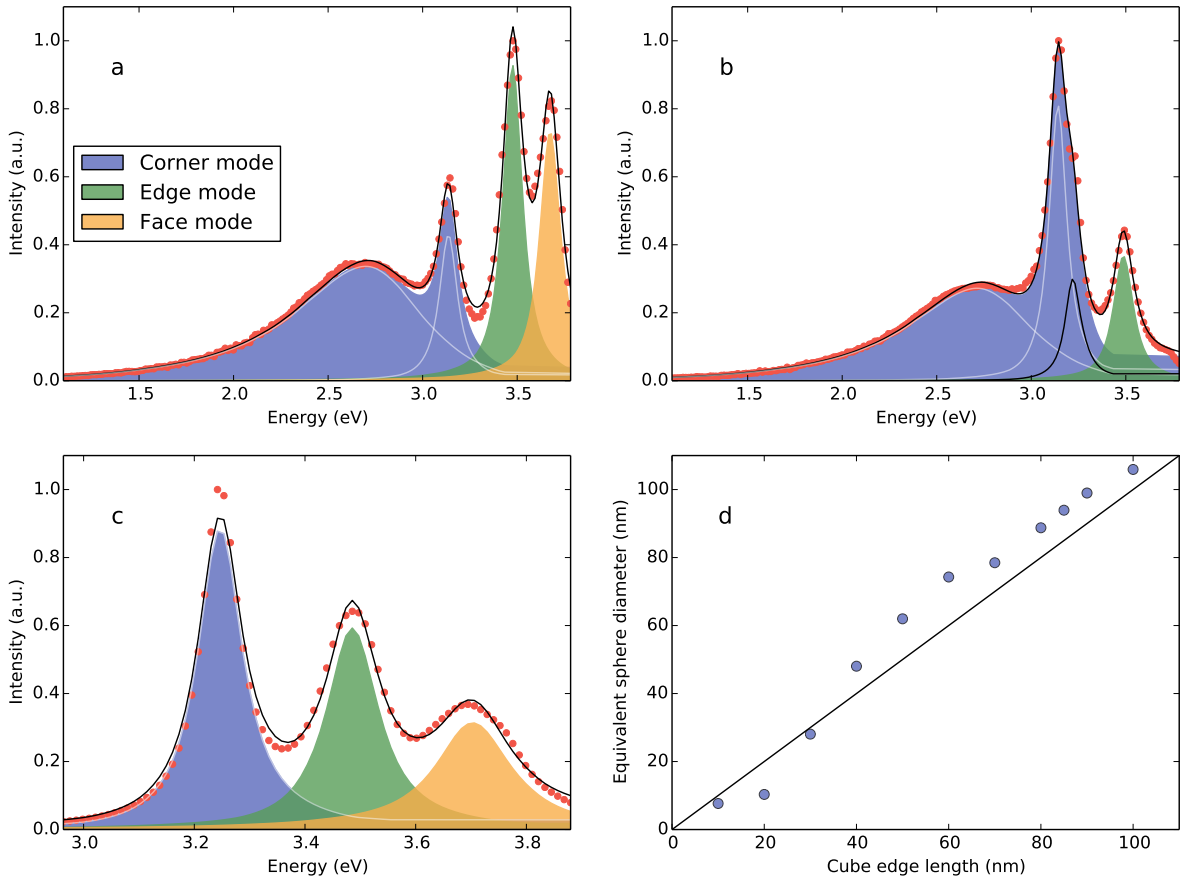


Fig. 3.6 Fitted EELS spectra of (a,b) 100nm and (c) 10nm edge length rounded silver cubes with the electron beam 10nm above the surface. Separate fitting components have been highlighted. In (a,b) the particle is sufficiently large to see retardation effects: the lowest energy component can be approximated only with at least two (three) spherical modes when the electron passes above the face (edge) of the cube. The octupole component (highlighted in b) is forbidden by symmetry for the face trajectory. In (d) "equivalent sphere diameter" versus the rounded cube edge length is plotted. A straight line with gradient 1 is shown to guide the eye.

Chapter 4

Large Multi-dimensional Data Analysis

With the best electron microscopes already pushing the spatial resolution beyond 1 Å, the analytical electron microscopy (EM) field is beginning to aim for not only increased accuracy and sensitivity, but also at probing more dimensions over larger fields of view. This will inevitably mean two things. Firstly, measuring micron-sized regions with sub-nanometer resolution will undoubtedly reveal new and exciting science, and help apply the full power of EM in many fields that were previously limited to measuring relatively small sample areas. The access to fine detail over large areas will offer previously inaccessible insights based, for example, on much improved statistical analyses of the specimen. Secondly, the size of the datasets will grow beyond what many conventional data analysis and tools are currently capable of handling.

In the following section I will briefly introduce two Analytical Imaging techniques that offer the most promise for large multi-dimensional data, and then explain them in more detail in sections 4.1.1 and 4.1.2. Section 4.2 will discuss the most common current and yet-to-be-encountered issues with these methods.

4.1 Analysis techniques

Having been used throughout the sciences [55–58], model fitting (described in more detail in Section 4.1.1) is arguably one of the most versatile analysis methods to date. Its huge success and analytical power can be credited to its ability to define a mathematical expression and then find the parameter values (often the constants in the expression) that match it to the data. This flexibility is however a two-sided coin – the method greatly benefits from, and relies on, previous knowledge, for example in the form of a theorem

describing the phenomenon. Fitting is also able to provide a direct link to the data, enabling comparison of the two, and extracting direct physical results. However, if no assumptions are made about the data or its origins, model fitting becomes significantly less useful. In the end, both the results and their interpretability relies heavily on the "set-up" of fitting analysis.

On the flip side, in some cases it is advantageous to not assume any knowledge of the data and avoid bias as much as possible. While the relevant mathematical formalism has been known for over a century [59], machine learning (ML) methods for EM data analysis have only been used extensively over the last decade. The reasons for this are twofold: first, relatively powerful and high-memory computers (historically speaking) are required to perform the calculations for typical microscopy datasets at reasonable times. Secondly, the general rule is that the more data the algorithm has to learn from, the better the end results are. However, before the modern computer era both performing sufficiently data-rich experiments and storing the said data were significant challenges. While machine learning is introduced in more detail in section 4.1.2, in essence the commonly used methods learn a model (components) by looking at all the data, and then calculate their respective weights. This allows extracting significant information that is simply not available from any one individual measurement, making it inaccessible to model fitting.

4.1.1 Model Fitting

The model fitting analysis requires making a series of decisions that highly influence the end results. This requires inserting previous knowledge into the process, allowing for a path for human bias. Also, when the datasets become very large, the final result often consists of a set of discrete smaller fits that are not ensured to be consistent with each other. This requires reviewing and rechecking the validity of these results, which is rarely done either automatically or manually. Nevertheless, a brief overview of the required steps will be given in the following sections.

Defining the model

When fitting, the first and often most important step is assigning a mathematical description to the data. It sets out what will be measured and thus should be chosen carefully. Fortunately, many different approaches can be taken, ranging from highly theory-based to data-based. A list of examples follows.

A full simulation of the system could be performed for every optimization step, in the end resulting in a fully coherent simulation model of the physical system. Such an approach, however, is often time-prohibitive due to the computational costs for most systems in question. If the phenomenon has a theoretical solution in the form of an equation [38], it can be used to recover the physical parameters of the measured system, allowing a comparison of physical and theoretical systems directly. Section 3.4 shows an example how this can be achieved even when the assumed and real systems do not match exactly, and are only used as approximations. Conversely, often the quantity of interest can be seen directly in the data, and the model is used only to measure it, for example the area under the curve. In such cases the functional form of the model does not matter, as long as it matches the data shape well [60, 61]. Finally, sometimes it is necessary to quantify the relative change of a measurement across the dataset. In such cases the full model consists of a datum and functions that perturb or modify it, as will be shown in Section 9.1.1.

Generally, the least complex model[1] that measures the required quantities should be picked to avoid over-fitting [60, 62]. Here model "complexity" should take into account its computational cost, interpretability, and how difficult it is to optimize.

The optimized cost function

Once a suitable model (or a family thereof) is decided, an optimization cost function has to be defined. The cost function is the measure that enables calculating how well the model represents the data. For most optimizers it should be a smooth, preferably analytically differentiable function that represents a best fit at its extrema (minima or maxima).

Traditionally the most common and general cost function is the sum of least squares of deviations of the model from the data [60, 66]:

$$S(\boldsymbol{\beta}) = \sum_{i=1}^{m} \left[y_i - f(x_i, \boldsymbol{\beta}) \right]^2 \tag{4.1}$$

where $f(x, \boldsymbol{\beta})$ and $\boldsymbol{\beta}$ are the model and the parameter value vector respectively, and the data is described by (x, y). This so-called least-squares cost function promotes models that match the data values as closely as possible, and is a suitable choice for many optimization problems.

[1] As determined by one of Pearson's χ^2 test [63], Akaike's Information Criterion (AIC) [64] or Bayesian Information Criterion (BIC) [65]. Discussed in more detail in section 5.2.3.

If the model can be interpreted as a probability distribution, a different cost function is often preferred. In such cases the goal is to find the probability distribution of the underlying mechanism that produced the data, and not just match the data itself. The maximum likelihood [60, 67] is a better-suited measure to optimize such models. Putting it simply, the maximum likelihood represents how likely the (probabilistic) model will result in the data distribution that was measured, which may differ from the solution given by optimizing the least-squares function.

The optimizer choice

Depending on the cost function being linear or non-linear with respect to the model parameters, a linear or non-linear optimizer has to be used. Linear optimization presents a significantly simpler problem, since the cost function by definition has only one minimum, which in some cases allows calculating a closed-form solution analytically [68]. The largest drawback is the linearity requirement, which greatly reduces the number of problems that can be tackled. As a result, even though linear optimization is routinely used for very simple problems, it is not the subject of this work as thus will not be discussed further.

Non-linear optimizers, conversely, are able to deal with any models and cost functions to find a good, but not necessarily best, match to the data. The effect is often named the "local" or "false" minima problem, and is the reason why care should be taken when picking the starting parameter values. All non-linear optimizers are iterative, meaning they look for the solution by searching the parameter space in the neighbourhood of the current guess. Consequently, the optimizer is able to get stuck in one of the local minima, failing to find the best fit. Due to the nature of this search process, the starting guess is often as important as a correct model for the data. A subset of non-linear optimizers that perform the so-called "global optimization" are able to go around this problem at the cost of significantly increased computational load [69].

A number of factors should be considered when deciding upon the choice of the optimizer. The ability to constrain the allowed parameter space, while requiring to know the viable parameter bounds beforehand, often eases the global solution search [66, 70–72]. Some optimizers also support assigning weights to the data, allowing better fits to be found if some data quality estimation is available (for example the variance of the data) [73]. Finally, due to the wide use of common cost functions (the best example being the least-squares), there are optimizers that perform these calculations faster and more robustly at the cost of not being general [66, 74].

4.1.2 Machine learning

Machine learning is a subfield of computer science that has recently received much attention from both the academic community [75] and industry [76]. In practical terms, ML attempts to solve tasks where hand-crafting a suitable function or model is unfeasibly complex, for example defining all rules how to distinguish a chair in an image. By letting the computer figure out the necessary generalisations and rules, many previously nearly impossible tasks become approachable. The main machine learning program characteristics have been succinctly defined by Tom Mitchell [77] as:

> "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E."

Various branches of ML have been successfully used to solve a host of different problems, from drug discovery [78] to creating new pieces of art [79, 80]. Nevertheless, while the field is indeed exceptionally diverse, it can be split into several categories based on what kind of information is available for the algorithm to learn from:

- **Supervised learning** considers algorithms that can learn from examples with known desired outputs. By generalising the knowledge encoded in the given examples, the final model is then able to predict the outcome for previously unseen input parameters. Best known supervised learning algorithms include spam filters [81] and handwritten digit recognition [82].

- **Unsupervised learning** considers ML algorithms where the desired output is either not known to begin with, or just not available when learning. The goal is to find any hidden structure in the raw supplied data. Examples of unsupervised learning include feature learning such as Independent Component Analysis and Non-negative Matrix Factorization (both discussed in more detail later) or anomaly detection [83, 84].

- **Reinforcement learning** algorithms are different to the two previous classes because instead of directly accessing the data, they continuously interact with a system to achieve a long-term goal. Usually such problems offer no ways to estimate the correctness of moves while the interaction occurs, and the success can only be determined at the end, for example when playing games [76, 85] or even recognising images [86].

- **Semi-supervised learning** is the area between supervised and unsupervised learning. It considers algorithms where just a small subset of total available data is labeled with known correct answers. Such datasets are generally common: for example millions of hours of audio recordings are easily available, but labeling each piece requires significant human effort. Semi-supervised learning algorithms are used, amongst other examples, for image recognition [87] and text classification [81]. SAMFire, presented in chapter 5, can be considered to be a semi-supervised machine learning algorithm.

Alternatively, ML methods can be divided into groups based on the expected outcome [60]. Classification algorithms divide the inputs into two or more known classes. These are normally tackled in a supervised way. If the classes are not known beforehand, the process is done in an unsupervised manner, and is called clustering. Regression algorithms, on the other hand, output a continuous function instead of (known or unknown) discrete classes. Finally, dimensionality reduction algorithms simplify the given data by mapping it into a lower-dimensional space.

With such a diverse field it is useful to concentrate on ML branches that are widely used for electron microscopy data, in particular regression and dimensionality reduction. The former, in the simplest form of curve fitting, has already been described in the previous section. The latter will be discussed as a combination of data compression and mixed signal unmixing ("blind source separation").

Machine learning for EM data

The goal of dimensionality reduction procedures is to give means to reduce the rank of the tensor representing the data: $\boldsymbol{X}^{[n]} \approx \boldsymbol{X'}^{[m]}$, where $\boldsymbol{X}$ and $\boldsymbol{X'}$ are the original and compressed tensors of ranks n and m respectively, with $m < n$. Principal Component Analysis (PCA), the most often used dimensionality reduction method in EM, does this by transforming the data tensor in such a way that it is possible to easily discard the irrelevant information. With the original idea published in 1901 by Pearson [59], according to Tipping and Bishop [88] the most common definition of PCA is as follows:

$$\boldsymbol{T} = \boldsymbol{X}\boldsymbol{W} \tag{4.2}$$

with $\boldsymbol{X}$ being the original data of n measurements each with f features, and $\boldsymbol{W}$ an orthogonal linear transformation. $\boldsymbol{W}$ is picked such that the greatest variance of $\boldsymbol{T}$ lies on the first coordinate (the first principal component), the second greatest on the second, and so on. Even though $\boldsymbol{T}$ is still of identical rank as the original $\boldsymbol{X}$, $\boldsymbol{W}$ can be

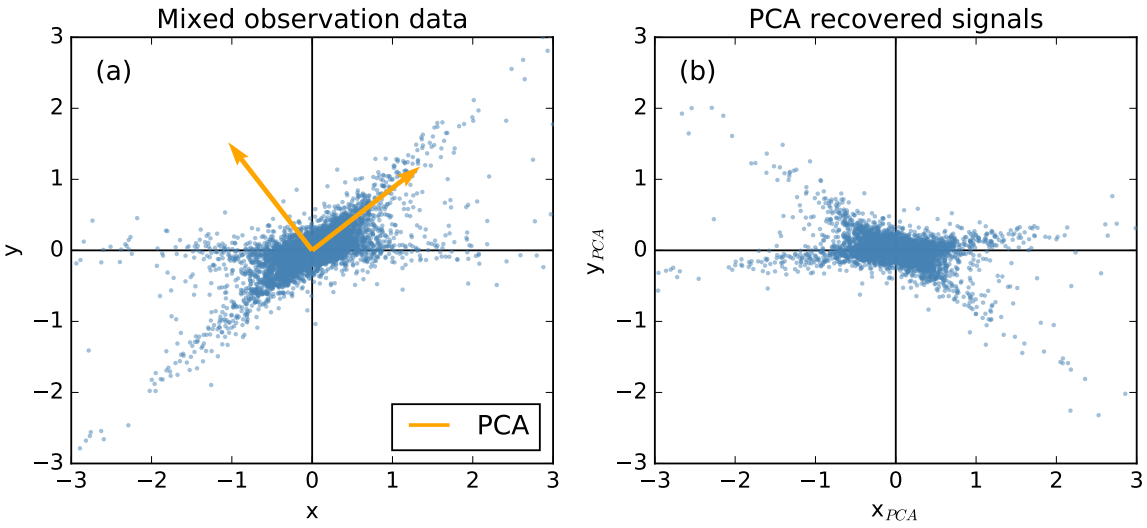


Fig. 4.1 Visualization of Principal Component Analysis (PCA) decomposition. In (a) two mixed distributions are shown in blue, orange arrows indicating PCA-computed component directions. (b) shows the PCA-unmixed values. Due to the components always being orthogonal, the unmixing is not complete.

truncated to some number m with $m < n$, leaving only m most significant components, and reducing the dimensionality of $\boldsymbol{T}$.

The clear statistical interpretability of each component significance is one of the main appeals of PCA, allowing components that do not contain statistically significant information, such as noise, to be discarded. In fact, due to the requirement that principal components (the transformation axes) are orthogonal, often individual components are not physically meaningful, and de-noising [89] is the main use of PCA in EM. Fig. 4.1 shows an example PCA decomposition for mixed two-dimensional observations. While the first PCA component direction can be seen to roughly correspond to one of the "clouds", the second component is constrained to be orthogonal and hence still corresponds to a mixture of sources.

The second reason PCA is widely used concerns the computational effort to calculate the transformation. In particular, Singular Value Decomposition (SVD), a highly optimized [90, 91] matrix factorization method used in a wide array of fields, provides a very useful way to compute PCA [92]. The SVD theorem states that for a real or complex $n \times f$ matrix $\boldsymbol{X}$ there exists a factorization such that

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{W}^* \tag{4.3}$$

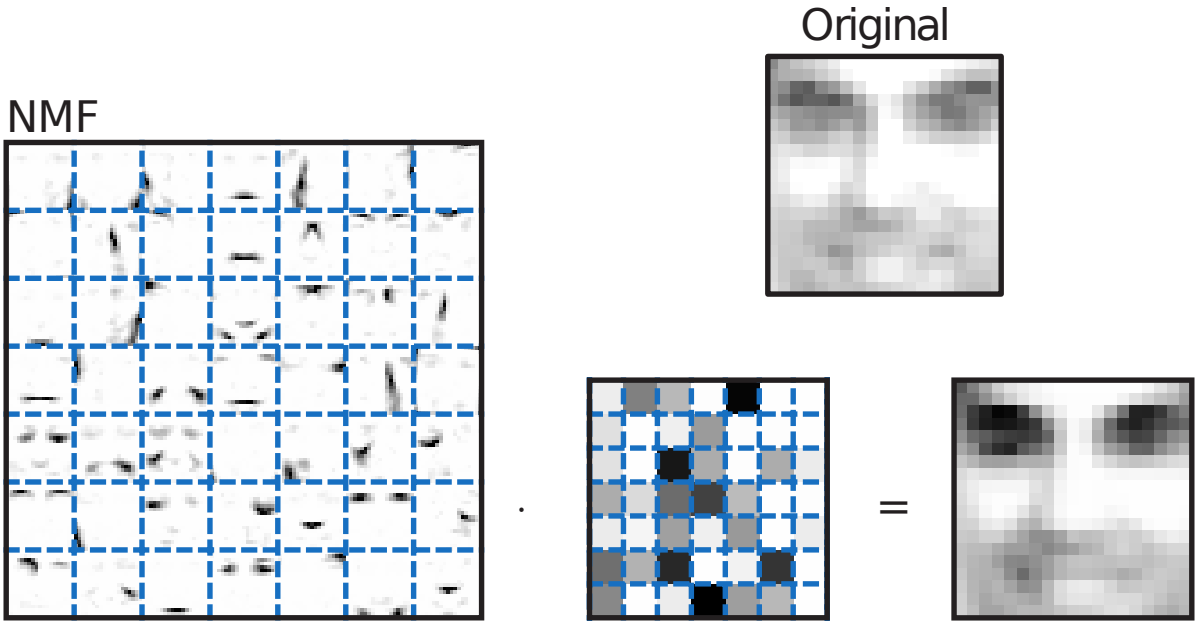


Fig. 4.2 NMF learns a parts-based representation of faces from the original dataset. A particular face instance, shown at top right, is approximately represented as a linear superposition of basis images. The basis image matrix is shown on the left, and their coefficients in the middle. Adapted from [93].

where $\boldsymbol{U}$ is $n \times n$ unitary matrix, $\boldsymbol{\Sigma}$ is a diagonal $n \times f$ matrix with non-negative real numbers on the diagonal, and $\boldsymbol{W}^*$ is a $f \times f$ unitary matrix. By substituting eq. (4.3) into eq. (4.2) we get that $\boldsymbol{T} = \boldsymbol{U}\boldsymbol{\Sigma}$, and hence efficient SVD algorithms allow fast PCA estimation.

Non-negative Matrix Factorization (NMF) is an often-used ML algorithm that does both compression and unmixing of the data at the same time [93, 94], Fig. 4.2. It is defined such that the original matrix $\boldsymbol{X}$ can be approximated by a product of two non-negative matrices $\boldsymbol{W}$ and $\boldsymbol{H}$, each of possibly significantly lower dimensions. Calculating the factor matrices can be done in a number of ways by using different cost functions when measuring how well the product represents the original matrix. Since the problem is significantly under-determined, usually some other constraints (in addition to non-negativity) are imposed on the factor matrices, for example, sparseness. One major appeal of using NMF on EM data comes from the requirement that the factor matrices are non-negative, which is not present for PCA. However, NMF is usually significantly more expensive to compute and requires guessing (or otherwise estimating) the number of components to keep for subjectively good results.

Independent Component Analysis (ICA) is arguably the most often used pure "blind source separation" algorithm for EM data. As mentioned before, the algorithm aims

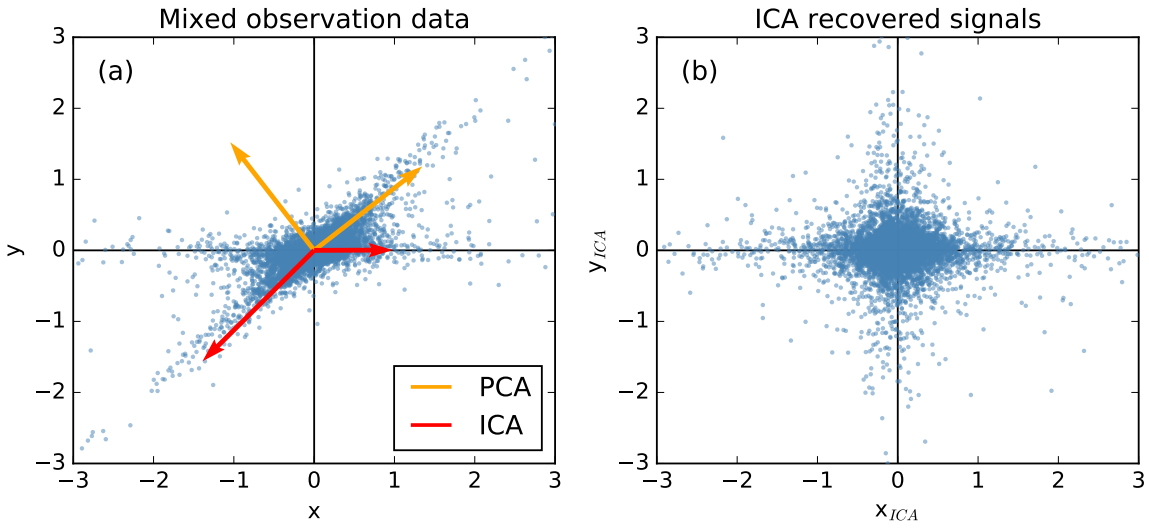


Fig. 4.3 Visualization of Independent Component Analysis (ICA) decomposition. In (a) the mixed data is shown in blue, while orange and red arrows indicate PCA and ICA component directions respectively. (b) shows the ICA-unmixed data by projecting the raw values on the estimated component directions

to separate the dataset of mixed signals into its individual, statistically independent contributions [95]. The two ways that the statistical independence has been defined for ICA is (i) the minimum of mutual information (defined as the amount of information that is obtained about some random variable by measuring a different one [96, 97]) and (ii) non-Gaussianity. The second criterion comes from the Central Limit Theorem, which states that under certain, often seen, conditions, when independent random variables are added, their sum tends towards a normal (Gaussian) distribution even if the variables themselves follow other distributions [98].

The most common mathematical form of ICA for noise-free data is identical to that of PCA, but with different constraints of $\boldsymbol{W}$ – it does not have to be orthogonal, and is instead calculated by minimizing the mixing according to one of the measures. Fig. 4.3 shows an example ICA unmixing. The data is shown in blue, while orange and red arrows correspond to PCA and ICA component directions respectively. Once the data is projected on the ICA components, the two sources are unmixed in Fig. 4.3(b).

While there are ICA algorithms that are able to deal with noisy data, the problem offers few shortcuts, and thus is quite difficult to solve. To remedy this and allow more practical use of the ICA algorithms, the data is usually pre-treated using PCA and truncated to only include the mixed orthogonal significant components. Since modern

implementations of SVD (and hence PCA) are exceptionally fast, it has become the standard procedure in EM data blind source separation analysis.

4.2 Common Issues

Many scientific fields have already experienced the so-called "data explosion" in the past decade [99], which both accelerated their growth and increased the importance of data analysis. Electron microscopy, currently only at the beginning of the phenomenon, is in a prime position to utilize tools that have emerged from other analytical imaging communities, and also learn from their mistakes.

This section will briefly describe two issues the EM field has already encountered or will encounter when performing data analysis, with proposed solutions.

4.2.1 Opening and manipulating the data

The first issue that the data explosion has already caused and that will likely become more severe, is not being able to open the data in the usual way on a personal computer. This limitation comes from the architecture of both most commercially and freely available software packages. Upon the instructions to open a file from the hard disk, the software attempts to decompress and lay out all the information of the dataset in the available computer memory [100, 101]. This is the best approach for performance if the datasets are just a small fraction of the available memory, as they have been for the past decade. However, as the data sizes grow beyond the available computer memory, it quickly becomes very limiting. It is not difficult to imagine the frustration of a scientist who is not able to access results just because there are too many of them! Moreover, very few data analysis approaches are able to perform in-place[2].

There are different ways to avoid or at least delay these problems. The obvious one involves using high-memory dedicated supercomputers. Its significant disadvantage is the cost of such facilities, which may serve as a detriment to performing high data volume experiments. An alternative way to delay this hardware limitation involves compressing the data while performing the experiments, for example with PCA-like algorithms [102–104]. This would significantly reduce the data volume while simultaneously de-noising it, albeit being subject to compression artefacts that may hide interesting parts of the data. Nevertheless, if a particular algorithm is able to extract the required features truthfully, such an approach is worth considering.

[2]Not requiring significantly more memory than that of the data to perform the computations

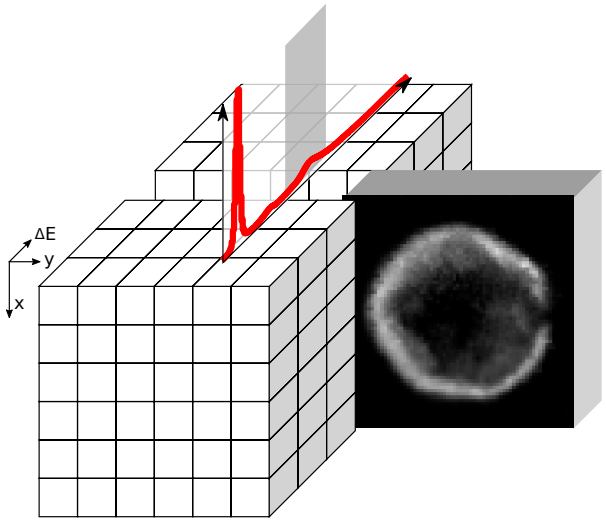


Fig. 4.4 Visualization of a STEM-EELS datacube. Adapted from [107]

An altogether different solution to the data size problem will be shown in Chapter 6. It involves treating the data as a collection of "chunks" (each of which easily fit in the computer memory) and the concept of "lazy computation", where operations are not performed immediately [105, 106]. Instead, a list of operations is stored, similar to writing down equations to be calculated on a sheet of paper. Only when (and if) a particular result is required, are all the required computations run. If the data loading is treated as one of the lazy operations, this combination allows most conventional data analyses to cease being memory-limited. The main examples where this approach does not work are the machine learning compression algorithms. In these cases, the original algorithm has to be replaced by an "online" version - one, where each row or column of the total data tensor is only visible to the algorithm one by one and without the resources to record them for later re-use.

4.2.2 Starting guess

The second issue for much-larger-than-traditional datasets concerns non-linear fitting. For illustrative purposes I will use examples from electron microscopy, but the described problems and solutions can be applied in other fields that use similar data structures (tensors).

Many large EM datasets are some form of mapping across dimensions, for example STEM EELS [4]. In such experiments the electron beam is focused to a small (often sub-nm) spot, which is then raster-scanned over the surface of the specimen, with a full

electron energy loss spectrum measured at every position of the scan. The result is a three-dimensional data tensor, with two dimensions for space (corresponding directly to the points on specimen) and one for energy, recording all the electron interaction information, as illustrated in Fig. 4.4. In order to analyse such data using curve fitting, each of the spectra has to be fitted independently, so the optimizer is only given access to one spectrum at a time. This presents opportunities and challenges: the order in which the spectra are passed to the optimizer and the starting parameters for each of the fits are left to the implementors of the algorithms. At this point it is worth reminding the reader that the starting guess is of paramount importance to the convergence and correctness of the fit result [58], as was stressed in section 4.1.1.

Most of the currently available non-linear optimization implementations, when given such a multi-dimensional data tensor, aim to run fast by doing as little as possible. The algorithms access the spectra in the order they are stored in memory (i.e. raster-order), and either always use the same starting guess, or re-use the end result of the prior spectrum. This approach implicitly results in starting guesses that are either constant by definition, or highly dependent on the order the spectra are stored on the hard disk, which clearly has no bearing on the data it contains. If the dataset is relatively simple and less sensitive to the starting guesses, these approaches indeed result in the fastest non-parallel way to analyse such data. However, many specimens of interest (and hence datasets) are sufficiently complex that the fast-access algorithm results contain numerous errors. With the data sizes growing rapidly, such errors increase in both number and proportion (due to more significant variations of information across the larger datasets), thus increasing in difficulty for the scientist to correct them.

In Chapter 5 a new Smart Adaptive Multi-dimensional Fitting algorithm “SAMFire” is proposed to significantly increase the robustness of starting guesses and ease such non-linear analysis. Instead of re-using the last result, SAMFire attempts to estimate the best starting guess and its confidence for each spectrum, based on all already-finished fits at the time, learning the starting guesses in a semi-supervised way. It analyses the dataset in the order that maximises the convergence for each subsequent spectrum and has in-built Goodness of Fit (GOF) tests to re-run the failed fits once more information for the starting guess estimate is available.

Chapter 5

Smart Adaptive Multi-dimensional Fitting (SAMFire)

Fitting is one of the most often encountered analysis techniques, especially when spectra are considered. It is able to provide a wealth of information about data if the observed phenomena can be accurately described by a mathematical model. Many analytical imaging fields use similar methods on a regular basis: integral field spectroscopy [108, 109] at the astronomical end, cathodoluminescence, electron energy-loss and other spectroscopies at the nanoscale. However, as shown in Section 4.2.2, most of the currently available non-linear optimization algorithms approach the fitting problems in a non-optimal way partly due to historic reasons. While it did not pose a significant problem just a few years ago, as the size and number of dimensions of a typical datasets increase and the specimens become more complex, better algorithms have to be found.

An illustrative example of the general problem is given in Section 5.1, with the proposed method of solution and its implementation in Sections 5.2 and 5.3 respectively. Finally, synthetic example datasets are analysed in Section 5.4.

5.1 Motivation

As shown in Section 4.1.1, model fitting requires making a number of steps that influence the end results. For the purposes of the example let us assume that the exact model is known, and choose to use the least-squares cost function (eq. (4.1)) and Levenberg-Marquardt [110] optimization algorithm (LMA).

LMA, like almost every other numeric minimization algorithm, is iterative. The search for a solution starts with a vector of initial guesses for parameters ($\boldsymbol{\beta}$), which is replaced in each iteration with a new estimate $\boldsymbol{\beta} + \boldsymbol{\delta}$. The $\boldsymbol{\delta}$ is determined by linearly

approximating the model $f(x_i, \boldsymbol{\beta})$ in the vicinity as a Taylor series:

$$f(x_i, \boldsymbol{\beta} + \boldsymbol{\delta}) \approx f(x_i, \boldsymbol{\beta}) + J_i \boldsymbol{\delta} \tag{5.1}$$

with J_i being the gradient:

$$J_i = \frac{\partial f(x_i, \boldsymbol{\beta})}{\partial \boldsymbol{\beta}} \tag{5.2}$$

By combining this approximation with eq. (4.1) we can write the cost function $S(\boldsymbol{\beta})$ as

$$S(\boldsymbol{\beta} + \boldsymbol{\delta}) \approx \sum_{i=1}^{m} \left(y_i - f(x_i, \boldsymbol{\beta}) - J_i \boldsymbol{\delta}\right)^2 \tag{5.3}$$

or, in vector notation and expanded for clarity

$$\begin{aligned} S(\boldsymbol{\beta} + \boldsymbol{\delta}) &\approx \|\boldsymbol{y} - \boldsymbol{f}(\boldsymbol{\beta}) - \boldsymbol{J}\boldsymbol{\delta}\|^2 \\ &= [\boldsymbol{y} - \boldsymbol{f}(\boldsymbol{\beta}) - \boldsymbol{J}\boldsymbol{\delta}]^T [\boldsymbol{y} - \boldsymbol{f}(\boldsymbol{\beta}) - \boldsymbol{J}\boldsymbol{\delta}] \\ &= [\boldsymbol{y} - \boldsymbol{f}(\boldsymbol{\beta})]^T [\boldsymbol{y} - \boldsymbol{f}(\boldsymbol{\beta})] - [\boldsymbol{y} - \boldsymbol{f}(\boldsymbol{\beta})]^T \boldsymbol{J}\boldsymbol{\delta} - (\boldsymbol{J}\boldsymbol{\delta})^T [\boldsymbol{y} - \boldsymbol{f}(\boldsymbol{\beta})] + \boldsymbol{\delta}^T \boldsymbol{J}^T \boldsymbol{J}\boldsymbol{\delta} \\ &= [\boldsymbol{y} - \boldsymbol{f}(\boldsymbol{\beta})]^T [\boldsymbol{y} - \boldsymbol{f}(\boldsymbol{\beta})] - 2[\boldsymbol{y} - \boldsymbol{f}(\boldsymbol{\beta})]^T \boldsymbol{J}\boldsymbol{\delta} + \boldsymbol{\delta}^T \boldsymbol{J}^T \boldsymbol{J}\boldsymbol{\delta}\,, \end{aligned} \tag{5.4}$$

where the two-norm is defined as $\|\boldsymbol{x}\|^2 = x_1^2 + x_2^2 + \ldots + x_n^2$. The derivative of eq. (5.4) with respect to $\boldsymbol{\delta}$ is zero at the minimum value of S, giving

$$\left(\boldsymbol{J}^{\boldsymbol{T}} \boldsymbol{J}\right) \boldsymbol{\delta} = \boldsymbol{J}^{\boldsymbol{T}} \left[\boldsymbol{y} - \boldsymbol{f}(\boldsymbol{\beta})\right] \tag{5.5}$$

where $\boldsymbol{J}$ is the Jacobian matrix, making eq. (5.5) a set of linear equations to be solved for $\boldsymbol{\delta}$. Eq. (5.5) is also known as the Gauss-Newton method for approximation, and is the starting point for LMA [111]. Levenberg's contribution was to replace it by a damped version:

$$\left(\boldsymbol{J}^{\boldsymbol{T}} \boldsymbol{J} + \lambda \boldsymbol{I}\right) \boldsymbol{\delta} = \boldsymbol{J}^{\boldsymbol{T}} \left[\boldsymbol{y} - \boldsymbol{f}(\boldsymbol{\beta})\right] \tag{5.6}$$

with $\boldsymbol{I}$ being the identity matrix. The damping factor λ is adjusted at each iteration to increase or decrease the reduction of the residual S. Marquardt followed by noting that scaling each component of the gradient according to the curvature ($\boldsymbol{J}^{\boldsymbol{T}} \boldsymbol{J}$) avoids slow convergence in small gradient direction. He replaced the identity matrix $\boldsymbol{I}$ with a diagonal matrix, consisting of the diagonal elements of $\boldsymbol{J}^{\boldsymbol{T}} \boldsymbol{J}$, completing what is now known as the Levenberg-Marquardt algorithm [110]:

$$\left(\boldsymbol{J}^{\boldsymbol{T}} \boldsymbol{J} + \lambda \operatorname{diag}\left(\boldsymbol{J}^{\boldsymbol{T}} \boldsymbol{J}\right)\right) \boldsymbol{\delta} = \boldsymbol{J}^{\boldsymbol{T}} \left[\boldsymbol{y} - \boldsymbol{f}(\boldsymbol{\beta})\right] \tag{5.7}$$

In fact, most non-linear optimizers search for optima in a very similar way by traversing the parameter space. The major differences are the criteria for choosing the direction and step size [91]. In Fig. 5.1 a two-dimensional cost function landscape is shown for a model consisting of Gaussian and Lorentzian curves. The goal of the optimizers is, starting from the starting guess (SG1-3), to make a series of steps $\boldsymbol{\delta}$ (shown as grey lines) to find a stationary point, which will hopefully be the global minimum (GM). Optimization paths for a number of different algorithms are shown, with all of them eventually finding just the local minima (LM1-3). Clearly, if a better starting guess was given, any of the optimizers would have been able to find the correct solution. Such behaviour is often called the "local minima problem".

While there are many different optimizers that search the parameter space for the optimum, the practical problem that most researchers face is different: finding the global minimum is trivial if the user starts the search close to the true value, often easy to set by hand. The difficulty arises when a typical dataset consists of thousands or more of such "pixels" (spectra in this example) that span many dimensions and have to be fitted individually—supplying the starting parameters by hand ceases to be viable, and an algorithmic approach has to be found. However, as shown in Section 4.2.2, the currently available algorithms are not suited to tackle such data. While the optimization methods continue to improve [115], little attention is paid to finding the optimal ways to apply these methods to datasets that are not analysed all at once.

The SAMFire (Smart Adaptive Multi-dimensional Fitting) algorithm eases the task of fitting datasets that suffer from the aforementioned local minima problem by automatically generating starting parameters from successfully fitted parts of the data. SAMFire significantly decreases the effort to analyse large datasets by requiring a fit to only a few pixels as "seeds" from which the algorithm automatically learns. Extensive result validation ensures only sufficiently good fits get propagated while the SAMFire operates, further increasing the method robustness. It has already been used for large multi-dimensional fitting problems with highly successful results [116].

5.2 Method

The SAMFire algorithm operates by using structures and patterns in data to predict both the order the fits should be performed in and the likely candidates for the starting parameters, which are then passed to an optimizer. Currently there are two strategies in SAMFire, both described in the subsequent sections. By creating an analysis workflow that consists of a chain of such strategies, each exploring different structures, the algorithm

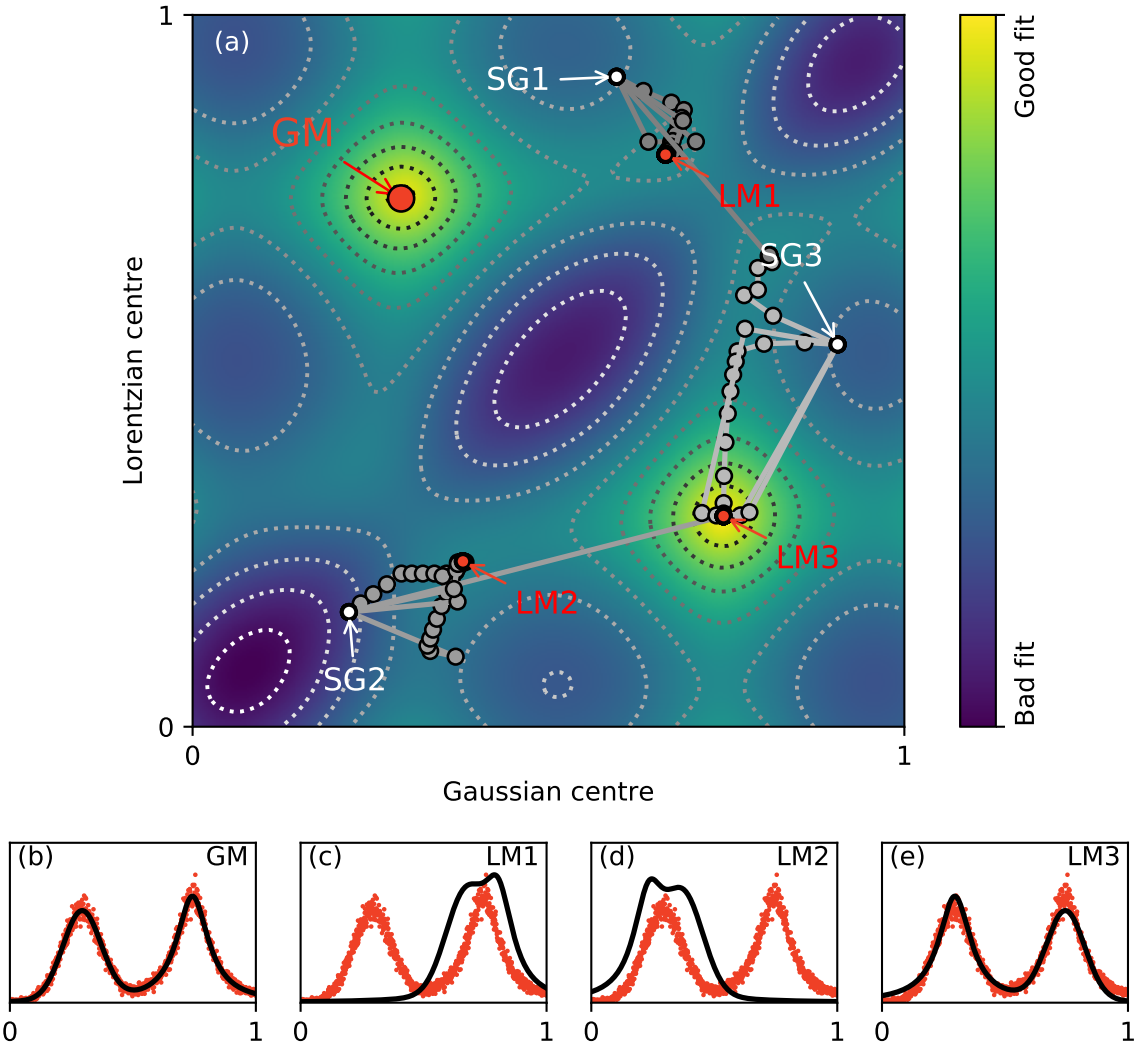


Fig. 5.1 (a) The optimization landscape when fitting the same data and varying just two parameters. Global minimum (GM) and local minima (LM1-3) are marked by red circles. A number of different optimization algorithms (Nelder-Mead Simplex [112], Levenberg-Marquardt's [110], Powell's [113], Polak-Ribiere's [114] and L-BFGS-B [72]) were given three sets of starting guesses (SG1-3). Their convergence towards local minima is shown: each step $\boldsymbol{\delta}$ for each algorithm is shown as a grey line, with resulting $\boldsymbol{\beta}$ marked as grey dots. (b) Global minimum and (c-e) local minima fits, corresponding to the red circle marks in (a). The constant data is shown in red, with the corresponding fits in black.

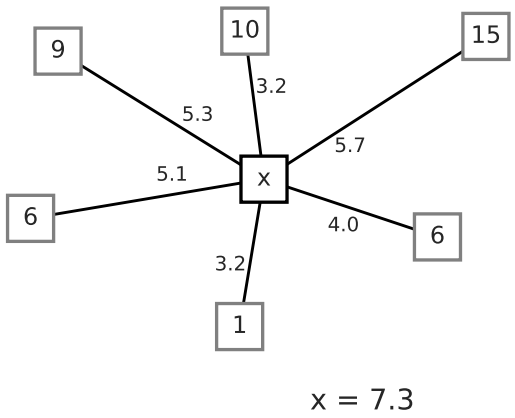


Fig. 5.2 A toy example of local parameter estimation. Fitted pixels are shown as boxes with the parameter values written in them, with distances to the central pixel marked. The inverse of the distance is used as weight when calculating the average.

enables a robust fitting of the datasets that would be extremely time-consuming and difficult—if not impossible—to fit using traditional methods [70]. All fits are checked by a user-optimized goodness of fit (GOF) test, so only valid points are allowed to propagate while SAMFire explores the dataset. In addition, SAMFire is able to use the GOF tests to determine if all components are required to fit a particular pixel and "switch off" the unnecessary ones, allowing for robust fits even with overcomplicated models.

5.2.1 Local strategy

Often multi-dimensional datasets that have to be fitted exhibit local similarity of pixels, which occurs naturally if the data were measured with a finite resolution. Examples include electron microscopy, astronomy, remote sensing and most other analytical imaging techniques. In these cases all fine features below the resolution limit get blurred [117], and the data exhibits a locally smooth landscape in the multi-dimensional space. SAMFire uses this structure to estimate how much fitting information is available about each point from its location. For example, if the fitted values for pixels surrounding a central pixel are already known, in most cases the values for the unknown pixel can be confidently predicted by just averaging with weights that decay with distance from the unknown pixel, for example inversely proportional to the Euclidean distances between pixels, Fig. 5.2. Such weights lessen the smoothing effect of the mean, allowing to follow the distribution

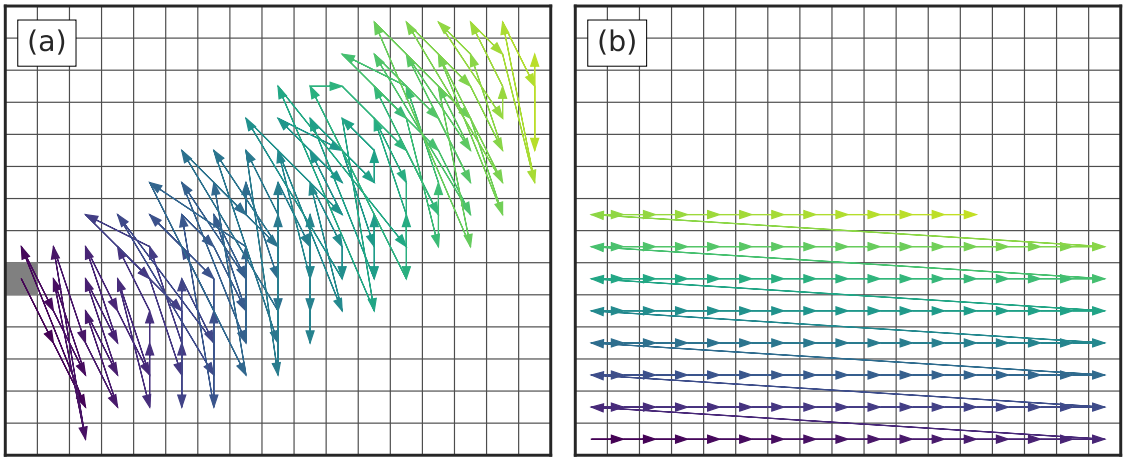


Fig. 5.3 (a) The order SAMFire fits pixels based on previous results. Better fits guide the algorithm to follow the underlying dataset structure, significantly increasing the chance of convergence. (b) conventional pixel fitting order.

more closely. GOF tests can act as a way to estimate the confidence in the fit results, providing additional useful information.

To combine these two criteria for all fitted pixels and use the local average, SAMFire assigns a scalar value that is high near better fits and decreases with distance. Such behaviour is present in natural phenomena such as gravity or electrostatics, hence the GOF measure can be interpreted as a positive theoretical "charge" (or "mass") w, associated with each pixel. By calculating the corresponding "potential" at unfitted pixels, it is possible to express the relevant knowledge by a scalar. The following expression can use any spatial decay function $f(r)$, which in gravity and electrostatics would be $f(r) = 1/r$:

$$\boldsymbol{P} = P_j = \sum_i P_{ij} = \sum_i w_i f(r_{ij}) \quad , \tag{5.8}$$

where i spans all fitted pixels, P_j is the potential at point j, and r_{ij} is the distance between i and j. When analysing real data, parameter smoothing in the spatial domain is often detrimental to the results, thus SAMFire performs better with $f(r)$ decaying faster than previously suggested classical examples. Because the exact form of the function does not matter, in the real implementation $f(r) = e^{-r}$, is used by default.

$\boldsymbol{P}$ can be crudely interpreted as the measure of the useful available information, and once calculated, the optimal pixel order to fit the full dataset can be trivially looked up by always selecting the pixel with the highest current value of P. If $\boldsymbol{P}$ is updated

after every fit, the optimizer can just follow the highest values until the full dataset is processed.

To calculate the average for the starting guess of each new pixel, the algorithm uses the individual $\boldsymbol{P}$ contributions of the fitted pixel as weights:

$$\alpha_j = \frac{\sum_i \alpha_i P_{ij}}{\sum_i P_{ij}} = \frac{\sum_i \alpha_i w_i f(r_{ij})}{\sum_i w_i f(r_{ij})} \quad , \tag{5.9}$$

where j is the pixel of interest, i spans the previously fitted pixels, and α_i is any fitting parameter at i. With the exponential decay function it is often practical to specify a "cut-off" radius r_c to speed up the computations by considering much fewer points. Then only pixels i for $r_{ij} \leq r_c$ are used in estimation. The overall approach provides a way to always pick the pixels about which SAMFire has the most information, enabling not only following the data structure and its suggested "path of the least resistance", but also offering the highest chance of convergence. Fig. 5.3 shows the SAMFire pixel fitting order, where selecting $\arg\max \boldsymbol{P}$[1]allows the algorithm to traverse it in the data-suggested order.

5.2.2 Global strategy

While the previously described approach allows most experimental data to be fitted straightforwardly, if the already fitted neighbouring pixels do not have the required information, the fit propagation stops. This might happen in data with any kind of sharp boundary in the parameter space (domain structure) or if part of a model was deemed unnecessary for the neighbours by GOF tests, but was required for the pixel in question. The parameter distributions with corresponding frequencies for both cases are shown in Fig. 5.4. More generally, if the spatial location of the pixel does not provide the necessary information, SAMFire tackles the problem differently.

The global approach is best used when the parameter values are significantly different with no or very few intermediate values across the neighbourhood, as shown in Fig. 5.4(b). The global strategy exploits such value separation by identifying the local peaks in the histogram (shown as the shaded regions) and then using their most frequent values to form a set of probable starting guesses for each parameter. The algorithm then attempts to fit the pixel in question by trying out all combinations of such starting guesses until a

[1] **Arguments of the maxima** are the points of the domain of some function at which the function values are maximised [118]:

$$\arg\max f(x) := \{x \mid \forall y : f(y) \leq f(x)\} \tag{5.10}$$

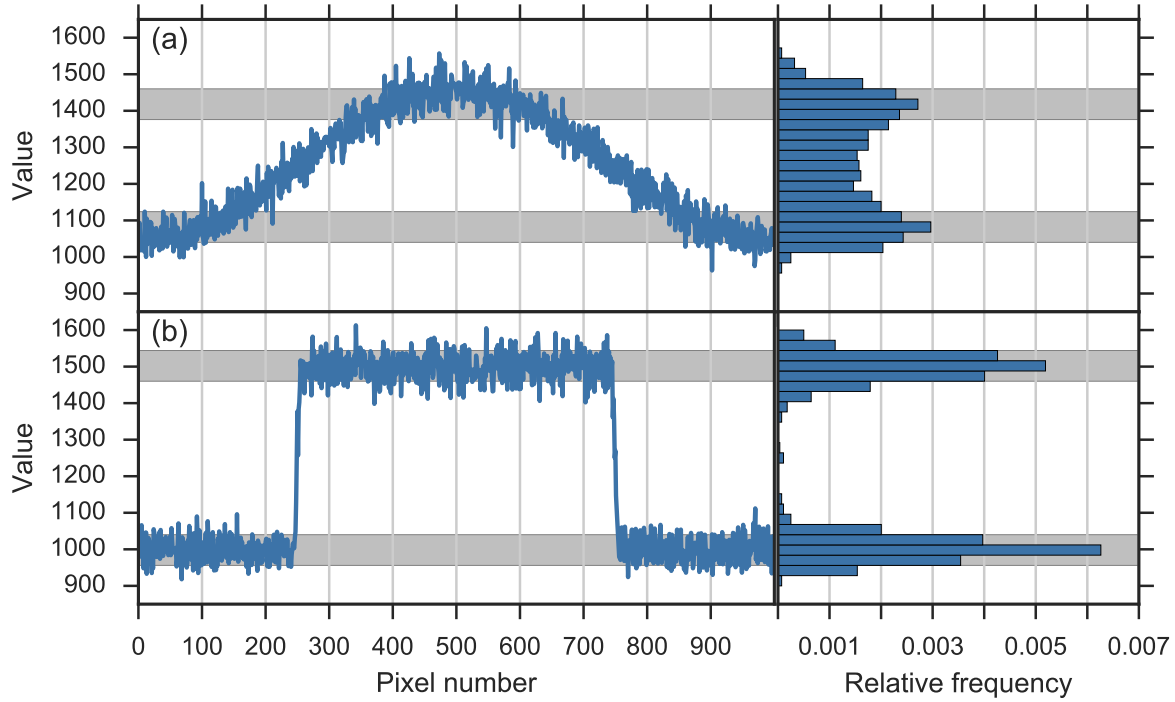


Fig. 5.4 Two example parameter value distributions across pixels with corresponding relative frequencies on the right. The shaded regions are selected by finding two largest local maxima in the relative frequencies (i.e. peak positions on the right), with the width of a histogram bin to either side. In (a) the parameter values are locally smooth, hence the "nearest neighbour" strategy is able to provide accurate estimates for all pixels. Conversely, while the frequency distribution shows clear maxima, a significant proportion of pixels have intermediate values and fall outside the shaded regions, limiting the usefulness of the relative frequency information for unknown value prediction. In (b) the parameter values show a clear domain structure with values suddenly jumping and dropping around pixels no. 250 and 750. The corresponding relative frequency distribution clearly shows two maxima with most of the values falling inside the shaded regions. While in each flat region (domain) the local average is able to provide an adequate estimate, crossing the boundaries between domains is not possible due to the lack of pixels with intermediate values. As a result, a better approach is to guess the unknown value to be inside one of the shaded regions and just discard the unsuitable one using trial and error and GOF tests.

sufficient fit is found, at which point the result is saved and the corresponding histograms are updated. In such case SAMFire does not assume any relation of a particular pixel position in the dataset to its value, so it fits them in random order to ensure a uniform sampling of the tensor.

It is worth noting that while the value separation in the frequency space is quite easily seen even for very smoothly changing values in Fig. 5.4(a), a significant portion of pixels fall outside the shaded regions. As a result, the intermediate-valued pixels would likely fail to converge if given one of the most frequent values.

5.2.3 Robustness

Fit

In order to ensure that a sufficiently good fit is found, each pixel's result has to pass a GOF test, irrespective of the active strategy. The test is set-up by the algorithm user, and should be adapted to match the data and the model. Based on my experience, SAMFire performs best when the GOF test is similar to the quantity being optimized – Pearson's χ^2 test [63] for least-squares family of optimizations, Akaike's Information Criterion (AIC) [64] or Bayesian Information Criterion (BIC) [65] for probability-based optimizations [119].

Local strategy

The local strategy choice is robust to noise and value landscape by design. In particular, there are four important factors when considering the robustness and scaling of the algorithm: noise in parameters, noise in goodness of fit estimates, the rate of change of parameters, and their number. As in many other natural phenomena, noise in parameter α_i is usually distributed normally[2]around the true value even if the underlying data follows other (such as Poisson) distributions. Robustness to parameter noise is due to the mean of normal distribution also being the most often encountered value. Using the local mean of parameters converges towards the true value with increasing number of samples, thus allowing reasonable estimates. When fitting noisy data with an appropriate model, the GOF estimates w_i usually follow some bell-shaped distribution around the mean GOF value. The range of acceptable values around the tails of this distribution is strictly controlled by the user, hence we do not considered it further.

[2]Normal (or Gaussian) distribution is one defined by the probability density

$$f(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}},$$

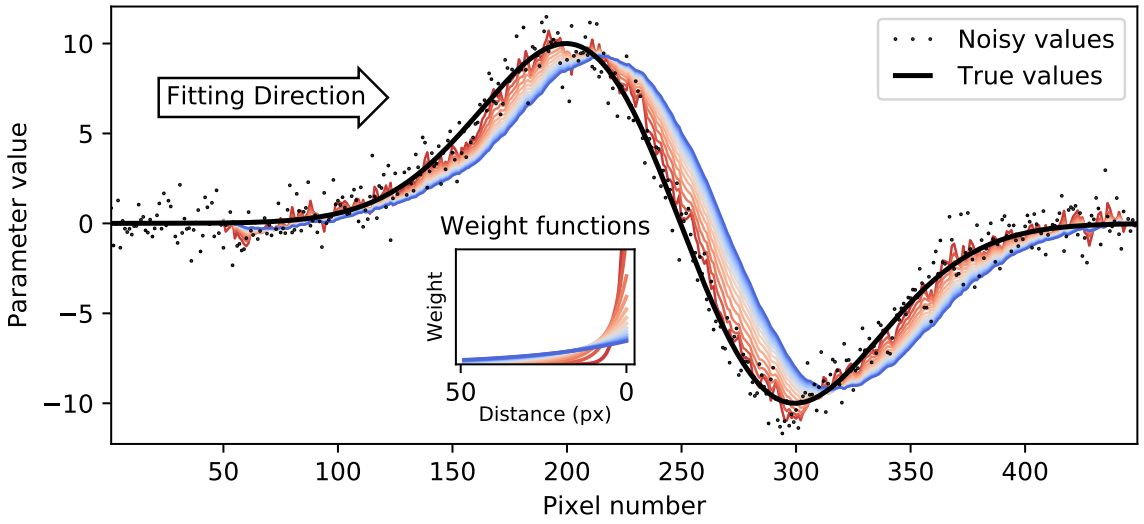


Fig. 5.5 Estimated parameter values when fitting a one-dimensional dataset from left to right. Cut-off distance $r_c = 50$. The estimates (shown in as coloured lines) lag behind the true parameter values more as less weight is assigned to the nearest pixels (blue $f(r)$ curves). The distance weight functions $f(r)$ are shown in the inset. With forward-heavy weight functions (red) the local mean lags less and more closely follows the noise pattern, while more evenly weighted functions (blue) are not as sensitive to noise but lag behind the true values by $\sim r_c/2$.

The algorithm's robustness to the rate of change of parameters is most important when estimating the starting value at the edge of the fitted region and is highly dependent on the particular fitting problem and optimizer. In essence, each such combination has a region around the sought global minimum, from where the optimizer is able to converge to the correct solution, such as the landscape shown in Fig. 5.1. If the strategy–suggested starting guess falls within that window of acceptable values, the algorithm is able to proceed. A one-dimensional simulated example is shown in Fig. 5.5. Keeping in mind that the weighted local average results in estimates that always "lag behind" the actual values, if the required window changes too fast and the lagging estimate is no longer close, the fit fails. Reducing the cut-off radius r_c or other distance function $f(r)$ parameters for less smoothing allows less lag between the estimate and real value at the price of less robustness to noise.

The local average with $\boldsymbol{P}$ values as weights can be interpreted as fitting (in one spatial and parameter space dimensions in Fig. 5.5) "horizontal" two-dimensional hyper-plane to the neighbouring values, and then assuming that the unknown is also part of that plane.

where μ is the mean and σ^2 the variance of the distribution.

A more generalized approach could be used instead by allowing the said plane to tilt in various directions. Then strategy estimates would be calculated by effectively extending the plane past the measured region and extrapolating. This would have two main consequences: on the one hand, the strategy would perform better with high parameter gradients, on the other, it would become significantly more sensitive to noise and other extrapolation artefacts. In practice, reducing r_c values and using the simpler average approach often gives sufficiently good estimates, while in addition being numerically faster to compute and more robust to noise.

Finally, the number of components (and hence parameters) has no bearing on the performance of the algorithm, as each parameter is estimated independently of all others.

Global strategy

The global strategy in essence is just a thinly-veiled histogram, and as a result it depends on the way the parameter histograms are estimated. A number of "rules of thumb" have been suggested over the years [120, 121] for samples of normally distributed data. However, the main task of the global strategy is to be able to identify when the underlying distribution consists of two or more such normal distributions, hence different bin estimation algorithms had to be used. Knuth [122] suggested an algorithm that allows for the calculation of the optimal bin width for the data using Bayesian probability theory. Whilst powerful, such an approach is constrained to use uniform bins across the dataset, which is unnecessary for the global strategy.

An alternative, one that is used in the global strategy, was suggested by Scargle et al. (2013) [123]. Called a "Block histogram", it allows estimating unbiased and optimal non-uniform bins based on similar Bayesian theory calculations. Its robustness in the original paper (and thus that of the global strategy) was measured by estimating the required amplitude above the background to detect a signal with normally distributed zero mean noise with variance σ^2 from N measurements, such as shown in Fig. 5.4(b). The authors based their analysis on the theoretically derived [124] lowest detection limit intensity

$$A_1 = \sigma\sqrt{2\log N}\,. \tag{5.11}$$

Scargle et al. empirically measured the amplitude requirements for the Block histogram algorithm, and defined the critical threshold to be

$$A_2 = 11.3\sigma\sqrt{\frac{\log N}{N}}\,. \tag{5.12}$$

It was argued to be roughly consistent with the theoretical limit, with the differences mainly due to eq. (5.11) being asymptotic in N while eq. (5.12) is for a specific N value.

We investigated the Block histogram further by measuring the number of detected signals compared to the known ground truth. While not considered in the original study, we found that in order to successfully detect all such signals (which corresponds to successfully identifying domains in the parameter space) the smallest mean amplitude differences had to be

$$A_3 = \Omega_{M,N} \cdot A_2 \,, \tag{5.13}$$

where N is again the number of points in each signal and M is the number of signals in the dataset. Values for Ω as well as an example dataset for one M, N combination are given in Fig. 5.6. The Ω table suggests that the algorithm struggles distinguishing large number of domains when each domain is either very small ($\leq$10 measurements) or very large (>500 measurements). In the first (large M and small N) case, M is underestimated if A_3 (and thus Ω) is small. In the second case, with both M and N large, M is overestimated. For the global strategy to function correctly, the estimated M has to be at least equal to the true value, thus the degrading Block histogram performance with large M values only impacts strategy's performance, and not correctness.

Similarly to the previously considered local strategy, each parameter is estimated independently and thus the number of components does not affect the algorithm performance.

5.3 Implementation

SAMFire was implemented to complement the HyperSpy [101] framework that already had convenient structures for data loading, preprocessing, creating models and fitting using various optimizers.

The algorithm was realised using a "one master – many workers" paradigm. Such architecture pattern was historically first used when large databases had to be replicated [125–127], but since then it has been widely adopted for many other uses as well, such as large data analysis [128, 129] and parallel execution [130] frameworks. The master-workers pattern is well suited for algorithms with many independent and parallel–running tasks. By dedicating a single process ("master") to manage and coordinate all the other processes ("workers"), the total task is completed efficiently: the master assigns each worker a relatively small task to run independently, resulting in a parallel execution. Once a particular task is finished, the result is sent to the master, which assigns the

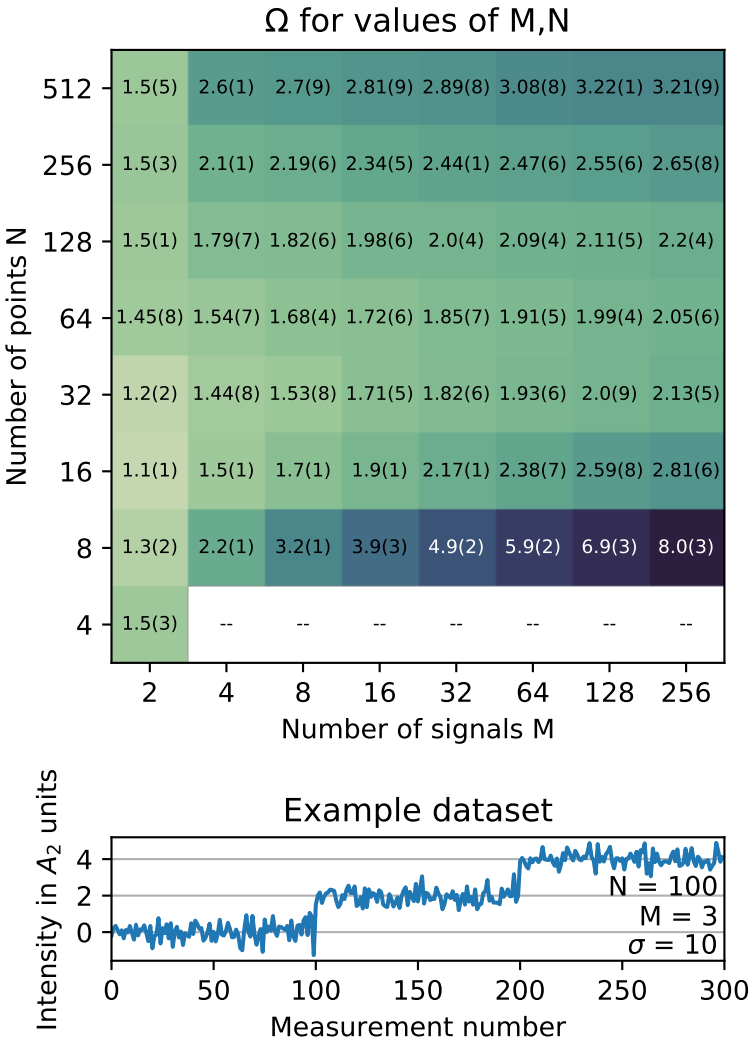


Fig. 5.6 Example dataset for global strategy test is shown at the bottom. The test requires determining the number of domains $M = 3$, where each domain consists of $N = 100$ points distributed normally around the mean with variance of $\sigma^2 = 100$. Ω value matrix with errors in the last digit for logarithmically increasing M, N values is shown at the top. We speculate that $\Omega \neq 1$ for $M = 2$ because only one value jump was present.

worker a new task. Such architecture elegantly deals with unequal execution times and acts as an effective load-balancer.

The master-worker communications are best implemented not directly, but as two queues: a worker-consumed "work queue", where the master process is able to put new tasks to be executed, and a master-consumed "result queue", where workers put the computed results. Such a design pattern is often named as "loose coupling", and is widely used in various cloud architectures [131]. Loosely coupled master-workers communications allow not only significantly simpler and robust implementation, but also changing the number of workers without stopping and restarting the algorithm.

SAMFire was straightforward to implement using the master-workers paradigm. The master process was assigned to decide which pixels should be fitted next and estimating their starting values, while workers were left to perform the actual fitting. This completely separated the optimization from decision making steps, allowing developing and improving each individually.

The SAMFire architecture and its decision tree are shown in Fig. 5.7. The master process consists of two loops: the outer loop, applying different strategies to solve the dataset, and the inner loop, looking for best pixels to fit and estimating starting guesses for them. As explained in Sections 5.2.1 and 5.2.2, the exact methods used in the inner loop depend on the strategy. Nevertheless, a pixel without currently known satisfactory solution is always chosen, and at least one starting guess (SG) is estimated. After that, this information is put in a queue that the worker processes consume. Any worker process takes the first item from the queue and generates all combinations of possible starting guesses for the parameters (often just one). The worker then enters its inner loop, where each combination is attempted as a starting guess for the optimization. If the fit result satisfies the GOF test, then the loop is terminated early, and the result is put in a result queue, consumed by the master process.

With the described architecture, the two kinds of processes end up having rather different properties. For example, only the master process is likely[3] to require significant amounts of memory, as it is the only process to require full access to the data and model. Each worker only deals with one pixel at a time, hence its memory requirements are significantly lower. Furthermore, while the worker computational load increases directly with the difficulty of the particular fit, the master process only performs work when a fitted result is submitted by one of the workers and a new starting guess estimate is required. These properties, combined with the loose coupling, mean that given there

[3]Depends on the underlying architecture of the data loading. An alternative to the common approach is given in Chapter 6

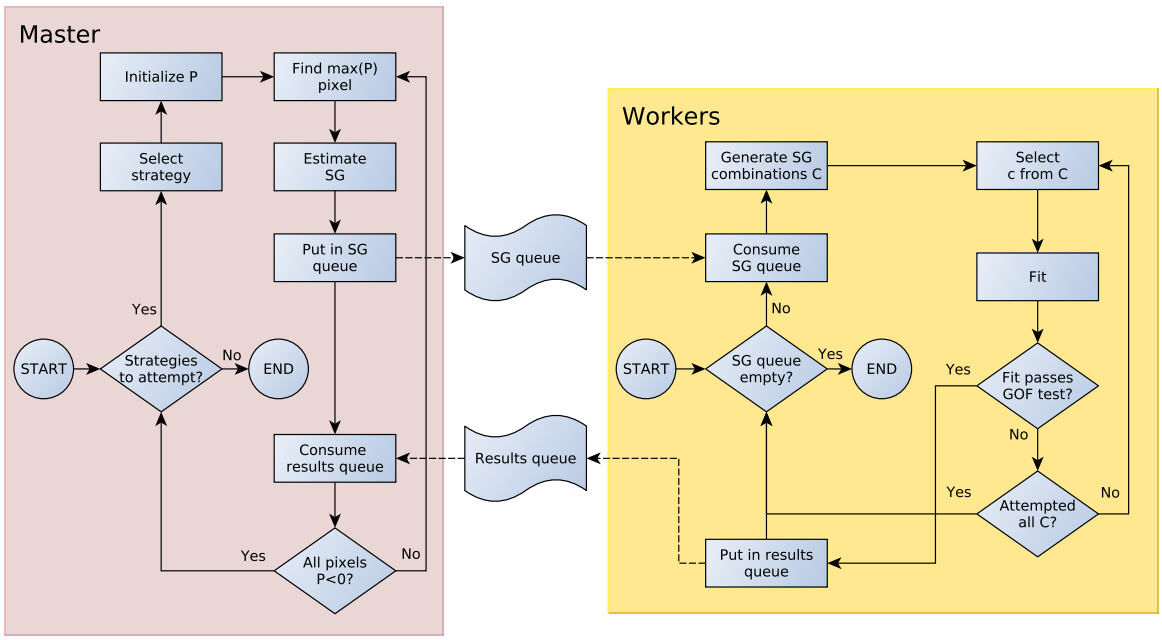


Fig. 5.7 SAMFire architecture and its decision tree. The master process consists of two loops: inner and outer. The outer loop selects strategies that are used for estimations. The inner loop performs the best pixel selection (the one with the highest $\boldsymbol{P}$ value) and starting guess (SG) estimation. The guess is then put in a queue, consumed by the worker processes. The workers grab the top SG from the said queue, generate all possible combinations of the given starting guesses, and attempt each in order. If any of the fit results pass the GOF test, the search is terminated, and the result is put into the result queue, consumed by the master process to update the $\boldsymbol{P}$ values.

are available processors, adding new worker processes is cheap and may significantly boost the overall performance. In addition, the two kinds of processes may benefit from different hardware, for example the master could be run on GPU-enabled machine to efficiently calculate new starting guess estimates, whilst workers are best run on relatively simple, but capable of fast numerical optimization machines. Finally, the loose coupling allows part or even all of the workers to be run on remote clusters, enabling a highly scalable approach to multi-dimensional fitting analysis.

5.4 Synthetic examples

In order to demonstrate SAMFire, three synthetic spectral images were created and fitted, comparing results to the known ground truth.

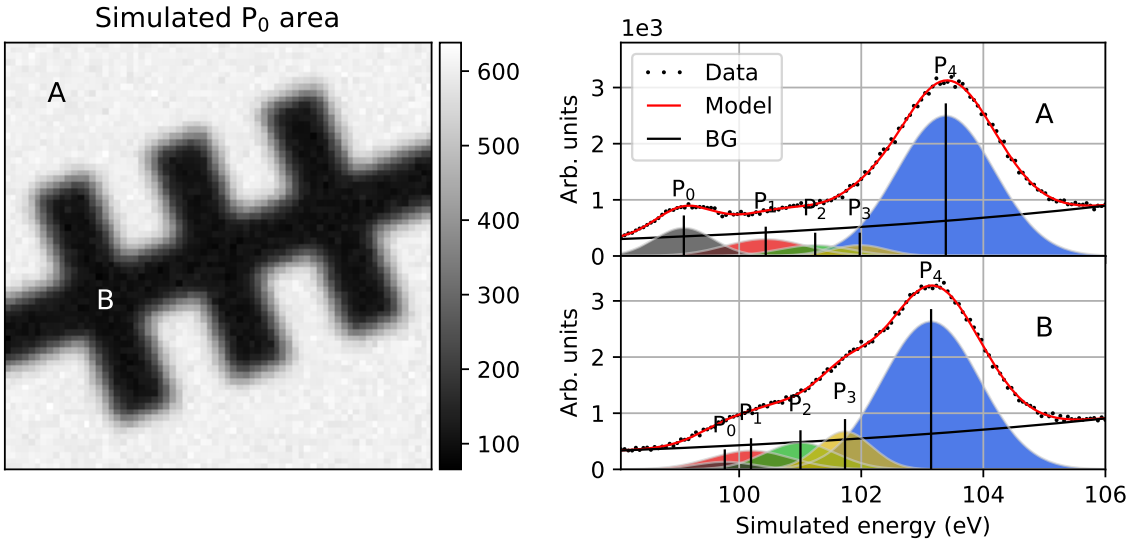


Fig. 5.8 The simulated photoemission dataset consisted of an exponential background and five Gaussian peaks labeled P_0 to P_4. All peak areas as well as P_0 and P_4 centre positions were free to change when fitting, because P_1 to P_3 were constrained to be a set distance from P_4, just like in the original study [58]. The P_0 simulated area map, showing the two domains, is shown on the left.

The first dataset was inspired by the work of Francisco de La Peña et al. [58], and is similar to what might be measured in a photoemission spectroscopy (PES) experiment. The original data from the paper was available, and the full presented analysis was repeated with SAMFire in a significantly more straightforward way.[4] However, to be able to verify the results, a synthetic version of the dataset was created with known true values. It contains an exponential background with multiple Gaussian-like peaks that move in the spectral dimension between two regions, as shown in Fig. 5.8. In the simulation all parameters had normal (Gaussian) variation around the mean intended values, and the final simulated spectrum had Poisson noise added. When fitting, the intensities of each peak, as well as the two background parameters, were free to float. There were two more free parameters – the positions of the P_0 and P_4 peaks. The rest, P_{1-3}, were constrained to be a set distance from P_4, just like in the original study.

In Fig. 5.9 the parameter distributions of the SAMFire fit results are compared to the true values, as well as the fit results if true values were given as the starting guesses. It can be seen that both the ideal fit solution and SAMFire suffered from a parameter distribution broadening due to the Poisson noise that was added to the data.

[4] In fact, the paper served as the initial inspiration for SAMFire.

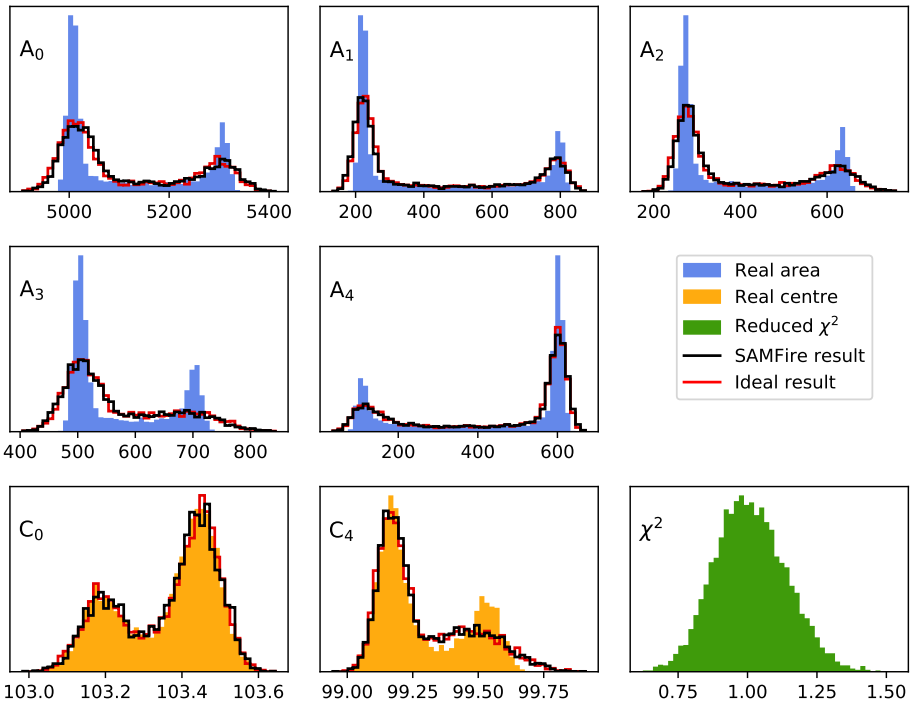


Fig. 5.9 Simulated photoemission fitting results. Results for seven parameters not associated with the background component are shown: peak areas in blue, centres in yellow. In each of those histograms, the true value distribution is shown as the filled in histogram, SAMFire results as a black line, and ideal fitting results as the red line. The reduced χ^2 of the SAMFire fit is shown in green in the bottom right.

Nevertheless, ideal and SAMFire results are nearly identical, showing that the algorithm was able to always provide a starting guess sufficiently close to the true solution.

The second synthetic example dataset is inspired by the experiment and results that will be presented in chapter 8. Again, in order to have the ground truth values to gauge the accuracy of the analysis, a similar dataset was simulated. The simulated EELS dataset considered a pure crystalline boron core, surrounded by a boron oxide enclosed in a boron nitride shell. The simulation did not contain any multiple scattering effects. Both pure B and oxide only require one component, but previous studies [61] showed that due to the BN anisotropy, two electron loss near-edge structure (ELNES) fingerprints have to be used for the outer shell. The final model consisted of a power-law background and four EELS edges that completely overlapped in the spectral dimension, corresponding to the four boron ELNES components that were used in the analysis. The spatial distribution was simulated by creating model 3D masks and then projecting them to the two measured dimensions, as shown in Fig. 5.10. The BN anisotropy was

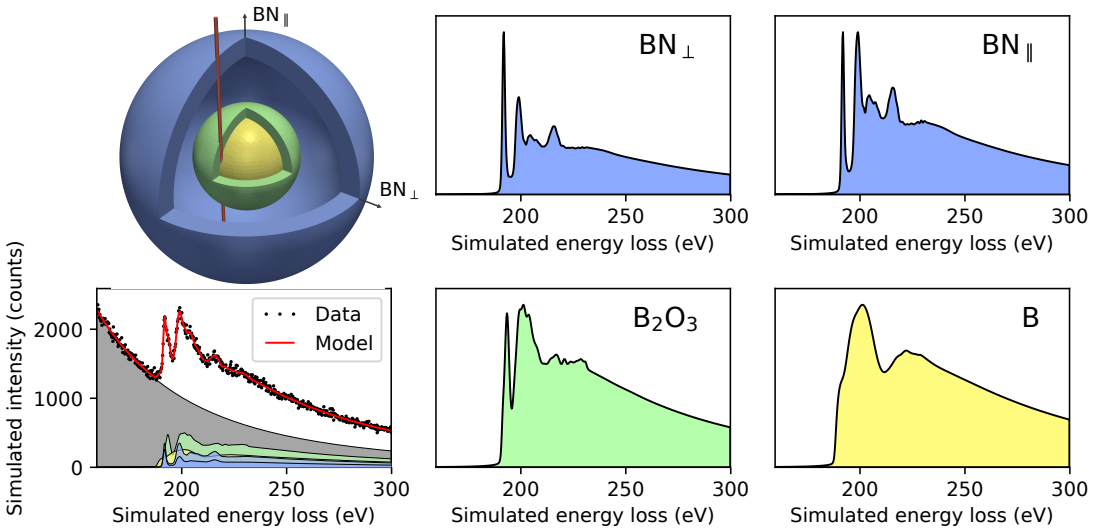


Fig. 5.10 Simulation of a core–double shell boron nanoparticle. The four EELS templates used for both simulation and analysis are shown in the two right columns. Due to the boron nitride anisotropy, two components are used for the outer shell simulation, with intensities as sine and cosine of the angle between the electron trajectory and the shell surface normal. The simulated data with added Poisson noise and the fitted model are shown in the bottom left for the trajectory marked with the red bar in the 3D visualisation.

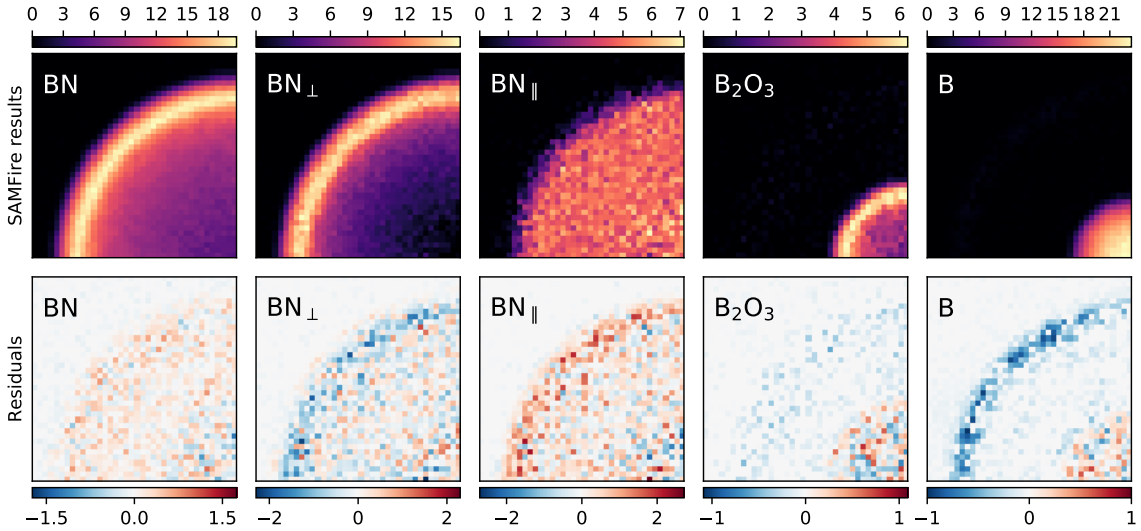


Fig. 5.11 (Top) results and (bottom) their residuals from the known true values for the total BN signal and the four components used in the simulated data analysis. Only one quarter of the result maps are shown due to the spherical symmetry of the simulation.

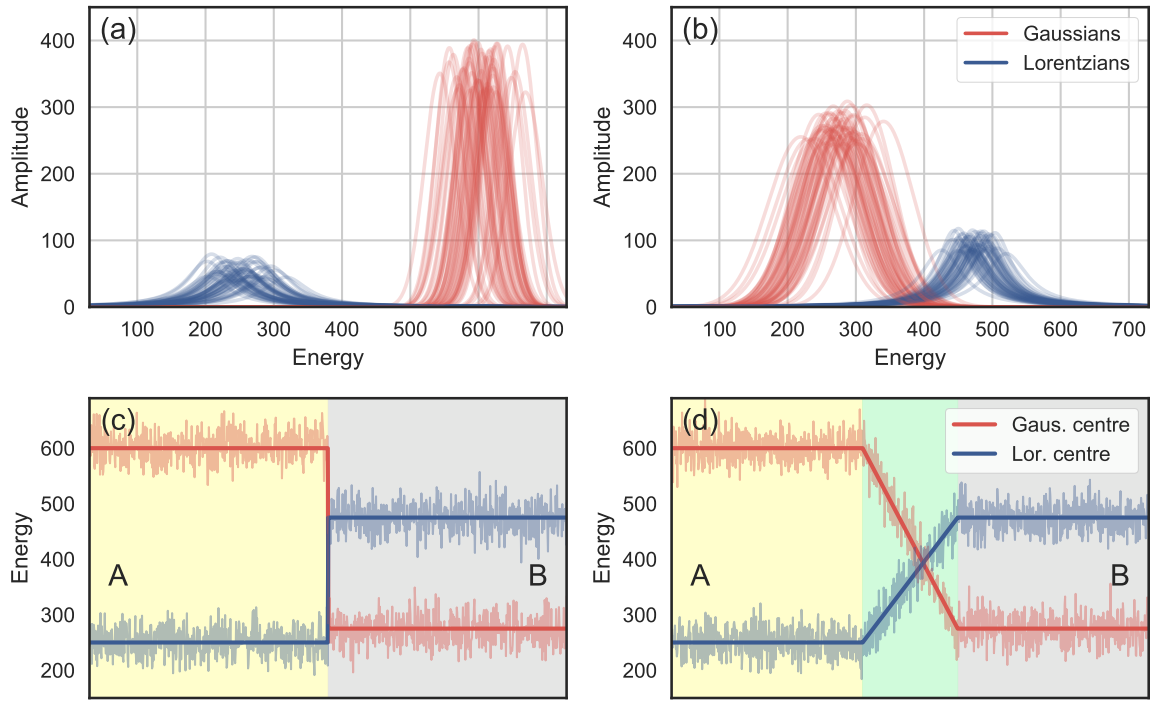


Fig. 5.12 Lorentzian and Gaussian curves exemplifying the parameter distributions A and B domains of the synthetic dataset in (a) and (b) respectively. The actual fitted spectra consisted of one Gaussian and Lorentzian pair with simulated Poisson statistics noise. (c) abrupt and (d) smooth domain boundary is shown by plotting the peak centre energies as a function of pixel positions. The shaded regions correspond to domains A and B, with the "transition" domain only appearing in (d).

simulated by modulating the strength of BN components by the cosine and sine of the angle between the virtual beam and the outer shell surface normal.

The analysis results and the residuals after subtracting the known true values for a quarter of the full map are shown in Fig. 5.11. The quality of the results shows the expected behaviour of components with more signal leading to more accurate results. We notice that the two BN components were not unmixed perfectly, each resulting in larger than expected and anti-correlated residuals. Nevertheless, the sum of the two results leads to a higher precision total BN signal map that could be interpreted as the number of boron atoms bonded with nitrogen.

Unlike the previous two examples, the final simulated spectrum image is not meant to show a typical use-case of SAMFire, but rather the data complexity that is still readily solved by the algorithm. The spectrum can be fully described by two peaks, a Lorentzian and a Gaussian[5], with their parameters (positions, widths and intensities) changing both

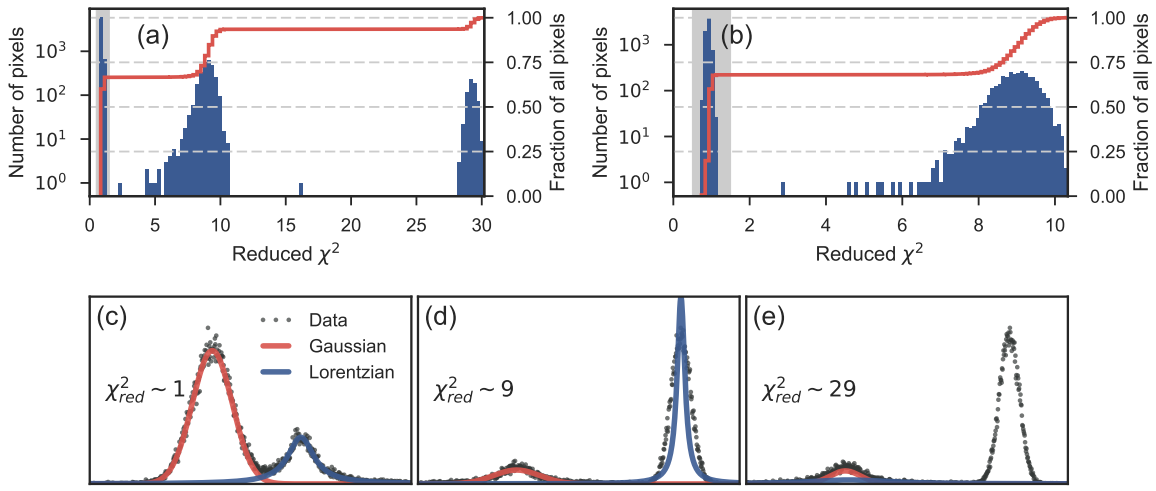


Fig. 5.13 Goodness of fit measure results when the synthetic dataset was analysed using regular fitting routines, which use (a) last results and (b) a constant values as starting guesses. The χ^2_{red} GOF measure was calculated for every pixel, its distributions are shown in blue. The red lines show cumulative fraction of pixels fitted better than the corresponding χ^2_{red} values. (c-e) Fits when $\chi^2_{\text{red}} \sim 1$, 9 and 29 respectively.

across the dataset and "jiggling" uniformly at random as shown in Fig. 5.12(a,b). The dataset is constructed such that in some pixels (domain A) the Gaussian peak is at higher energy, and in others (domain B) – lower. Fig. 5.12(c,d) shows how the two curve centres shift in a sequence of spatial pixels that start in domain A and end in B. The dataset was constructed such that both instant and gradual parameter change was present. Due to the choice of the curves, the fitting landscape situation closely resembles the problem shown in Fig. 5.1, where even a slightly wrong starting guess quickly leads to a local minimum and a bad fit. As a result, the algorithm has to be able to deal with both a smooth and an abrupt parameter value change in order to solve the dataset. To minimize accidentally correct starting guesses, the dataset consists of domain boundaries in many different orientations.

First, two regular fitting routines were run as control experiments. Both traversed the dataset in the traditional raster–order row by row from top left, as it was stored

[5]Gaussian and Lorentzian functions for real constants a, b and c are defined as

$$G(x) = ae^{-\frac{(x-b)^2}{2c^2}},$$
$$L(x) = \frac{ac}{(x-b)^2 + c^2},$$

where a is the height of the curve, b the position of the center of the peak, and c controls the width of the peak.

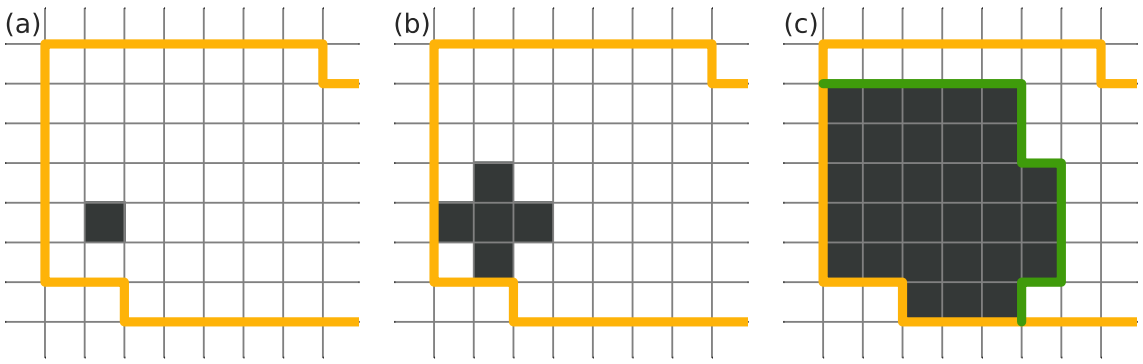


Fig. 5.14 The local SAMFire strategy fitting dataset. The yellow line marks the sudden A to B domain boundary, as shown in Fig. 5.12(c). (a) The seed pixel. (b) As defined, the seeds neighbours have the highest $\boldsymbol{P}$ values and are fitted next. (c) Propagating fitting front (marked in green) emerges as SAMFire proceeds.

in memory. The first approach used the last known optimization value as the starting guess for the next pixel. In the second case, a constant starting guess (matching the B domain) was used throughout. Note that the synthetic dataset was intentionally created such that even with the peaks reversed, the conventional optimizers managed to find a local minimum (corresponding to LM2 in Fig. 5.1). If this was not the case, optimizers would have diverged in most pixels (as often happens with real-life examples), preventing comparisons of the results. The χ^2_{red} distributions for both methods are shown in Fig. 5.13. According to its definition, best fits correspond to $\chi^2_{\text{red}} = 1$, with overfitting and underfitting occurring below and above this value respectively. Based on my experience fitting real spectral datasets, $0.5 < \chi^2_{\text{red}} < 1.5$ typically corresponds to sufficiently good fits. Both conventional methods managed to fit around 70% of pixels well, which corresponds to the fraction of pixels in the domain B. The rest were fitted poorly, $\chi^2_{\text{red}} >> 1$, due to the algorithms not being able to adapt to different domains and converging on a local minima. Crucially, using the (a) "last-result" starting guess estimates around 7% of pixels were fitted extremely poorly, with $\chi^2_{\text{red}} \approx 28$.

Seed pixels, if possible, should be chosen to contain as much information as possible by selecting pixels that require the most degrees of freedom and components to fit. For the dataset all possible seed pixels in this sense were equal, thus one that resulted in a visually interesting fitting path was chosen, as seen in the supplementary movie. SAMFire was initialized with just one pixel already fitted as a seed to learn from, shown in Fig. 5.14(a). As the dataset structure is known, a yellow line marks the sudden change from domain A to B, as shown in Fig. 5.12(c). Once the starting pixel was given, the local strategy was used to calculate the potential as described previously. As $\boldsymbol{P}$ was

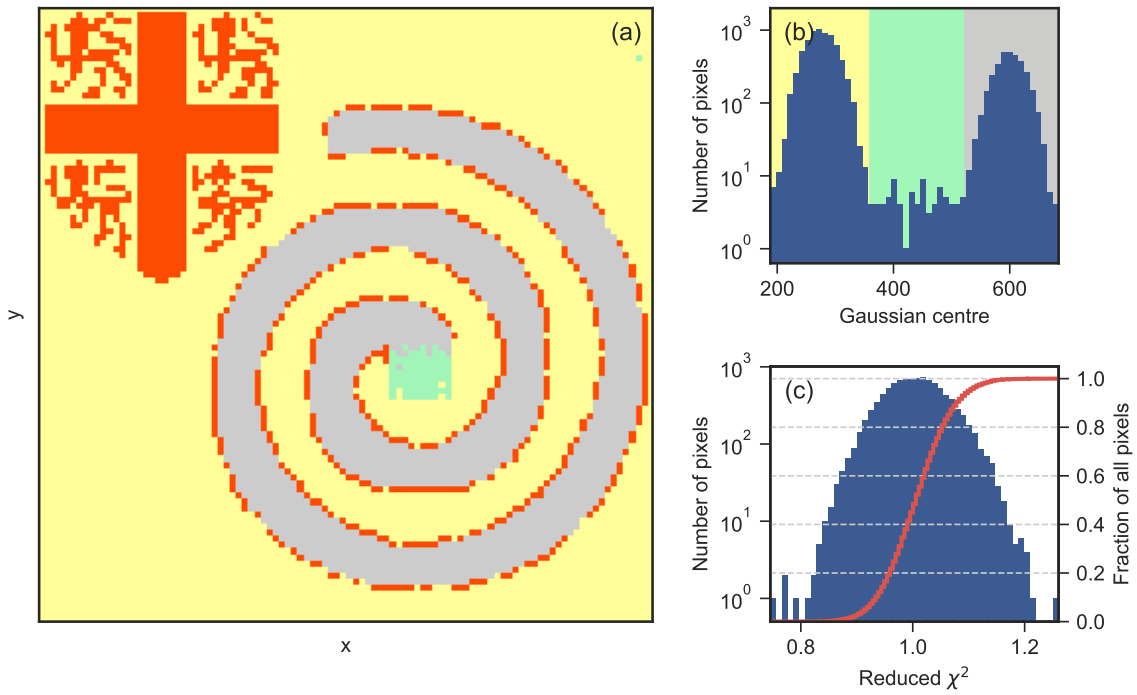


Fig. 5.15 (a) Fitted pixel spatial distribution when local strategy finished. (b) Gaussian centre parameter value distribution at the same point as (a). The two domains can be clearly identified as peaks in the histogram. In (a) unsuccessful fits are shown in red, with other colours corresponding to regions in (b). (c) The final χ^2_{red} SAMFire distribution, with all pixels successfully fitted.

highest near the seed, the pixel's neighbours were fitted next, shown in panel (b). As the local strategy proceeded, good fits resulted in high $\boldsymbol{P}$ values and accurate starting guesses, which in turn often allowed better fits. On the other hand, with only the local information available for each pixel, SAMFire failed to find a good fit for domain B and was not able to cross the domain boundary. Such positive feedback and containment created a propagating "fitting front", marked in green in Fig. 5.14(c). When the front reached the dataset region with smooth A to B transition as shown in Fig. 5.12(d), the local strategy was able to successfully follow parameter shifts, allowing the fitting front to propagate into domain B.

Once finished, the local strategy ended up not fitting 6% of pixels, shown in red in Fig. 5.15(a), and the global strategy was employed to finish fitting the dataset. Fig. 5.15(b) shows the relative frequency for one of the parameters from the model. The two domains can be clearly seen from the histogram. Regions, corresponding to parameter value ranges are marked with identical colours in both (a) and (b) panels of the figure. Once

Table 5.1 Table giving performance metrics when performing the fits for the three synthetic datasets given in section 5.4 on a laptop with Intel® Core™ i5-3320M CPU @ 2.60GHz × 4.

Dataset	Metric	Method		Improvement
		Traditional	SAMFire	
PES	time (s)	189	350	x1.8
	correct (perc)	67	100	33
EELS	time (s)	676	219	x0.3
	correct (perc)	99.9	100	0.1
Spiral	time (s)	714-1362	1112	x0.8-1.5
	correct (perc)	22-32	100	68-78

histograms were evaluated, this global information was used as described in section 5.2.2 to successfully fit the remaining pixels. Fig. 5.15(c) shows the final χ^2_{red} distribution, with all pixels well within the $[0.5, 1.5]$ "good fit" window. A movie of SAMFire fitting the synthetic dataset is available in the supplementary CD.

5.5 Performance

The fitting performance when analysing the three synthetic datasets on a laptop with Intel® Core™ i5-3320M CPU @ 2.60GHz × 4 is given in table 5.1. Traditional fitting running times for the simulated BN EELS data are relatively low due to the optimizer getting stuck in a false minimum, thus converging on an incorrect solution for all subsequent pixels abnormally quickly. Depending on the particular fitting problem, SAMFire improves the fraction of correct fits or reduces the running time, or both.

Chapter 6

Big Data

As measurements become increasingly data-rich and specimens more complex, many sciences experience "data explosion" [99]. The phenomenon always presents itself similarly, however the peculiarities differ from one field to the next. While previously found solutions often cannot be applied directly in other sciences, many general approaches can be reused and learnt from. The big data problem for electron microscopy in the light of other fields will be described in section 6.1. An overview of the available tools and the implemented solution will be laid out in sections 6.2 and 6.3. An example workflow that previously would have required expensive dedicated hardware is described in section 6.4.

6.1 Motivation

The modern computer can be said to be the main tool of many scientists. Computers are used throughout all stages of a modern scientific discovery, from running most experiments, some completely virtual in the form of simulations, to the final data analysis and visualization. While many large–scale international projects rely on significant computational resources (CERN, LIGO and others), few electron microscopy labs can offer dedicated data analysis hardware for all its users. As a result, being able to use consumer-grade personal computers (PCs) is of paramount importance.

Until recently, the self-fulfilling prophecy of Moore's law [132] allowed many analysis methods to be developed and easily applied to the data, as it usually fit comfortably in the computer memory. However, in recent years the analysis of the EM data has become significantly more demanding and important [133–135], sometimes justifying the term "computational electron microscopy". With both the specimens and experiments becoming more complex, and microscopes operating faster and achieving higher resolution, the average PC memory is quickly outpaced by the growing scientific needs.

As briefly mentioned in section 4.2.1, one of the primary issues is that many of the current analysis and data-handling software solutions rely on being able to store the full dataset in the computer memory. If this first step is not possible, no further analysis or visualization can take place, effectively rendering the data useless unless a more powerful machine is used. Even if the software is able to open the dataset, in most cases further memory is required to store the results of any calculations, limiting the largest datasets that can be analysed on average workstations.

The matters are often complicated further if a machine learning algorithm is to be used for the analysis. Such methods greatly benefit from large datasets, thus incentivizing scientists to collect more data for better results. On the other hand, sometimes due to the nature of the experiment only a subset of the acquired dataset is needed for the analysis, but the measurements cannot be targeted sufficiently well to acquire the needed data. With such overly-rich data, many redundant calculations (requiring additional memory) have to be performed as intermediate steps in order to draw the final conclusions. Fortunately, a number of different solutions exist to facilitate both types of analyses.

6.2 Frameworks

In the data processing fields there are numerous ways to approach "big data". Historically, large datasets were first analysed on distributed clusters. Such structures can be thought of as an array of nodes, where each node is similar to a normal PC, with its own memory and processing units. By connecting them to an exceptionally fast network, the full structure can be effectively treated as one supercomputer. Crucially, each node could perform its computations in parallel with all other nodes, as long as the whole process was orchestrated well.

"MapReduce" [128] framework by Google, shown in Fig. 6.1, resulted in a major break-through in the field. It offered a way to orchestrate and parallelize operations on the many-node supercomputers. In particular, the algorithm managed loading a chunk of the large dataset in each node, then applying the same function for each chunk across all nodes ("map"). The map results then had to be "reduced" in groups (e.g. counting items in each group), which was done by the intermediate result shuffling step to appropriate workers, before actually running the reduction function. After that, the result was written back to disk. The split-load-map-shuffle-reduce-write workflow is suitable for many big data applications in both industry and sciences [136–140], however the largest MapReduce limitation was the inability for the process to reuse previous

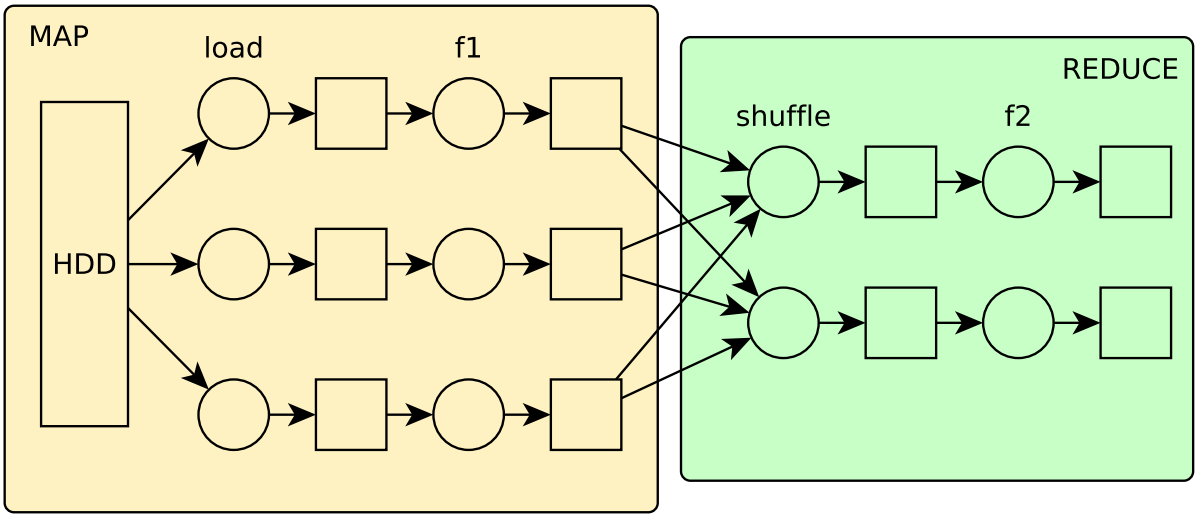


Fig. 6.1 An example MapReduce diagram. Circles and squares mark functions and their results respectively. The user is able to supply functions f1 for "mapping" and f2 for "reduction". After that, the results are saved back to the hard disk.

results without storing them to disk, hence rendering most iterative algorithms unsuitable for the framework.

Inspired by these MapReduce limitations, an improved implementation, known as "Spark" [129] (later "Apache Spark") was suggested and quickly popularized by the Berkeley group. The main idea of the framework involves expressing the computations as operations on "Resilient Distributed Datasets" (RDDs) [141] that the Spark engine is able to optimize and execute in parallel on the cluster. This is only possible due to the RDDs keeping track of their lineage: each RDD keeps track of the function that generated it. By writing computations using RDDs, algorithms were effectively expressed as Directed Acyclic Graphs (DAGs), as shown in Fig. 6.2. Having the full DAG is useful on many different levels – it allows culling unnecessary operations that are not used for the end result, offers more insight for memory management, for example keeping intermediate results if they are required for the next function, and enables lazy[1] evaluation while constructing the algorithm.

While Spark and RDD offer significant improvement over MapReduce, a number of design decisions still limit the usefulness of the framework. Not giving the user direct access to the data chunks is arguably the most important Spark drawback – all allowed operations have to operate on the full RDD and not its parts. This restriction severely

[1]Lazy evaluation is an evaluation strategy, where an expression is only computed when (and if) the results are requested [142–144]. This allows potentially infinite data sources and various other optimizations.

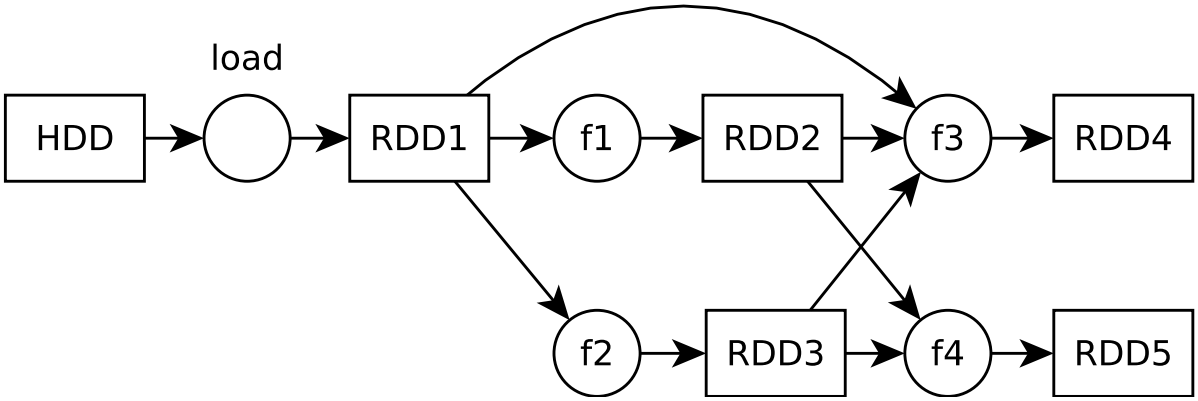


Fig. 6.2 An example Spark directed acyclic diagram (DAG). Circles and squares mark functions and their results respectively. The user is able to define all functions and reuse previous results (RDDs) for future calculations, however all internal chunking of the data is hidden and not accessible.

limits the use of the tiled (blocked) algorithms[2] which have been shown to perform well on such clusters [146, 147]. Unless it has been explicitly implemented in the Spark engine, adding new blocked algorithms is relatively difficult. Finally, while it is possible to run Spark on a single machine, it is generally only recommended for testing purposes, further limiting its usefulness for EM data analysis on traditional PCs.

In late 2014 Rocklin presented another similar framework called "dask" [105]. The idea of dask was to use the DAG as the core concept, and build the structure from there without abstractions like the RDD (Fig. 6.3). This resulted in a framework that not only inherited the laziness, but also allowed direct access and manipulation of the DAG and its members – most often the data chunks. Such freedom allowed users to implement blocked algorithms easily and straightforwardly. The downside of dask is that the framework is not aware of the large-scale computations, meaning only the DAG (not the algorithm) can be optimized automatically, and many possible optimizations are left to the algorithm implementor. Another great advantage of dask is its administrative side. From the beginning, dask was intended to efficiently run on a single machine (and only later expanded to be able to run on thousands of cores in a cluster), thus the setup is virtually non-existent and very user-friendly.

[2]Tiled (blocked) algorithm is an algorithm that performs matrix operations by dividing it into many smaller submatrices and their operations to construct the final result block by block [145]

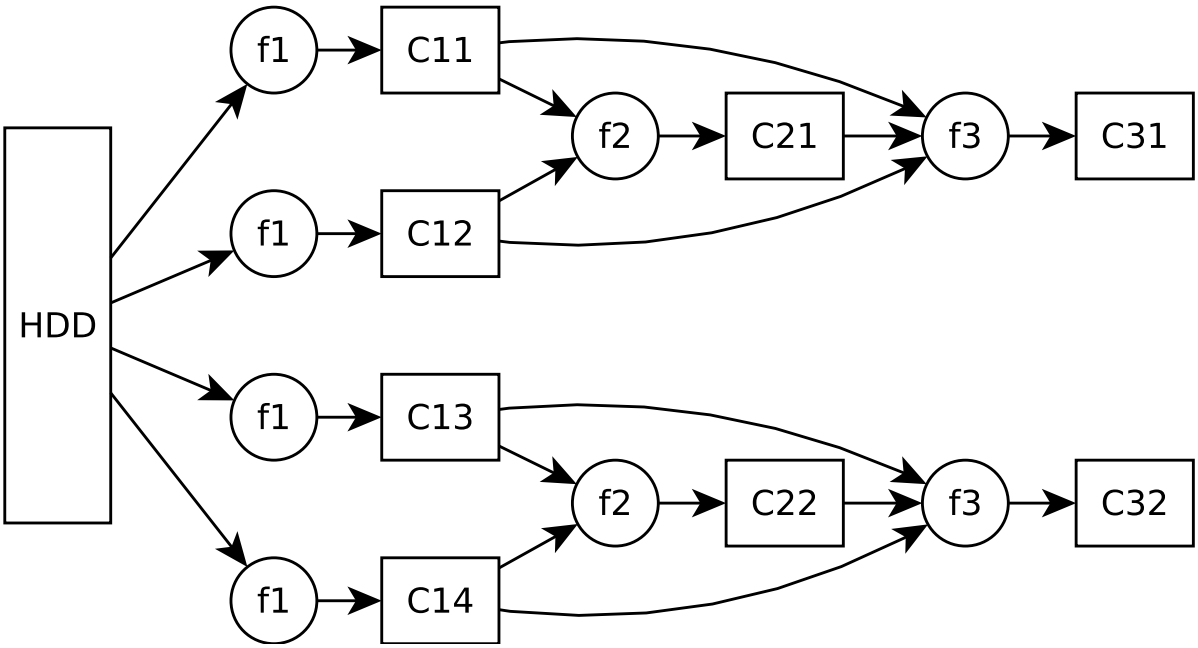


Fig. 6.3 An example dask DAG. Circles and rectangles mark functions and their results respectively. The user has full control over functions (f1-3) and chunks (C11-32), allowing flexible algorithm design. In dask framework loading the data is treated just like any other function (in this case it's f1). If only C32 is required, the DAG is simplified: C11 and C12 are not loaded to the memory, and C21 and C31 are skipped.

6.3 Implementation

The proposed implementation uses dask to solve the big data problem. Named "LazySignal", it has already been added to the HyperSpy toolbox [101]. Instead of attempting to load the full dataset, LazySignal constructs a dask DAG, where each node only loads a particular chunk of the data, as expected by the tiling algorithms. Any further operations are then just added to the graph lazily. Once a result that cannot be left lazy (for example, the visualization of the dataset) is required, dask uses the graph to identify branches that can be run in parallel. If suitably small chunks were chosen, each branch and its results fit comfortably in memory, allowing performing operations on the large datasets. In practice, the best-performing chunk size is significantly smaller than the standard available memory, allowing running the computations on multiple computer cores at once.

Standard machine learning algorithms, as described in section 4.1.2, are, however, more problematic. While there are matrix decomposition implementations [148] for tiled matrices, for best performance the total data tensor should be "tall and skinny" [149]. Unfortunately, most experiments produce fairly square data tensors, leading to the

algorithms still requiring too much memory for common computers. Instead, the traditional PCA and NMF formulations were replaced by "online" versions [102, 103]. Online machine learning algorithms are created for potentially infinite data sources, such as the Internet. They approximate a conventional algorithm by iteratively refining the learnt results (in PCA and NMF case components) with each new datum that is presented. By supplying such an algorithms with small data chunks, LazySignal is able to extract the components from very large datasets. The main drawback of such an approach is that for accurate loading maps the dataset should be read twice: the first time to learn the components, and the second – to project the data on them and calculate the loadings with the final component versions that hopefully converged. On the other hand, a potentially infinite data source (a microscope) could be used to generate data and extract the components while the experiment is still being performed.

6.4 Example workflow

In order to illustrate the possibilities of the LazySignal framework, I will describe a typical workflow that was previously effectively impossible due to the size of the dataset. The data in question was acquired on a transmission electron microscope (TEM) by scanning the focused beam across the sample and measuring a 2D diffraction pattern (DP) at each location, resulting in a 4-dimensional scanning electron diffraction (SED) dataset.

With the data recorded as integers in $0 - 255$ range, the full dataset amounts to over 32 GiB and is already prohibitive for conventional data loading approaches on most consumer-grade computers. Nevertheless, LazySignal allows opening multiple such files on a conventional laptop. In fact, until further operations are performed, only the general information about the data is read: the dimensions of the tensor and how many bytes each element would require if loaded. This in effect sets up the internal infrastructure for future processing. In particular, it chunks the dataset in such a way that each DP is always whole in one chunk, and real-space is subdivided into sub-regions.

Next the dataset is explored by plotting. This can be done in a number of ways, but the most often encountered method plots a particular DP from the selected real-space position, recreating the exact image measured during the experiment. As this follows both acquisition and chunking schemes, the operation is perceptually instantaneous and no different if data were loaded conventionally. Behind the scenes, only the chunk that contains the currently plotted DP is loaded at any point. This can also be extended with Regions Of Interests (ROIs) of various shapes, where all real-space pixels inside the ROI

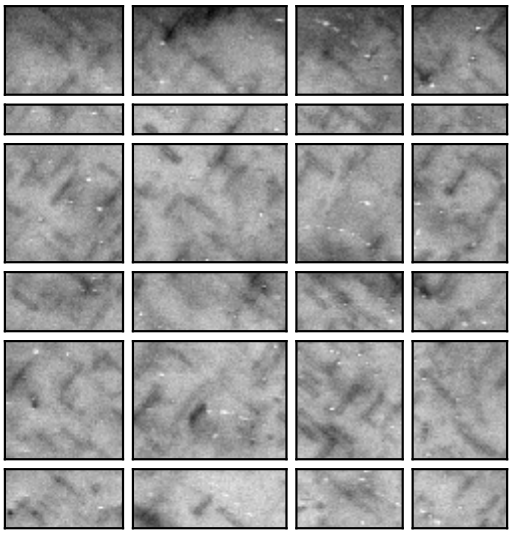


Fig. 6.4 An example VDF with different chunks in real-space separated. In practice chunks are much smaller, often around 10 to 20 pixels in each direction.

are averaged, for example, before plotting. In such case all chunks that span the ROI have to be read from disk. If there is enough available memory, all required chunks are loaded at once to perform the operation and plot the mean DP. Otherwise, the chunks are read in sequence, storing the intermediate result of the operation before displaying the final image.

Another often much-revealing way to plot SED data involves forming a Virtual Dark Field (VDF) image. It entails selecting a pixel in the DP and forming the VDF at all real-space positions using the selected reciprocal-space pixel. As shown in Fig. 6.4, even a single VDF formation requires loading all chunks of the dataset. While the process takes around 2 hours when using other software, HyperSpy with LazySignal performance is bound by the disk reading speed and takes under a minute to perform an identical operation.

Once the goal of the analysis is clear, the data is usually re-cast as floating-point values for higher precision. If the dataset was loaded conventionally, this would increase the required memory 8-fold to 260 GiB. In the LazySignal framework, however, such re-casting operation merely gets added to the end of the DAG and is performed on each chunk only if it is required for further processing.

Most analysis and other processing methods are defined for one DP, and applied repeatedly on all real-space pixels of the dataset. For example, if the direct beam spot drifted from the centre of the DP during the course of the experiment, it could be

easily fixed for each pattern in post-processing. Using the LazySignal "compute-only-when-needed" framework such alignments become much cheaper to perform. Instead of replacing the loaded data with the post-processing result or doubling the required memory to store it, each DP is only aligned when required. Chaining such operations allows forming complex processing routines on large datasets without expensive hardware. If the mentioned data were plotted after alignment, first the chunk with the required real-space pixel would be loaded from the disk, then the integer-type data would be re-cast in higher precision and aligned as required. Because both re-casting and alignment are relatively cheap operations, both can be performed faster than perceptually noticeable, irrespective of the size of the dataset.

LazySignal framework also works well with significantly more complex analysis methods. For example, both traditional non-linear optimization and SAMFire can be successfully run on such data without any changes. As already mentioned, if machine learning methods such as PCA or NMF are required, their online versions are also included in LazySignal: this allows us to iteratively perform the decompositions in two steps. First, the "factors" are learnt by loading each chunk (also performing any required pre-processing) and supplying this information to the online algorithm. After the data were read once and suitably accurate factors are estimated, each chunk is loaded the second time to project it in the learnt factor space. While such processing is slower and only an approximation of the traditional algorithms, it requires significantly less computer memory.

Finally, LazySignal framework supports operations in a distributed computing environment, which speeds up computations by loading the dataset into the distributed memory. Even though groups of chunks are read in each machine's memory, complex operations requiring information transfer (such as forming a VDF) can still be run. Most importantly, such distributed environments are cheap and simple to setup on online services providers, making them available for most scientists.

6.5 Performance

The LazySignal and traditional method performances when summing all values in various datasets are shown in table 6.1. Due to the laptop[3] running usual background processes, only around 8GB were available for computations. While the additional requirement of more memory for operations and result storage prevented from opening even 8GB

[3] A laptop with Intel® Core™ i5-3320M CPU @ 2.60GHz × 4 with 12GB of RAM memory, running Linux.

Table 6.1 Table giving running times summing all pixels of respective datasets. Performed on a laptop with Intel® Core™ i5-3320M CPU @ 2.60GHz × 4 with 12GB of RAM memory, running Linux.

Dimensions	Representation	Size (GB)	Method (s)	
			Traditional	LazySignal
2048 × 2048	u-int (8bit)	0.004	0.004	0.013
(Traditional	u-int (16bit)	0.008	0.003	0.013
TEM image)	float (32bit)	0.016	0.002	0.016
	float (64bit)	0.032	0.003	0.022
1024 × 1024 × 2048	u-int (8bit)	2	1.8	1.5
(Traditional	u-int (16bit)	4	1.7	2.0
STEM EELS)	float (32bit)	8	–	3.9
	float (64bit)	16	–	4.4
256 × 256 × 256 × 256	u-int (8bit)	4	3.9	3.7
(Traditional	u-int (16bit)	8	–	7.3
SPED map)	float (32bit)	16	–	15.7
	float (64bit)	32	–	61.5
256 × 256 × 2048 × 2048	u-int (8bit)	256	–	161
(Potential	u-int (16bit)	512	–	271
SPED map)	float (32bit)	1024	–	504
	float (64bit)	2048	–	982

datasets, LazySignal framework was able to successfully finish the operations, while never requiring more than 2GB additional memory.

The datasets were generated using either `numpy` or `dask.array` libraries, containing only ones in each position. The sum result was checked to be equal to the number of elements in the datset. The dataset dimensions and representations were picked to be representative of typical or potential EM datasets.

Chapter 7

Monitoring the Stark effect in quantum disks

This chapter includes work published in:

> L. F. Zagonel, L. H. G. Tizei, G. Z. Vitiello, G. Jacopin, L. Rigutti, M. Tchernycheva, F. H. Julien, R. Songmuang, T. Ostaševičius, F. de la Peña, C. Ducati, P. A. Midgley, and M. Kociak. Nanometer-scale monitoring of quantum-confined Stark effect and emission efficiency droop in multiple GaN/AlN quantum disks in nanowires. *Phys. Rev. B*, 93:205410, May 2016

In particular, TO pre-processed and analysed the datasets and produced the figures unless otherwise noted.

By growing semiconductor crystals with a high degree of control, a new generation of potential optoelectronic devices with exciting properties have been achieved in laboratories [150–152]. One example is nanowires (NW) grown with quantum-confined heterostructures of different materials in layers called Quantum Disks (QDisks). One such class uses III-nitride structures for light emitting diodes (LEDs) in the visible-ultraviolet range [153, 154].

To engineer such NWs for practical use, a more thorough understanding of the various effects changing their performance is required. It has been shown [156, 157] that a high electric field is present inside such crystals due to strain. In turn, the high field gives rise to the so-called Quantum Confined Stark Effect (QCSE) [158, 159], where the electron and hole energy levels (p-doped and n-doped bands) are pushed closer together in energy, leading to a redshift of the emission wavelength of these structures (Fig. 7.1).

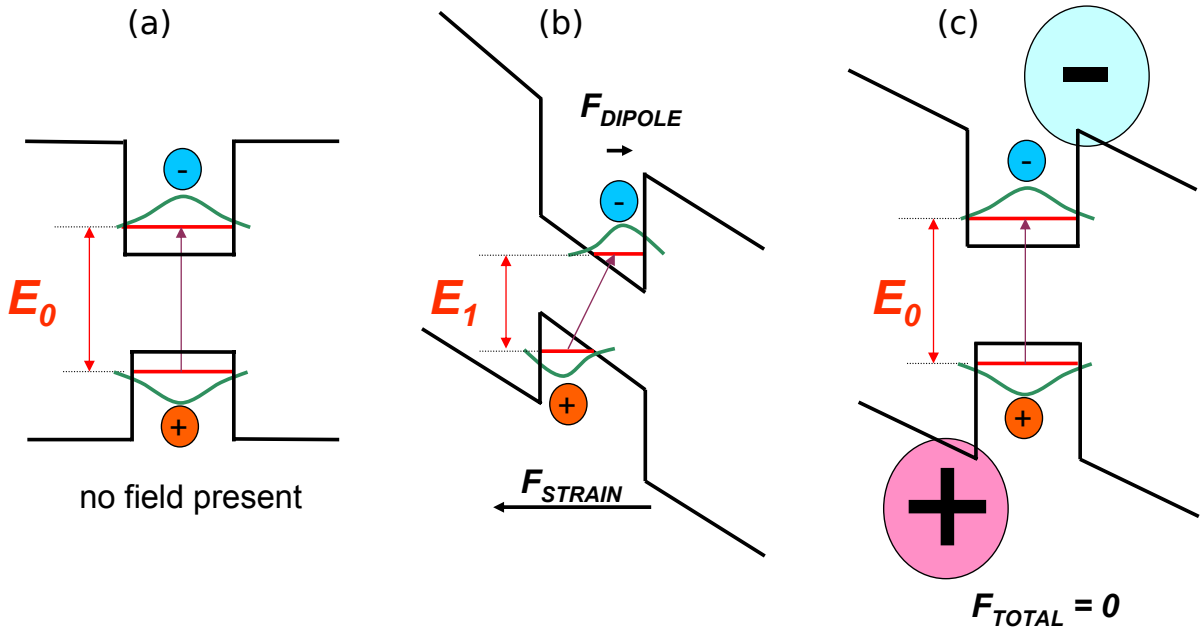


Fig. 7.1 (a) Electronic band structure sketch with no electric field. Red lines and green curves mark energy levels and wavefunctions respectively. (b) Band structure with the strain-induced electric field (QCSE). Both transition energy and wavefunction overlap are reduced. (c) Accumulating charge carriers screen the strain-induced field, undoing the QCSE. Adapted from [155].

Additionally, if the physical width of such quantum wells (or in this case the thickness of the quantum disk) is large enough, the electron and hole wavefunctions are "squeezed" in the opposite directions, reducing the overlap and hence transition probability between the bands. The described QCSE changes if the quantum structure is driven strongly enough, complicating attempts to study it. Namely, if the carrier injection into the crystal rate exceeds that of the recombination, electrons and holes accumulate in the bands. The free charge carriers (CC) screen the aforementioned internal electric field, reducing the effects of QCSE. Thus as the CC density increases, the transition energy redshift gets undone (effectively blueshifting the emission). The screening also makes the energy levels flatter, hence increasing the carrier wavefunction overlap, increasing the transition probability and the photon emission rate (Fig. 7.1(c)). On the other hand, with increasing carrier density, high-order effects become dominant. One such example is the Auger effect, where the excess energy is not released as a photon, but instead is transfered to a third charge carrier, detrimental to the quantum structure performance [160].

Although the interplay of these two effects is thought to be the main cause of the so-called "efficiency droop" for QDisks and similar crystal structures [160–162], the contributions of each effect has proved difficult to isolate and evaluate in 3D materials

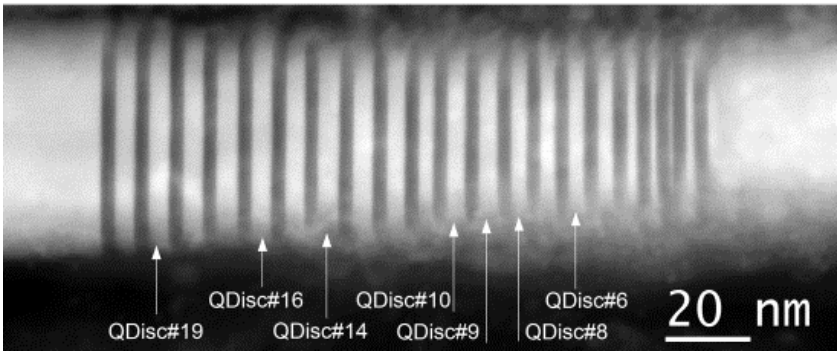


Fig. 7.2 HAADF image of a NW with individual QDisks studied later marked.

using the traditional photoluminescence methods [163–165]. Instead, we used a different approach by probing such NWs using CL-STEM.

7.1 Experiment

The specimens of interest were GaN NWs, each containing 20 GaN/AlN QDisks, shown in Fig. 7.2. The NWs were grown by catalyst-free plasma-assisted molecular beam epitaxy, as described in [116, 166]. Each GaN QDisk increased in thickness from ∼1 nm to ∼4.5 nm in the growth direction, while the AlN barrier thicknesses varied much less and randomly between 2.6 nm and 3.6 nm. The spectroscopic measurements were performed using a CL-STEM setup described in [166], with the sample kept at ∼150 K using a cold finger during experiments.

In order to achieve highly different CC densities in the sample, both beam current and dwell times per pixel were varied over orders of magnitude: 0.1 to 600 pA and 20 ms to 10 s respectively. The experimental beam current was measured using the EEL spectrometer as a Faraday cup.

The aim of the study was to investigate how CC density influences the emission peak properties. Under the QCSE interpretation, two possible scenarios are possible for a QDisk and some electron beam current:

1. The beam induces CCs at a rate lower than the recombination, and emission at constant energy and proportional to the current is measured.

2. The beam induces CCs faster than the recombination rate and electrons and holes start to accumulate. With high CC densities not only high-order effects become more pronounced, but also the free CCs partially screen the internal electric field, resulting in two separate measurable effects:

(a) QCSE gets "undone", effectively blueshifting the emission.

(b) Electron and hole wavefunction overlap increases, increasing the emission intensity.

If the electron beam currents allow a probing of the second regime, the emission energy can be used as a proxy to monitor the CC density while observing the intensity variations. A transition from constant to intensity-dependent emission energy marks the break point between the two regimes, allowing an estimate of the recombination rate.

7.2 Analysis and Results

As the electron probe scanned the specimen, a smoothly varying CC density was induced at any one point on the sample. This led to emission peaks smoothly appearing (due to different QDisks being hit with different intensities by the probe) and shifting in energy. As ML methods are not suitable for tracking smooth changes in energy space, the analysis was performed by fitting a Lorentzian peak in the wavelength domain.

The main analysed dataset consisted of ten spectral images (SIs) of the same NW acquired at different currents and dwell times. Due to vastly different experimental parameters, some SIs were of significantly worse quality than others. In order to provide an unbiased collection of fitting results, we used the SAMFire algorithm (chapter 5) to facilitate changing models, robust fitting suitable for different data quality, and a large number of spectra to fit.

The first experimental evidence of a possible link between carrier density and the emission energy and intensity can be seen from a separate NW with an isolated emitting QDisk measurement, shown in Fig. 7.3(a-d). Pixels within the corresponding total intensity windows in (b,c) were extracted and spectra averaged in (d). The expected trend of higher intensities corresponding to higher energies can be seen. As the corresponding selected pixels tend to be further from the geometrical centre of the QDisk, such an effect could also be explained by different energy emission of different parts of the QDisk. To disprove such a possibility, a virtual dataset was formed from the ten SIs of the main dataset. By extracting the pixels closest to the geometrical centre of QDisk#10 across all SIs with different beam energies, a similar behaviour was recovered, shown in Fig. 7.3(e). Individual measurements of energy and intensity of the emission from all parts of QDisk#10 across all ten SIs are shown in panel (f), with no signs of beam damage. For further verification, collaborators performed theoretical simulations of energy-intensity relations for previously measured NWs [116], supporting the QCSE interpretation.

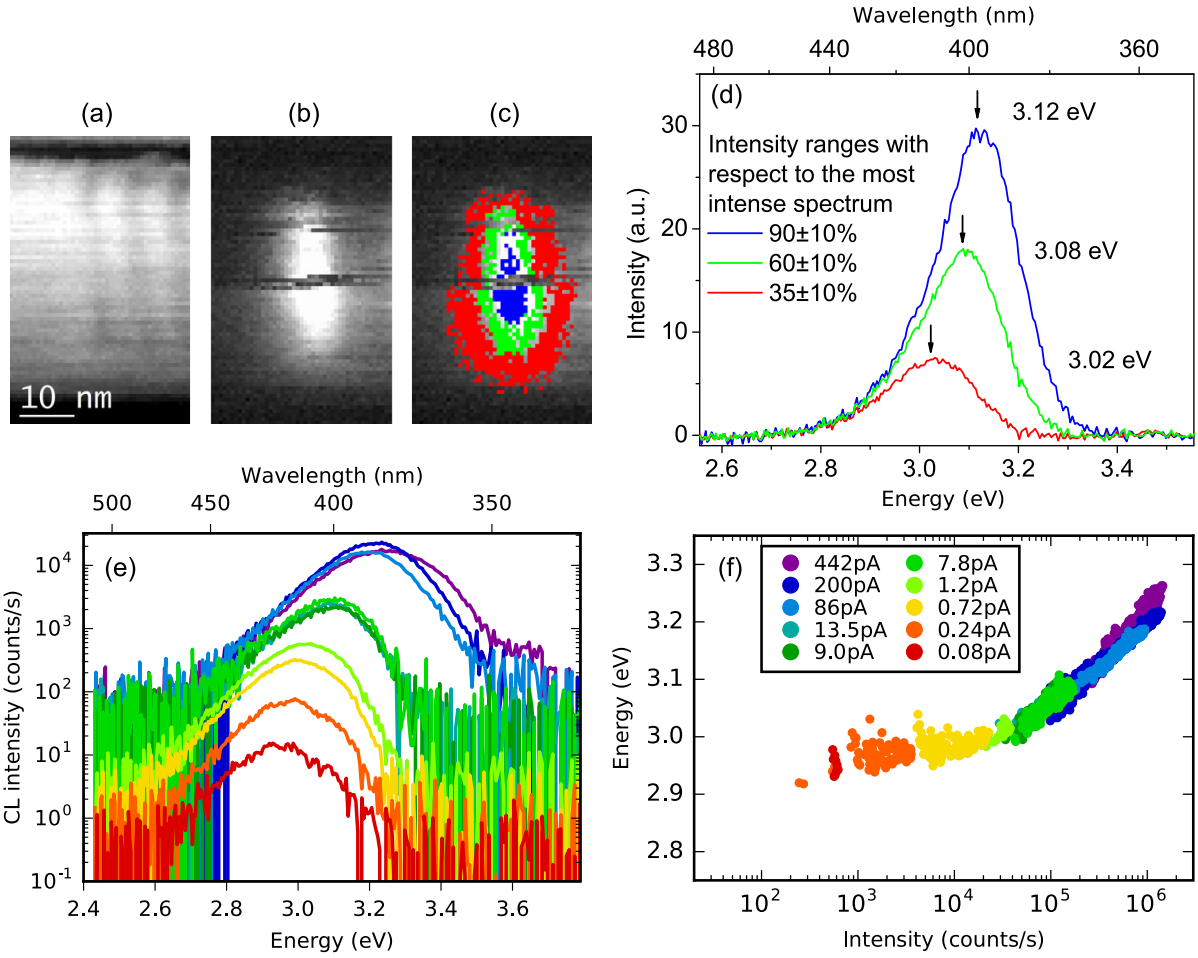


Fig. 7.3 (a) ADF image of a NW containing two QDisks. (b) CL emission map for the same region. Only one of the QDisks emit. (c) Regions of the map with emission in the corresponding intensity windows, and (d) the averaged lineshapes for those regions. (e) CL intensity from the geometrical centre of QDisk#10 with varying beam currents, formed in post-processing. Similar emission energy–intensity behaviour is recovered. (f) Emission energy as a function of emission intensity from all parts of QDisk#10, considering all ten SIs. Colours in (e,f) indicate different electron beam currents.

The fitting analysis results of seven selected QDisks from the ten SIs are presented in Fig. 7.4. QDisk#6 displays a flat energy-intensity region, where the emission energy stayed roughly constant with the intensity increasing over two orders of magnitude. Within the QCSE interpretation this corresponds to carrier injection being lower than the recombination rate, and the "break point" in the curve at $\sim 3 \times 10^4$ counts/s allowed us to estimate the recombination lifetime to be $\sim$100 ns [116]. This was consistent with similar results in literature [159]. The estimated energy-intensity curves for QDisks#8-10 show a maximum intensity of $\sim 2 \times 10^6$ counts/s that that was measured at multiple

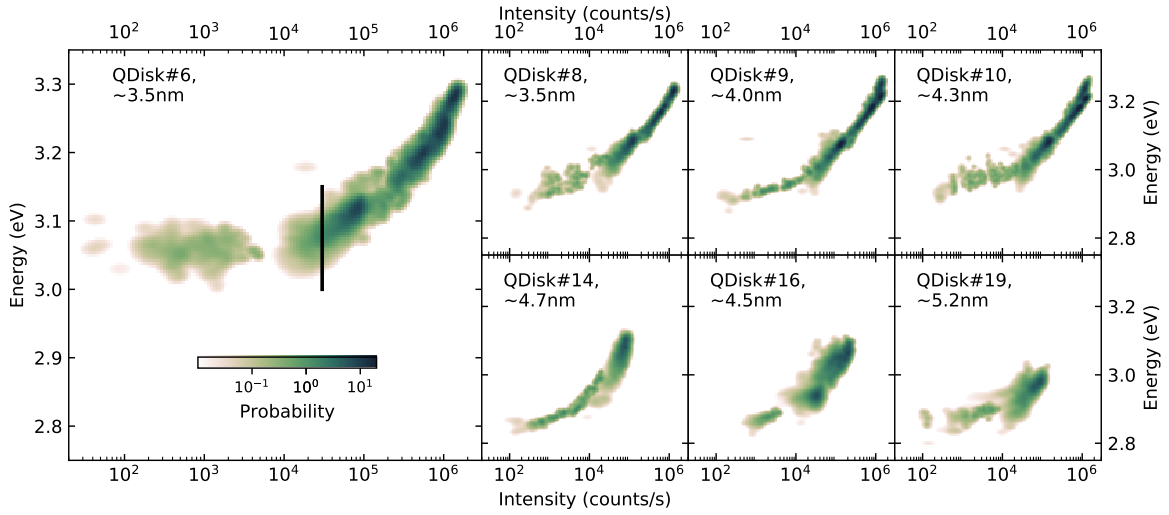


Fig. 7.4 Bivariate histograms of emission energy as a function of the emission intensity for seven QDisks, with different electron beam current measurements combined. Each data point has been spread according to its standard deviation. This allows interpreting the shown colours as probability of finding a spectrum with particular emission energy and intensity. Values larger than 1 correspond to more than one such spectrum in the full dataset, thus the sum of the images equals the total number of spectra considered, 15790. Only QDisk#6 shows a flat region, where energy stays approximately constant with increasing intensity. The "break point" (indicated by a line) allows to estimate total recombination rate.

excitation energies and with different driving currents, as more explicitly shown for QDisk#10 in Fig. 7.3(f). As the internal electric field (via the QCSE) was the only factor changing the emission wavelength, this upturn shows that increasingly larger CC densities were created during the measurements, however the emission intensity did not increase, leading to the first sign of the efficiency droop. Finally, QDisks#14,16,19 show that thicker QDisks contained more non-radiative paths for the CC recombination, hence significantly reducing the overall emission.

Band filling was considered as one of the possible explanations for the constant intensity with increasing injection rate. Previous photoluminescence studies of similar structures found that when the electronic bands are full, the FWHM of the emission increases and the energy saturates [167]. To compare, emission properties from geometrical centres of the QDisks in question were carefully extracted. Fig. 7.5 shows the energy increase with current with the FWHM varying little over the range.

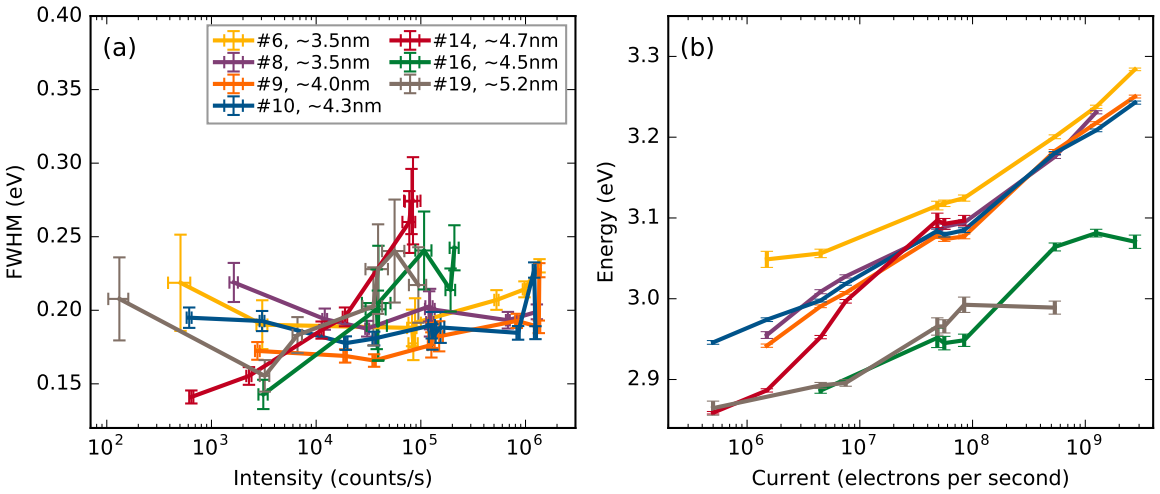


Fig. 7.5 (a) FWHM as a function of intensity of the emissions from the centres of the QDisks. Little variation can be seen for most QDisks across over three orders of magnitude of intensity. Although with decreasing estimation confidence, QDisks #14 and #16 show some systematic increase in FWHM. (b) emission energy as a function of the beam current, with no consistent saturation energy.

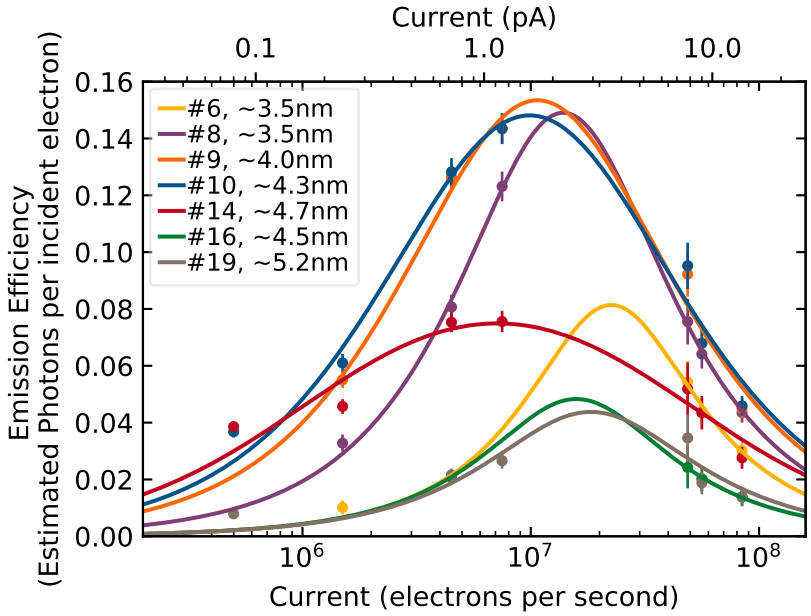


Fig. 7.6 Estimated emission efficiency with the beam hitting the geometrical centre of each QDisk. Lines correspond to fitted ABC models. A drop in efficiency past 1 pA current can be clearly seen, reminiscent of the droop in nitride LEDs.

External Quantum Efficiency (EQE) is an important measure for optoelectronic devices, in this case corresponding to the number of emitted photons per incident electron. Emission rates with the beam directly hitting the geometrical centre of the QDisks were extracted and then converted to photons per second. The calculations of conversion factor to estimate emitted photons per measured counts are explained in more detail in [116]. Considering all factors, the sample emitted approximately 30 photons per detected CCD count. The measured EQE factors for various electron currents are shown in Fig. 7.6, indicating that 7 to 100 electrons are required to generate a single photon. A clear droop in emission probability can be seen above 4×10^7 electrons per second. It cannot be attributed to the QDisks changing due to irradiation damage, as the process is reversible.

A similar and well researched droop may be observed in nitride LEDs. To explain it, the so-called "ABC" model is used [161], shown as fitted lines in Fig. 7.6. With no CC leakage due to the AlN barriers, three mechanisms contribute to EQE with different CC density dependencies. Non-radiative defect contributions, represented by the A term, are modeled to have linear dependency on the density n: dominant at low, but negligible at high n values. The B term usually denotes the radiative efficiency. It is considered to have n^2 dependency, however it has been shown to tend towards n^1 at high densities [168]. While this prevents theoretically perfect EQE, it cannot account for the observed droop [161]. Finally, the C term in the model represents the detrimental Auger effect contribution with n^3 dependency. It becomes dominant at high n values, successfully explaining the observed efficiency droop.

It is much more difficult to apply such analysis to the measured QDisks due to the internal electric field screening, which changes the wavefunction overlap, adding extra complexity to A and B terms in the model. Nevertheless, the model can be used as a framework for reasoning about the cause of the measured effect. In particular, the changes in interaction probability are supposed to be the same for both non-radiative (A) and radiative (B) contributions, and thus are not able to account for the droop. What is left is the C term of the model, the Auger effect, supported by similar findings in LEDs [160].

Fig. 7.6 also allows estimating emitted photons per second. In particular, it shows that these systems could not emit at higher than the measured 10^7 photons per second no matter how strongly driven, with the maximum EQE achieved at roughly 1 pA. Considering that the electron velocities in SEMs are significantly smaller and interactions are stronger, it is likely that many conventional CL-SEM setups create much higher carrier

densities even with low currents, thus operating in the droop regime and overshooting the optimal efficiency.

7.3 Conclusions

A nanowire with multiple GaN QDisks separated by AlN barriers was investigated measuring the CL emission in a TEM across four orders of excitation intensity magnitude. The measured emission energy–intensity relations were investigated with 1 nm spatial resolution for seven QDisks, supporting the QCSE interpretation. The emission efficiency droop was measured to be present with beam currents above approximately 10 pA, and tentatively attributed to the Auger effect.

Chapter 8

Quantifying elemental and bonding maps in 3D in a TEM

This chapter includes work that is prepared for publication as:

> Francisco de La Peña, Tomas Ostaševičius, Rowan K. Leary, Caterina Ducati, Paul A. Midgley, and Raúl Arenal. Quantitative three-dimensional elemental and bonding mapping of a complex hybrid nanoparticle

In particular, RA and FdlP conceived the experiment, RA, FdlP and RKL performed the tomo-EELS measurements, TO came up with the fingerprinting algorithms, TO fitted the spectra, FdlP and TO performed the EELS quantification, FdlP and RKL performed the 3D-CS tomography.

Nanoparticles and other nanostructures have been a major part of many research fields due to their potential to have a high impact on our lives – from medical applications [25], to LEDs [153, 170]. The ability to efficiently determine nano material properties and structure is of paramount importance. The behaviour of such materials is controlled not only by their chemical composition and electronic state (bonding), but also by their shape and size. As a result, measurement techniques able to provide all the required information are of great interest.

The Transmission Electron Microscope (TEM) has been the cornerstone of nanocharacterization because of its ability to quantify chemical composition and determine specimen morphology with atomic spatial resolution [1]. In particular, the most general way to determine the atomic species that constitute the sample involves either energy-dispersing

the emitted X-rays, or measuring the electron energy loss spectrum (EELS) as the electron passes through the sample.

However, as in bulk materials, in order to fully describe the particle and its properties, the chemical composition alone is often not enough. An experiment that not only determines the positions and species of atoms, but also their immediate surroundings is significantly more useful. There have been two ways to attempt to extract such information. The first involves atomic resolution scanning TEM (STEM), where each atom species is determined from the measured signal (either high-angle annular dark field (HAADF) [171, 172] or EELS [4]). For such an experiment, the specimen should be very thin, often just tens of layers of atoms, making the technique less than ideal for morphologically-complex large particles.

The alternative measurement uses STEM-EELS, which is capable of both absolute quantification and determining the chemical composition, all without the need to use standards [4]. For the local atomic surroundings, it uses the fine EELS spectral features in the few tens of eV following the elemental edge onset. The so-called energy loss near edge structure (ELNES), described in section 2.1.2, can be directly related to the local density of states of the measured atom. Importantly, the ELNES features stay present and meaningful even if the spatial resolution of the EELS map is significantly worse than atomic. This opens a way for a much faster nanocharacterization, where the necessary experimental spatial resolution is determined only by the specimen morphology.

Finally, the 3D morphology for sufficiently complex structures has been shown to be of key importance for their properties. Even though the STEM-EELS measurement produces two-dimensional elemental or bonding maps, electron tomography [173] provides a way to reconstruct 3D information. Such EM tomography of individual nanoparticles is usually performed by measuring the quantities of interest in as many different directions as possible, often with 1° steps. Many samples are not able to withstand such radiation damage without significant changes, breaking the key assumptions of tomography [173]. Compressed-sensing (CS) techniques have been used to successfully side-step this requirement by reducing the number of required projections by using various assumptions about the information content of the projections [52, 174, 175].

8.1 Methods

To the author's knowledge, there are just three examples of bonding tomography in the scientific literature [21, 176, 177] to date. Jarausch [21] investigated the 3D distribution of silicon oxidation states in a cylindrical nanopillar of diameter ∼200 nm (Fig. 8.1).

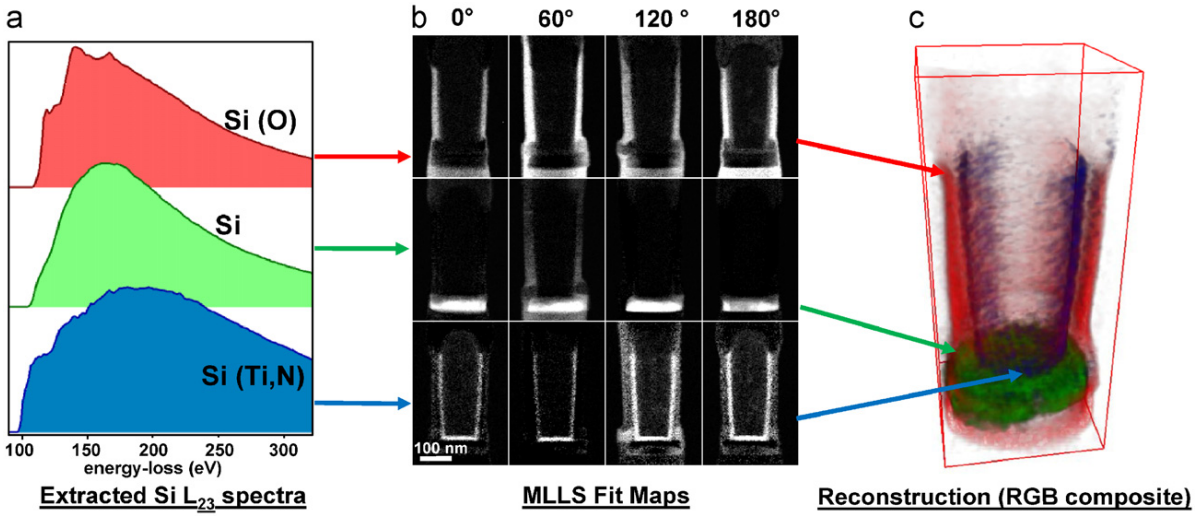


Fig. 8.1 (a) Extracted fingerprint EEL spectra, (b) corresponding fitting results and (c) tomographic reconstructions from the said results. Figure from [21].

This allowed the researchers to assume that multiple scattering was approximately constant throughout the dataset and use the traditional curve fitting approach to extract qualitative tilt-series by fitting oxidation states with respective fingerprints. The signature EELS signals were estimated from selected areas of the specimen, where each state was dominant. This step again relied heavily on the fact that the effects of multiple scattering were approximately identical across all spectra. The bonding maps are the input for the tomographic approach of choice, reconstructing each state 3D distribution independently.

The second approach, shown by Goris [176], follows a significantly more computationally demanding route. The specimen of interest was composed of ∼10 nm ceria nanoparticles, where multiple scattering effects were negligible (Fig. 8.2). One tomographic reconstruction per energy channel of the full recorded EEL spectrum was performed, such that their combination allowed extracting a spectrum from any voxel. Fitting those spectra with a linear combination of oxidation state fingerprints from thin standards allowed an estimation of their abundance in 3D. This approach is not only exceptionally computationally demanding (and hence subject to tomographic reconstruction artefacts more than others), but also applicable only to specimens where multiple scattering, to a good approximation, is not present.

The final ELNES quantification, presented by Torruella [177], reconstructed the 3D abundance distributions of iron oxidation states in a 40 nm cubic core-shell nanoparticle (Fig. 8.3). The corresponding tilt-series maps in this case were extracted from the full EELS dataset using a blind source separation machine learning technique without using standards or references. While such an approach has the advantage of learning the

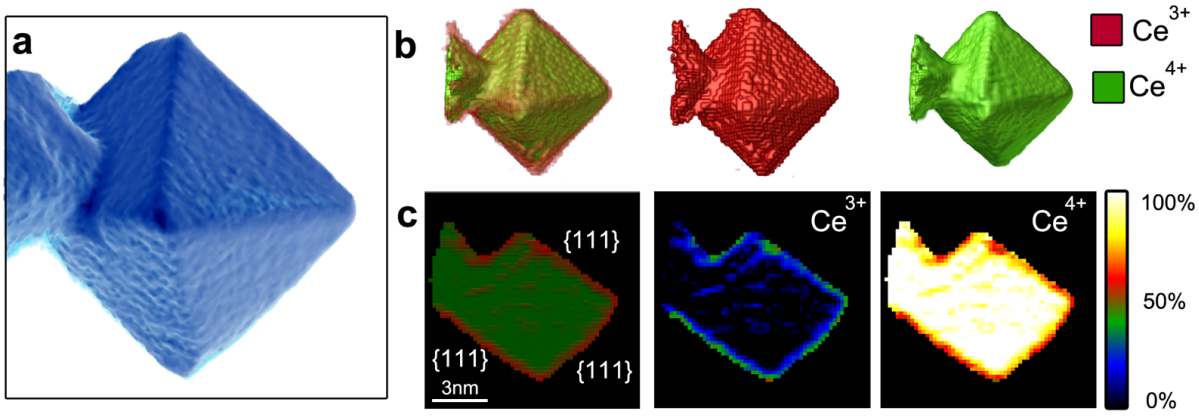


Fig. 8.2 (a) HAADF-STEM reconstruction of the investigated particle, (b) reconstruction visualizations of ceria with different valency, and (c) slices through the volumes in (b). Figure from [176].

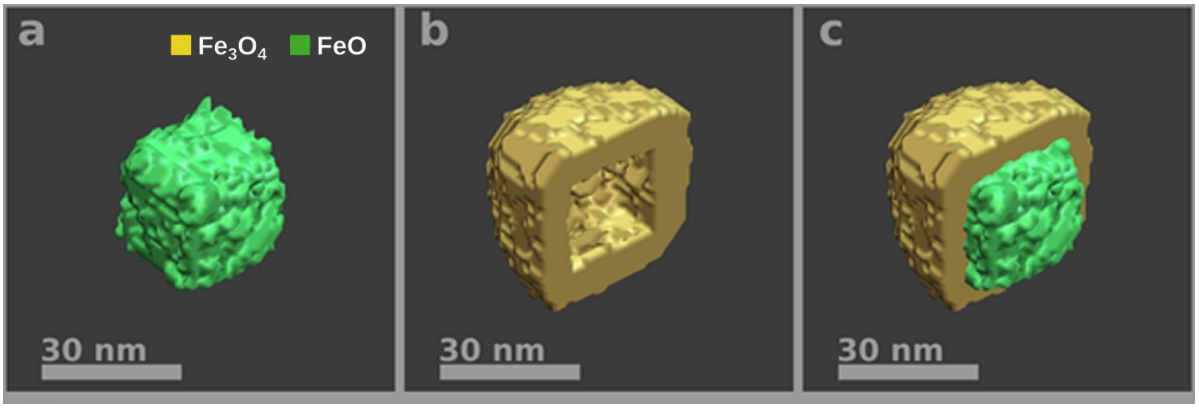


Fig. 8.3 3D surface visualization of iron nanoparticle core and shell, with (a) only core, (b) only shell and (c) both core and shell visible. Figure adapted from [177].

fingerprints automatically, it is applicable only in cases without significant multiple scattering. The 3D reconstructions were performed using a compressed sensing (CS) electron tomography algorithm, with additionally imposed mirror symmetry to double the number of effective tilts for the reconstruction.

It is important to note that in all three examples the final intensity was in arbitrary units. Furthermore, in all cases multiple scattering was assumed to be either negligible or constant throughout the specimen, hence significantly restricting the applicability of these techniques.

This chapter describes how a combination of the quasi-simultaneous acquisition of low-loss (LL) and high-loss (HL) EEL spectra and novel analysis techniques has enabled truly quantitative analytical tomography without the need for standards and not limited

to specimens with constant or weak multiple scattering. First, after pre-processing the dataset for measurement artefacts, the fingerprints were extracted using one of the algorithms suggested in section 8.1.1. The procedures do not put any constraints on multiple scattering conditions by using the simultaneously measured LL signal. Then the reference single scattering distributions (SSDs) were fitted as described in section 2.1.2 for the full dataset. As the resulting model contained multiple elements, each with multiple corresponding SSDs (up to four), the SAMFire algorithm (presented in chapter 5) was used to increase the stability of the solutions and ease the analysis. The resulting intensity maps can be directly converted to atoms with the particular oxidation state per pixel area. In addition, curve fitting allowed a calculation of an estimate for the errorbars of the results. The 3D reconstruction was performed using a CS tomographic algorithm, explained in section 8.1.2, capable of using both the tilt-series and the corresponding confidence weights, calculated from the errorbars that were estimated whilst fitting. Therefore the resultant 3D distributions are measured in numbers of atoms of particular species and oxidation state in each voxel, an example of state-of-the-art quantitative analytical tomography of highly complex nanoparticles.

8.1.1 Extracting "fingerprint" spectra

As the thickness of the investigated particle varied from just a few to almost 100 nm, multiple scattering contributions changed drastically throughout the area of interest. As a result, conventional blind source separation methods to extract the corresponding "unmixed signals" were unsuitable and a more robust algorithm was sought.

We propose two curve fitting based approaches to extract SSDs from EELS measurement maps, where HL and LL signals are available. Neither algorithm uses any form of deconvolution, but instead relies on modelling an already broadened SSD, as will be explained in more detail in section 8.3, eq. (8.5). Both start with preparing the datasets by removing various measurement artefacts, such as beam energy drift, X-ray spikes, removing the detector afterglow effects and normalising detector pixel gain. Both the LL and HL signals are de-noised using Principal Component Analysis (PCA) and examined. The LL signal (or EELS maps) will be used as an effective point-spread function for each of the SSDs, and thus should be as artefact-free (both PCA and measurement) as possible in both energy-loss and energy-gain regions of the spectrum. In addition to ensuring that the HL spectra are artefact-free in the energy windows of interest, both the cleaned data and PCA results are used as a guide to identify the elements and their oxidation states. While the elements present can also be determined by examining the

total average spectrum, the different oxidation states would overlap and all real-space information would be lost.

After the real-space positions of the different oxidation states are estimated, the SSDs are ordered in a list in such a way that allows unambiguous fitting if following the order. That is, for each SSD in the list, in the energy range of the particular SSD and in its associated real-space pixels (or just a subset of pixels), all other bonding states present are already estimated and above in the list. Any known compounds found in the specimen can be incorporated to help create the list.

The first proposed analysis approach makes use of specific favourable conditions, where a particular spectrum (with the appropriate edges from above in the list subtracted) is assumed to be very close to the required SSD convolved with the corresponding LL spectrum. These pixels can be just adequate estimates. By fitting the convolved SSDs to their respective spectra, a library of first guesses is formed. If required, the next two steps can be repeated for possibly increased accuracy. First, the edge intensity maps for the full dataset (or just the first EELS map in the series in this case) are estimated by fitting the library of SSDs and convolving appropriately. Then the maps are used to determine the pixels that have the most favourable conditions to be used to fingerprint each of the edges, and the SSD library is updated with the new estimate. For example, to determine the boron in a BN fingerprint SSD, a pixel with the most nitrogen (for BN) and, for example, the least oxygen (for BO/B_2O_3) could be selected.

The second approach does not rely on finding particular "magic" pixels in the EELS dataset. Instead, all spectra that satisfy the conditions for the previously created fingerprint list are used, with the assumption that if multiple scattering was not present or deconvoluted [4], the mean spectrum would correspond to the required SSD. To achieve this, two optimization problems are solved at different scales simultaneously over the particular subset of pixels. The "outer" loop optimizes the particular SSD in a global way, where the outer optimization parameters ("superparameters" further) are constant throughout the map. For each goodness of fit evaluation in the outer loop, the inner optimization is run at least one full cycle, where each pixel is fitted individually with appropriate convolutions with LL, resulting in intermediate intensity maps for all considered edges. Therefore the solution consists of superparameters that have just one value throughout the fitted dataset, and normal parameters with one value per pixel. With such a scheme, multiple scattering is taken into account by using the LL spectrum on a per-pixel basis, but the appropriate SSD is determined for the full map, hence the region of interest can be selected without much consideration for specimen thickness.

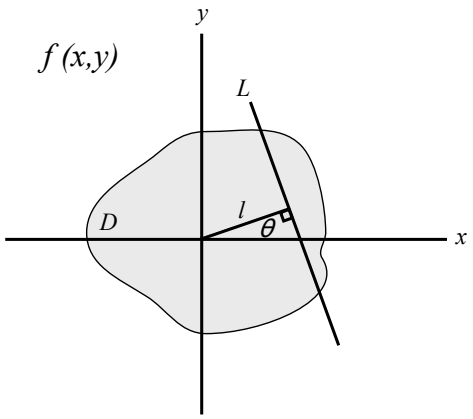


Fig. 8.4 The 2D Radon transform R can be visualised as the integration through a body D in real space $f(x, y)$ along all possible line integrals L with its normal at an angle θ to the horizontal. Figure taken form [173].

8.1.2 Compressed-sensing tomography with weights

Probably the best known examples of tomography in everyday life come from its use in medicine, namely the Computer Assisted Tomography (CAT-scan). The first real application of the technique, however, came from astronomy in 1956 by Bracewell [178], proposing reconstructing a 2D map of solar microwave emission from a series of 1D profiles. Midgley and Weyland [173] reviewed the history, developments and limitations of tomography in electron microscopy, pointing out that the first EM tomography papers came out in 1968. However, relatively few experimental results followed due to important limiting factors that have since been overcome: lack of processing power and goniometer precision.

The mathematical principles behind tomography were first outlined by Radon in 1917 [179], defining the Radon transform, see Fig. 8.4. It described mapping the original real-space function $f(x, y, z)$ to projections in $(\boldsymbol{r}, \theta)$ space via appropriate line integrals. Here θ defines the projection direction and $\boldsymbol{r}$ the position of the integral in the 2D projection. In principle, if some measurement produced a Radon space representation of an object, an inverse of the transform could then reconstruct the real-space structure. In practice, experiments only subsample Radon space (most notably in θ), and hence the reconstruction is always just an approximation. Thus the main challenge is recovering the best real-space reconstruction from the limited measurement data.

A measurement is suitable for use in tomographic reconstruction if it satisfies the projection requirement [173, 180]. In particular, the detected signal should be a monotonic function of the measured phenomenon or material. Such a requirement limits the usefulness of bright-field TEM measurements for EM tomography, as the detected signal

in general is highly dependent on particular diffraction conditions. HAADF measurements, on the other hand, are mostly formed from incoherently scattered electrons, and hence are much better suited for recording tomograms. In this work the EEL measurements were converted to numbers of atoms in particular beam trajectories before the tomographic reconstruction took place, perfectly satisfying the projection requirement. If the EELS signals were used directly, only compounds with an isotropic dielectric function (that is with the detected intensity not dependent on the direction of the swift electron trajectory) would be suitable for a reconstruction, which is known to not be the case for BN [61].

In many cases the difficulty of tomographic reconstruction is directly related to, for example, the number of sampled angles in the original dataset. With finer sampling (<5°), simpler reconstructions such as Filtered Back projection [181] or Simultaneous Iterative Reconstruction Tomography [182] can be used relatively straightforwardly. However, as the tilt step increment increases, the quality of such reconstructions quickly deteriorates beyond an acceptable level. Furthermore, often in EM tomography the measurement system is physically limited to only a subset of angles, contributing to the so-called "missing wedge" problem [173]. To increase the quality of the final result, prior information can be incorporated into the reconstruction algorithm. The field of Compressed Sensing (CS) [174] discovered the mathematical foundations for such approaches in the 2000's suggesting that the object can be reconstructed from a small subset of measurements if the object is sparse in some domain. In this context an image is said to be sparse if most of its pixels are zero, and the fraction of zero coefficients measures the sparseness of the image. Examples include reconstructing a spectrum from a truncated Fourier spectrum (frequency domain) or compressing an image in a JPEG format (discrete cosine transform domain) [183]. An often reasonable assumption in EM tomography states that the true object consists of constant density regions. It can be directly related to sparsity in the gradient of the density, called the total variation (TV) of the object. The final solution is then found iteratively by penalising reconstructions with large TV values:

$$\boldsymbol{x} = \arg\min_{\boldsymbol{x}} \left[\frac{1}{2} \|\boldsymbol{y} - \boldsymbol{P}\boldsymbol{x}\|^2 + \beta_{TV} TV(\boldsymbol{x}) \right] , \tag{8.1}$$

$$TV(\boldsymbol{x}) = \sum_{\text{pixels}} \sqrt{\sum_i (\partial_i \boldsymbol{x})^2} \, , \tag{8.2}$$

where $\boldsymbol{x}$ is the reconstruction, $\boldsymbol{P}$ is the projection operator and $\boldsymbol{y}$ are the measurements. Here β_{TV} is a constant that controls how strongly the TV regularisation is taken into account. In this work in particular we used a modified version of PyHST2 [175]. The

main minimization problem in eq. (8.1) was extended to support 3D reconstructions and weighting each projection by its error $\boldsymbol{\sigma}_{\boldsymbol{x}}$:

$$\boldsymbol{x} = \arg\min_{\boldsymbol{x}} \left[\frac{1}{2} \left\| \frac{\boldsymbol{y} - \boldsymbol{P}\boldsymbol{x}}{\boldsymbol{\sigma}_{\boldsymbol{x}}} \right\|^2 + \beta_{TV} TV(\boldsymbol{x}) \right] . \tag{8.3}$$

In contrast to often previously used "slice-by-slice" algorithms, this allowed not only to perform the reconstruction on the full volume all at once, but also take into account the confidence of each quantification result.

8.2 Specimen and experiment

The specimen in this study was a boron nitride core-shell nanoparticle, often called a "cage" or "nano-cocoon". Such particles are a by-product of one of the BN nanotubes manufacturing processes and are believed to play a significant role in their growth [61, 184]. In particular, the samples were synthesized using a laser vaporization technique described in [170, 184]. Briefly, an h-BN target was vaporized by a continuous laser under 1 bar of flowing nitrogen. Importantly, the target was not pure and contained 4.5% by weight B_2O_3 that was used as a binder, as well as other impurities such as carbon, silica and calcium. The vapour from the target condenses in the form of soot, which was ultrasonicated in ethanol, and the solution pipetted onto TEM grids with a holey carbon film.

A HAADF image, low-loss and high-loss STEM-EELS tilt-series from an isolated nanoparticle were recorded quasi-simultaneously using a Tecnai Osiris Gatan Enfinium ER spectrometer equipped with DualEELS™. In order to minimize beam damage, the microscope acceleration voltage was set to 80 keV, and the tilt range was $-70°$ to $70°$ with acquisition every 17.5°. A separate HAADF tilt-series was acquired by recording just the HAADF image of the particle before the simultaneous HAADF and EELS measurement, to be used later to correct sample drift during the longer acquisition.

When the goal of the analysis is determination of areal densities of atoms, the size of the probe and the step size of the mappings become particularly important to consider when setting up the experiment. Namely, the diameter of the probe should be very similar to the experimental step size in order to avoid both counting the sampled atoms multilpe times when the probe is much larger than the step, and under-estimating the measured densities, when the probe is much smaller than the step. Unfortunately, the latter effect is also present when the probe is larger than the smallest detectable atom clusters, and should be taken into account when reasoning about the final results.

8.3 Analysis

All measured EELS data was corrected for detector readout noise[1] and dark current[2] by subtracting the exposure-scaled dark current and constant readout noise spectra from the raw values [185]. Acceleration voltage fluctuations and similar effects were corrected by applying a sub-channel energy shift to both spectra on a pixel basis so that the highest zero-loss peak intensity was at exactly 0 eV [101]. Once a noise-free high-loss part of the data was available (described later), the energy channel width and lowest energy were calibrated by comparing detected edge onset energies and their differences for B-K and N-K with previously established values [186].

The HAADF image at 0° tilt as well as EEL spectra from three real-space locations and the total summed spectra are shown in Fig. 8.5. Six different edges are marked at their respective ionisation energies, corresponding to Si, B, C, Ca, N and O. Due to the necessary wide spectral range to detect all these elements, the EELS signal counts were very different at high and low energies, leading to the signal to noise ratio (SNR) ranging from 17 for boron to 0.5 for oxygen. In addition, even the edges with relatively high SNR can be seen to have different ELNES shapes throughout the series, showing additional complexity of the data that has to be unraveled. Fig. 8.6 shows two LL spectra from the shell and core of the particle, corresponding to spots (b) and (c) in Fig. 8.5, respectively.

In order to de-noise the data as described in section 4.1.2, principal component analysis (PCA) was performed on each spectral image (SI) of the series. The important results for 0° tilt are shown in Figs. 8.7 and 8.8. While the component spectra are clearly mixed and not physical, the results help to determine how many fitting components will be needed when performing the quantitative analysis as well as real-space regions where various edges have the strongest and weakest signal. In particular, the decomposition split the BN signals into two components, one of which seems to form a ring around the outer shell (number 5), while the other is more uniform throughout the particle (number 3). It has been shown previously in [61] that such separation is due to the anisotropy of the BN, where the ELNES shape varies with orientation of the electron beam with respect to the anisotropic axis. The PCA results further confirm that the outer shell BN *c*-axis is perpendicular to the surface of the particle.

[1] Readout noise is a fixed noise value each time a particular pixel is read out. It arises from the conversion of CCD charge carriers to a voltage signal, as well as any further processing. It does not depend on the exposure time. Readout noise is measured by recording a spectrum with the shortest possible exposure.

[2] Dark current arises from thermally generated electrons within the CCD. It increases with exposure time. The dark current reference is measured by recording a spectrum from a long, for example 10 seconds, exposure, but with the beam blanked, from which the readout noise is subtracted.

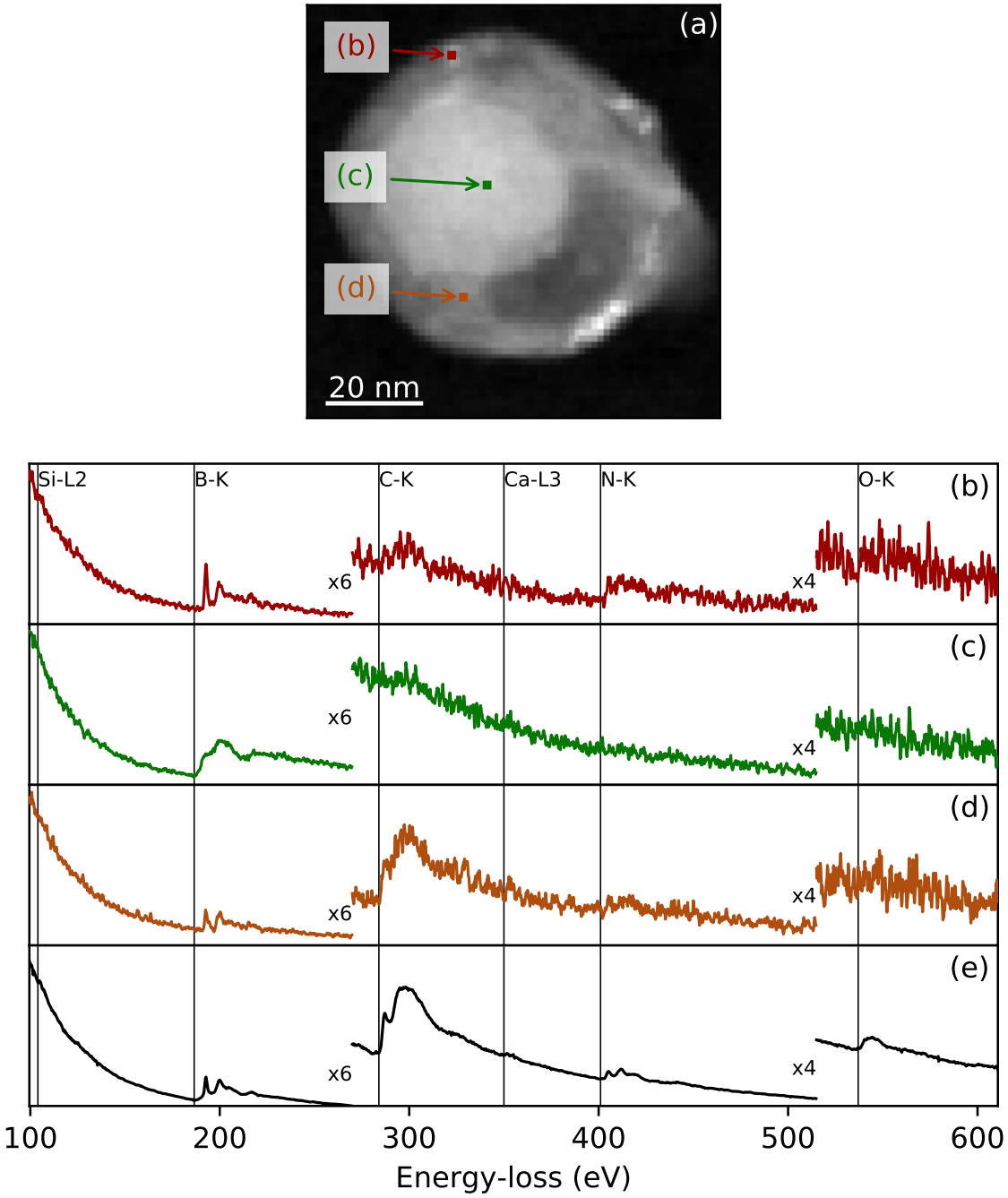


Fig. 8.5 (a) HAADF image of the nanoparticle at 0° tilt. Spectra from three pixels marked in (a) are shown in panels (b-d). Clear B-K ELNES shape differences can be seen, showing sensitivity to the local environment. In (e) the total average over full EELS map spectrum is shown. Edge onsets of the elements are marked with labeled vertical lines. The data was multiplied by 6 and 4 at 270 eV and 515 eV, respectively, for clarity.

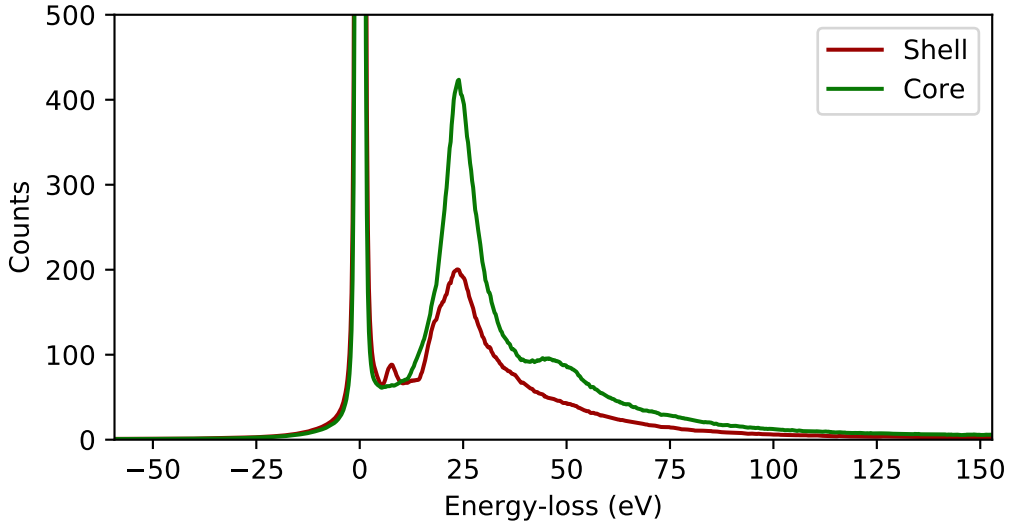


Fig. 8.6 Low-loss spectra from the shell and core of the particle, corresponding to (b) and (c) in Fig. 8.5, respectively. Plasmon contributions (5 eV to 50 eV) significantly increase with electrons passing through more material. Both spectra have been denoised using PCA.

The curve-fitting analysis described in section 2.1.2 requires low-loss (LL) EELS spectra for each real-space pixel with as little noise and few artefacts as possible. The de-noising of the LL part of the dataset was performed as previously described, by using PCA and truncating the component list. Unfortunately, the experimental setup introduced a spectral artefact that had to be corrected. Due to how the DualEELS™ spectrometer operates by acquiring both energy ranges quasi-simultaneously, significant leakage of the high-loss part of the spectrum was visible in low-loss spectra where the HL signal was strong as the beam trajectory intersected the particle, as shown in Fig. 8.9. The artificial energy-gain bump in the LL shape was fixed by first calculating the mean LL spectrum of 810 pixels without the particle and hence the artefact, and then fitting this reference shape to all pixels of the LL part of the signal. Only the spectral range corresponding to the artificial bump, −60 to 0 eV, was replaced by the fitted template. The overall procedure resulted in a LL EELS map for every tilt with very little noise or known de-noising or experimental artefacts. Lastly, the LL signal intensities were calibrated with high-loss. After converting both signals from raw experimental counts to counts per second, at identical energies the two signal magnitudes were different by a factor of (2.663 ± 0.078) due to a combination of experimental factors such as gain change and spectral CCD binning.

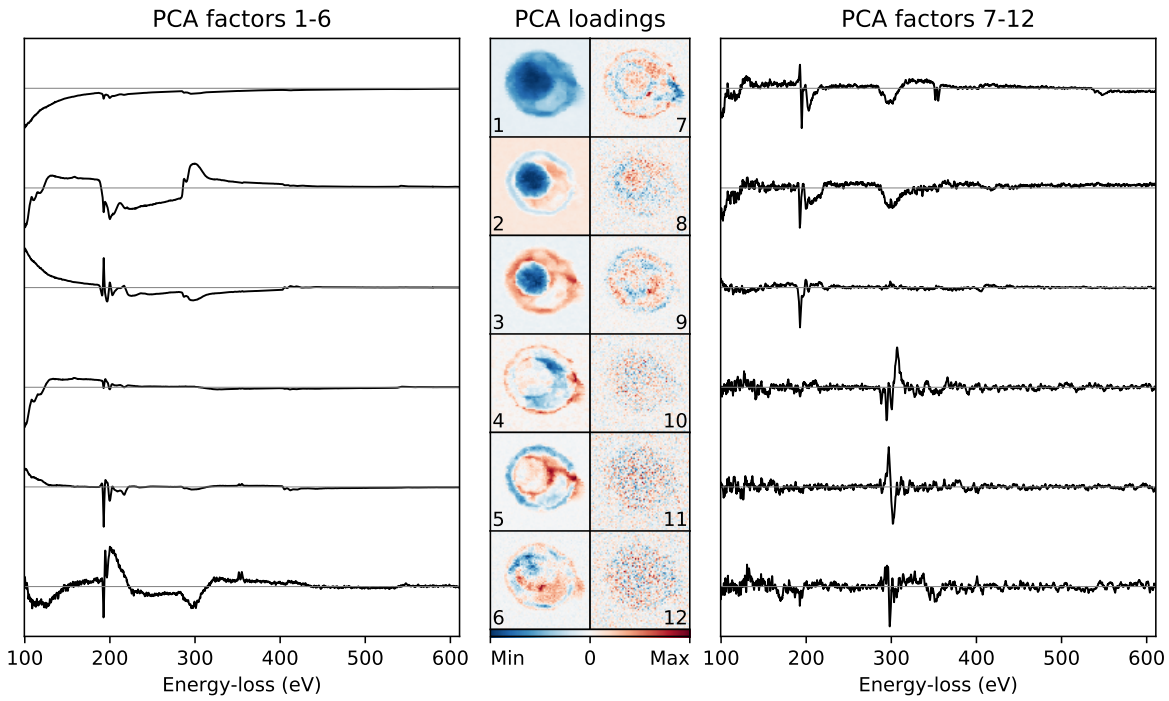


Fig. 8.7 First 12 PCA loadings and corresponding factors. The first 9 components were used to reconstruct close to noise-free EELS dataset.

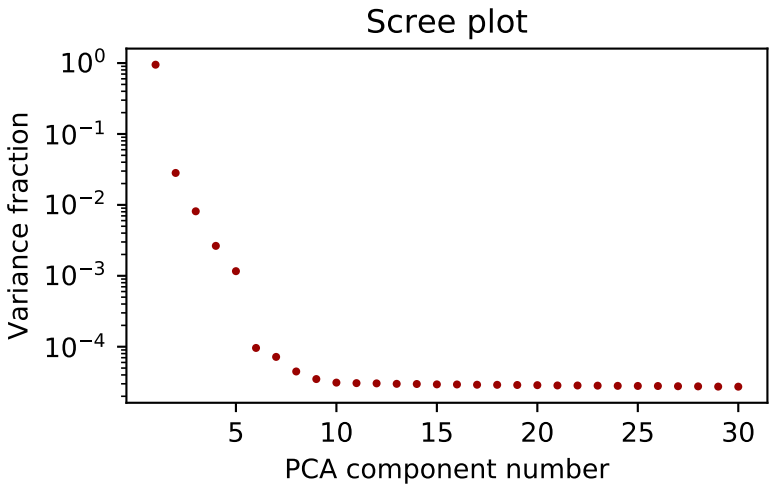


Fig. 8.8 Scree plot, showing the fraction of total variance that is included in each component. The first 9 components were used to reconstruct close to noise-free EELS dataset.

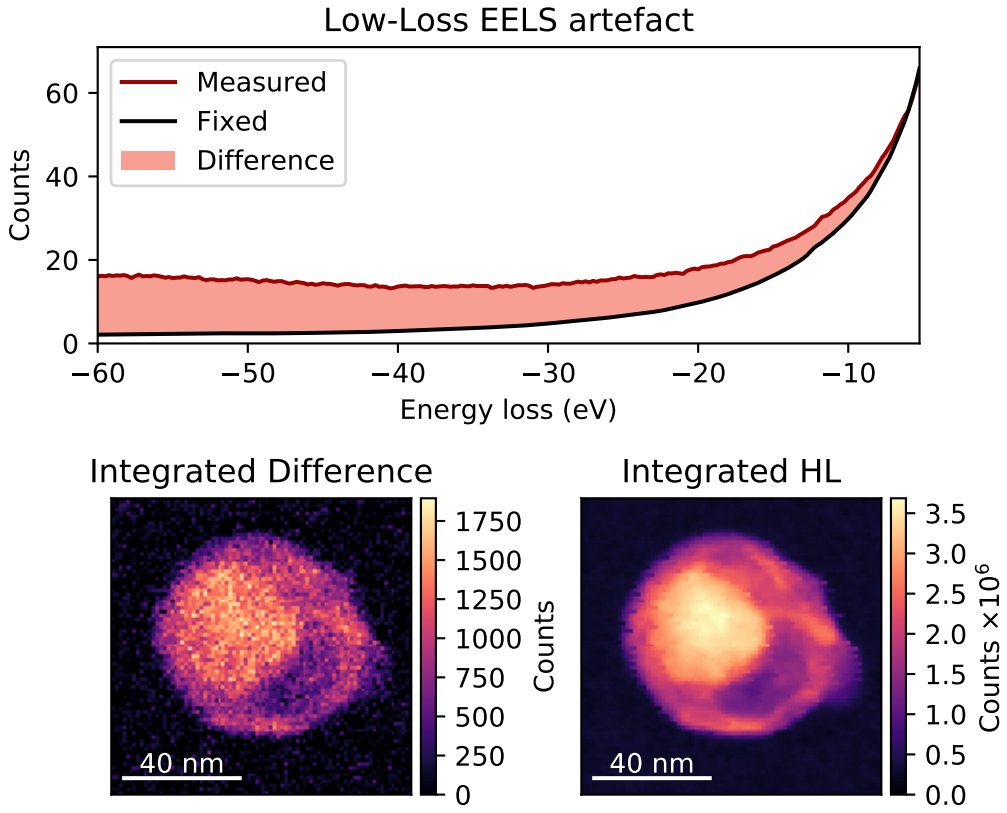


Fig. 8.9 The EELS low-loss spectrum energy gain tail as measured (red) and after template fixing (black) for one pixel are shown at the top. The difference, shown as the shaded region, was an artefact of a high intensity signal from the high-loss part of DualEELS™. The integrated difference for the 0° tilt (left) is proportional to the total measured high-loss intensity (right).

`EELSCLEdge` components from HyperSpy [101] with Hartree-Slater cross-sections [18, 19] were used for fitting as fingerprints unless stated otherwise. Each component was able to model the fine structure intensity modulations by a spline[3] in the first few tens of eV as described by Verbeeck [23]. The fitting method is described in detail in de la Peña's thesis [188]. Briefly, the full model consists of a linear combination of all edges that are present in the energy range and a power law background:

$$J'_{\mathrm{SSD}}(E) = A \cdot E^{-r} + I_{\mathrm{ZLP}} \sum_{i=1}^{n} N_i \sum_{j=1}^{m_1} \sigma_{i,j}(E)\,, \tag{8.4}$$

[3] **A spline** is a numeric function that is piecewise-defined by polynomial functions and which is highly smooth at the places where the polynomial piece connect, called "knots" [187]

where $J'_{\mathrm{SSD}}(E)$ is the fitted core loss spectrum as a function of energy loss E, and I_{ZLP} the zero loss peak intensity, N_i are the areal densities of atoms, n is the number of chemical elements in the energy range, m_i the number of ionisation edges of the i element, $\sigma_{i,j}$ gives the ionisation cross-section of an atom of element i and excitation of shell j, and A and r model the power law background. Eq. (8.4) can be used directly for very thin samples with negligible multiple scattering, but in general these effects have to be either removed from the data before fitting, or incorporated into the model. We model the multiple scattering by not just multiplying the cross-sections by the ZLP intensity, but instead convolving with the LL spectrum $J_{\mathrm{LL}}(E)$ on a per-pixel basis

$$J_{\mathrm{HL}}(E) = A \cdot E^{-r} + J_{\mathrm{LL}}(E) \otimes \sum_{i=1}^{n} N_i \sum_{j=1}^{m_1} \sigma_{i,j}(E)\,, \tag{8.5}$$

This gives the final modeled core loss spectrum. The ELNES structure spline model is included in each $\sigma_{i,j}$. Once a fingerprint was determined, the fine structure modulations were fixed and only relative scaling of intensity N_i (in addition to the effects of the convolution with LL) was allowed.

Ten spectral fingerprints for five elements were extracted as described in section 8.1.1:

- **Boron (4)** for B_2O_3, $BN_{\parallel}$, $BN_{\perp}$ and $B_{\beta-\mathrm{rhombic}}$;
- **Oxygen (2)** for SiO_2 and B_2O_3;
- **Nitrogen (2)** for $BN_{\parallel}$ and $BN_{\perp}$;
- **Calcium (1)** Ca-$L_{2,3}$;
- **Carbon (1)** C-K;

Silica (SiO_2) was found everywhere in the background of the specimen, hence the oxygen O(SiO_2) SSD was estimated first from a background region. Even though carbon was also present in the background, a more accurate SSD estimate was fitted from the middle of the particle, where the C-K edge signal was particularly strong. Afterwards B and O in B_2O_3 SSDs were fitted to the pixels in the region where no nitrogen (for BN) was visible. Even though oxygen SSDs overlap, the silica oxygen edge was already estimated. The process was then repeated for two boron and nitrogen SSDs in BN (one from the edge of the shell and second from the middle, as required by the anisotropy of BN). In each case pixels with the least detected oxygen were sought to make estimations as independent of others as possible. Finally, crystalline boron and calcium SSDs were fitted from the core of the particle.

The Si-$L_{2,3}$ edge is tabulated to start around 99 eV with a delayed onset [189], and due to the experimental setup, the lowest detected EEL in the high-loss part of the dataset was at 99.8 eV. This meant that the data contained insufficient information for the fitting algorithm to accurately fit both the exponential background and the edge, thus a Si edge fingerprint was extracted from parts of dataset that did not contain the particle using machine learning methods used in the previous studies [61]. Assuming the support and silica layer were sufficiently thin, the fingerprint was assumed to be a good approximation of the silicon SSD multiplied by a scalar factor.

Once all fingerprints were determined, noisy calibrated data from each experimental tilt was fitted using the SAMFire algorithm described in chapter 5. The analysis was performed using a least-squares optimizer MPFIT [74], which also returns the standard deviation of the solution. In order to minimize the errors of the power law background model over large energy ranges [4], each tilt was fitted in energy windows. For example, 150-250 eV energy window was used for all boron edges, 250-275 eV for carbon and calcium, and so forth.

One fitted spectrum with highlighted components is shown in Fig. 8.10. All edge components except C-K were tripled in intensity for visual clarity. Note that for both boron and nitrogen in BN the total corresponding signal is encoded in a two-dimensional ("perpendicular" vs "parallel") space. To access both dimensions, two components were fitted to the required fingerprints. Nevertheless, the two SSD pairs can only be interpreted as physical when summed, which is how they are shown in the figure. In addition to the high degree of control over the model, fitting also allows us to estimate the standard deviation of the result, shown in the figure as a darker region around the final fitted spectrum.

Finally, performing tomographic reconstruction using CS requires tilt-series maps without significant backgrounds. As carbon and silica were found everywhere, three background tilt-series had to be estimated and subtracted. It was further complicated by beam damage to the substrate, leading to holes right next to the particle at the later (high angle) tilts. First, a binary mask of the particle at all tilts was created to define the boundary between background signal that was visible (outside) and underneath the particle (inside), shown in Fig. 8.11. The masked region was iteratively filled in from the boundary inwards. The estimated background values for each iteration were calculated using weighted mean with weights proportional to the inverse distance squared [190].

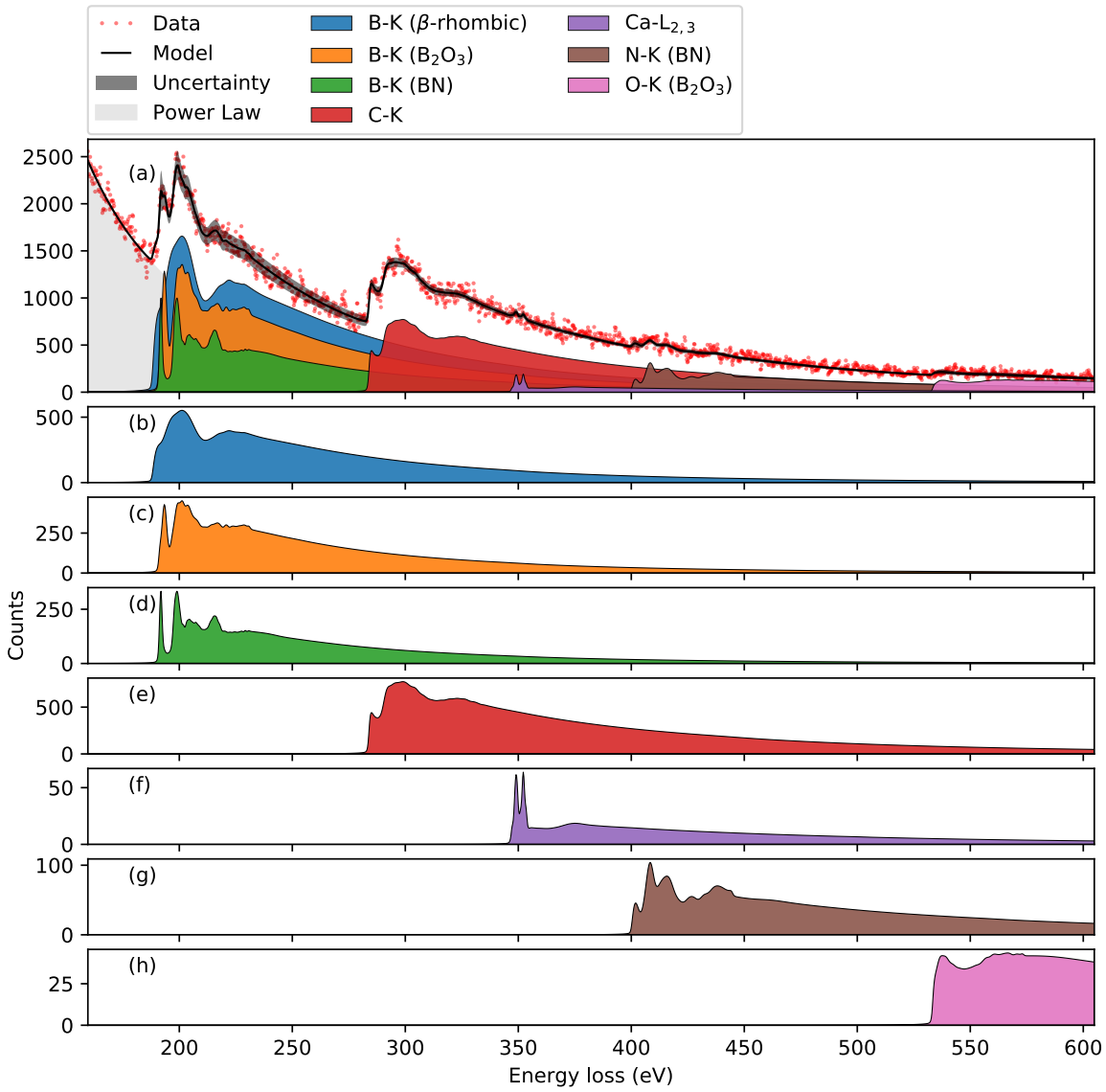


Fig. 8.10 A fit for one EEL spectrum (a) with individual edge components (b-h). In (a) all edges except carbon C-K were tripled in intensity for visual clarity. The data is shown as red dots, and the fitted model as a black line. The dark shaded region around the model marks the uncertainty of the fit. The fitted pixel did not contain any silica, thus the two relevant edges are not shown.

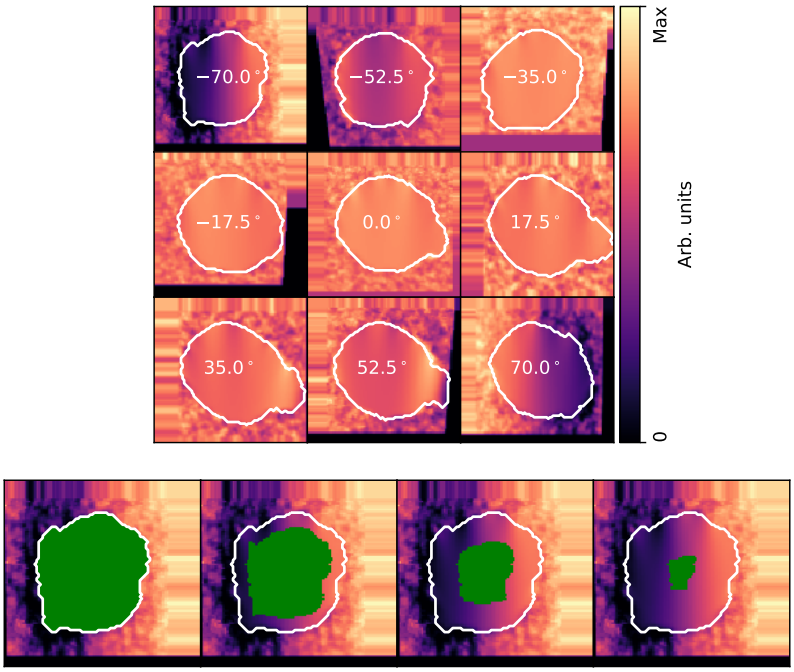


Fig. 8.11 Estimated backgrounds for the carbon C-K fit result tilt series. Tomographic alignment artefacts around the edges of the maps are intentionally shown. Outline of the particle mask is shown in white in each corresponding tilt. Three iterative background filling-in steps are shown in the bottom for the −70° tilt.

8.4 Results and discussion

The quantification maps for the first tilt (0°) in the tilt-series are shown in Fig. 8.12. Nine independent tomographic reconstructions, one per bonding or elemental map, were performed following the method outlined in section 8.1.2. If the amplitudes of the fitted tilt series are proportional to the number of atoms per nm^2, then the reconstructions, by definition, are proportional to the number of atoms per nm^3.

Fig. 8.13 shows composite particle reconstruction iso-surfaces for five of its elements. Due to measurement errors being used as weights, high fidelity reconstructions were achieved with only 9 experimental tilts and no favourable symmetries. The particle is best explained using the simplified scheme on the right of Fig. 8.13. In particular, the crystalline boron forms the ∼38 nm diameter core (red) surrounded by carbon (grey), enclosed by boron nitride as the main ∼5 nm thick shell (green). Two cavities were found in the carbon. The first one corresponds exactly to the crystalline core, the second, while of similar volume, was reconstructed to be almost empty. The BN shell is not continuous and has well reconstructed holes. Carbon was reconstructed to “leak” through the shell

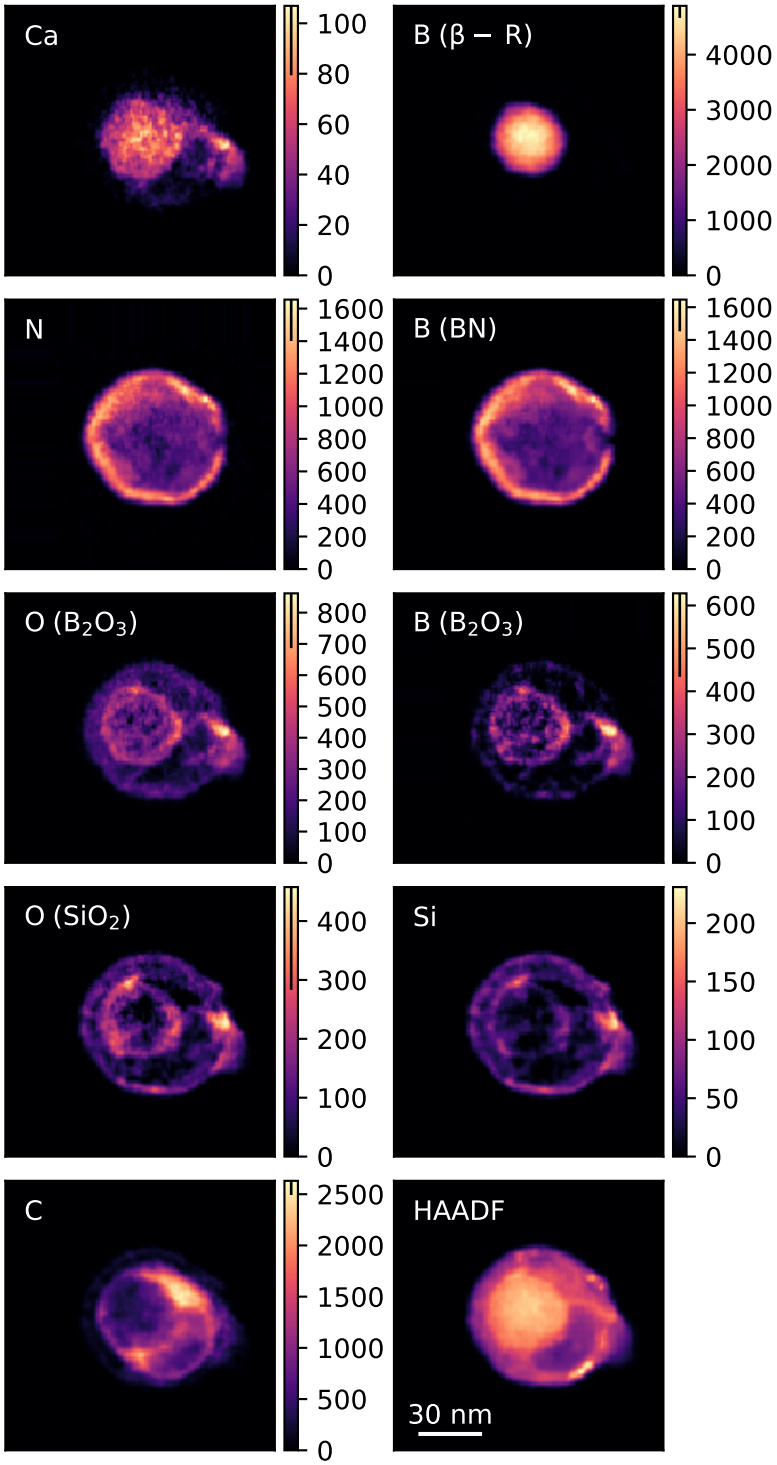


Fig. 8.12 Quantification maps and HAADF image at 0° tilt. The colourbars are in atoms/nm^2. Colourbars include a vertical line with the length of the magnitude of the errorbar for the largest intensity in the map. The error estimation was performed during the curve fitting, and thus is not available for Si, which relied on machine learning methods.

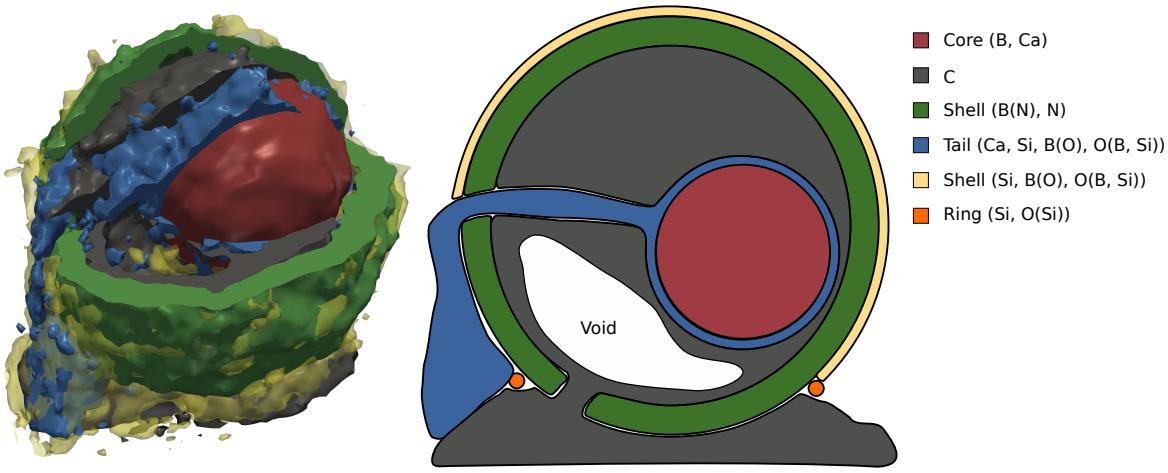


Fig. 8.13 Reconstructed elemental iso-surfaces on the left and a simplified scheme of the investigated composite particle on the right. The ring structure in the experimental results emerges only when considering the absence of boron in the corresponding voxels, and thus is incorporated in the yellow (Si) isosurface. For clarity, the scheme shows the ring separately. Each colour in the scheme corresponds to a mixture of elements indicated in the legend. To decrease the figure complexity, only iso-surfaces of the first element in each colour are shown. The corresponding atoms/nm^3 for the surfaces are: 18, 20, 18, 1.2, 1.2

gap nearest the support and form a disk around the base of the particle. The interface between the crystalline boron core and carbon surrounding it was reconstructed to contain a thin shell of oxygen bonded with both boron, silicon and calcium, shown in blue. The oxides also form an elongated continuous ∼6 nm diameter wire-like structure from the core via a well reconstructed path in the carbon and finally through a different gap in the BN shell, where it "flows" on the outer surface of the shell towards the support. For the sake of simplicity, this elongated structure will be called "the tail". Trace quantities of calcium were reconstructed on the walls of the nearly empty void in the carbon, as well as in both the core and the tail. Two more structures are observed in the reconstructions. First, the BN shell is encapsulated in a very thin (at the detection limit) outer shell containing boron, oxygen and silicon (yellow). Secondly, a thin (∼3.5 nm thickness) ring of silica (SiO_2) is reconstructed on top of the carbon ring that "leaked" from the BN shell and formed a support at the bottom (orange).

Slices through the nine reconstructed volumes are plotted in Fig. 8.14, showing distinct separation of the core, the inner shell, the carbon filling, the main BN shell and the thin outer oxide shell. In addition, a gap in the BN shell with corresponding higher intensities in the tail element reconstructions is visible.

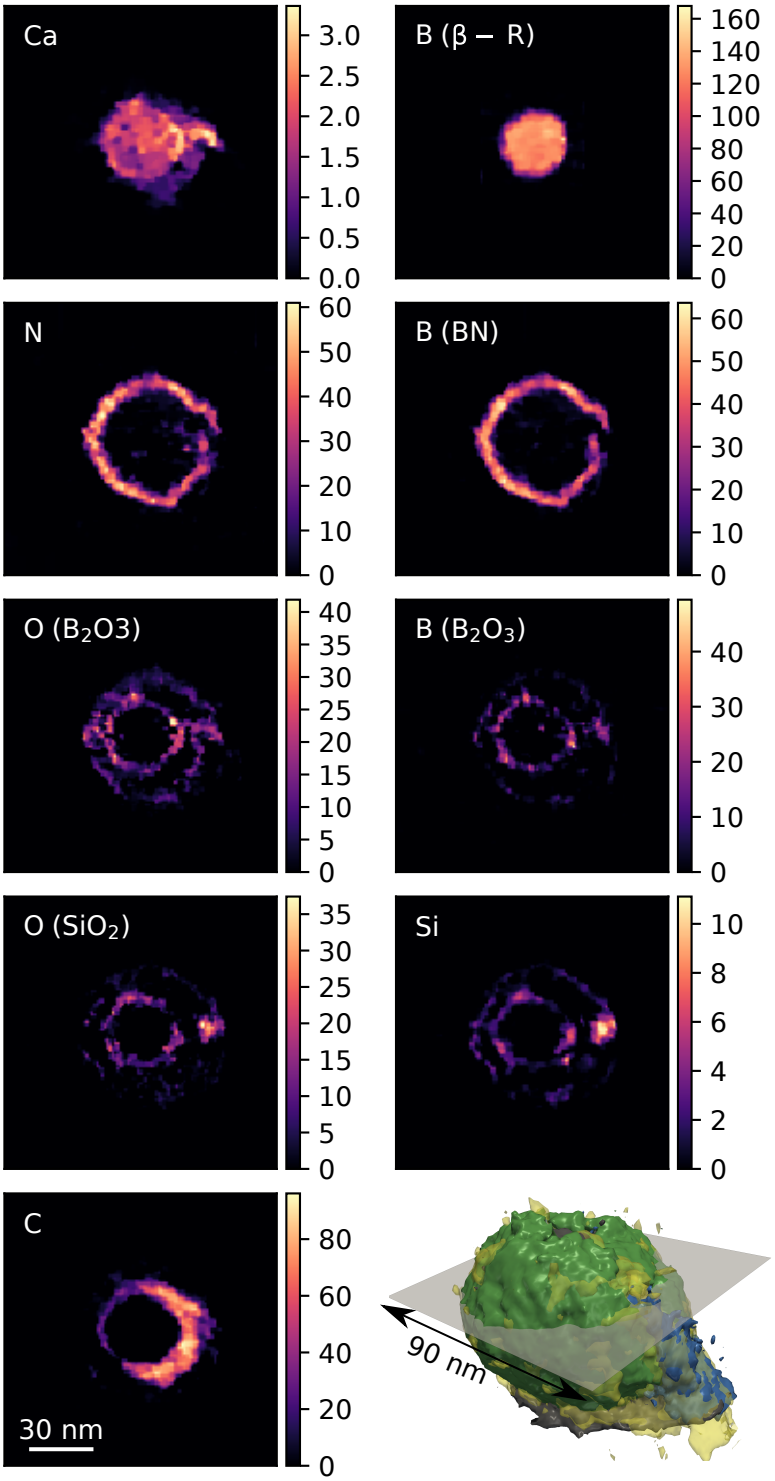


Fig. 8.14 Quantitative reconstruction slice. The shown colourbars are in atoms/nm^2. Bottom right shows the slice plane location.

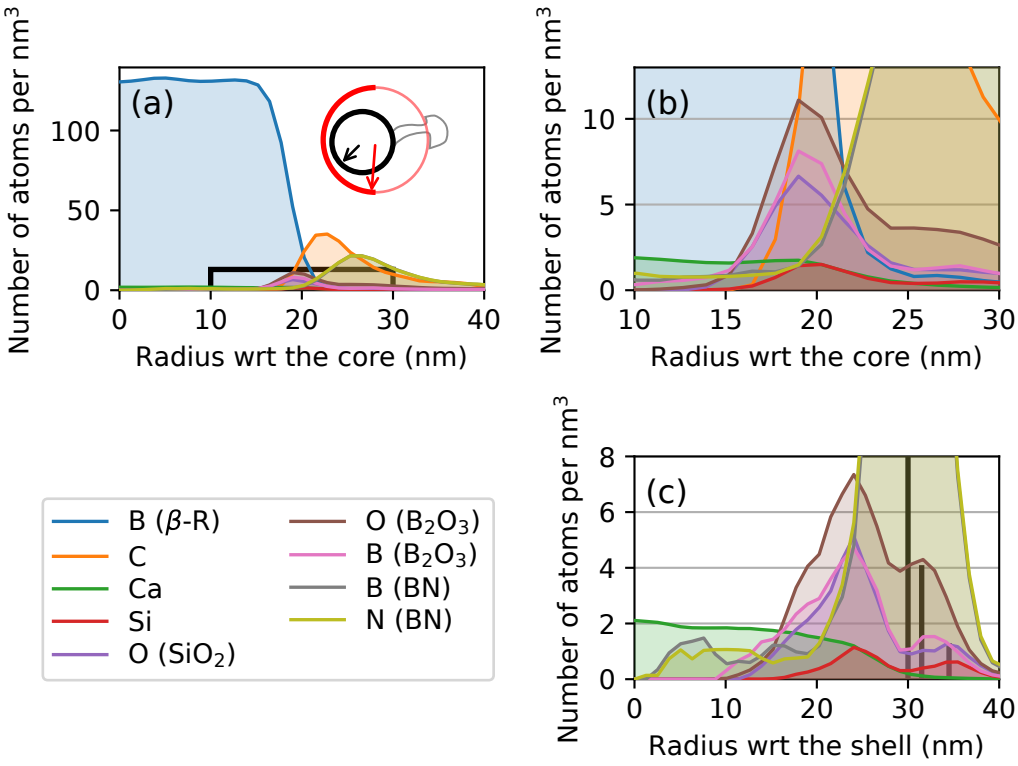


Fig. 8.15 (a) Mean number of atoms as a function of distance from the centre of the core, averaged over all directions. The inset diagram displays the relative positions of the centres of the core and the shell. Arrows mark one of the averaged radial profiles for clarity. (b) Zoomed-in version of the marked region in (a). Boron, silicon and calcium oxides can be seen between the metallic core and carbon. (c) Mean number of atoms as a function of distance from the centre of the shell, considering only half of the particle to exclude the "tail" structure. Three vertical black lines mark the peak positions for BN (30 nm), B_2O_3 (32 nm) and SiO_2 (34.5 nm).

Radially averaged profiles showing number of atoms/nm^3 from the centres of the core and the shell are shown in Fig. 8.15. Panels (a,b) show profiles when the zero is at the centre of the core (marked as black in the inset) and averaged over 4π, whereas (c) corresponds to using the centre of the shell as the origin (red in the inset) and considering only 2π. (a) shows that the metallic boron is contained almost entirely within the first 20 nm. Also, boron and silicon oxides can be detected between the core and it surrounding carbon, forming a thin shell. (b) shows the zoomed-in on the oxides version of (a). The measured mean number of atoms at 20 nm are 7.5 ± 3.0, 10.5 ± 5.0, 1.5 ± 1.1 and 6.0 ± 2.8 for B(B_2O_3), O(B_2O_3), Si(SiO_2) and O(SiO_2) respectively. Finding the oxygen in SiO_2 signal to be double the expected value, we include calcium with 1.8 ± 0.3 atoms at the

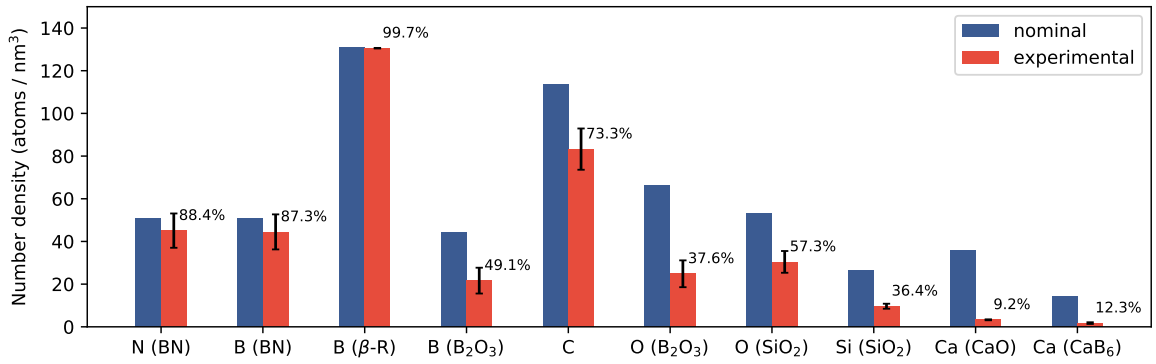


Fig. 8.16 Nominal and experimental (extracted from the 3D reconstructions) number densities for elements in corresponding compounds. Fractions of the theoretical values are shown as percent [191].

same radius into consideration and conclude that on the surface of the core there are, in fact, three oxides – boron, silicon and calcium, with the oxygen signal corresponding to the last oxide incorporated into what we thought to be just O(SiO_2).

Because the structure of the particle contains an elongated "tail" of similar oxide composition as the inner shell, we exclude it by considering only half of angles for Fig. 8.15(c) (thick red line in the inset). The profiles suggest a thin B_2O_3–SiO_2 shell on the outside of the thick BN shell. Black vertical lines mark the peak positions for BN (30 nm), B_2O_3 (32 nm) and SiO_2 (34.5 nm).

The results were verified to be consistent by comparing relative fitted intensities for the expected compounds – for example boron and nitrogen atom counts in BN were verified to be equal within ±5% where the SNR was sufficient. Fig. 8.16 compares the reconstruction results to theoretical compound atomic densities. We note that the errorbars only represent the error of the estimation from the reconstruction, and thus do not take into account the error of reconstruction itself or, for that matter, the error of the fitting analysis. The measured density of the core of the particle is just 0.3% below the theoretical density of the beta-rhombic boron crystal, one of its four known allotropes. The difference could be attributed to the small but detectable quantity of calcium in the core, which is estimated to be around 1.6%. The estimated number of carbon atoms was found to be 73% of the nominal graphite value, suggesting that a fraction of the graphene sheets might be rolled. While both boron and nitrogen in BN densities were underestimated by only around 12%, carbon, silica and boron oxide number densities were on average 50% lower than expected. The values are speculated to be a result of finite-sized beam measuring atom clusters of comparable or smaller spatial

dimensions, resulting in accurate numbers of atoms, but overestimated corresponding volumes. Calcium in CaO results suggest that compared to the beam size, CaO crystals were of significantly smaller dimensions. The seemingly unfavourable CaB_6 result is in fact consistent with the current knowledge, as both reported and simulated Ca–B phase diagrams support very low mole fractions of Ca with β-rhombic boron [192, 193].

The described morphology agrees with the previously suggested findings and growth mechanisms of such nanoparticles [61, 170, 184]. Namely, the particles were made by evaporating a h-BN target under a partial pressure of nitrogen gas. As the boron vapor cooled it condensed into boron droplets. Liquid boron is known to be highly reactive [61] with both carbon and oxygen, thus both elements (originally coming from the target itself [184]) dissolve in the boron droplet. As the droplets cool past ∼2700 K, the surface boron atoms react with the gaseous nitrogen to form the BN networks. If the droplet happened to not contain any dissolved oxygen, the BN networks form root-growing single-walled BN nanotubes. If oxygen is present in the droplet, as was the case for the investigated specimen, the B-N_2 reaction is claimed to be highly inhibited [184], and BN only forms at the surface as the boron droplet solidifies at 2000 K, creating a "cage" around the core. As carbon is not soluble in a now solid boron, it diffuses and precipitates at the surface of the core, filling the BN cage. In this particular specimen the carbon is reconstructed to either punch a hole in the BN shell, or just form around an already present gap in the cage. The dissolved oxygen precipitates when the particle cools past ∼1000 K, forming yet another shell around the nearly pure boron core. Again, the reconstructed B_2O_3 spatial distribution supports these claims. The oxide is found to seemingly flow through the possibly porous carbon and via an already present gap in the BN cage, or deform the said structures to create such path. In addition, the oxide is found to contain not only boron and oxygen as claimed in the previous studies, but also other trace elements that were initially present in the evaporation target, namely calcium and silicon. Interestingly, while no Si was reconstructed inside the kernel, a trace amount of Ca was still present.

We speculate that the unambiguously detected thin shell of Si, B(O) and O(B, Si) on the very outside of the measured particle was created during the STEM-EELS measurement. A thin layer of silica was found on the holey carbon film everywhere around the particle. We suggest that the silica layer was in fact created when preparing the specimen, covering both the particle and the support. As the high-energy electron beam scanned the specimen, it supplied the necessary energy to break the B–N bonds and allowed to form a shell of boron oxide by using SiO_2 as the source of oxygen. Even though the combination of spectral SNR and the spatial resolution of the reconstruction

are too low to confidently confirm it, borosilicate glass ($B_2O_3 - SiO_2$) may have formed as the outmost shell. The hypothesis of reaction during experiment is supported by the ring of silica at the base of the particle: because silica did not touch the BN shell, the boron oxide–forming reaction could not take place.

8.5 Conclusions

A complex multi-layered BN core-shell nanoparticle was investigated using EELS where both high-loss and low-loss regions of the spectrum were acquired quasi-simultaneously. Owing to the SAMFire algorithm significantly increasing curve-fitting stability, two new ELNES fingerprinting approaches were presented and used to quantitatively unravel both elemental and bonding maps from the measured dataset. Crucially, due to having both LL and HL of the dataset, both fingerprinting approaches are not subject to particle thickness limitations. Finally, a full 3D-TV compressed sensing tomography was performed to reconstruct the number of atoms of each species with high fidelity. The resulting 3D distributions confirmed previously suggested particle growth mechanisms and also revealed new, previously unnoticed details of the process. Because the reconstructions were also quantitative, atomic densities were estimated and compared with nominal values for corresponding compounds. The combination of the data analysis approaches serves as the first example where numbers of atoms were measured in 3D with nanometer precision but without sub-atomic experimental resolution. The analysis is not limited to small or symmetric particles, and hence can be extended to significantly larger structures and regions of interests.

Chapter 9

Strain mapping in diffraction cartography

This chapter includes work by Tomas Ostaševičius, Duncan N. Johnstone, and Sigurd Wenner. In particular, SW performed the experiment, DNJ developed the strain mapping ideas that were then extended with SAMFire by TO. TO analysed the datasets and produced the figures.

Strain has long been used in many areas to adapt and improve material properties: from the semiconductor industry to improve electronic devices [194] or to tune the emitted photon frequency [195] in opto-electronics, to the more traditional and widely-used mechanical strengthening of alloys [196, 197]. In all cases strain mapping with high spatial resolution and precision is the key to better models and understanding of the effects.

In the last few years the transmission electron microscope (TEM) has become the tool of choice for experimental measures of strain in thin specimens. A number of techniques that enable access to such information have been developed or recently improved. Some rely on high resolution real-space imaging by comparing unstrained atom positions with the strained region of interest [198], while others analyse atomic-resolution image geometry using so-called geometrical phase analysis [199]. Other measurements use electron holography, interfering two waves following different trajectories, one of which is through the strained specimen [199, 200]. Finally, two diffraction-based TEM strain measurement techniques are available: nanobeam electron diffraction (NBED) and scanning precession electron diffraction (SPED) [5], the technique used in this work.

The main focus of the chapter are new NBED and SPED data analyses, and experimental data and results are used only to demonstrate their capabilities.

9.1 Strain in diffraction

Strain is generally defined as the relative change of the object shape due to outside forces [201]. For crystals this directly corresponds to change in the planar spacing, which is readily visible in the reciprocal space section of a diffraction pattern. Writing a strain component as $\boldsymbol{\varepsilon} = \frac{d_0-d}{d} = \frac{g-g_0}{g_0}$, where d,g corresponds to real and reciprocal lattice spacings respectively, the main task when mapping is often simplified to measuring how diffraction spots move compared to an unstrained crystal lattice.

The traditional way to perform such a task is via peak finding, where each measured spot is registered and its centre position is determined with sub-pixel accuracy. The estimated coordinates can then be used to find the basis vectors of both strained and reference diffraction patterns, which are directly used in the strain definition. Such an approach relies heavily on the peak finding algorithm, which may be unstable with noisy data. In addition, each peak location is estimated independently of all others, allowing for more ways of error propagation. Nevertheless, it has been shown that such an approach, while slow, is able to produce precise strain maps [202].

In this work, two alternative ways to estimate strain from diffraction patterns are shown. The first one, described in section 9.1.1, operates purely in reciprocal space and neither requires nor provides any knowledge of the crystal structure apart from the relative deformation of the reference. The method was tested by analysing an age-hardened Al sample, as described in section 9.2. Section 9.1.2 briefly describes the second approach, where the strain is found by perturbing the locations of atoms in a simulated crystal lattice and comparing with the experimental result. While very promising, the practical side of the method is not fully realised and is still work in progress. Importantly, both approaches are, in effect, non-linear optimization problems, and hence both are best used with SAMFire.

9.1.1 Reference diffraction pattern

An atom in the strained crystal lattice is described by a position vector $\boldsymbol{x}$. If the same atom is at $\boldsymbol{X}$ in the unstrained reference lattice, a mapping $\boldsymbol{x} = \phi(\boldsymbol{X})$ can be constructed. The deformation gradient is then defined as

$$\boldsymbol{F} = \nabla\phi\,. \tag{9.1}$$

While $\phi(\boldsymbol{X})$ is a general mapping that also includes arbitrary particle motion, $\boldsymbol{F}(\boldsymbol{X})$ operates on infinitesimal parts of the crystal, and hence represents the relative element coordinate shift: $d\boldsymbol{x} = \boldsymbol{F}d\boldsymbol{X}$. The gradient can be further decomposed into a product of a rotation matrix $\boldsymbol{R}$ and a stretch matrix $\boldsymbol{U}$ using a polar decomposition $\boldsymbol{F} = \boldsymbol{R}\boldsymbol{U}$ [201]. By noting that generally a displacement (importantly, without rotation) can be defined as $\boldsymbol{x} = \boldsymbol{X} + \boldsymbol{u}$, we can expand eq. (9.1):

$$\begin{aligned} \boldsymbol{U} &= \frac{\partial}{\partial \boldsymbol{X}}(\boldsymbol{X} + \boldsymbol{u}) \\ &= \boldsymbol{I} + \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{X}}\,, \end{aligned} \tag{9.2}$$

which in the matrix notation is written as

$$U_{ij} = \delta_{ij} + \frac{\partial u_i}{\partial X_j}\,. \tag{9.3}$$

Here the second term is the displacement gradient and corresponds directly to the strain tensor, thus the problem is formulated in terms of finding the deformation gradient $\boldsymbol{F}$ and does not require peak finding.

To find $\boldsymbol{F}$ in each pixel, we use an affine transformation in two (reciprocal space) dimensions, defined as

$$\begin{aligned} x &= a_0 X + a_1 Y + a_2\,, \\ y &= b_0 X + b_1 Y + b_2\,, \end{aligned} \tag{9.4}$$

where (x, y) and (X, Y) are the resultant and original coordinates. Such a transformation fully describes rotation, shift and shear in both directions of the diffraction pattern plane. Importantly, writing the transformation as a matrix we realise that it is exactly the deformation gradient:

$$F = \begin{pmatrix} a_0 & a_1 & a_2 \\ b_0 & b_1 & b_2 \\ 0 & 0 & 1 \end{pmatrix}\,. \tag{9.5}$$

The final algorithm to map strain with respect to some reference pattern then involves finding an affine transformation matrix $\boldsymbol{F}$ for each of the experimentally measured DPs, decomposing each into rotation $\boldsymbol{R}$ and displacement $\boldsymbol{U}$, and then calculating the strain components from eq. (9.3). The affine transformation is found using a non-linear optimizer with the matrix elements as parameters and the correlation as the cost function.

9.1.2 Forward model

The second approach to estimate strain from a SPED dataset approaches the problem from the material, and not image processing, side. It requires knowing the crystallographic structure of the specimen – the type and size of the lattice, and the species of atoms that the crystal is made of. If this information is known, then the strain mapping for one particular DP can be done the following way.

The task is again formulated as a non-linear optimization problem. First, a model of the investigated lattice is created, where a theoretical diffraction pattern in some direction can be calculated, for example using the kinematic theory of electron diffraction [203]. By applying geometrical (atom position) transformations that correspond to rotations and strain effects, DPs for different strains can be simulated and compared to the experimental data.

While the process itself is fairly straightforward from the theoretical point of view, the biggest difficulty lies in applying such analysis to real data. Because the model is very sensitive to the starting parameters of the simulation, in particular the direction of electron trajectories, normally each pixel in a SPED dataset would require a great deal of attention. Fortunately, SAMFire was created to solve this exact problem and becomes particularly useful when analysing polycrystalline samples. By first identifying different grains in the specimen, each can be given one "seed" pixel for the particular orientation, letting SAMFire propagate them outwards to the grain boundaries. In this particular framework the local SAMFire strategy can be interpreted in a very physical sense: it assumes that the strain field is continuous when considering neighbouring real-space pixels in the same grain. In addition, if many similar grains are visible in the field of view, the global SAMFire strategy is able to propagate the direction information from distant similar grains automatically.

The remaining challenge of using such forward model to analyse SPED data in practice involves comparing the model to the data. The difficulty arises from different representations: the model usually consists of a list of coordinates and the respective intensities of the diffraction spots, whilst each DP is measured, and thus represented, as a pixelated image. Comparing these two results efficiently is challenging and better approaches are needed. The currently attempted method compared the real diffraction intensities at the coordinates returned by the simulation with the theoretical results. The method, however, proved to be insufficiently precise for practical strain analyses if conventional DP resolutions, for example $512\,\mathrm{px} \times 512\,\mathrm{px}$, were used. We speculate that interpolating the measured DP up to a much higher resolution (as well as acquiring higher resolution patterns experimentally to begin with) should increase the method effectiveness

for strain mapping. Another alternative involves calculating image representation of the simulated DPs for the comparison with data, however it may be prohibitively computationally expensive.

The biggest strength of the forward model strain mapping is the ability to directly relate and compare DPs from completely different directions, which makes the approach more robust to buckling and similar non-ideal samples. In addition, while the idea is not yet extensively explored, the forward model allows mapping strain in all three dimensions. This would complement 3D X-ray diffraction strain mapping [204–206] but with nanometer spatial resolution.

9.2 Example: mapping strain in Al alloy

As an example of the type of analysis made possible by the methods described in section 9.1, we present a study of an Al–Mg–Si alloy. It is widely used in many industries due to the excellent mix of strength, cost, weight and corrosion properties. The exceptional increase in strength of the material has long been shown to be a result of precipitation hardening [196, 207]. During the age hardening, for this alloy in particular, long needle-like ($4\,\text{nm} \times 4\,\text{nm} \times 50\,\text{nm}$) β'' precipitates form in the Al matrix. Because the precipitates have a positive misfit with respect to the bulk Al, they strain the matrix, significantly inhibiting deformation propagation and thus strengthening the alloy. However, a number of mechanisms are present that reduce the coherency of the precipitates and thus the alloy strength [197]. In this work we demonstrate the affine transformation approach described in section 9.1.1 to extract strain fields from a SPED dataset of a large field of view of such a sample.

9.2.1 Experiment

SPED was performed using a NanoMEGAS DigiSTAR scan generator [208] fitted to a JEOL JEM 2100F FEG-(S)TEM operated at 200 kV, with a precession angle of 1° and a step size of 1.92 nm. The PED patterns were recorded using an externally mounted StingRay camera to capture the image on the phosphor viewing screen of the microscope. The microscope was operated in nano-beam diffraction mode using a probe size of 1 nm and a convergence angle $\alpha = 4$ mrad. A PED pattern is recorded at each probe position yielding 160 000 (400×400) diffraction patterns covering areas up to $\sim$1 µm × 1 µm. The diffraction patterns were recorded with a camera length of 20 cm and an exposure time of 40 ms per pattern.

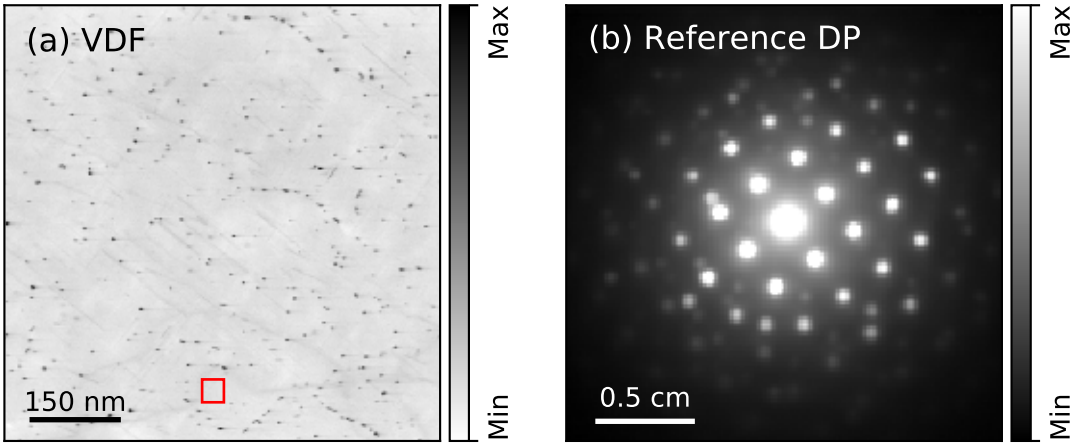


Fig. 9.1 (a) Virtual dark field (VDF) image with inverted contrast, where precipitates are darker than the surrounding matrix. (b) Reference DP used in strain analysis as described in section 9.1.1. (b) was calculated by taking the mean DP from an unstrained region, marked with red rectangle in (a).

The sample was an Al-Mg-Si 6xxx series alloy aged for 4 h at 195 °C to achieve the peak-aged condition. In this condition the predominant precipitate phase is β''–$Al_2Mg_5Si_4$ as well as β' formed at dislocation cores. β'', the main phase of interest of the experiment, is described by the monoclinic $C2/m$ space group. Its lattice parameters have been measured to be $a = 1.516$ nm, $b = 0.405$ nm, and $c = 0.674$ nm, with a monoclinic angle of 105.3° [209, 210]. The specimen was prepared so that $[001]_{Al} \parallel [010]_{\beta''}$ is parallel to the electron beam trajectory, with the long needle axis corresponding to $[010]_{\beta''}$.

9.2.2 Analysis

Strain in the large (ca. 0.5 µm^2) field of view was mapped using an affine transformation of the reference pattern as described in section 9.1.1 using the SAMFire algorithm for the necessary starting parameter stability. The virtual dark field (VDF) image with inverted contrast, where precipitates are darker than the surrounding matrix, is shown in Fig. 9.1(a). The VDF allowed to calculate the mean DP of an unstrained region, marked with red rectangle in (a), to be used as a reference DP in further analysis. Strain and rotation components calculated from fitted eq. (9.5) matrix elements are shown in Fig. 9.2. The strain is given in percent and calculated in crystal coordinates, where (x, y) correspond to $([100], [010])$ of the aluminium matrix. This required rotating the strain basis vectors by 52.7° to align with the image axes. The rotation about the central spot

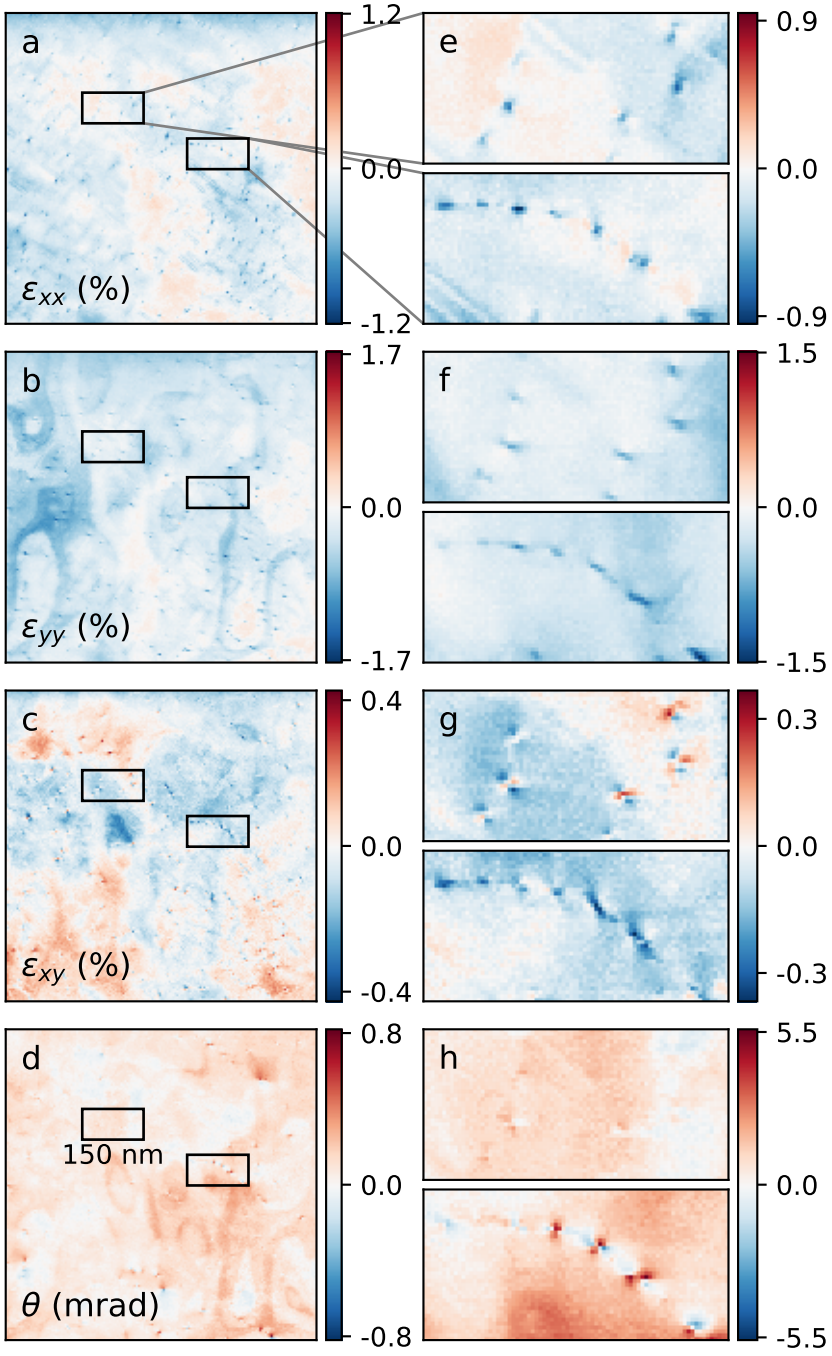


Fig. 9.2 Strain components ($\varepsilon_{xx}, \varepsilon_{yy}, \varepsilon_{xy}, \theta$) estimated according to eq. (9.5). The full field of view is shown in (a-d), with two 150 nm × 75 nm marked regions in (e-h). The angle is given in mrad clockwise from the vertical, and strain in percent.

is given in the clockwise direction in milliradians. In the figure the local strain fields around small precipitates are visible and reconstructed, however a slow smoothly varying background prohibits accurate evaluation. We speculate that the background is caused by the sample preparation for TEM procedure. To remove it, a Gaussian blurring with large (3 pixel) radius was applied and subsequently subtracted from the strain fields. This effectively removed low spatial frequency variations and allowed background-independent fields to be extracted around most of the precipitates, as shown in Fig. 9.3.

It is useful to note that the diffraction spots from the precipitates themselves were neither used nor required to be visible in the original dataset. Nevertheless, most needles were readily identifiable in the background-subtracted strain components. The strain values around each precipitate were found to be roughly a factor of 2 lower than previously reported [197]. We believe it to be due to a combination of the previously mentioned non-ideal background subtraction, limited SPED spatial resolution (blurring the rapidly decaying strain field), and the necessary oversaturation of the Al matrix DP spots to have sufficient signal to image the needles. Despite these issues, the results can be used for further analyses using high resolution strain field maps to investigate precipitate interaction (Fig. 9.3 (h)) or dislocation dynamics.

A strain map with ∼200 precipitates with 1 nm spatial resolution, while rich in information, is difficult to interpret. Fortunately, the experiment was set up to allow sufficient diffraction intensity from the precipitates to be used in a parallel analysis that can be later combined with the strain results. Machine learning, in particular non-negative matrix factorisation (NMF) was used to factor out DPs corresponding to the precipitates in question. We found that because the total number of pixels with needles was just a small fraction and in each such DP Al matrix spots were at least 100 times more intense, the fraction of total counts attributed to the precipitates was not sufficient for NMF to straightforwardly separate out the required components from the full dataset. Instead, we used a VDF image (Fig. 9.1) to construct a binary mask containing the precipitates and their surrounding pixels. By then performing NMF on only the masked pixels of the full dataset, four different DP components associated with a subset of the considered needles were extracted, shown in Fig. 9.4. The four spatial distributions of precipitate sub-classes were then used as class markers in real-space. The patterns themselves were investigated by DNJ and matched to two pairs of matrix-precipitate lattice alignments with mirror symmetry, $[3\bar{1}0]_{Al} \parallel [001]_{\beta''}$ and $[230]_{Al} \parallel [100]_{\beta''}$, supporting previous results [209, 210].

In addition to the physical locations of each class of the precipitate, NMF results also reveal their crystallographic directions via the learnt diffraction patterns. Keeping

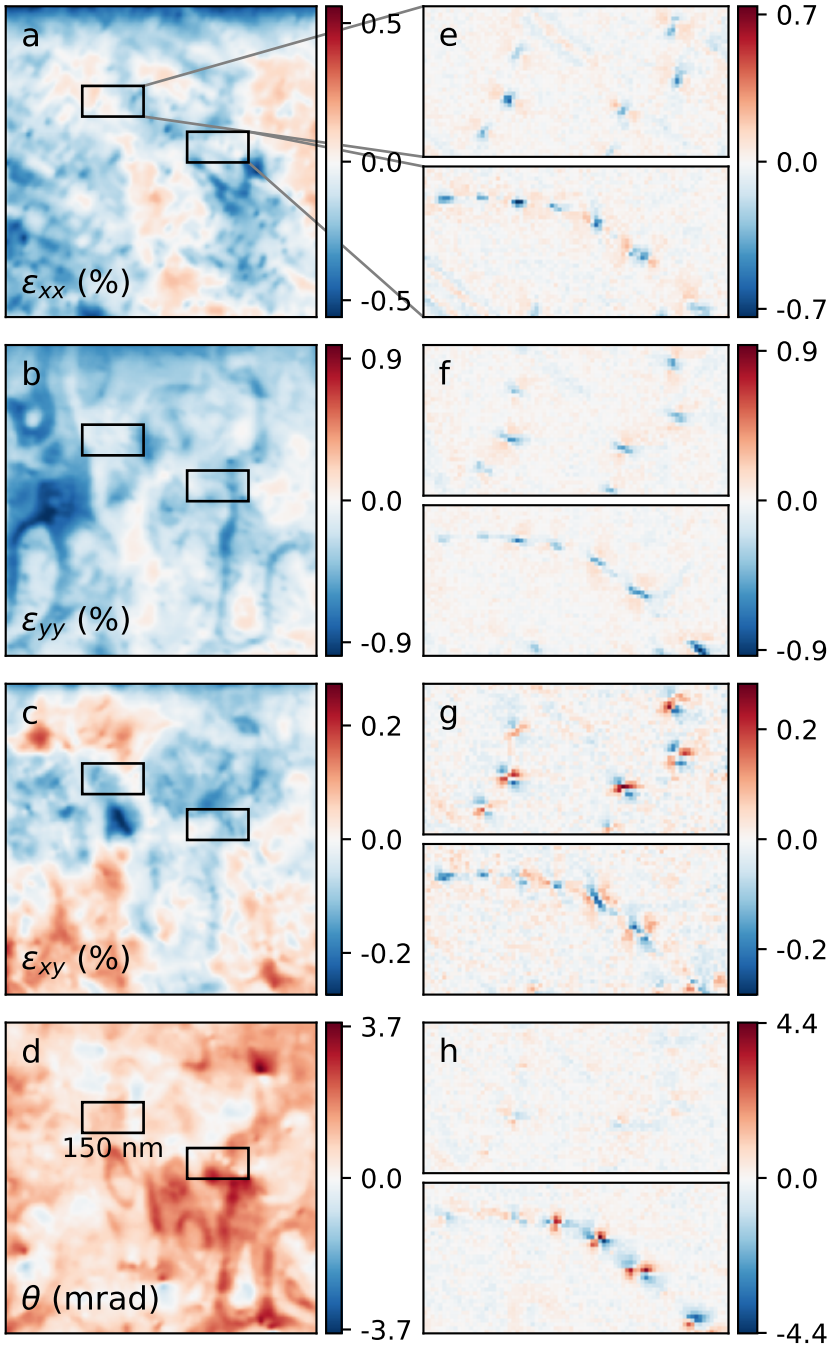


Fig. 9.3 The Gaussian-smoothing estimated backgrounds (a-d) and background-subtracted strain components (e-h)

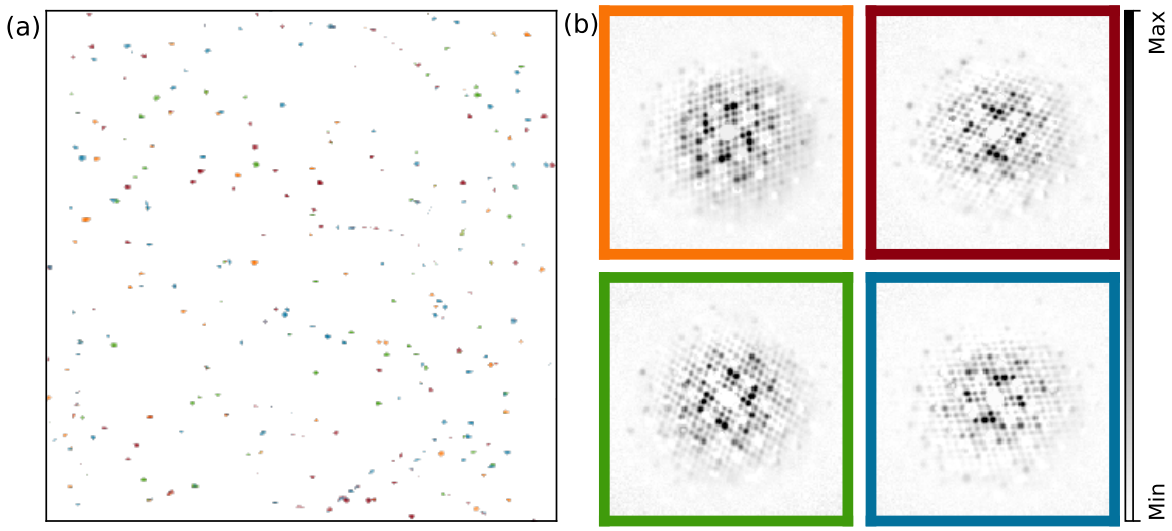


Fig. 9.4 The four precipitate diffraction pattern NMF components (right) and their corresponding real-space locations. Notably, some precipitates show multiple phases even if just 5 pixels are considered, confirming the method sensitivity.

in mind that NMF results correspond to needles while strain was measured in the Al matrix, there are two ways to combine the information and perform a crude statistical analysis. Both require first to identify the needles where each strain field can be readily isolated from any surrounding artefacts or other strong strains and segmenting the result map into the four subsets based on the learnt real-space locations. The first and more straightforward analysis involves calculating the mean strain field from all four classes combined. It can be interpreted as the strain field around a statistically average (in terms of both crystallographic direction and size) precipitate. In this case the original strained lattice is kept as the strain basis coordinates. Alternatively, a similar analysis could be performed using the β'' lattice as the basis. By determining relative rotations and mirror symmetries from the learnt DPs, each subset of strain maps can undergo independent coordinate transformation to align the strain basis vectors with the β'' lattice. The average strain fields for each detected β'' direction after appropriate alignments are shown in Fig. 9.5(a-d). Panel (f) shows the mean across all four classes (a-d), and can be interpreted as the mean strain environment of a β'' needle in the matrix. Fig. 9.5(e) shows the estimated distribution of needle cross-section areas for the four classes. In agreement with the previous studies [197], most precipitate cross-sections are 17 ± 3 nm^2, which corresponds to around 4.5 px^2.

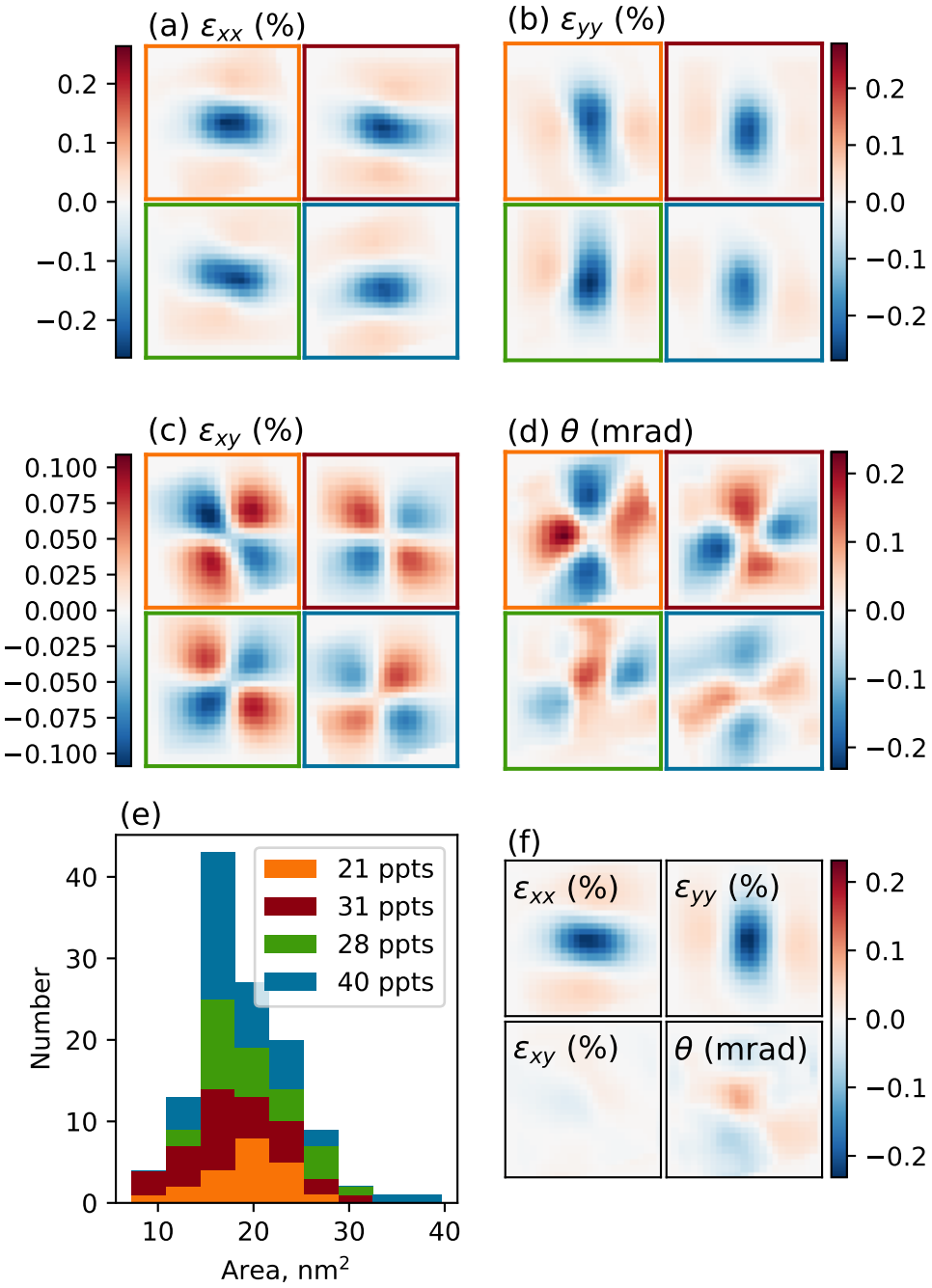


Fig. 9.5 (a-d) show mean strain fields across classes, each marked with corresponding colour. Here the strain basis vectors of all classes were aligned according to the DPs shown in Fig. 9.4 by rotating and mirroring where required. (e) shows needle cross-section area distribution across the four NMF-learnt components. (f) shows the total average strain field across all four classes (that is the mean of (a-d)), interpreted as the mean strain environment around such precipitates.

9.3 Conclusions

We presented a new strain mapping from SPED data approach that was very sensitive to starting parameters and required using SAMFire to operate. Strain in an age-hardened aluminium alloy was mapped. The dataset was also decomposed using the NMF machine learning algorithm into component-loading pairs, corresponding to diffraction patterns and spatial maps respectively. NMF components revealed four distinct DPs associated with precipitates. Strain results from the four classes of precipitates were extracted using the associated loadings as real-space markers. Because crystallographic directions within classes were identical by definition, mean strain fields for each class were estimated. By rotating each class strain field bases to align the corresponding β'' lattices, the mean strain environment of the needle in the matrix could be estimated.

Chapter 10

Conclusions

In this thesis I showed that data analysis tools play an increasingly important role in electron microscopy irrespective of the measured signal nature. In particular, chapter 5 presented a new algorithm for non-linear optimizations called SAMFire (Smart Adaptive Multi-dimensional Fitting), which was then used throughout the work to tackle challenging data analysis problems for a variety of experimental datasets, with analysis diagrams shown in Appendix A:

- Chapter 7 considered Cathodoluminescence (CL) measurements of a nanowire with Quantum Disks (QDisks). SAMFire was key to robustly and consistently extracting energy–intensity relationships of many QDisks across the ten considered CL maps. The results offered a better understanding of the efficiency droop of the structures by attributing it to the Auger emission, which enables further optimizations in such structures.

- Chapter 8 presented Electron Energy Loss Spectrum (EELS) analysis of a complex BN core-shell nanoparticle. Two new Energy Loss Near-Edge Structure (ELNES) estimation algorithms were developed that neither require standards nor limit the thickness of the specimen. Combined with SAMFire, it enabled extracting quantitative bonding maps across the particle without atomic resolution. By performing the quantification over a tilt-series and using a 3D total variation (TV) compressed sensing (CS) tomography, the first absolutely quantitative bonding tomography without atomic resolution was performed. These technique developments act as a milestone for electron tomography and for the first time allow such quantitative information to be extracted from thick irregular specimens.

- Chapter 9 showed how SAMFire was used for Scanning (Precession) Electron Diffraction (S(P)ED) strain analysis, where it was essential for two new strain

mapping approaches from such data. Strain in an age–hardened aluminium alloy was mapped over large (ca. 0.5 µm^2) area containing many precipitate needles. The dataset was also decomposed using a Machine Learning (ML) algorithm, in particular Non-negative Matrix Factorisation (NMF), which allowed the precipitates to be classified. Strain and NMF results were combined to estimate the mean displacement fields around each class of precipitates.

Large data sizes arose as the one common problem when dealing with the aforementioned experimental results, and a solution was proposed in chapter 6. Named "LazySignal", the framework allows seamless analysis of datasets that do not fit into the computer memory and would prove impossible to access without special high-memory hardware. It uses state-of-the-art Python libraries to enable almost any type of analysis on both consumer-grade computers and large distributed computing environments, as long as the dataset fits on the disk. LazySignal makes the electron microscopy field well-equipped to deal with most "big data" problems that are likely to arise in the near future.

Finally, chapter 3 considered cube and sphere plasmon EELS responses and showed by example that first-order cube modes can be approximated by theoretical sphere plasmon solutions. It offers a more robust way to unravel plasmonic large nanocube EELS signals, which in turn allows more control when designing nanoplasmonic devices.

10.1 Further work

As electron microscopes and computer hardware grow in effective data throughput, it is left to the analysis and further data interpretation to make the most of it. As a result, various data analysis and treatment routines are likely to get significantly closer to the experiment itself.

The global parameter fitting, presented as one of the ELNES estimation methods in chapter 8, is in fact a completely general approach. As such, in addition to the mentioned use for EELS data, it also allows measurement of any other experiment or sample property that is constant throughout, but unreliable to estimate from just a single measurement. Another example might be lens distortions in S(P)ED experiments for high-angle reflections. As the non-linearity of the experimental setup is constant throughout the dataset, it can be modelled in terms of global parameters and corrected.

With faster yet still accurate algorithms, many analyses can be performed live while the experiment is still running. This would reduce guesswork and human factors when optimizing experimental parameters for the sought result by making it immediately

available. In particular, performing online ML to de-noise and optionally decompose the data would allow an accurate way to gauge experimental dwell times or the specimen composition. Another example may be performing the reference-based strain analysis presented in chapter 9. Both approaches require minimal setup to get started, but provide information that is usually only available once the experiment is finished.

Compressed sensing may allow making large strides towards lowering the necessary experimental electron doses. Shown to be able to reasonably reconstruct the ground truth with just a few percent of information [174], CS algorithms may benefit EM in a multitude of ways. From simply providing much faster initial previews when searching for areas of interest, to greatly reducing the total electron dose in any EM experiment. Even if the specimen does not suffer from beam damage, performing experiments using CS approaches may increase the experimental "time resolution" by simply lowering the number of required measurements.

Finally, with increasingly powerful analyses becoming routine, previously "too challenging" experiments can be attempted. One example follows from SPED, where the de-scan coils are either not used at all or their current is less than in normal SPED experiments. This would result in measuring diffraction rings instead of spots across the sample. If de-scan is adjusted to minimise ring overlap while still keeping the directional information available, forming Virtual Dark Field (VDF) images around the ring may allow virtually rocking the sample. In addition, most strain mapping algorithms are likely to perform better by having more information from each diffracted beam – be it measuring the centre of the ring, or transforming it to match a different one.

10.2 Open source data analysis tools

With analysis tools becoming significantly more complex and important for the final results, more attention should be paid to ensure both their correctness and longevity. This is best achieved by releasing research tools as new or contributing them to already active open-source and preferably open-development projects. There are many benefits in making the algorithm and its implementation available to everyone. Open source analysis tools:

- allow analysis and hence experimental result replication.
- save time for other scientists that are interested in the algorithm or its modifications. Often it is simpler to modify an existing solution to suit the particular experiment than writing one from scratch.

- often serve as the groundwork upon which new tools and algorithms are built.
- encourage collaborations both among peers and between different disciplines.
- have to pass the scrutiny of peer review and hence both the algorithms and their implementations are often of higher quality.
- are encouraged to have tests. Combined with the peer review, it often means that improving the algorithms performance or general approach can be done in a reliable way that still allows result replication.

HyperSpy [101] is an open-source and open-development Python based analysis framework that benefits from all these points. Both SAMFire and LazySignal were contributed to HyperSpy and are therefore publicly available to use, modify, study, test, and build upon.

References

[1] Philip E Batson, Niklas Dellby, and Ondrej L Krivanek. Sub-ångstrom resolution using aberration corrected electron optics. *Nature*, 418(6898):617–620, 2002.

[2] Ondrej L Krivanek, Tracy C Lovejoy, Niklas Dellby, Toshihiro Aoki, RW Carpenter, Peter Rez, Emmanuel Soignard, Jiangtao Zhu, Philip E Batson, Maureen J Lagos, et al. Vibrational spectroscopy in the electron microscope. *Nature*, 514(7521):209–212, 2014.

[3] N Zabala and A Rivacoba. Electron energy loss near supported particles. *Physical Review B*, 48(19):14534, 1993.

[4] Ray F Egerton. *Electron energy-loss spectroscopy in the electron microscope*. Springer Science & Business Media, 2011.

[5] Roger Vincent and PA Midgley. Double conical beam-rocking system for measurement of integrated electron diffraction intensities. *Ultramicroscopy*, 53(3):271–282, 1994.

[6] M Kociak and LF Zagonel. Cathodoluminescence in the scanning transmission electron microscope. *Ultramicroscopy*, 174:50–69, 2017.

[7] Hans Geiger and Ernest Marsden. On a diffuse reflection of the α-particles. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 82(557):495–500, 1909.

[8] Hideki Yukawa. On the interaction of elementary particles. i. *Nippon Sugaku-Buturigakkwai Kizi Dai 3 Ki*, 17(0):48–57, 1935.

[9] Mitio Inokuti. Inelastic collisions of fast charged particles with atoms and molecules—the Bethe theory revisited. *Reviews of modern physics*, 43(3):297, 1971.

[10] R. Ritchie. Plasma Losses by Fast Electrons in Thin Films. *Physical Review*, 106(5):874–881, June 1957.

[11] RH Ritchie and A Howie. Inelastic scattering probabilities in scanning transmission electron microscopy. *Philosophical Magazine A*, 58(5):753–767, 1988.

[12] S.A. Maier. *Plasmonics: Fundamentals and Applications: Fundamentals and Applications*. Springer, 2007.

[13] F. J. García de Abajo. Optical excitations in electron microscopy. *Reviews of Modern Physics*, 82(1):209–275, February 2010.

[14] Ben G Yacobi and David Basil Holt. *Cathodoluminescence microscopy of inorganic solids*. Springer Science & Business Media, 2013.

[15] N Yamamoto, K Araya, and FJ García de Abajo. Photon emission from silver particles induced by a high-energy electron beam. *Physical Review B*, 64(20):205419, 2001.

[16] R Gómez-Medina, N Yamamoto, M Nakano, and FJ García de Abajo. Mapping plasmons in nanoantennas via cathodoluminescence. *New Journal of Physics*, 10(10):105009, 2008.

[17] S Meuret, LHG Tizei, T Cazimajou, R Bourrellier, HC Chang, F Treussart, and M Kociak. Photon bunching in cathodoluminescence. *Physical review letters*, 114(19):197401, 2015.

[18] Peter Rez. Cross-sections for energy loss spectrometry. *Ultramicroscopy*, 9(3):283–287, 1982.

[19] Peter Rez. Accurate cross sections for microanalysis. *Journal of research of the National Institute of Standards and Technology*, 107(6):487, 2002.

[20] S Manson. The calculation of photoionization cross sections: An atomic view. *Photoemission in Solids I*, pages 135–163, 1978.

[21] Konrad Jarausch, Paul Thomas, Donovan N Leonard, Ray Twesten, and Christopher R Booth. Four-dimensional STEM-EELS: Enabling nano-scale chemical tomography. *Ultramicroscopy*, 109(4):326–337, 2009.

[22] J Verbeeck and G Bertoni. Deconvolution of core electron energy loss spectra. *Ultramicroscopy*, 109(11):1343–1352, 2009.

[23] J Verbeeck and S Van Aert. Model based quantification of EELS spectra. *Ultramicroscopy*, 101(2):207–224, 2004.

[24] G Baffou, R Quidant, and C Girard. Heat generation in plasmonic nanostructures: Influence of morphology. *Applied Physics Letters*, 94(15):153109, 2009.

[25] LR Hirsch, RJ Stafford, JA Bankson, SR Sershen, B Rivera, RE Price, JD Hazle, NJ Halas, and JL West. Nanoshell-mediated near-infrared thermal therapy of tumors under magnetic resonance guidance. *Proceedings of the National Academy of Sciences*, 100(23):13549–13554, 2003.

[26] SR Sershen, SL Westcott, NJ Halas, and JL West. Temperature-sensitive polymer–nanoshell composites for photothermally modulated drug delivery. *Journal of biomedical materials research*, 51(3):293–298, 2000.

[27] Jeffrey N. Anker, W. Paige Hall, Olga Lyandres, Nilam C. Shah, Jing Zhao, and Richard P. Van Duyne. Biosensing with plasmonic nanosensors. *Nature Materials*, 7(6):442–453, Jun 2008.

[28] G Raschke, S Kowarik, T Franzl, C Sönnichsen, TA Klar, J Feldmann, A Nichtl, and K Kürzinger. Biomolecular recognition based on single gold nanoparticle light scattering. *Nano letters*, 3(7):935–938, 2003.

[29] Adam D McFarland, Matthew A Young, Jon A Dieringer, and Richard P Van Duyne. Wavelength-scanned surface-enhanced Raman excitation spectroscopy. *The Journal of Physical Chemistry B*, 109(22):11279–11285, 2005.

[30] Sandy Owega, Edward PC Lai, and Wayne M Mullett. Laser desorption ionization of gramicidin S on thin silver films with matrix isolation in surface plasmon resonance excitation. *Journal of Photochemistry and Photobiology A: Chemistry*, 119(2):123–135, 1998.

[31] Sougata Sarkar, Surojit Pande, Subhra Jana, Arun Kumar Sinha, Mukul Pradhan, Mrinmoyee Basu, Joydeep Chowdhury, and Tarasankar Pal. Exploration of electrostatic field force in surface-enhanced Raman scattering: an experimental investigation aided by density functional calculations. *The Journal of Physical Chemistry C*, 112(46):17862–17876, 2008.

[32] Lukas Novotny. Effective wavelength scaling for optical antennas. *Physical Review Letters*, 98(26):266802, 2007.

[33] Thomas Søndergaard and Sergey Bozhevolnyi. Slow-plasmon resonant nanostructures: Scattering and field enhancements. *Physical Review B*, 75(7):073402, 2007.

[34] Tineke Thio, KM Pellerin, RA Linke, HJ Lezec, and TW Ebbesen. Enhanced light transmission through a single subwavelength aperture. *Optics Letters*, 26(24):1972–1974, 2001.

[35] H Ditlbacher, JR Krenn, A Hohenau, A Leitner, and FR Aussenegg. Efficiency of local light-plasmon coupling. *Applied Physics Letters*, 83(18):3665–3667, 2003.

[36] John David Jackson. *Classical electrodynamics*. Wiley, New York, NY, 3rd ed. edition, 1999.

[37] James Baker-Jarvis and Sung Kim. The interaction of radio-frequency fields with dielectric materials at macroscopic to mesoscopic scales. *Journal of research of the National Institute of Standards and Technology*, 117:1, 2012.

[38] FJ García de Abajo. Relativistic energy loss and induced photon emission in the interaction of a dielectric sphere with an external electron beam. *Physical Review B*, 59(4), 1999.

[39] Christian Matyssek, Jens Niegemann, Wolfram Hergert, and Kurt Busch. Computing electron energy loss spectra with the Discontinuous Galerkin Time-Domain method. *Photonics and Nanostructures-Fundamentals and Applications*, 9(4):367–373, 2011.

[40] FJ García de Abajo and J Aizpurua. Numerical simulation of electron energy loss near inhomogeneous dielectrics. *Physical Review B*, 56(24):15873, 1997.

[41] Nicholas W Bigelow, Alex Vaschillo, Vighter Iberi, Jon P Camden, and David J Masiello. Characterization of the electron- and photon-driven plasmonic excitations of metal nanorods. *ACS nano*, 6(8):7497–504, August 2012.

[42] Bruce T Draine and Piotr J Flatau. User guide for the discrete dipole approximation code DDSCAT 7.3. *arXiv preprint arXiv:1305.6497*, 2013.

[43] Guillaume Boudarham and Mathieu Kociak. Modal decompositions of the local electromagnetic density of states and spatially resolved electron energy loss probability in terms of geometric modes. *Physical Review B*, 85(24):245447, June 2012.

[44] Bruce T Draine and Piotr J Flatau. Discrete-dipole approximation for scattering calculations. *JOSA A*, 11(4):1491–1499, 1994.

[45] D Gutkowicz-Krusin and Bruce T Draine. Propagation of electromagnetic waves on a rectangular lattice of polarizable points. *arXiv preprint astro-ph/0403082*, 2004.

[46] Maxim A. Yurkin and Alfons G. Hoekstra. The discrete-dipole-approximation code ADDA: Capabilities and known limitations. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 112(13):2234 – 2247, 2011. Polarimetric Detection, Characterization, and Remote Sensing.

[47] Nicolas Geuquet and Luc Henrard. EELS and optical response of a noble metal nanoparticle in the frame of a discrete dipole approximation. *Ultramicroscopy*, 110(8):1075–1080, 2010.

[48] Jaysen Nelayah, Mathieu Kociak, Odile Stéphan, F Javier García de Abajo, Marcel Tencé, Luc Henrard, Dario Taverna, Isabel Pastoriza-Santos, Luis M Liz-Marzán, and Christian Colliex. Mapping surface plasmons on a single metallic nanoparticle. *Nature Physics*, 3(5):348–353, 2007.

[49] J Nelayah, M Kociak, Odile Stephan, N Geuquet, L Henrard, F Javier García de Abajo, Isabel Pastoriza-Santos, Luis M Liz-Marzan, and C Colliex. Two-dimensional quasistatic stationary short range surface plasmons in flat nanoprisms. *Nano letters*, 10(3):902–907, 2010.

[50] D Rossouw, M Couillard, J Vickery, E Kumacheva, and GA Botton. Multipolar plasmonic resonances in silver nanowire antennas imaged with a subnanometer electron probe. *Nano letters*, 11(4):1499–1504, 2011.

[51] Stefano Mazzucco, Nicolas Geuquet, Jian Ye, Odile Stephan, Willem Van Roy, Pol Van Dorpe, Luc Henrard, and Mathieu Kociak. Ultralocal modification of surface plasmons properties in silver nanocubes. *Nano letters*, 12(3):1288–1294, 2012.

[52] Olivia Nicoletti, Francisco de la Peña, Rowan K Leary, Daniel J Holland, Caterina Ducati, and Paul a Midgley. Three-dimensional imaging of localized surface plasmon resonances of metal nanoparticles. *Nature*, 502(7469):80–4, October 2013.

[53] Anton Hörl, Andreas Trügler, and Ulrich Hohenester. Tomography of particle plasmon fields from electron energy loss spectroscopy. *Physical review letters*, 111(7):076801, 2013.

[54] SL Altmann and AP Cracknell. Lattice harmonics I. Cubic groups. *Reviews of Modern Physics*, 37(1):19, 1965.

[55] Bethany A Bradley, Robert W Jacob, John F Hermance, and John F Mustard. A curve fitting procedure to derive inter-annual phenologies from time series of noisy satellite NDVI data. *Remote Sensing of Environment*, 106(2):137–145, 2007.

[56] Aimee M Morris, Murielle A Watzky, and Richard G Finke. Protein aggregation kinetics, mechanism, and curve-fitting: a review of the literature. *Biochimica et Biophysica Acta (BBA)-Proteins and Proteomics*, 1794(3):375–397, 2009.

[57] Junichiro Shiomi, Keivan Esfarjani, and Gang Chen. Thermal conductivity of half-Heusler compounds from first-principles calculations. *Physical Review B*, 84(10):104302, 2011.

[58] F De la Peña, N Barrett, LF Zagonel, M Walls, and O Renault. Full field chemical imaging of buried native sub-oxide layers on doped silicon patterns. *Surface Science*, 604(19):1628–1636, 2010.

[59] Karl Pearson. Principal components analysis. *The London, Edinburgh and Dublin Philosophical Magazine and Journal*, 6(2):566, 1901.

[60] Christopher M Bishop. Pattern recognition. *Machine Learning*, 128, 2006.

[61] R Arenal, F De la Peña, O Stephan, M Walls, M Tence, A Loiseau, and C Colliex. Extending the analysis of EELS spectrum-imaging data, from elemental to bond mapping in complex nanostructures. *Ultramicroscopy*, 109(1):32–38, 2008.

[62] Douglas M Hawkins. The problem of overfitting. *Journal of chemical information and computer sciences*, 44(1):1–12, 2004.

[63] Karl Pearson. X. on the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 50(302):157–175, 1900.

[64] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, Dec 1974.

[65] Gideon Schwarz. Estimating the dimension of a model. *Ann. Statist.*, 6(2):461–464, 03 1978.

[66] Kenneth Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics*, 2(2):164–168, 1944.

[67] Ronald Aylmer Fisher. *Statistical methods for research workers*. Genesis Publishing Pvt Ltd, 1925.

[68] David G Luenberger, Yinyu Ye, et al. *Linear and nonlinear programming*, volume 2. Springer, 1984.

[69] Arnold Neumaier. Complete search in continuous global optimization and constraint satisfaction. *Acta numerica*, 13:271–369, 2004.

[70] Jorge Nocedal and Stephen J Wright. Numerical optimization 2nd. 2006.

[71] Stephen G Nash. Newton-type minimization via the Lanczos method. *SIAM Journal on Numerical Analysis*, 21(4):770–788, 1984.

[72] Ciyou Zhu, Richard H Byrd, Peihuang Lu, and Jorge Nocedal. Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software (TOMS)*, 23(4):550–560, 1997.

[73] Noel Cressie. Fitting variogram models by weighted least squares. *Journal of the International Association for Mathematical Geology*, 17(5):563–586, 1985.

[74] Craig B Markwardt. Non-linear least squares fitting in IDL with MPFIT. *arXiv preprint arXiv:0902.2850*, 2009.

[75] Margaret A Shipp, Ken N Ross, Pablo Tamayo, Andrew P Weng, Jeffery L Kutok, Ricardo CT Aguiar, Michelle Gaasenbeek, Michael Angelo, Michael Reich, Geraldine S Pinkus, et al. Diffuse large b-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning. *Nature medicine*, 8(1):68–74, 2002.

[76] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

[77] Tom M Mitchell. Machine learning. 1997. *Burr Ridge, IL: McGraw Hill*, 45(37):870–877, 1997.

[78] Robert Burbidge, Matthew Trotter, B Buxton, and Sl Holden. Drug design by machine learning: support vector machines for pharmaceutical data analysis. *Computers & chemistry*, 26(1):5–14, 2001.

[79] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[80] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015.

[81] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47, 2002.

[82] Yann LeCun, Bernhard E Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne E Hubbard, and Lawrence D Jackel. Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems*, pages 396–404, 1990.

[83] Eleazar Eskin. Anomaly detection over noisy data using learned probability distributions. In *In Proceedings of the International Conference on Machine Learning*. Citeseer, 2000.

[84] Robin Sommer and Vern Paxson. Outside the closed world: On using machine learning for network intrusion detection. In *Security and Privacy (SP), 2010 IEEE Symposium on*, pages 305–316. IEEE, 2010.

[85] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[86] Jimmy Ba, Volodymyr Mnih, and Koray Kavukcuoglu. Multiple object recognition with visual attention. *arXiv preprint arXiv:1412.7755*, 2014.

[87] Jason Weston, Frédéric Ratle, Hossein Mobahi, and Ronan Collobert. Deep learning via semi-supervised embedding. In *Neural Networks: Tricks of the Trade*, pages 639–655. Springer, 2012.

[88] Michael E Tipping and Christopher M Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, 1999.

[89] M Bosman, M Watanabe, DTL Alexander, and VJ Keast. Mapping chemical and bonding information using multivariate analysis of electron energy-loss spectrum images. *Ultramicroscopy*, 106(11):1024–1032, 2006.

[90] Gene H Golub and Christian Reinsch. Singular value decomposition and least squares solutions. *Numerische mathematik*, 14(5):403–420, 1970.

[91] William H Press, SA Teukolsky, WT Vetterling, and BP Flannery. Numerical recipes in C: the art of scientific computing, second edition, 1992.

[92] Daniel P Berrar, Werner Dubitzky, Martin Granzow, et al. *A practical approach to microarray data analysis*. Springer, 2003.

[93] Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[94] Pentti Paatero and Unto Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2):111–126, 1994.

[95] Pierre Comon. Independent component analysis, a new concept? *Signal processing*, 36(3):287–314, 1994.

[96] Claude Elwood Shannon. A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(1):3–55, 2001.

[97] Andrew M Fraser and Harry L Swinney. Independent coordinates for strange attractors from mutual information. *Physical review A*, 33(2):1134, 1986.

[98] Pierre Simon marquis de Laplace. *Théorie analytique des probabilités*. V. Courcier, 1820.

[99] Gordon Bell, Tony Hey, and Alex Szalay. Beyond the data deluge. *Science*, 323(5919):1297–1298, 2009.

[100] Gatan Inc. Digital micrograph™ software. http://www.gatan.com/products/tem-analysis/gatan-microscopy-suite-software.

[101] Francisco de la Peña, Tomas Ostasevicius, Vidar Tonaas Fauske, Pierre Burdet, Petras Jokubauskas, Magnus Nord, Eric Prestat, Mike Sarahan, Katherine E. MacArthur, Duncan N. Johnstone, Joshua Taillon, Jan Caron, Tom Furnival, Alberto Eljarrat, Stefano Mazzucco, Vadim Migunov, Thomas Aarholt, Michael Walls, Florian Winkler, Ben Martineau, Gaël Donval, Eric R. Hoglund, Ivo Alxneit, Ida Hjorth, Luiz Fernando Zagonel, Andreas Garmannslund, Christoph Gohlke, Ilya Iyengar, and Huang-Wei Chang. hyperspy/hyperspy: Hyperspy 1.3, May 2017.

[102] Jiashi Feng, Huan Xu, and Shuicheng Yan. Online robust pca via stochastic optimization. In *Advances in Neural Information Processing Systems*, pages 404–412, 2013.

[103] Renbo Zhao and Vincent YF Tan. Online nonnegative matrix factorization with outliers. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2662–2666. IEEE, 2016.

[104] Naiyang Guan, Dacheng Tao, Zhigang Luo, and Bo Yuan. Online nonnegative matrix factorization with robust stochastic approximation. *IEEE Transactions on Neural Networks and Learning Systems*, 23(7):1087–1099, 2012.

[105] Matthew Rocklin. Dask: Parallel computation with blocked algorithms and task scheduling. In Kathryn Huff and James Bergstra, editors, *Proceedings of the 14th Python in Science Conference*, pages 130 – 136, 2015.

[106] Dask Development Team. *Dask: Library for dynamic task scheduling*, 2016.

[107] J Nelayah, L Gu, W Sigle, CT Koch, I Pastoriza-Santos, LM Liz-Marzán, and PA Van Aken. Direct imaging of surface plasmon resonances on single triangular silver nanoprisms at optical wavelength using low-loss EFTEM imaging. *Optics letters*, 34(7):1003–1005, 2009.

[108] RI Davies, F Müller Sánchez, R Genzel, LJ Tacconi, EKS Hicks, S Friedrich, and A Sternberg. A close look at star formation around active galactic nucleibased on observations at the european southern observatory vlt (60. a-9235, 070. b-0649, 070. b-0664, 074. b-9012, 076. b-0098). *The Astrophysical Journal*, 671(2):1388, 2007.

[109] E. K. S. Hicks, R. I. Davies, M. A. Malkan, R. Genzel, L. J. Tacconi, F. Müller Sánchez, and A. Sternberg. The role of molecular gas in obscuring Seyfert active galactic nuclei. *The Astrophysical Journal*, 696(1):448, 2009.

[110] Jorge J Moré. The Levenberg-Marquardt algorithm: implementation and theory. In *Numerical analysis*, pages 105–116. Springer, 1978.

[111] Donald W Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the society for Industrial and Applied Mathematics*, 11(2):431–441, 1963.

[112] John A Nelder and Roger Mead. A simplex method for function minimization. *The computer journal*, 7(4):308–313, 1965.

[113] Michael JD Powell. A fast algorithm for nonlinearly constrained optimization calculations. In *Numerical analysis*, pages 144–157. Springer, 1978.

[114] Elijah Polak and Gerard Ribiere. Note sur la convergence de méthodes de directions conjuguées. *Revue française d'informatique et de recherche opérationnelle, série rouge*, 3(1):35–43, 1969.

[115] Luis Miguel Rios and Nikolaos V. Sahinidis. Derivative-free optimization: a review of algorithms and comparison of software implementations. *Journal of Global Optimization*, 56(3):1247–1293, 2013.

[116] L. F. Zagonel, L. H. G. Tizei, G. Z. Vitiello, G. Jacopin, L. Rigutti, M. Tchernycheva, F. H. Julien, R. Songmuang, T. Ostaševičius, F. de la Peña, C. Ducati, P. A. Midgley, and M. Kociak. Nanometer-scale monitoring of quantum-confined Stark effect and emission efficiency droop in multiple GaN/AlN quantum disks in nanowires. *Phys. Rev. B*, 93:205410, May 2016.

[117] Lord Rayleigh. Xxxi. investigations in optics, with special reference to the spectroscope. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 8(49):261–274, 1879.

[118] Wikipedia. Arg max — wikipedia, the free encyclopedia, 2017. [Online; accessed 17-July-2017].

[119] Ken Aho, DeWayne Derryberry, and Teri Peterson. Model selection for ecologists: the worldviews of AIC and BIC. *Ecology*, 95(3):631–636, 2014.

[120] David Freedman and Persi Diaconis. On the histogram as a density estimator: L 2 theory. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, 57(4):453–476, 1981.

[121] David W Scott. On optimal and data-based histograms. *Biometrika*, pages 605–610, 1979.

[122] Kevin H Knuth. Optimal data-based binning for histograms. *arXiv preprint physics/0605197*, 2006.

[123] Jeffrey D Scargle, Jay P Norris, Brad Jackson, and James Chiang. Studies in astronomical time series analysis. vi. bayesian block representations. *The Astrophysical Journal*, 764(2):167, 2013.

[124] Ery Arias-Castro, David L Donoho, and Xiaoming Huo. Near-optimal detection of geometric objects by fast multiscale methods. *IEEE Transactions on Information Theory*, 51(7):2402–2425, 2005.

[125] mongoDB. Documentation: Replication, 2016. [Online; accessed 13-December-2016].

[126] Redis. Cluster specifications, 2016. [Online; accessed 13-December-2016].

[127] AB MySQL. MySQL, 2001.

[128] Jeffrey Dean and Sanjay Ghemawat. MapReduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.

[129] Matei Zaharia, Mosharaf Chowdhury, Michael J Franklin, Scott Shenker, and Ion Stoica. Spark: cluster computing with working sets. *HotCloud*, 10:10–10, 2010.

[130] William Gropp, Ewing Lusk, Nathan Doss, and Anthony Skjellum. A high-performance, portable implementation of the MPI message passing interface standard. *Parallel computing*, 22(6):789–828, 1996.

[131] Chunye Gong, Jie Liu, Qiang Zhang, Haitao Chen, and Zhenghu Gong. The characteristics of cloud computing. In *2010 39th International Conference on Parallel Processing Workshops*, pages 275–279. IEEE, 2010.

[132] Robert R Schaller. Moore's law: past, present and future. *IEEE spectrum*, 34(6):52–59, 1997.

[133] Paul A Midgley and Rafal E Dunin-Borkowski. Electron tomography and holography in materials science. *Nature materials*, 8(4):271–280, 2009.

[134] Olivia Nicoletti, Francisco de La Peña, Rowan K Leary, Daniel J Holland, Caterina Ducati, and Paul A Midgley. Three-dimensional imaging of localized surface plasmon resonances of metal nanoparticles. *Nature*, 502(7469):80–84, 2013.

[135] Alexander S Eggeman, Robert Krakow, and Paul A Midgley. Scanning precession electron tomography for three-dimensional nanoscale orientation imaging and crystallographic analysis. *Nature communications*, 6, 2015.

[136] Bowen Meng, Guillem Pratx, and Lei Xing. Ultrafast and scalable cone-beam CT reconstruction using MapReduce in a cloud computing environment. *Medical physics*, 38(12):6603–6609, 2011.

[137] Michael C Schatz. CloudBurst: highly sensitive read mapping with MapReduce. *Bioinformatics*, 25(11):1363–1369, 2009.

[138] Aaron McKenna, Matthew Hanna, Eric Banks, Andrey Sivachenko, Kristian Cibulskis, Andrew Kernytsky, Kiran Garimella, David Altshuler, Stacey Gabriel, Mark Daly, et al. The genome analysis toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data. *Genome research*, 20(9):1297–1303, 2010.

[139] Keith Wiley, Andrew Connolly, Jeff Gardner, S Krughoff, Magdalena Balazinska, Bill Howe, Y Kwon, and Yingyi Bu. Astronomy in the cloud: using MapReduce for image co-addition. *Publications of the Astronomical Society of the Pacific*, 123(901):366, 2011.

[140] Hadoop Wiki. Powered by Apache Hadoop, 2016. [Online; accessed 10-November-2016].

[141] Matei Zaharia, Mosharaf Chowdhury, Tathagata Das, Ankur Dave, Justin Ma, Murphy McCauley, Michael J Franklin, Scott Shenker, and Ion Stoica. Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. In *Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation*, pages 2–2. USENIX Association, 2012.

[142] Christopher Peter Wadsworth. *Semantics and Pragmatics of the Lambda-Calculus.* PhD thesis, University of Oxford, 1971.

[143] Peter Henderson and James H Morris Jr. A lazy evaluator. In *Proceedings of the 3rd ACM SIGACT-SIGPLAN symposium on Principles on programming languages*, pages 95–103. ACM, 1976.

[144] D.P. Friedman and D.S. Wise. `cons` should not evaluate its arguments. 1976.

[145] Erik Elmroth, Fred Gustavson, Isak Jonsson, and Bo Kågström. Recursive blocked algorithms and hybrid data structures for dense matrix library software. *SIAM review*, 46(1):3–45, 2004.

[146] Monica D Lam, Edward E Rothberg, and Michael E Wolf. The cache performance and optimizations of blocked algorithms. In *ACM SIGARCH Computer Architecture News*, volume 19, pages 63–74. ACM, 1991.

[147] Alfredo Buttari, Julien Langou, Jakub Kurzak, and Jack Dongarra. A class of parallel tiled linear algebra algorithms for multicore architectures. *Parallel Computing*, 35(1):38–53, 2009.

[148] Eran Rabani and Sivan Toledo. Out-of-core SVD and QR decompositions. In *PPSC*, 2001.

[149] Austin R Benson, David F Gleich, and James Demmel. Direct QR factorizations for tall-and-skinny matrices in MapReduce architectures. In *Big Data, 2013 IEEE International Conference on*, pages 264–272. IEEE, 2013.

[150] Neil P Dasgupta, Jianwei Sun, Chong Liu, Sarah Brittman, Sean C Andrews, Jongwoo Lim, Hanwei Gao, Ruoxue Yan, and Peidong Yang. 25th anniversary article: semiconductor nanowires–synthesis, characterization, and applications. *Advanced materials*, 26(14):2137–2184, 2014.

[151] Yat Li, Fang Qian, Jie Xiang, and Charles M Lieber. Nanowire electronic and optoelectronic devices. *Materials today*, 9(10):18–27, 2006.

[152] Charles M. Lieber. Semiconductor nanowires: A platform for nanoscience and nanotechnology. *MRS Bulletin*, 36(12):1052–1063, Dec 2011.

[153] Shunfeng Li and Andreas Waag. GaN based nanorods for solid state lighting. *Journal of Applied Physics*, 111(7):5, 2012.

[154] Hieu Pham Trung Nguyen, Shaofei Zhang, Kai Cui, Xueguang Han, S Fathololoumi, M Couillard, GA Botton, and Z Mi. p-type modulation doped InGaN/GaN dot-in-a-wire white-light-emitting diodes monolithically grown on Si (111). *Nano letters*, 11(5):1919–1924, 2011.

[155] Dmitry Turchinovich. *Study of ultrafast polarization and carrier dynamics in semiconductor nanostructures: a THz spectroscopy approach*. PhD thesis, Ph. D. Thesis, University of Freiburg, 2004.

[156] Fabio Bernardini, Vincenzo Fiorentini, and David Vanderbilt. Spontaneous polarization and piezoelectric constants of III-V nitrides. *Physical Review B*, 56(16):R10024, 1997.

[157] Mathieu Leroux, Nicolas Grandjean, M Laügt, Jean Massies, Bernard Gil, Pierre Lefebvre, and Pierre Bigenwald. Quantum confined Stark effect due to built-in internal polarization fields in (Al, Ga) N/GaN quantum wells. *Physical Review B*, 58(20):R13371, 1998.

[158] D. A. B. Miller, D. S. Chemla, T. C. Damen, A. C. Gossard, W. Wiegmann, T. H. Wood, and C. A. Burrus. Band-edge electroabsorption in quantum well structures: The Quantum-Confined Stark Effect. *Phys. Rev. Lett.*, 53:2173–2176, Nov 1984.

[159] Pierre Lefebvre and Bruno Gayral. Optical properties of GaN/AlN quantum dots. *Comptes Rendus Physique*, 9(8):816–829, 2008.

[160] Justin Iveland, Lucio Martinelli, Jacques Peretti, James S Speck, and Claude Weisbuch. Direct measurement of Auger electrons emitted from a semiconductor light-emitting diode under electrical injection: identification of the dominant mechanism for efficiency droop. *Physical review letters*, 110(17):177406, 2013.

[161] Joachim Piprek. Efficiency droop in nitride-based light-emitting diodes. *physica status solidi (a)*, 207(10):2217–2225, 2010.

[162] Hideaki Murotani, Hiroya Andoh, Takehiko Tsukamoto, Toko Sugiura, Yoichi Yamada, Takuya Tabata, Yoshio Honda, Masatoshi Yamaguchi, and Hiroshi Amano. Emission wavelength dependence of internal quantum efficiency in InGaN nanowires. *Japanese Journal of Applied Physics*, 52(8S):08JE10, 2013.

[163] Jochen Bruckbauer, Paul R Edwards, Jie Bai, Tao Wang, and Robert W Martin. Probing light emission from quantum wells within a single nanorod. *Nanotechnology*, 24(36):365704, 2013.

[164] James R Riley, Sonal Padalkar, Qiming Li, Ping Lu, Daniel D Koleske, Jonathan J Wierer, George T Wang, and Lincoln J Lauhon. Three-dimensional mapping of quantum wells in a GaN/InGaN core–shell nanowire light-emitting diode array. *Nano letters*, 13(9):4317–4325, 2013.

[165] Lorenzo Rigutti, Ivan Blum, Deodatta Shinde, David Hernández-Maldonado, Williams Lefebvre, Jonathan Houard, François Vurpillot, Angela Vella, Maria Tchernycheva, Christophe Durand, et al. Correlation of microphotoluminescence

spectroscopy, scanning transmission electron microscopy, and atom probe tomography on a single nano-object containing an InGaN/GaN multiquantum well system. *Nano letters*, 14(1):107–114, 2013.

[166] Luiz Fernando Zagonel, Stefano Mazzucco, Marcel Tencé, Katia March, Romain Bernard, Benoît Laslier, Gwénolé Jacopin, Maria Tchernycheva, Lorenzo Rigutti, Francois H Julien, et al. Nanometer scale spectral imaging of quantum emitters in nanowires and its correlation to their atomically resolved structure. *Nano letters*, 11(2):568–573, 2010.

[167] Georg Rossbach, Jacques Levrat, G Jacopin, Mehran Shahmohammadi, J-F Carlin, J-D Ganière, Raphael Butté, Benoit Deveaud, and Nicolas Grandjean. High-temperature Mott transition in wide-band-gap semiconductor quantum wells. *Physical Review B*, 90(20):201308, 2014.

[168] Aurélien David and Michael J Grundmann. Droop in InGaN light-emitting diodes: A differential carrier lifetime analysis. *Applied Physics Letters*, 96(10):103504, 2010.

[169] Francisco de La Peña, Tomas Ostaševičius, Rowan K. Leary, Caterina Ducati, Paul A. Midgley, and Raúl Arenal. Quantitative three-dimensional elemental and bonding mapping of a complex hybrid nanoparticle.

[170] RS Lee, J Gavillet, M Lamy de La Chapelle, A Loiseau, J-L Cochon, D Pigache, J Thibault, and F Willaime. Catalyst-free synthesis of boron nitride single-wall nanotubes with a preferred zig-zag configuration. *Physical Review B*, 64(12):121405, 2001.

[171] A Howie. Image contrast and localized signal selection techniques. *Journal of Microscopy*, 117(1):11–23, 1979.

[172] SJ Pennycook. Z-contrast STEM for materials science. *Ultramicroscopy*, 30(1-2):58–69, 1989.

[173] PA Midgley and M Weyland. 3D electron microscopy in the physical sciences: the development of Z-contrast and EFTEM tomography. *Ultramicroscopy*, 96(3):413–431, 2003.

[174] David L Donoho. Compressed sensing. *IEEE Transactions on information theory*, 52(4):1289–1306, 2006.

[175] Alessandro Mirone, Emmanuel Brun, Emmanuelle Gouillart, Paul Tafforeau, and Jerome Kieffer. The PyHST2 hybrid distributed code for high speed tomographic reconstruction with iterative reconstruction and a priori knowledge capabilities. *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, 324:41–48, 2014.

[176] Bart Goris, Stuart Turner, Sara Bals, and Gustaaf Van Tendeloo. Three-dimensional valency mapping in ceria nanocrystals. *ACS nano*, 8(10):10878–10884, 2014.

[177] Pau Torruella, Raul Arenal, Francisco de la Peña, Zineb Saghi, Lluís Yedra, Alberto Eljarrat, Lluis Lopez-Conesa, Marta Estrader, Alberto Lopez-Ortega, Germán Salazar-Alvarez, et al. 3D visualization of the iron oxidation state in FeO/Fe_3O_4 core–shell nanocubes from electron energy loss tomography. *Nano Letters*, 16(8):5068–5073, 2016.

[178] Ronald N Bracewell. Strip integration in radio astronomy. *Australian Journal of Physics*, 9(2):198–217, 1956.

[179] J Radon. On determination of functions by their integral values along certain multiplicities. *Ber. der Sachische Akademie der Wissenschaften Leipzig,(Germany)*, 69:262–277, 1917.

[180] Joachim Frank. *Electron tomography*. Springer, 1992.

[181] Roy A Crowther and Linda A Amos. Three-dimensional image reconstructions of some small spherical viruses. In *Cold Spring Harbor symposia on quantitative biology*, volume 36, pages 489–494. Cold Spring Harbor Laboratory Press, 1972.

[182] Jeannot Trampert and Jean-Jacques Leveque. Simultaneous iterative reconstruction technique: physical interpretation based on the generalized least squares solution. *J. geophys. Res*, 95(12):553–9, 1990.

[183] Gregory K Wallace. The JPEG still picture compression standard. *IEEE transactions on consumer electronics*, 38(1):xviii–xxxiv, 1992.

[184] Raul Arenal, Odile Stephan, Jean-Lou Cochon, and Annick Loiseau. Root-growth mechanism for single-walled boron nitride nanotubes in laser vaporization technique. *Journal of the American Chemical Society*, 129(51):16183–16189, 2007.

[185] Gatan Inc. Optimize spectrum. http://www.eels.info/how/spectroscopy/optimize-spectrum.

[186] Raúl Arenal de la Concha. *Synthèse de nanotubes de nitrure de bore : études de la structure et des propriétés vibrationnelles et électroniques*. PhD thesis, Université Paris-Sud 11 Orsay, 2005. Thèse de doctorat dirigée par Loiseau, Annick Physique des solides Paris 11 2005.

[187] Carl De Boor. On calculating with b-splines. *Journal of Approximation Theory*, 6(1):50–62, 1972.

[188] Francisco Javier de la Peña Manchón. *Advanced methods for Electron Energy Loss Spectroscopy core-loss analysis*. PhD thesis, Université Paris-Sud 11 Orsay, 2010. Thèse de doctorat dirigée par Colliex, Christian et Walls, Michael.

[189] CC Ahn and OL Krivanek. *EELS atlas*. Gatan, 1983.

[190] Technariumas. Inpainting. https://github.com/Technariumas/Inpainting.

[191] John Emsley. *Nature's building blocks: an AZ guide to the elements*. Oxford University Press, 2011.

[192] Thaddeus B Massalski, Hiroaki Okamoto, PR Subramanian, Linda Kacprzak, and William W Scott. *Binary alloy phase diagrams*, volume 1. American society for metals Metals Park, OH, 1986.

[193] Shunli Shang and Zi-Kui Liu. Thermodynamics of the B–Ca, B–Sr, and B–Ba systems: Applications for the fabrications of CaB_6, SrB_6, and BaB_6 thin films. *Applied Physics Letters*, 90(9):091914, 2007.

[194] Min Chu, Yongke Sun, Umamaheswari Aghoram, and Scott E Thompson. Strain: A solution for higher carrier mobility in nanoscale MOSFETs. *Annual Review of Materials Research*, 39:203–229, 2009.

[195] Oliver Stier, Marius Grundmann, and Dieter Bimberg. Electronic and optical properties of strained quantum dots modeled by 8-band $k \cdot p$ theory. *Physical Review B*, 59(8):5688, 1999.

[196] John D Verhoeven. *Fundamentals of physical metallurgy*. John Wiley & Sons Inc, 1975.

[197] Sigurd Wenner and Randi Holmestad. Accurately measured precipitate–matrix misfit in an Al–Mg–Si alloy by electron microscopy. *Scripta Materialia*, 118:5–8, 2016.

[198] PH Jouneau, A Tardot, G Feuillet, H Mariette, and J Cibert. Strain mapping of ultrathin epitaxial ZnTe and MnTe layers embedded in CdTe. *Journal of applied physics*, 75(11):7310–7316, 1994.

[199] MJ Hÿtch, E Snoeck, and R Kilaas. Quantitative measurement of displacement and strain fields from HREM micrographs. *Ultramicroscopy*, 74(3):131–146, 1998.

[200] Christoph T Koch, V Burak Özdöl, and Peter A van Aken. An efficient, simple, and precise way to map strain with nanometer resolution in semiconductor devices. *Applied Physics Letters*, 96(9):091901, 2010.

[201] Javier Bonet and Richard D Wood. *Nonlinear continuum mechanics for finite element analysis*. Cambridge university press, 1997.

[202] David Cooper, Armand Béché, Jean Michel Hartmann, Veronique Carron, and Jean-Luc Rouvière. Strain mapping for the semiconductor industry by dark-field electron holography and nanobeam electron diffraction with nm resolution. *Semiconductor Science and Technology*, 25(9):095012, 2010.

[203] John Brian Pendry. Low-energy electron diffraction. In *Interaction of Atoms and Molecules with Solid Surfaces*, pages 201–211. Springer, 1990.

[204] BC Larson, Wenge Yang, GE Ice, JD Budai, and JZ Tischler. Three-dimensional X-ray structural microscopy with submicrometre resolution. *Nature*, 415(6874):887–890, 2002.

[205] Mark A Pfeifer, Garth J Williams, Ivan A Vartanyants, Ross Harder, and Ian K Robinson. Three-dimensional mapping of a deformation field inside a nanocrystal. *Nature*, 442(7098):63–66, 2006.

[206] Marcus C Newton, Steven J Leake, Ross Harder, and Ian K Robinson. Three-dimensional imaging of strain in a single ZnO nanorod. *Nature materials*, 9(2):120–124, 2010.

[207] GA Edwards, K Stiller, GL Dunlop, and MJ Couper. The precipitation sequence in Al–Mg–Si alloys. *Acta materialia*, 46(11):3893–3904, 1998.

[208] NanoMEGAS. http://www.nanomegas.com.

[209] HW Zandbergen, SJ Andersen, and J Jansen. Structure determination of Mg_5Si_6 particles in Al by dynamic electron diffraction studies. *Science*, 277(5330):1221–1225, 1997.

[210] SJ Andersen, HW Zandbergen, J Jansen, C Traeholt, U Tundal, and O Reiso. The crystal structure of the β'' phase in Al–Mg–Si alloys. *Acta Materialia*, 46(9):3283–3298, 1998.

Appendix A

Analysis diagrams

This appendix provides data analysis diagrams for the three results chapters, chapters 7 to 9, with corresponding Figs. A.1 to A.3. Circles and rectangles represent operations and their results respectively. Steps that require SAMFire and LazySignal are highlighted in yellow and green respectively.

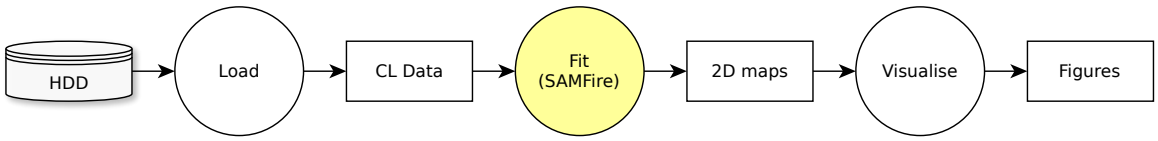


Fig. A.1 The diagram showing chapter 7 analysis workflow. Circles and rectangles represent operations and their results respectively. Steps that require SAMFire are highlighted in yellow.

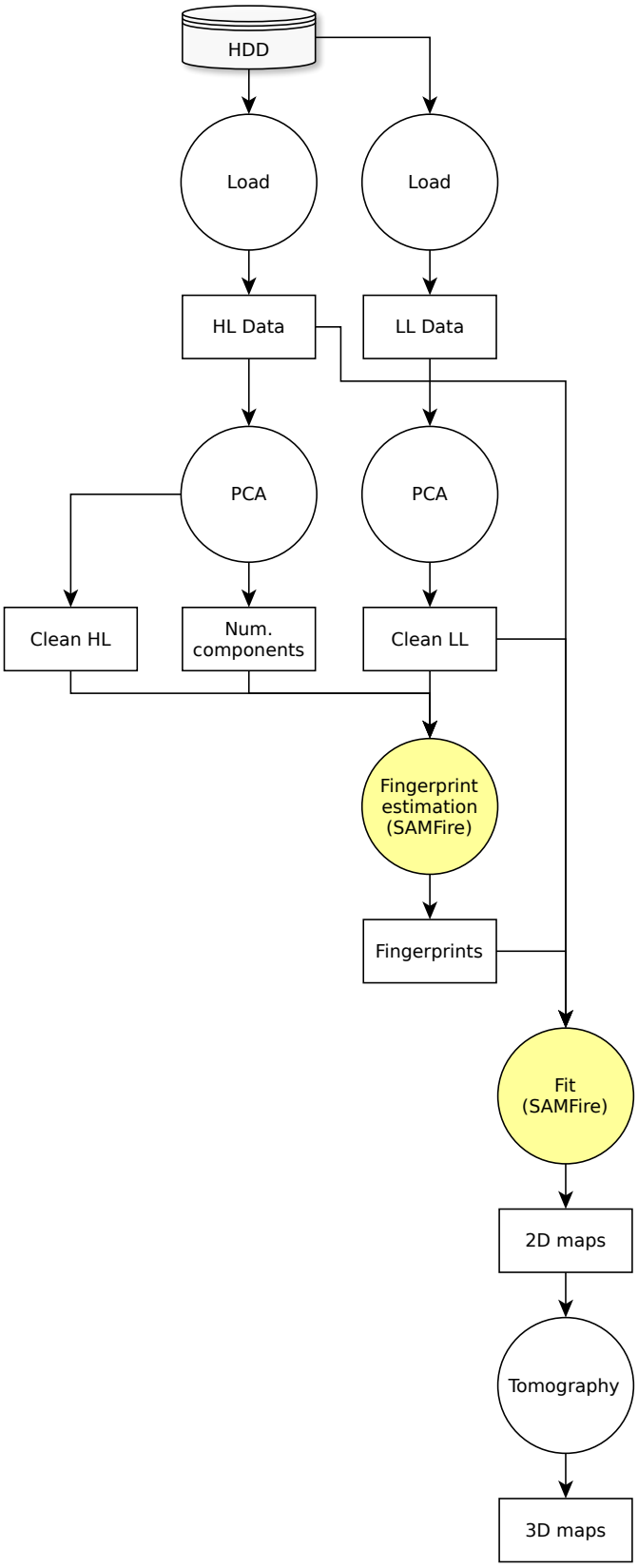


Fig. A.2 The diagram showing chapter 8 analysis workflow. Circles and rectangles represent operations and their results respectively. Steps that require SAMFire are highlighted in yellow.

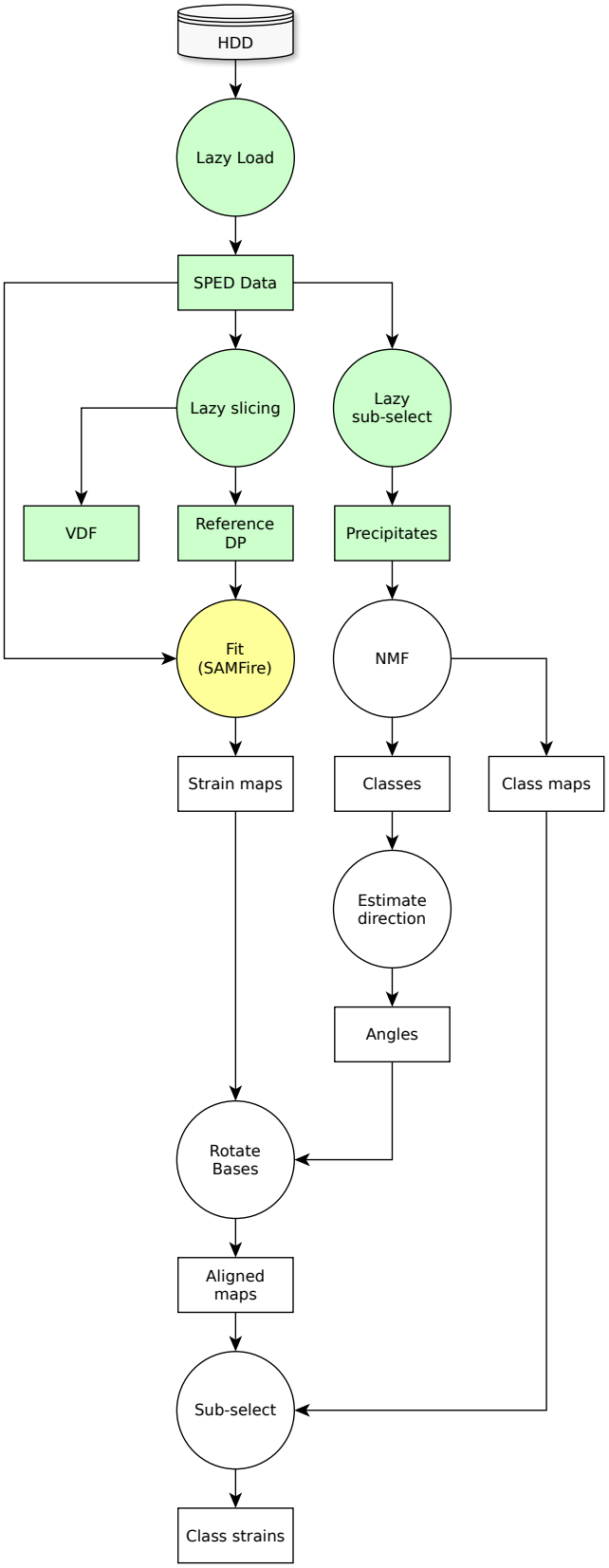


Fig. A.3 The diagram showing chapter 9 analysis workflow. Circles and rectangles represent operations and their results respectively. Steps and results that require SAMFire and LazySignal are highlighted in yellow and green respectively.

Appendix B

Extracting fingerprints code

```
In [ ]: import hyperspy.api as hs
```

```
In [ ]: import numpy as np
        %matplotlib
```

```
In [ ]: cl = hs.load('clean_calibrated_cl_stacked.hdf5', lazy=True)
        ll = hs.load('clean_normalised_fixed_calibrated_stacked_ll.hdf5', lazy=True)
```

```
In [ ]: m = cl.create_model(ll=ll, auto_add_edges=True, auto_background=True)
```

```
In [ ]: m.suspend_auto_fine_structure_width()
```

```
In [ ]: m[1].name = 'O substrate'
        m.append(hs.model.components1D.EELSCLEdge('O_K'))
        m[-1].name = 'O part'


        m.components.B_K.name = 'BN 1'
        m.append(hs.model.components1D.EELSCLEdge('B_K'))
        m[-1].name = 'BN 2'
        m.append(hs.model.components1D.EELSCLEdge('B_K'))
        m[-1].name = 'B metallic'
        m.append(hs.model.components1D.EELSCLEdge('B_K'))
        m[-1].name = 'BO'

        m.components.N_K.name = 'N 1'
        m.append(hs.model.components1D.EELSCLEdge('N_K'))
        m[-1].name = 'N 2'
```

```
In [ ]: m.components

In [ ]: for cn in ['O substrate',
                   'N 1',
                   'Ca_L3',
                   'C_K',
                   'BN 1',
                   'B metallic',
                   "BO"]:
            m[cn].intensity.bmin = 0

        m.components.BN_2.intensity.bmin = None
        m.components.N_2.intensity.bmin = None

In [ ]: m.components.Ca_L3.fine_structure_smoothing = 0.27
        m.components.Ca_L3.fine_structure_width = 7

        m.components.N_1.fine_structure_width = 45.
        m.components.N_1.fine_structure_smoothing = 0.1

        m.components.N_2.fine_structure_width = 45.
        m.components.N_2.fine_structure_smoothing = 0.1

        m.components.O_substrate.fine_structure_width = 40.
        m.components.O_substrate.fine_structure_smoothing = 0.15

        m.components.O_part.fine_structure_width = 40.
        m.components.O_part.fine_structure_smoothing = 0.15

        m.components.BN_1.fine_structure_width = 40.
        m.components.BN_1.fine_structure_smoothing = 0.35

        m.components.BN_2.fine_structure_width = 40.
        m.components.BN_2.fine_structure_smoothing = 0.35

        m.components.B_metallic.fine_structure_width = 40.
        m.components.B_metallic.fine_structure_smoothing = 0.1
```

```
        m.components.B0.fine_structure_width = 40.
        m.components.B0.fine_structure_smoothing = 0.2

        m.components.C_K.fine_structure_width = 50
        m.components.C_K.fine_structure_smoothing = 0.12
```

```
In [ ]: m.components.B0.onset_energy.value = 190.9732998459634
        m.components.BN_1.onset_energy.value = 190.4451714029937
        m.components.BN_2.onset_energy.value = 190.4451714029937

        m.components.N_1.onset_energy.value = 399.7069556719910
        m.components.N_2.onset_energy.value = 399.7069556719910

        m.components.Ca_L3.onset_energy.value = 347.0857902879798

        m.components.O_substrate.onset_energy.value = 533.3136858795248
        m.components.O_part.onset_energy.value = 533.3136858795248
```

B.0.1 Getting fine structure fingerprints

O$_{substrate}$

```
In [ ]: m.axes_manager.indices = (69, 14, 4)
        m.enable_fine_structure(edges_list=['O substrate'])
```

```
In [ ]: m.disable_edges()
        m.components.O_substrate.active = True
        m.set_signal_range(480., 590.)
        m.remove_signal_range(573., 575.)
```

```
In [ ]: m.two_area_background_estimation()
        m.fit(fitter='mpfit', bounded=True)
```

```
In [ ]: m.components.O_substrate.fine_structure_coeff.free = False
        m.assign_current_values_to_all(components_list=['O substrate',])
```

B_2O_3 and $O_{particle}$

```
In [ ]: m.axes_manager.indices = (74, 48, 4)
        el = ['BO', 'O part', 'O substrate']
        m.enable_fine_structure(edges_list=el[:-1])

In [ ]: m.disable_edges()
        m.enable_edges(edges_list=el[1:])

In [ ]: # for the Oxigen edge fitting:
        m.set_signal_range(480., 590.)
        m.remove_signal_range(573., 575.)
        m.two_area_background_estimation()
        m.fit(fitter='mpfit', bounded=True)

In [ ]: # for the Boron edge fitting:
        m.set_signal_range(None, 270.)
        m.disable_edges()
        m.enable_edges(edges_list=el[:1])
        m.components.PowerLaw.r.value = 3.4
        m.components.PowerLaw.A.value = 9.3e10
        m.components.BO.intensity.value = 2e-5
        m.two_area_background_estimation()
        m.fit(fitter='mpfit', bounded=True)

In [ ]: ox_ratio = m.components.O_part.intensity.value / m.components.BO.intensity.va
        print(ox_ratio)

In [ ]: m.components.BO.fine_structure_coeff.free = False
        m.components.O_part.fine_structure_coeff.free = False

        m.assign_current_values_to_all(components_list=['BO', 'O part'])

In [ ]: m.plot()
```

BN and N2 - first edge

```
In [ ]: m.axes_manager.indices = (35, 60, 4)
        el = ['BN 1', 'N 1']
        m.enable_fine_structure(edges_list=el)
```

```
In [ ]: m.disable_edges()
        m.enable_edges(edges_list=el)

In [ ]: # for the Nitrogen edge fitting:
        m.set_signal_range(370., 480.)
        m.components.BN_1.active = False
        m.two_area_background_estimation()
        m.fit(fitter='mpfit', bounded=True)

In [ ]: # for the Boron edge fitting:
        m.set_signal_range(None, 270.)
        m.components.BN_1.active = True
        m.components.N_1.active = False
        m.two_area_background_estimation()
        m.fit(fitter='mpfit', bounded=True)

In [ ]: n1_ratio = m.components.N_1.intensity.value / m.components.BN_1.intensity.val
        print(n1_ratio)

In [ ]: m.components.N_1.fine_structure_coeff.free = False
        m.components.BN_1.fine_structure_coeff.free = False

        m.assign_current_values_to_all(components_list=el)
```

BN and N2 - Second edges

```
In [ ]: # m.axes_manager.indices = (54, 40, 4)
        m.axes_manager.indices = (54,41, 4)
        el = ['BN 1', 'BO', 'BN 2']
        m.disable_edges()
        m.enable_fine_structure(edges_list=['BN 2', 'N 2'])
        #

In [ ]: # for the Oxigen edge fitting:
        m.set_signal_range(480., 800.)
        m.disable_edges()
        m.components.O_part.active = True
        m.components.O_substrate.active = True
        m.two_area_background_estimation()
```

```
        m.fit(fitter='mpfit', bounded=True)
        #
```

In [ ]:
```
        # for the Boron edge fitting:
        m.set_signal_range(None, 270.)
        m.components.BN_1.intensity.free = True
        m.components.BO.intensity.free = True
        m.disable_edges()
        m.enable_edges(edges_list=el[:-1])

        m.two_area_background_estimation()
        m.fit(fitter='mpfit', bounded=True)
        m.remove_signal_range(206., 280.)
        m.remove_signal_range(188., 196.)
        m.fit(fitter='mpfit', bounded=True)

        m.components.BN_1.intensity.free = False
        m.components.BO.intensity.free = False
        m.set_signal_range(None, 270.)
        m.components.BN_2.active = True
        m.components.BN_2.fine_structure_coeff.value = tuple(np.zeros(61).tolist())
        m.fit()
        m.components.BN_1.intensity.free = True
        m.components.BO.intensity.free = True
        #
```

In [ ]:
```
        # for the Nitrogen edge fitting:
        m.set_signal_range(370., 480.)
        m.disable_edges()
        m.components.N_1.active = True
        m.components.N_2.active = True
        m.components.N_2.intensity.value = 0.
        m.two_area_background_estimation()
        m.fit(fitter='mpfit', bounded=True)
```

In [ ]:
```
        m.components.BN_2.fine_structure_coeff.free = False
        m.components.N_2.fine_structure_coeff.free = False
```

```
        m.assign_current_values_to_all(components_list=['BN 2' ,'N 2'])
```

```
In [ ]: print (m.components.BN_1.intensity.value / m.components.BN_2.intensity.value)
        print (m.components.N_1.intensity.value / m.components.N_2.intensity.value)
```

```
In [ ]: bn_n1 = m.components.BN_1.intensity.value / m.components.N_1.intensity.value
        bn_n2 = m.components.BN_2.intensity.value / m.components.N_2.intensity.value
        print (bn_n1)
        print (bn_n2)
```

Metallic B

```
In [ ]: m.axes_manager.indices = (41, 32, 4)
        m.disable_edges()
        el = ['BN 1', 'B metallic', 'BO', 'BN 2']
        m.components.B_metallic.fine_structure_active = True
        m.components.B_metallic.fine_structure_coeff.free = True
```

```
In [ ]: # for the Nitrogen edge fitting:
        m.set_signal_range(370., 480.)
        m.components.N_1.active = True
        m.components.N_2.active = True
        m.two_area_background_estimation()
        m.fit(fitter='mpfit', bounded=True)
```

```
In [ ]: # for the Oxigen edge fitting:
        m.set_signal_range(480., 590.)
        m.disable_edges()
        m.components.O_part.active = True
        m.components.O_substrate.active = True
        m.two_area_background_estimation()
        m.fit(fitter='mpfit', bounded=True)
```

```
In [ ]: # for the Boron edge fitting:
        m.set_signal_range(None, 270.)
        m.disable_edges()
        m.enable_edges(edges_list=el)
```

```
        m.components.BN_1.intensity.value = m.components.N_1.intensity.value * bn_n1
        m.components.BN_2.intensity.value = m.components.N_2.intensity.value * bn_n2
        m.components.BN_1.intensity.free = False
        m.components.BN_2.intensity.free = False

        m.components.BO.intensity.value = m.components.O_part.intensity.value
        m.components.BO.intensity.value /= ox_ratio

        # BO free to float, but starting point is the previous ratio
        m.components.BO.intensity.free = True

        m.two_area_background_estimation()
        m.components.B_metallic.intensity.value = 5e-3
        m.fit(fitter='mpfit', bounded=True)
```

```
In [ ]: m.components.B_metallic.fine_structure_coeff.free = False

        m.components.BN_1.intensity.free = True
        m.components.BN_2.intensity.free = True
        m.components.BO.intensity.free = True

        m.assign_current_values_to_all(components_list=['B metallic'])
```

C edge

```
In [ ]: m.axes_manager.indices = (56, 74, 4)
        # for the Carbon and Calcium edge fitting:
        m.set_signal_range(250., 380.)
        m.disable_edges()
        el = ['Ca_L3', 'Ca_L2', 'Ca_L1', 'C_K']
        m.enable_edges(edges_list=el[-1:])
```

```
In [ ]: m.enable_fine_structure(edges_list=el[-1:])
        m.two_area_background_estimation()
        m.fit(fitter='mpfit', bounded=True)
```

```
In [ ]: m.components.C_K.fine_structure_coeff.free = False
        m.assign_current_values_to_all(components_list=['C_K'])
```

Ca edge

```
In [ ]: m.axes_manager.indices = (71, 45, 4)
        m.set_signal_range(None, 380.)
        el = ['Ca_L3', 'C_K', 'BO', 'BN 1', 'BN 2']
        m.enable_edges(edges_list=el)

In [ ]: m.enable_fine_structure(edges_list=el[:1])

In [ ]: m.two_area_background_estimation()
        m.fit(fitter='mpfit', bounded=True)

In [ ]: m.components.Ca_L3.fine_structure_coeff.free = False
        m.assign_current_values_to_all(components_list=['Ca_L3'])
```

Appendix C

Extracting strain code

In [ ]:
```
%matplotlib
import hyperspy.api as hs
import numpy as np
from scipy import ndimage as ndi
```

In [ ]:
```
s = hs.load('./20160623-Sigurd5.blo',
            lazy=True).as_lazy()
st = s.transpose(signal_axes=s.axes_manager.navigation_axes)
```

In [ ]:
```
st.plot()
```

In [ ]:
```
m = s.create_model()
rect = hs.roi.RectangularROI(left=343.9,
                             top=651.7,
                             right=381.9,
                             bottom=691.6)
ref = rect(s).mean()
if ref._lazy:
    ref.compute()
ref.change_dtype('float')
comp = hs.model.components2D.ScalableFixedPattern2D(ref)
m.append(comp)
```

In [ ]:
```
ref.plot()
```

In [ ]:
```
m.fit()
```

In [ ]:
```
m.print_current_values()
```

C.1 SAMFire

```
In [ ]: samf = m.create_samfire()
        samf.remove(1)
        samf.plot_every = 30

In [ ]: samf.refresh_database()

In [ ]: samf.plot()

In [ ]: samf.start()
        while samf.pool.collect_results():
            pass

In [ ]: m.save_parameters2file('AlMgSi_parameters_bottom_ref')

In [ ]: m.corr.save('bottom_ref_correlation')
```

C.2 Plotting Results

```
In [ ]: sm = m.as_signal()

In [ ]: hs.plot.plot_signals([s, sm])
```

C.3 Strain tensor

```
In [ ]: import numpy as np
        from scipy import linalg
        import math

        dp =s
        def construct_displacement_gradient(ref):
            shape = (dp.axes_manager.navigation_shape[1],
                     dp.axes_manager.navigation_shape[0],
                     3,
                     3)
            D = hs.signals.BaseSignal(np.ones(shape))
            D.axes_manager.set_signal_dimension(2)
```

```
        D.data[:,:,0,0] = ref.d11.map['values']
        D.data[:,:,1,0] = ref.d12.map['values']
        D.data[:,:,2,0] = 0.
        D.data[:,:,0,1] = ref.d21.map['values']
        D.data[:,:,1,1] = ref.d22.map['values']
        D.data[:,:,2,1] = 0.
        D.data[:,:,0,2] = 0.
        D.data[:,:,1,2] = 0.
        D.data[:,:,2,2] = 1.

        return D


    def _transform_basis2D(D, angle):
        """Method to transform the 2D basis in which
        a 2nd-order tensor is described.

        Parameters
        ----------
        D : 3x3 matrix

        angle : float
            The anti-clockwise rotation angle between
            the diffraction x/y axes and the x/y axes
            in which the tensor is to be specified.
        Returns
        -------
        T : TensorField
            Operates in place, replacing the original
            tensor field object with a tensor field
            described in the new basis.
        """

        a=angle*np.pi/180.0
        r11 = math.cos(a)
```

```
    r12 = math.sin(a)
    r21 = -math.sin(a)
    r22 = math.cos(a)
    R = np.array([[r11, r12, 0.],
                  [r21, r22, 0.],
                  [0.,  0.,  1.]])

    T = np.dot(np.dot(R, D), R.T)
    return T


def polar_decomposition(D, side='right'):
    shape = (dp.axes_manager.navigation_shape[1],
             dp.axes_manager.navigation_shape[0],
             3,
             3)
    R = hs.signals.BaseSignal(np.ones((shape))
    R.axes_manager.set_signal_dimension(2)
    U = hs.signals.BaseSignal(np.ones(shape))
    U.axes_manager.set_signal_dimension(2)

    for z, indices in zip(
        D._iterate_signal(),
        D.axes_manager._array_indices_generator()
    ):
        thing = linalg.polar(D.data[indices], side=side)
        R.data[indices] = thing[0]
        U.data[indices] = thing[1]

    return R, U


def get_rotation_angle(R):
    """Return the
    Parameters
    ----------
```

```
            R : RotationMatrix
                RotationMatrix two dimensional signal object of the form:
                 cos x  sin x
                -sin x  cos x
            Returns
            -------
            angle : float
            """
            arr_shape = (R.axes_manager._navigation_shape_in_array
                         if R.axes_manager.navigation_size > 0
                         else [1, ])
            T = np.zeros(arr_shape, dtype=object)

            for z, indices in zip(
                R._iterate_signal(),
                R.axes_manager._array_indices_generator()
            ):
                T[indices] = -math.asin(R.data[indices][1,0])

            X = hs.signals.Signal2D(T.astype(float))

            return X
```

In [ ]:
```
disp = construct_displacement_gradient(comp)

R, U = polar_decomposition(disp)

theta = get_rotation_angle(R)
```

In [ ]:
```
disp.T.plot()
```

In [ ]:
```
theta.plot()
```

In [ ]:
```
e11 = U.isig[0,0]
e11 = e11.as_signal2D(image_axes=e11.axes_manager.navigation_axes)
e11.data = 1. - e11.data

e12 = U.isig[0,1]
```

```
        e12 = e12.as_signal2D(image_axes=e12.axes_manager.navigation_axes)
        e12.data = e12.data

        e21 = U.isig[1,0]
        e21 = e21.as_signal2D(image_axes=e21.axes_manager.navigation_axes)
        e21.data = e21.data

        e22 = U.isig[1,1]
        e22 = e22.as_signal2D(image_axes=e22.axes_manager.navigation_axes)
        e22.data = 1. - e22.data

        strain_results = hs.signals.Signal2D(np.ones((4,
                dp.axes_manager.navigation_shape[1],
                dp.axes_manager.navigation_shape[0])))

        strain_results.data[0] = e11.data
        strain_results.data[1] = e22.data
        strain_results.data[2] = e12.data
        strain_results.data[3] = theta.data

        for ax1, ax2 in zip(strain_results.axes_manager.signal_axes,
                            s.axes_manager.navigation_axes):
            ax1.scale = ax2.scale
            ax1.offset = ax2.offset
            ax1.units = ax2.units
            ax1.name = ax2.name

        strain_results.plot(cmap='RdBu_r')#, vmin=-0.03, vmax=0.03)
```

In [ ]:
```
strain_results.save('./strain_results_bottom_ref', overwrite=True)
```

In [ ]:
```
strain_results.plot(cmap='RdBu_r', scalebar_color='k')
```