Superlattices in van der Waals materials: a low-energy electron microscopy study

Jong, T.A. de

3

Quantitative analysis of spectroscopic LEEM data

By aiming for better microscopy pictures,
new science will typically follow.

For many complex materials systems, low-energy electron microscopy (LEEM) offers detailed insights into morphology and crystallography by naturally combining real-space and reciprocal-space information. Its unique strength, however, is that all measurements can easily be performed energy-dependently. Consequently, one should treat LEEM measurements as multi-dimensional, spectroscopic datasets rather than as images to fully harvest this potential. Here we describe a measurement and data analysis approach to obtain such quantitative spectroscopic LEEM datasets with high lateral resolution. The employed detector correction and adjustment techniques enable measurement of true reflectivity values over four orders of magnitudes of intensity. Moreover, we show a drift correction algorithm, tailored for LEEM datasets with inverting contrast, that yields sub-pixel accuracy without special computational demands. Finally, we apply dimension reduction techniques to summarize the key spectroscopic features of datasets with hundreds of images into two single images that can easily be presented and interpreted intuitively. We use cluster analysis to automatically identify different materials within the field of view and to calculate average spectra per material. We demonstrate these methods by analyzing bright-field and dark-field datasets of few-layer graphene grown on silicon carbide and provide a high-performance Python implementation.

Parts of this chapter have been published as T. A. de Jong, D. N. L. Kok, A. J. H. van der Torren, H. Schopmans, R. M. Tromp, S. J. Van der Molen and J. Jobst, Ultramicroscopy **213** 112913 (2020) [35].

3.1 Introduction

In this Chapter, we describe the methodology developed to optimize the extraction of information out of LEEM data. LEEM has proven to be a versatile tool, due to its damage-free, real-time imaging capabilities and its combination of electron diffraction with spectroscopic, and real-space information. This enables more advanced LEEM-based techniques such as dark-field imaging, where electrons from a single diffracted beam are used to create a real-space image, revealing spatial information on the atomic lattice of the sample [36, 37].

Aside from usage as an imaging tool, LEEM is frequently used as a tool for quantitative analysis of physical properties of a wide range of materials. Multi-dimensional datasets can be created by recording LEEM images as a function of one or more parameters such as interaction energy E_0, angle of incidence, or temperature [29, 38]. Using this, a wide range of properties can be studied, for example, layer interaction, electron bands [34], layer stacking [37], catalysis [39], plasmons [40], and surface corrugation [41].

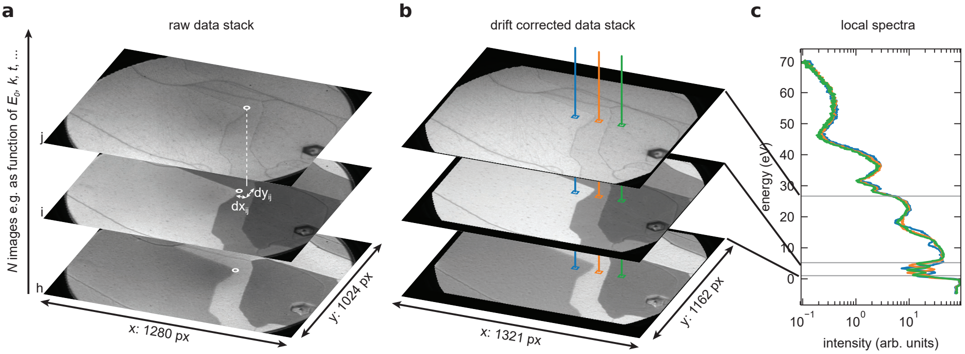

Figure 3.1: ***a,*** *A stack of raw LEEM images where images are shifted with respect to each other due to experimental drift.* ***b,*** *Drift correction aligns features in the detector-corrected images to compensate for this drift.* ***c,*** *Spectra corresponding to pixels indicated in (b).*

However, to unlock the true potential of quantitative analysis of multi-dimensional LEEM data, post-processing of images and combination with meta-data is needed. In particular, it is necessary to correct for detector artifacts and image drift and to convert image intensity to physical quantities.

To this end, we here present a modular data acquisition and analysis pipeline for multi-dimensional LEEM data, combining techniques well established in other fields such as general astronomy or transmission electron microscopy (TEM), that yields high resolution spectroscopic datasets and visualizations thereof. In particular, starting with the raw data (shown in Figure 3.1**a**), we correct for detector artifacts using flat field and dark current correction. Combining these corrections on the images with active feedback on detector gain enables High Dynamic Range (HDR) spectroscopy, which makes it possible to measure spectra over four orders of magnitude of intensity. Subsequently, we demonstrate that compensation of detector artifacts also enables drift correction with sub-pixel accurate image registration, yielding a fully corrected data stack (Figure 3.1**b**).

This creates a true pixel-by-pixel spectroscopic dataset, as shown in Figure 3.1**c**, i.e. every pixel contains a reflectivity spectrum of the corresponding position on the sample. Finally, we explore the potential for more advanced computational data analysis. We show that by using relatively simple dimension reduction techniques and clustering, these datasets can be intuitively analyzed and visualized, enabling semi-automatic identification of areas with different spectra. The implementation of the methodology presented here, is programmed in Python and published as open source at Ref. [42].

To demonstrate these features and quantify the accuracy, we apply the drift correction algorithm to artificial data and then apply the full pipeline to a real dataset.The sample of the dataset is few-layer graphene grown by thermal decomposition of silicon carbide (SiC) [43], followed by hydrogen intercalation to decouple the graphene from the SiC substrate [44, 45]. [1] Bright Field LEEM spectra can be used to distinguish the resulting mixture of bilayer, trilayer and thicker graphene, as interlayer states cause distinct minima in the reflectivity spectra [30, 46]. In addition, the growth process causes strain-induced stacking domains, which can be distinguished using Dark Field LEEM spectra [37, 47]. The sample dataset consists of bright-field and dark-field LEEM images of the same area for a range of landing energies (sometimes referred to as LEEM-I(V)). The dark-field dataset uses a first order diffraction spot and tilted illumination such that the incident beam has the opposite angle to the normal as the diffracted beam, as described in more detail in Chapter 4. The data is available as open data [48] and is interpreted and investigated in detail in Chapter 4.

3.2 DETECTOR CORRECTION

No physical detector system is perfect, i.e. each detector system introduces systematic errors and noise. Knowledge of the sources of these imperfections enables the correction of most of them. The ESCHER LEEM has the classical detector layout: A chevron microchannel plate array (MCP, manufactured by Hamamatsu) for electron multiplication, a phosphor screen to convert electrons to photons and a CCD camera (a PCO sensicam SVGA) to record images of the phosphor screen.

The CCD introduces artifacts in the form of added dark counts and a non-uniform gain [49–52]. Furthermore, the MCP gain is also spatially non-uniform, for example due to overexposure damaging of the MCP, resulting in locally reduced gain. Therefore we describe the measured intensity I_{CCD} on the CCD as the following combination of the previously named detector artifacts and the 'true' signal I_{in}:

$$I_{\text{CCD}}(x,y) = DC(x,y) + I_{\text{in}}(x,y) \cdot G(x,y) \tag{3.1}$$

Where $DC(x,y)$ is the intensity caused by dark current and $G(x,y)$ is the position-dependent and as-of-yet unknown gain factor comprising *all* modifications to the gain due to the complete detector system comprised of the MCP, phosphor screen and CCD camera together.

To compensate for these detector artifacts, we employ techniques well-established in astronomy (and other fields using CCD cameras) to effectively invert the relation in

[1] We thank Christian Ott and Heiko Weber for the fabrication of the graphene on SiC samples

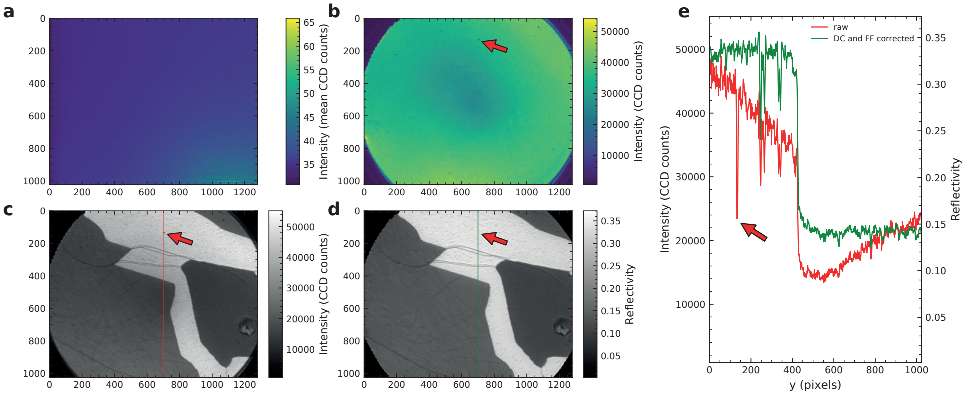

***Figure 3.2:* a,** *Dark Count image taken on ESCHER averaged over* 19 × 16 *images with* 250 ms *exposure time.* **b,** *Flat field image with visible edges of the round microchannel plate and damaged areas (arrow).* **c,** *Uncorrected bright-field LEEM image from the sample dataset. The field of view corresponds to* 3.5 µm*.* **d,** *Dark count and flat field corrected version of the image in (c).* **e,** *Line cut through the raw image (c) and the corrected image (d) shown in red and green, respectively. Note that the dip due to MCP damage at* $y = 140$ *(arrow) is removed and the profiles for similar areas are flattened.*

Eq. (3.1), to extract $I_{\text{in}}(x, y)$ without the deleterious effects of background dark counts $DC(x, y)$ and local gain variations $G(x, y)$.

First, the dark current of the CCD is compensated by pixel-wise subtracting a non-illuminated *dark count* image, i.e. an image with the same exposure time as used for the measurement, but no electron illumination at all. A pixel-wise average of a set of such dark count images is shown in Figure 3.2**a**. The dark current arises from thermal excitations in the sensor and varies over time with an approximately Gaussian distribution. The mean of this distribution is dependent on the pixel, i.e. the x, y location, for example visible in Figure 3.2**a** as a slight increase in the lower right corner. To suppress the thermal fluctuations in the template dark current image, it is desirable to average over several dark count images to prevent the introduction of systematic errors. We assume that the per-pixel dark currents are identically distributed with a variance $\text{Var}_{\text{therm}}$ except for a spatial variance $\text{Var}_{\text{spatial}}$ of the mean. This is mathematically equivalent to assuming the dark current fluctuates around its mean with both spatially dependent (but fixed in time) noise and time-dependent thermal noise. By averaging multiple dark images, we reduce the thermal variance but not the spatial variance. The remaining variance is given by:

$$\begin{aligned}\text{Var}_{\text{tot}}(n) &= \text{Var}_{\text{therm}}(n) + \text{Var}_{\text{spatial}} \\ &= \frac{1}{n}\text{Var}_{\text{therm}}(1) + \text{Var}_{\text{spatial}}\end{aligned}$$

Where $\text{Var}(n)$ is used to denote the variance of n pixel-wise averaged images. By determining $\text{Var}_{\text{tot}}(n)$ and $\text{Var}_{\text{tot}}(1)$ experimentally we can isolate the thermal noise on a single image:

$$\mathrm{Var}_{\mathrm{therm}}(1) = \left[\mathrm{Var}_{\mathrm{tot}}(1) - \mathrm{Var}_{\mathrm{tot}}(n)\right] \cdot \frac{n}{n-1} \tag{3.2}$$

For the ESCHER system with its Peltier-cooled camera, we find:

$$\mathrm{Var}_{\mathrm{therm}}(16 \times 250\,\mathrm{ms\ image}) = 114.3.$$

Therefore, by taking $n = 120$, a set of 120×16 images (a total exposure time of 8 minutes) is sufficient to suppress the systematic errors to values smaller than the discretization error. We find that the dark count image does not significantly change over time, and therefore remeasuring dark count images is seldomly needed.

Second, to compensate for spatial gain variations, which are mostly due to the MCP, a (conventional) *flat field* correction is performed, dividing the full dark count-corrected dataset by an evenly illuminated image [53]. In LEEM, when the potential of the sample corresponds to a higher potential electron energy than the kinetic energy of the electrons of the beam, all incoming electrons turn around before they reach the sample. In this situation the electron landing energy is negative and the sample behaves as a mirror. Imaging at a landing energy of $E_0 \approx -20\,\mathrm{eV}$ yields an almost perfect flat field image as approximation of $G(x, y)$ in Eq. (3.1). A relatively large value for the negative energy is taken to prevent artifacts from local in-plane electric field components, e.g. due to work function or height differences in the sample [54–56]. For the ESCHER system, it is necessary to take flat field images within hours of the measurement, as the MCP wears over time and the gun emission profile and system alignment change on relatively short timescales [57]. Furthermore, taking a flat field image at the same precise alignment as the measurement is preferred for two reasons. First, barring absolutely perfect alignment of the system as well as a perfectly uniform emission from the electron gun, the beam intensity is not spatially uniform. As illumination inhomogeneities are dependent on the precise settings of the lenses, these will also be compensated for if the flat field is recorded in the exact same configuration. Second, for proper normalization of the data, as explained in Section 3.3, the same magnification (projector settings) is needed.

An alternative to this mirror mode flat fielding is to average over a sufficiently large set of images of different positions on the sample and use the resulting average as a flat field image. In most cases however, mirror mode flat fielding is preferred over such ensemble-average flat fielding since for the latter many images of different locations are required. Even when such a set is already available, it is hard to rule out any systematic (statistical) errors. Lastly ensemble-average cannot provide proper normalization of the data to convert to true reflectivity.

3.3 HIGH DYNAMIC RANGE SPECTROSCOPY

In LEEM and LEED, large variations occur in the amplitude of the signal, both within individual images and from image to image. For example, in quantitative LEED, features of interest are often orders of magnitude less bright than primary Bragg peaks. This necessitates a detector system with a large dynamic range. The CCD-camera of the ESCHER setup has a bit depth of 12 bits and a possibility to accumulate 16 images in hardware, yielding an effective bit depth of 16 bits for singular images.

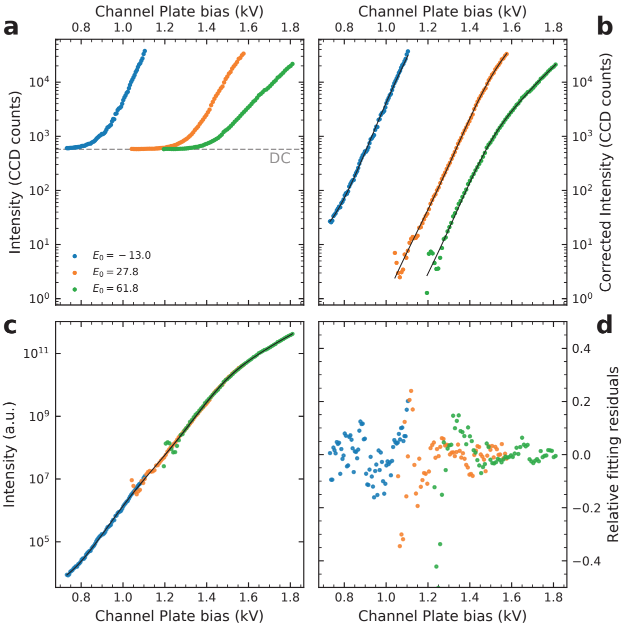

Figure 3.3:* a,** *Calibration curves as measured with 16 × 250 ms exposure time per image measured on graphene on SiC.* ***b, *Calibration curves corrected for dark count.* ***c,*** *Calibration curves with matched intensity and normalized by joint curve fit of Eq. (3.3) and resulting best fit (black line).* ***d,*** *Residuals of the joint fit in (c).*

For most materials, the reflected intensity $I(E_0)$ changes over orders of magnitude as a function of E_0. Starting in mirror mode with unity reflectivity, the reflected intensity tends to decrease orders of magnitude for $E_0 \lesssim 100\,\text{eV}$. To obtain spectra with such a large dynamic range, the dynamic range offered by the bit depth of the CCD alone is not sufficient.

However, the gain G of the MCP, i.e. the ratio of outgoing electrons to incoming electrons, can be tuned by the voltage V_{MCP} applied over the MCP. This gain scales approximately exponential in V_{MCP} (over a reasonable range, see next section), enabling image formation of approximately constant intensity on the CCD, for a wide range of incident electron intensities. We use this property to develop a scheme to further increase the dynamic range in which $G(V_{\mathrm{MCP}})$ is adjusted by setting a new MCP bias for each new image, i.e. increasing the gain for images where the reflected intensity is low. Measuring V_{MCP} for each recorded image and calibrating $G(V_{\mathrm{MCP}})$ makes it possible to employ the full dynamic range of the CCD-camera for all landing energies, without losing the information of the absolute magnitude of the measured intensities, thus extending the range of spectroscopy without significant decrease in signal-to-noise ratio.

3.3.1 CALIBRATION

Hamamatsu Photonics K.K., the manufacturer of the microchannel plate in the ESCHER setup, specifies an exponential gain as function of voltage for a part of the range of possible biases [58]. To extend the useful range beyond this limit and thus enable the use of the full bias range up to the maximal 1800 V, the gain versus bias curve was calibrated as follows:

1. First, in mirror mode, V_{MCP} is adjusted such that the maximum intensity in the image corresponds to the full intensity on the CCD, staying just below intensities damaging the MCP.

2. While decreasing V_{MCP}, images are acquired for evenly spaced bias values. The intensities of these images form the dataset for calibration of the low bias part.

3. Returning V_{MCP} to the previous maximum value, E_0 is increased until the intensity of the image is so low that it is barely distinguishable from the dark count.

4. Again V_{MCP} is turned up until the maximum image intensity corresponds to the maximum CCD intensity.

5. Steps 2. to 4. are repeated until a dataset is acquired starting at maximum MCP bias V_{MCP}. The resulting average intensity curves are shown in Figure 3.3**a**.

6. These datasets are then corrected for dark count as discussed above, resulting in the average intensity curves shown in Figure 3.3**b**. Comparing to the uncorrected curves, the increase in accuracy for low intensity values, crucial for accurate calibration, is very apparent. Averaging over a sufficiently large area ensures sufficient reduction of other noise sources such as MCP noise.

7. A joint fit of Eq. (3.3), allowing for a different amplitude A_i for each curve, is performed to the corrected data to obtain a general expression for MCP gain G as a function of V_{MCP}. The fit is performed using least squares on the logarithm of the original data with no additional weights, to ensure a good fit over the large range of orders of magnitude. The fitted curve is then normalized to a convenient value, e.g. $G(1\,\text{kV}) = 1$. This normalization can be freely chosen, as G will be applied equally to datasets and flat field images, yielding absolute reflectivity as resulting data.

A first choice for a fitting function would be a simple exponential, but this would not account for any deviation from perfect exponential gain, visible as deviations from a straight line in Figure 3.3**b**. For the ESCHER setup we therefore choose to add correction terms of odd power in the exponent:

$$G(V_{\text{MCP}}) = A_i \exp\left(\sum_{k=0}^{8} c_k {V_{\text{MCP}}}^{2k+1}\right) \tag{3.3}$$

Only odd powers were used to accurately capture the visible trends in the data. For the ESCHER setup correction terms up to order V_{MCP}^{17} ($k < 9$) turned out to give a satisfactory good approximation, as illustrated by the residuals in Figure 3.3**d**.

3.3.2 ACTIVE PER-IMAGE OPTIMIZATION OF MCP BIAS

The resulting curve with calibration coefficients is then used to actively tune the MCP bias during spectroscopic measurements: A desired range is defined for the maximum intensity on the camera, corresponding to a maximum safe electron intensity on the MCP to prevent damage on the one hand, and a minimum desired intensity of the image on the CCD on the other hand. Whenever the maximum intensity of an image falls outside this range, the MCP gain $G(V_{\mathrm{MCP}})$ will be adjusted such that the intensity of the next image again falls in the center of this range. Assuming the intensity changes continuously, this method ensures the use of the full intensity range of the camera for each image, while protecting the MCP against damage.

Additionally, after the measurements, the calibration curve is used to calculate the real, relative intensity from the image intensity and the recorded V_{MCP}. By dividing this intensity by the intensity of the flat field image (taken in mirror mode and corrected for dark current and the MCP bias), we calculate a (floating point) conversion factor to true *reflectivity* values for each image. These ratios are added to the metadata of every image. By applying this conversion as a final step after any analysis of the data, errors due to discretization of highly amplified, and therefore low true intensity, images are minimized. Note that this procedure makes the conversion to true reflectivity possible even for datasets with no mirror mode in the dataset itself, such as dark field measurements.

3.3.3 Comparison of results

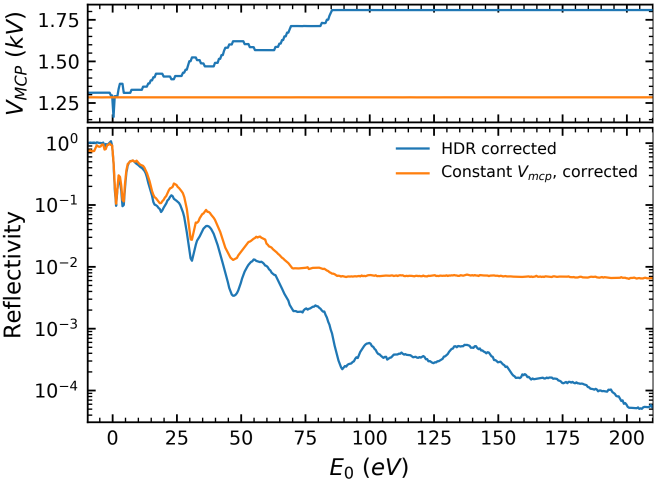

Figure 3.4: *Regular spectroscopic reflectivity curve (orange) of bilayer graphene on SiC, corrected for dark count and flat field, but with a single setting of V_{MCP} (top panel). The HDR measurement of the same area with active MCP bias tuning (blue) can resolve details down to lower intensity.*

Spectroscopic LEEM-I(V) curves on bilayer graphene on silicon carbide are measured both with constant MCP bias and with adaptive MCP bias as described above. A comparison between the resulting curves is shown in Figure 3.4. While the regular, constant MCP, curve starts to lose detail around $E_0 = 50$ eV, i.e. after a factor of 100 decrease in signal, the adaptive measurement captures intensity variations in the spectrum almost 4 orders of magnitude lower than the initial intensity. We thus call the adaptive method high dynamic range (HDR) imaging.

3.4 Drift correction by image registration

In LEEM imaging, the position of the image on the detector tends to shift during measurement as shown in Figure 3.1**a**. This prevents per-location interpretation of the data, both for spectroscopic measurements and measurements with varying temperature. Although the shift can be minimized by precise alignment of the system, we find that a significant shift always remains, especially in tilted illumination experiments such as DF-LEEM or angle-resolved reflected-electron spectroscopy (ARRES) [37, 38, 56], which makes the compensation of this image drift necessary.

This problem has been studied in depth in the field of image registration, motivated by wide-ranging applications such as stabilization of conventional video, combination of satellite imagery and medical imaging [59–61]. Techniques generally rely on defining a measure of similarity between a template and other images, either by some form of cross correlation, or by identifying specific matching features in both. The image is then deformed by a fixed set of transformations (either affine, i.e. purely shifts and rotations or non-rigid, i.e. additional deformation), until the match between the features in the images and the features in the template is maximal. For LEEM data, the measurement drift is almost completely described by in-plane shifts, significantly reducing the space of expected transformations. A common approach in this case is to use the (two-dimensional) cross-correlation as a measure of similarity between two shifted images and to find the maximum for all images compared to a template, as the location of maximum of the cross-correlation corresponds directly to the shift between the image and the used template.

The cross correlation of two $n \times n$ pixels images $I_1(x, y)$ and $I_2(x, y)$ is defined as follows:

$$\mathscr{C}(I_1, I_2)(x, y) = \frac{1}{n^2} \sum_{x'=0}^{n-1} \sum_{y'=0}^{n-1} I_1(x', y') I_2(x + x', y + y') \tag{3.4}$$

where the coordinates can be wrapped around, i.e. all spatial coordinates are modulo n. Furthermore, we can relate this to the convolution operation (denoted as $\circ$):

$$\begin{aligned}\mathscr{C}(I_1, I_2)(x, y) &= \frac{1}{n^2} \sum_{x'=0}^{n-1} \sum_{y'=0}^{n-1} I_1(x', y') I_2(x - (-x'), y - (-y')) \\ &=: \big(I_1(x', y') \circ I_2(-x', -y')\big)(x, y)\end{aligned} \tag{3.5}$$

Using this, the cross correlation can be expressed in terms of (two-dimensional) Fourier transforms $\mathscr{F}$:

$$\mathscr{C}(I_1, I_2) = I_1(x', y') \circ I_2(-x', -y') = \mathscr{F}^{-1}\left(\mathscr{F}(I_1) \cdot \overline{\mathscr{F}(I_2)}\right) \tag{3.6}$$

Where $\overline{\mathscr{F}(I_2)}$ denotes the complex conjugate of the Fourier transform of I_2. This makes the cross-correlation extra suitable as a measure of similarity, since it can be computed efficiently using the two-dimensional Fast Fourier Transform (FFT). Determining the local maximum of the cross-correlation yields the integer shift for which the two input images are most similar, with the height of the maximum an indication of the quality of the match. To further increase accuracy, several variants, such as gradient cross-correlation and phase-shift cross-correlation, have been shown to achieve sub-pixel accuracy for pairs of images [59, 62–64].

For LEEM data however, the straightforward cross-correlation approach is often hindered by the physics underlying the electron spectra, resulting in contrast changes (cf. Figure 3.5**a,b**) and even inversions for different values of E_0. The problem can be slightly

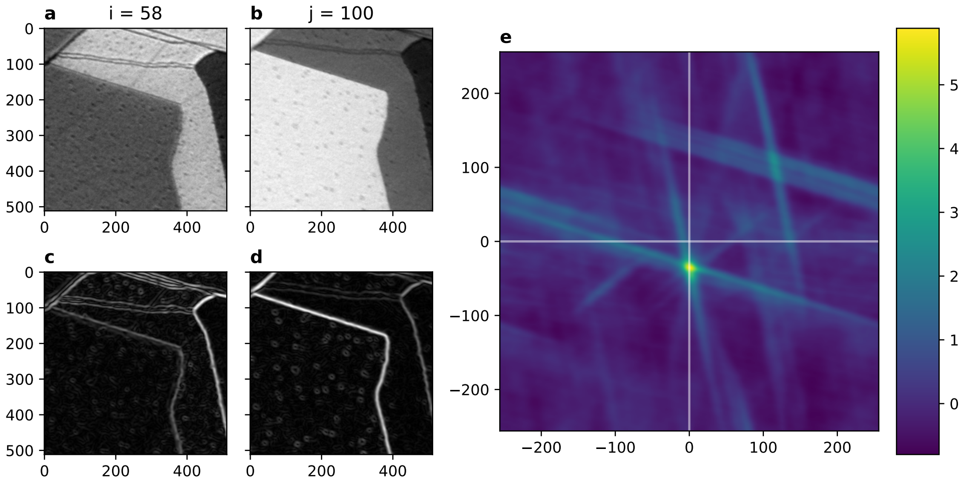

Figure 3.5: ***a,b,*** *Two bright-field LEEM images of few-layer graphene obtained at different E_0.* ***c,d,*** *Their Gaussian and Sobel-filtered versions with a Gaussian standard deviation of* 3 *pixels highlights the edges and erases the contrast inversion.* ***e,*** *The cross-correlation of the filtered images exhibits a clear maximum. Its position compared to zero (white lines) corresponds to the relative shift of the images.*

alleviated by using multiple templates, but in general this approach is unsatisfactory. Instead we present another approach here: We first apply digital filters and then compare each image to all other images, similar to the algorithm by Schaffer et al. for energy filtered transmission electron microscopy [65, 66]. It then uses a statistical average of the found integer shifts between all pairs of images to achieve sub-pixel accuracy.

We analyze the accuracy of this algorithm using an artificial test dataset and show that the accompanying Python implementation [42] is fast enough to process stacks of hundreds of images in mere minutes by performing benchmarks on a real dataset, followed by a discussion of the algorithm and results.

The algorithm consists of the following steps:

1. Select an area of each of the (detector-corrected) N images, suitably sized for FFTs (i.e. preferably $2^n \times 2^n$ pixels).

2. Apply Gaussian smoothing with standard deviation of σ pixels to reduce Poissonian noise in the images.

3. Apply a (magnitude) Sobel filter to highlight edges only, as shown in Figure 3.5**c** and **d**. As such, images with inverted contrast (cf. Figure 3.5**a,b**) become similar to each other.

4. Using Eq. (3.6), compute the cross-correlation, as shown in Figure 3.5**e**. Do this for all pairs (i, j) of images.

5. Compute the location $(DX, DY)_{ij}$ and value W_{ij} of the maximum of the cross-correlation for all image pairs (i, j). DX_{ij} and DY_{ij} form the anti-symmetric matrices of found relative shifts in either direction, while W_{ij} is a symmetric matrix of weights of the found matches, as shown in Figure 3.6.

6. Normalize the maximum values W_{ij} to be used as weights in step 8: $\overline{W}_{ij} = \frac{W_{ij}}{\sqrt{W_{ii} \cdot W_{jj}}}$.

7. Pick a threshold $W_{\min}$ to remove any false positive matches between images. A threshold of $W_{\min} = 0.15$, based on DX, DY and W_{ij}, is shown in Figure 3.6 as gray shading. Set $\overline{W}_{ij} = 0$ for all $\overline{W}_{ij} < W_{\min}$.

8. To reduce the N^2 relative shifts DX to a length N vector of horizontal shifts dx, minimize the errors $\left(dx_i - dx_j - DX_{ij}\right)\overline{W}^4_{ij}$ (using least squares). Do the same with DY to obtain the vertical shift vector dy.

9. Apply these found shifts dx and dy to the original detector corrected images, interpolating (either bi-linearly or via Fourier) for non-integer shifts.

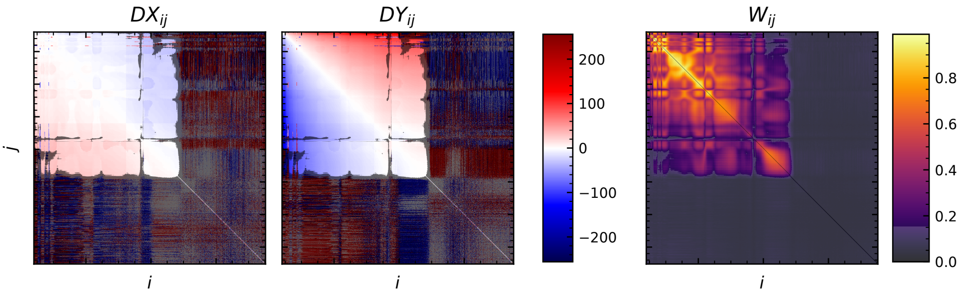

Figure 3.6: *Calculated shift matrices DX and DY and weight matrix $\overline{W}_{ij}$ for the bright-field dataset. Matches of a weight below $W_{\min} = 0.15$ (shaded in gray) are mostly false positives. Consequently, they are set to zero weight in the algorithm.*

3.4.1 IMPLEMENTATION DETAILS

The implementation makes extensive use of `dask` [67, 68]. This open source library, developed and maintained by Anaconda Inc., in particular its `Array` submodule, enables easy parallelization of array operations in common `numpy` syntax. It allows the lazy definition of computational operations on data, forming a task graph describing the computations to be performed. Delaying actual computation until explicitly called for enables `dask` to easily parallelize and stream computations, efficiently using all cores of a single computer. It should even allow to easily scale up to compute clusters.

The implementation uses of `dask` in combination with `scikit-image` code for blocked filtering of images [69].

The complete set of correlations is written in `dask` via the FFT functions and the multiplication in Fourier space. This, via `dask`'s task graph, allows every forward FFT to be computed only once for each image, as opposed to the N times for the naive implementation.

The optimization step uses `scipy` code, in particular the `least-squares` routine in the `optimization` module in combination with an explicit `scipy.sparse` Jacobian. [69–72]. We found that the use of an explicit Jacobian significantly reduced computation time and memory use for larger optimizations.

For easy interfacing with the user, `jupyter-notebook` and `ipython-widgets` are used [73]. For the dimension reduction using principal component analysis and clustering described in Section 3.5, `dask-ml` and `scikit-learn` are applied [74].

3.4.2 ACCURACY TESTING

To validate and benchmark the accuracy of the drift correction algorithm beyond visual inspection of resulting drift corrected datasets, an artificial test dataset with known shifts was created. This enables exact comparison of results to a 'true' drift.

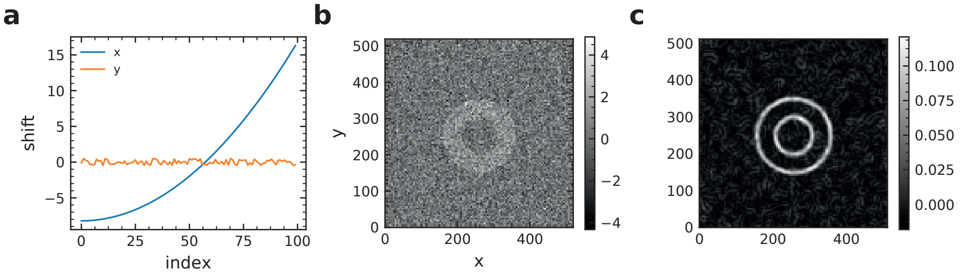

Figure 3.7: ***a,*** *Image shifts in x and y used for the synthetic dataset.* ***b,*** *Image 0 of the synthetic dataset with Gaussian noise with standard deviation of* $A = 1.0$. ***c,*** *Gaussian- and Sobel-filtered version of (b), with a Gaussian filter width of 5 pixels, highlighting edges.*

The test dataset, as shown in Fig. 3.7, consists of $N = 100$ copies of an annulus of intensity 1.0 on a background of 0.0. The dataset is shifted over a parabolic shift in the x-direction and random shifts uniformly chosen from the interval $[-0.5, 0.5]$ pixels in the y direction (see Figure 3.7**a**). Finally pixel-wise Gaussian (pseudo-) random noise is added to all images. The standard deviation A of the added random noise is then varied to simulate images with different signal-to-noise ratios (SNR).

BENCHMARK METHODOLOGY

Benchmarks were run on a desktop PC with an Intel `i7-7700K` running at 4.20 GHz with 32 GB of RAM and Windows 10 x64 installed. The full stack of images was read from separate `TIFF` files on a Toshiba XG5 NVMe SSD for each run during the filter and cross correlation phase and written to a uncompressed `zarr` archive on the same SSD during the shift and write phase.

The benchmark was run in a Jupyter notebook, with an installation of the Anaconda environment of Python 3.6 with at least the following packages installed:

```
numpy=1.16.3, matplotlib=3.0.3, dask=1.2.0, distributed=1.27.1,
scikit-image=0.15.0, scipy=1.2.1, jupyterlab, scikit-learn=0.20.3,
dask-ml=0.12.0, xarray=0.12.0, h5py=2.9.0, mkl=2019.3, mkl_fft=1.0.12,
numba=0.43.1, zarr=2.2.0
```

Note: Accuracy benchmarks were performed with images of 512 × 512 pixels, the time benchmarks in the next session on subimages of 256 × 256 pixels.

Results

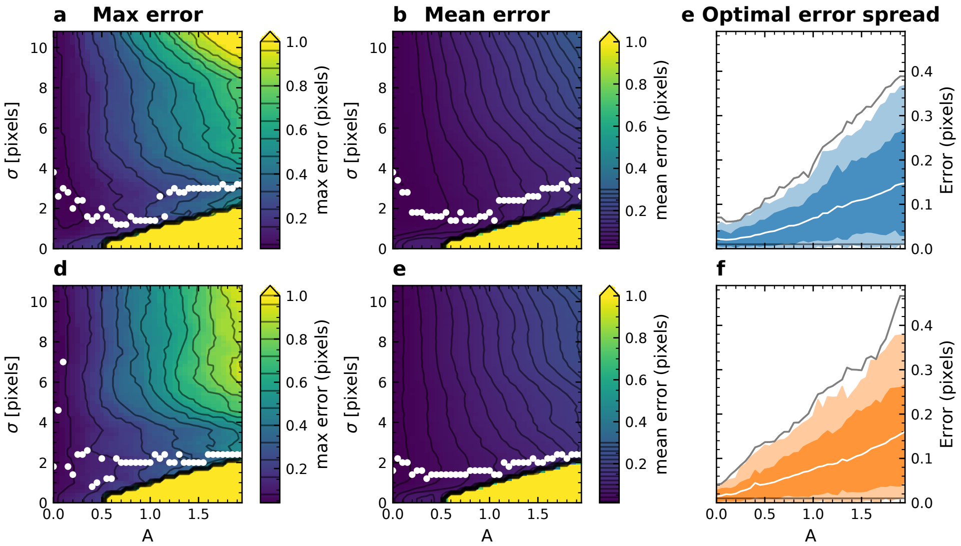

__Figure 3.8:__ __a,b,__ Maximum and mean error in dx shift as calculated by the algorithm for different values of noise amplitude A and smoothing parameter σ. The optimal value of the Gaussian smoothing σ as a function of added noise amplitude A is drawn in white. Black contour lines are added as a guide to the eye. __c,__ Spread of the error for the optimal values of σ for varying A. Dark and light bands are respectively 1 and 2 standard deviations, maximum error is indicated as gray line. __d,e,f,__ Same for the y direction.

The resulting maximum error in the found shift compared to the original, 'true' shift, as well as the resulting mean error for different values of A and σ is shown in Figure 3.8, separately for the x and y directions. These results verify that, at least for synthetic datasets, the algorithm achieves sub-pixel accuracy, with the mean absolute error in pixels of about 0.1 times the relative noise amplitude A for the optimal value of smoothing σ, and the maximum absolute error just reaching 0.5 pixel for the extreme value of $A = 2$. As expected, the error is strictly increasing for decreasing SNR, i.e. increasing A. After an initial cutoff, visible in saturated yellow in Figure 3.8, the accuracy of the algorithm is also generally decreasing for increasing smoothing width σ. However, after this initial cutoff, there is a comfortably large range of σ where the algorithm performs well.

The choice of smoothing parameter σ has significant influence on the analysis of real data, as is visible in Fig. 3.6: for high E_0, the noise level is so high that no feasible matches were found for the used value of smoothing σ. Increasing σ alleviates this, but reduces the match quality for images with low noise level.

We found that most features visible in the high-σ, high-A regime of Figure 3.8**b,d** are dependent on the initialization of the random generator for the added pixel-wise noise and are thus not significant.

3.4.3 Time complexity

To benchmark the computational complexity of the algorithm, it was applied to subsequently larger parts of the real dataset, while measuring the computation times for the least squares optimization (step 8 above) and the shifting and writing of images (step 9) separately.

The results show calculating the cross-correlations takes the most time, as it scales almost perfectly quadratically in the number of images N, as shown in Fig. 3.9. The shifting and writing of images scales linearly and is not significant for larger datasets. The total time therefore scales nearly perfectly quadratically, with a dataset of 500 images drift corrected in less than 7 minutes of computation time. As such, LEEM spectroscopy datasets can be comfortably and regularly drift corrected on a desktop PC.

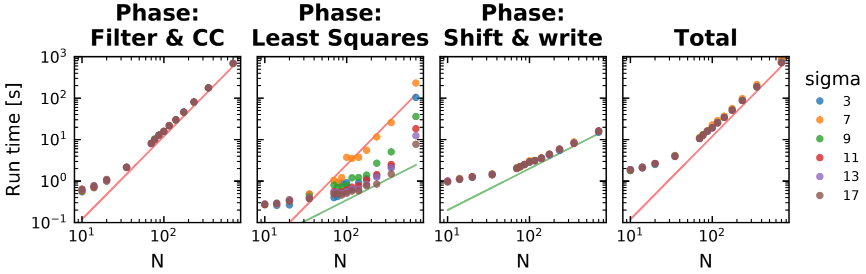

Figure 3.9: *Run times of different phases of the drift correction algorithm on* 256 × 256 *pixel images. Linear (green) and quadratic (red) slopes are added as a guide to the eye, illustrating that the cross correlations scales quadratically in the number of images N, while the shifting and saving of images itself is linear.*

3.4.4 Discussion

We elaborate here on the choices made in the algorithm. The use of the magnitude Sobel filter has multiple benefits, similar to using the gradient cross-correlation: Contrast inversions between areas with different spectroscopic properties nonetheless result in similar images (cf. Figure 3.5**c,d**). In addition, the constant zero background reduces errors due to wrap-around effects due to performing the calculation in the Fourier domain.

The exponent 4 for the weighing matrix W_{ij} in the least squares minimization step 8 was empirically found to give the best results for real datasets.

As already noted by Schaffer et al., the use of cross-correlation between multiple image pairs and combining the returned integer shift values enables sub-pixel accuracy. The maximum theoretical accuracy is $\frac{1}{N}$ pixels, but is reduced for images where false positive matches are thresholded out.

Alternative methods of obtaining sub-pixel accuracy in determining shifts include a combination of upscaling and matrix-multiplication discrete cosine transforms [63] and a rather elegant interpolation of the phase cross-correlation method proposed by Foroosh et al. [59]. However our current method is less complex and combines robustness against global drift with handling of changing contrasts, which is crucial for spectroscopic LEEM data. Although the sub-pixel precise phase correlation method seems a straightforward extension of regular cross correlation, it is less suitable for datasets with changing noise levels and not suitable for false positive detection by normalization, both properties we found essential to handle spectroscopic LEEM data.

Fourier interpolation for non-integer shifts, corresponding to Whittaker-Shannon interpolation, is optimal when the image resolution is limited by the optical transfer function of the intstrument, instead of pixel sampling-limited [75]. In this case, almost always true for LEEM measurements, the imaginary part is zero up to floating point precision and can be discarded. If however the resolution is limited by the detector, e.g. at very low magnifications, bi-linear interpolation is the better choice.

Contrary to Schaffer et al., we found smaller values of Gaussian smoothing width σ yield the best results, with larger values yielding artificial shifts around contrast inversions for real datasets and generally performing worse for the synthetic dataset, as visible in Fig. 3.8.

Schaffer et al. found their approach at the time (2004) not computationally feasible for large amounts of images but, as shown in the previous section, the current implementation is able to drift correct a stack of several hundreds of images comfortably on a single desktop computer. We want to emphasize that the use of `Python` gives flexibility and makes it easy to adapt the code. For instance, increasing performance even more lies within reach by performing the FFT cross-correlations and maximum search on one or more graphical processing units using one of several libraries or by using a cluster running a `dask` scheduler. Further speedup would be possible by pruning which pairs of images are to be cross-correlated. An avenue not explored here, but used in the image stitching described in Appendix B, is the use of pyramid methods to create a multi-step routine where firstly a fast estimate of the shift is computed on a smoothened and reduced-size image before using consecutively larger images to refine the estimate [76, 77].

Beyond drift correction, the same method presented here can also be applied to create precisely stitched overview images of areas much larger than the electron beam spot size. Although, as no contrast inversions or large feature differences are expected for the matching areas, the added value of using a gradient filter is nullified. Additionally the number of images that can be matched to the same template is limited, forcing a low upper bound on the sub-pixel accuracy of the optimization part of the algorithm. Instead, we found that an algorithm based on more regular phase-shift cross-correlation is sufficient for sub-pixel accurate stitching (See Appendix B).

3.5 DIMENSION REDUCTION

The sub-pixel accuracy drift correction now makes it possible to reinterpret a LEEM-I(V) dataset as a truly per-pixel set of spectroscopic curves, opening up possibilities for further data analysis. For a dataset of N images, each such curve (cf. Fig. 3.1) can be seen

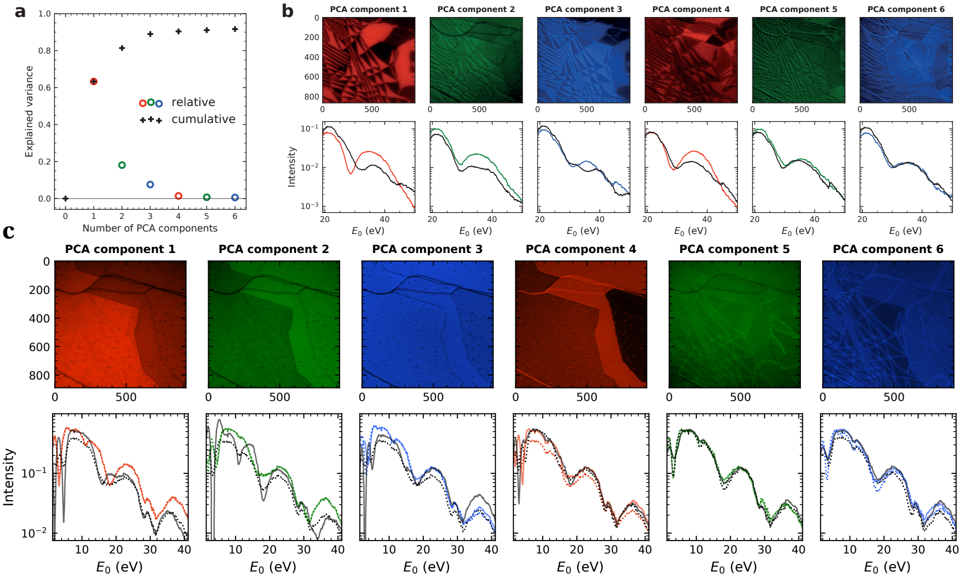

Figure 3.10: ***a,*** *Scree plot indicating the retained variance per PCA component for the Dark Field dataset.* ***b,*** *Images of the first six PCA components for the Dark Field sample dataset and the spectra corresponding to the maximum and minimum of the respective components occurring in the dataset.* ***c,*** *Images of the first six PCA components for the Bright Field sample data set and the spectra corresponding to the maximum and minimum of this component occurring in the dataset (dashed) and the spectrum corresponding to the corresponding eigen-spectrum itself.*

as a vector in a N-dimensional vector space of the mathematically possible spectra. Even for moderate datasets of a few hundred images this is a huge vector space. In almost all cases however, the physical behavior of the data can be described with a model with far fewer degrees of freedom, i.e. the vector space of physically possible spectra has a much lower number of dimensions. Therefore, it should be possible to summarize all significant behavior in a much smaller dataset, which can be analyzed (and visualized) much more easily.

Here, we use Principal Component Analysis (PCA), a linear technique based on singular value decomposition (SVD), often used for dimension reduction in data science fields [78–80]. The randomized iterative variant of PCA allows for efficient computation of the largest variance components without performing the full SVD decomposition [81]. This technique therefore projects the spectroscopic data to a lower dimensional subspace, in such a way that maximal data variance is retained. It does so in a computationally efficient way, making it well suited for, and popular in, data science. Before applying PCA, we crop the dataset to remove any areas that lie outside of the detector for any image inside the used range of E_0. Additionally, each image is scaled to zero mean and unit variance, to not let brighter images contribute more strongly to the analysis as they have larger variance. A lot of other choices for standardization of the data are possible, most of them with useful results, but for the scope of this paper we adhere to this standard choice.

After performing PCA, the lower dimensional subspace or PCA-space, is now spanned by orthogonal ‘eigen-spectra’, referred to as PCA components. Since the projection map onto this subspace retains most of the variance in the dataset, it is possible to build an approximate reconstruction of the full physical spectra from the reduced PCA spectra.

Although the aim is dimension reduction, the number of PCA dimensions should be chosen large enough, such that the mathematical PCA-space contains all physical behaviour up to a certain noise level [82]. A so-called scree plot, displaying the captured variance, aids to pick the right number of dimensions. For the sample dataset of dark-field images of $N = 300$ energies, a scree plot is shown in Figure 3.10**a**. In general, for spectroscopic LEEM datasets, we find that reducing down to 6 dimensions is often enough to capture more than 90% of all variance.

The dataset can be projected onto a single PCA component by taking the per-pixel inner product with the corresponding ‘eigen-spectrum’. This yields images visualizing the variance retained by the respective components, as shown in the top half of Figure 3.10**b** and **c,** for all 6 PCA components of the dark field and bright field data respectively. Below each image, the spectra corresponding to the pixels with the minimum and maximum value of this projection are shown in black and color, respectively.

3.5.1 VISUALIZATION

Reducing a spectrum from hundreds of dimensions to a few opens up new opportunities for data visualization. In particular, it allows for the visualization of nearly all of the variation in spectra of an entire dataset in only two images, as shown in Fig. 3.11. Here, the values of the six principal components (cf. Fig. 3.10), are displayed as the RGB color channels of two pictures per dataset. To lift the degeneracy in the possibilities of the sign of the PCA components (a PCA eigen-spectrum with the opposite signs retains as much variance), we change the signs such that the a positive projection onto the PCA component corresponds to having the higher average relative brightness in the images. This way, areas that are bright in the majority of the original images also appear bright in the visualization. To compensate for the human eyes’ preference for green, a scaling of colors as proposed by Kovesi is applied [83]. It is given by the following matrix:

$$\begin{pmatrix} R' \\ G' \\ B' \end{pmatrix} = \begin{pmatrix} 0.90 & 0.17 & 0.00 \\ 0.00 & 0.50 & 0.00 \\ 0.10 & 0.33 & 1.00 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

The results are striking. All the sample features are directly visible in Fig. 3.11: In the bright field dataset, bilayer and thicker graphene are clearly separated in orange and green, respectively in the first three PCA components (Figure 3.11**a**). Moreover, SiC step edges, domain walls and point-like defects are clearly visible. The next three PCA components (Figure 3.11**b**) highlight the difference between bilayer (green), trilayer (orange) and four-layer graphene (dark green) and in addition separates step edges (orange), domain boundaries (turquoise) and the defects in different colors. Furthermore, two types of domain boundaries can be observed in the four-layer area that are hard to tell apart in conventional LEEM images (See also Section 5.3). The light green and dark ones are presumably domain boundaries in the top-most and lower layers, respectively.

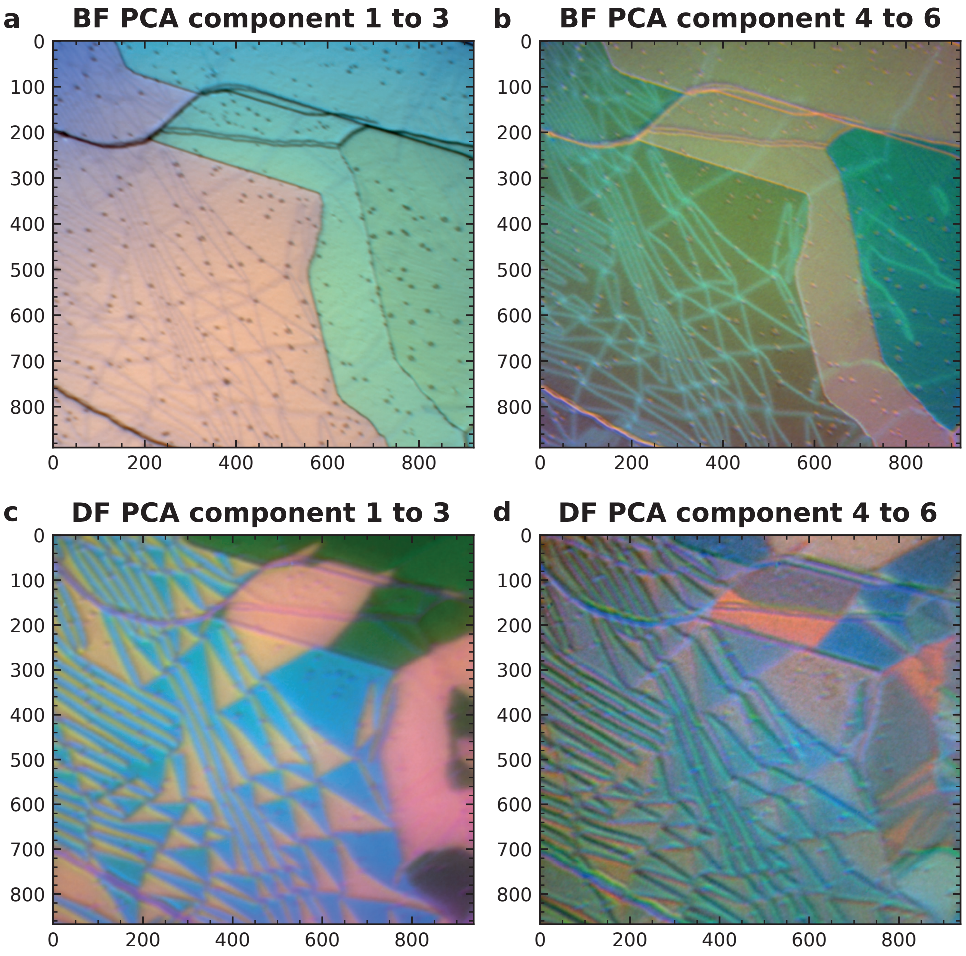

Figure 3.11: The first six PCA components can be used to summarize a spectroscopic LEEM measurement in two RGB pictures. ***a,b,*** *Visualization of a spectroscopic bright-field LEEM measurement of quasi-freestanding few-layer graphene on SiC. Different layer counts, stacking boundaries of two types and point defects are distinguishable.* ***c,d,*** *Visualization of a spectroscopic dark-field LEEM measurement of the same area. All six different possible stacking orders for up to trilayer graphene are easily distinguishable.*

This visualization using the first three PCA components of the dark field dataset (Figure 3.11**c**) clearly separates the different stacking orders in bilayer (AB in orange and AC in blue) and trilayer graphene. The PCA components 4 to 6 (Figure 3.11**d**) highlights the different stacking orders in trilayer and four-layer areas (different shades of orange and blue) and display an interference effect causing double lines at one type of (corresponding to one direction of) domain edge. This clear visualization is particularly remarkable as the dark-field dataset presents a worst case scenario due to its extreme off-axis alignment (see Chapter 4 for full details), which causes strong image drift and relative shifts of features.

3.5.2 CLUSTERING AND AUTOMATIC CLASSIFICATION

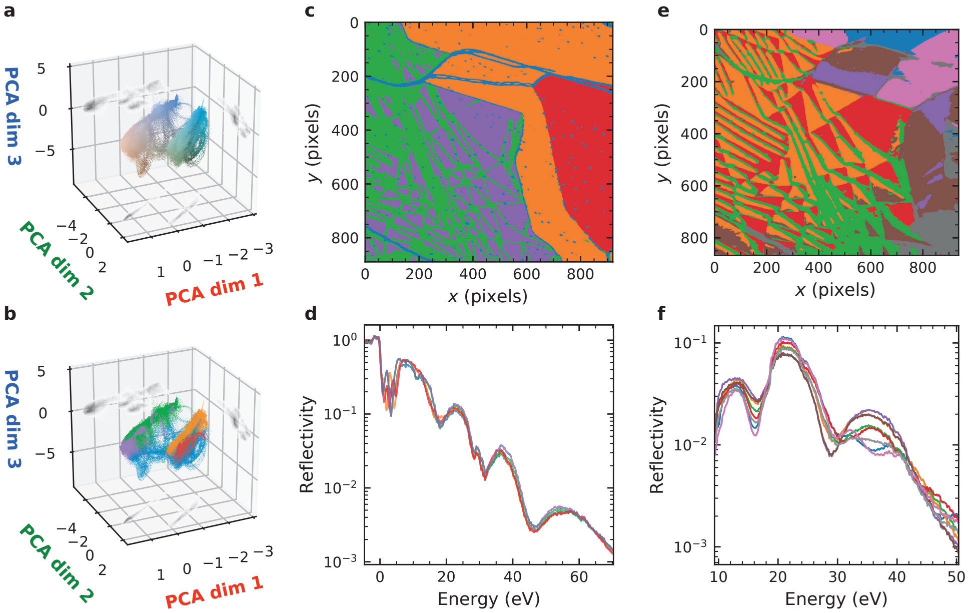

Figure 3.12: ***a,*** *Bright-field dataset visualized as point cloud in the space of the first three PCA components. The points are colored according to the mapping in Fig. 3.11(a) and are projected onto the planes in gray scale.* ***b,*** *Point cloud as in subfigure (a), but colored according to the computed clustering.* ***c,*** *Indication of the cluster labels in the real-space image.* ***d,*** *Mean bright-field spectroscopy curves for each cluster, automatically recovering layer count and domain walls.* ***e,f,*** *Same as subfigures (c,d) for the dark-field dataset reveals all stacking orders present as well as two sets of edge-case curves. Cf. Figure 4.2.*

In addition to the visualization possibilities explored in the previous section, the dimension reduction by PCA lowers the complexity of the data enough to enable the use of other, more quantitative data analysis techniques. In particular, reduction to less than ten dimensions is enough to perform unsupervised classification or clustering on the entire dataset. Here, we show that a relatively simple clustering algorithm, the classical k-means, also known as Lloyd's algorithm [84], applied to the PCA reduced dataset, can already be used to distinguish the relevant, different areas. The structure of the bright field dataset is visualized in terms of the first three PCA components in Figure 3.12**a**, both as a point cloud with colors corresponding to Figure 3.11**a** and as density projections (gray scale) on the three planes. The resulting classification from the application of k-means to the six PCA components is visualized in the same way in Figure 3.12**b**, where the color of the points now corresponds to the assigned labels. These same label colors are shown in real space in Figure 3.12**c**. In the real space visualization it is clear that the different layer counts are separated (bilayer, trilayer and four-layer as purple, orange and red, respectively) from a class with the point defects and step edges (blue) and a class containing the domain boundaries in the bilayer (green). The cluster labels can now be used to calculate spectra of each area, e.g. all trilayer pixels without the defects. For this, we take the mean over all pixels belonging to one cluster for each energy in the full N-dimensional dataset. This can be done even for energies outside the range used for the initial clustering as well as for energies where we only have partial data due to drift. The resulting spectra for the clustering in Figure 3.12**c** are plotted in Figure 3.12**d**.

The same clustering method is applied only to the first 4 PCA components of the dark-field dataset since component 5 and 6 show virtually no distinguishing features and corresponds to very little variance (cf. Figure 3.10**a,b**). The resulting real space labeling of the clusters and the spectra are shown in Fig. 3.12**e** and **f**, respectively. Here, although not perfect, the clustering algorithm manages to mostly separate the different possible stacking orders (red and orange for the bilayer and purple, brown, pink and blue for the trilayer). The green and gray areas correspond to areas where clear classification as a stacking order is not possible due to phase contrast and non-uniform illumination artefacts due to the tilted illumination.

Thanks to the proper calibration and mutual registration of the data, this relatively simple algorithm classifies the areas in the dark-field data set with only minor errors (e.g. the incorrect assignment of brown trilayer spectrum in the lower left), *without* any input of the positions of each spectrum in the image or any input about the expected differences between spectra. We anticipate that this classification using unsupervised machine learning will be useful for identifying unknown spectra in new datasets. In Chapter 8 for example, it is applied to distinguish the different spectra found on TaS_2 samples.

3.6 CONCLUSION

We have shown that treating (energy-dependent) LEEM measurements as multi-dimensional datasets rather than as collection of images, opens rich opportunities for detailed and quantitative insights into complex material systems that go well beyond morphological and crystallographic characterization.

Three key steps are necessary to convert a stack of raw LEEM images into spectroscopic dataset with a greatly increased body of quantitative information. First, we compensate for common detector artefacts such as camera dark count and non-uniform detector gain, which is crucial to quantitatively interpret LEEM images. Second, by calibrating the channel plate gain and adjusting it during spectroscopic measurements, we can not only extend the dynamic range of the dataset by two orders of magnitude, but also convert image intensity into absolute reflectivity or electron intensity (provided the beam current is accurately measured). Third, we describe a drift-correction algorithm that is tailored for spectroscopic LEEM datasets where contrast inversions make many other approaches infeasible. It relies on digital filtering and cross-correlation of every image to all other images and, without requiring large computation times, yields sub-pixel accuracy. It thus produces spectral LEEM data with high spatial resolution, i.e., true pixel-wise spectra.

This suite of techniques is already in regular use to obtain data from the ESCHER system in Leiden [25–27, 31, 34, 37]. In addition, the resulting spectral datasets enable more sophisticated data classification and visualization methods that rely on the spectrum (I(V)-curve) in every pixel. We demonstrate how we can use dimension reduction on the spectra to automatically compose images from only the six strongest spectroscopic features (PCA components). This approach produces rich color images that capture most of the features of the dataset and can thus give an intuitive view on complex material systems. Furthermore, we show that a relatively simple cluster analysis on those data sets of reduced dimensionality yields a quantitative representation of this information. Different materials within a field of view are automatically identified and statistical information such as the mean spectra and their spread per material can be extracted.

Treating LEEM measurements as multidimensional datasets as presented here will further strengthen the role of LEEM as a quantitative spectroscopic tool rather than as a pure imaging instrument, thus deepening its impact in the research and discovery of novel material systems. Furthermore, the presented techniques can be applied to related spectroscopic imaging techniques, such as energy-filtered PEEM [85] or even adapted for use in scanning probe techniques such as scanning tunneling spectroscopy [79, 86]. To facilitate the use of the approaches discussed here, the test data as well as Python code is available online [42, 48].

3.7 Outlook: Drift correction improvements

This is an addendum to the drift correction algorithm and the image stitching algorithm as described in this chapter and in Appendix B. The implementations of both are published in Ref. [42].

The drift correction as presented in this chapter can straightforwardly be extended in several ways, to increase computation speed or accuracy or to adopt to different datasets. I will list some ideas to do so here. Some of these are borrowed from the work done for stitching of images, others have been planned for a long time, but were never prioritized enough for time to be available to implement them.

3.7.1 GPUs for faster computation

Graphical Processing Units (GPUs) are very efficient at massively parallel computation in general, and image handling specifically. Therefore moving computation from the CPU to one or more GPUs can be a way to increase performance by orders of magnitude if the workload allows. However, traditionally programming for GPUs has been more involved, as it relied on specific extensions of general purpose programming languages to allow computation on GPU. With this came a lot of specific programming paradigms to accommodate the significantly different architecture of GPUs (compared to CPUs) and the moving of data from main memory to the GPU memory and back.

However, the software stack has been significantly maturing, even for more high-level languages such as Python, in particular for NVIDIA GPUs. This has led to Python being adopted as the language of choice for the quickly emerging field of data science, greatly enhancing the ecosystem by the sheer number of programmers working in it. Therefore it would now be relatively straightforward to implement a GPU-accelerated version of the drift correction by utilizing cupy [87] to handle array computations where numpy is used now. Massive work has been done to push the API of cupy towards higher compatibility with numpy and scipy, meaning that for (almost) all used functions, drop-in cupy replacements exist. Experience from rewriting some core pyGPA [88] routines to use cupy confirms that significant gains in speed can be achieved without much (programming) effort. Furthermore, it should work transparently with dask.

3.7.2 ALGORITHMIC SPEED IMPROVEMENTS

The core of the statistical drift correction algorithm is the $\mathcal{O}(N^2)$ comparison of every pair of images. While this enables subpixel accuracy, it is a major Achilles heel in computation speed and scalability. Not only does this mean a quadratic increase in computations for more images, it also prevents early release from memory of computed Fast Fourier Transforms (FFTs) of images, as any block of images is needed in combination with every other block of images. By drawing a parallel to the stitching algorithm, two solutions to this present itself:

- For larger smoothing filters, images can be scaled down, significantly reducing memory load and needed computation for the FFTs.
- Instead of comparing all pairs of images, we can compare each image only to k 'nearest' neighbors, with $k \ll N$, resulting in a $\mathcal{O}(N \times k)$ algorithm, now linear in N. The natural 'distance' to determine the nearest neighbors is now not the stage coordinates, but whichever other (set of) parameters is changed, e.g. landing energy. Even just the time at which the images were taken, comparing images taken after each other would yield a reasonable 'distance' to use. A non-optimized version of this linear variant has been implemented in the repository [42].

It is even possible to combine the two approaches in a two-pass algorithm, to create a faster algorithm without losing too much accuracy. This can work, because the 'quality' of a match between two images, depends on the overlap between them. A first, linear pass, will be used to generate an initial estimate of the absolute shift vectors. The subsequent, quadratic pass can now be done on crops based on the first pass, allowing for smaller crops to be used, decreasing memory and computational load. For a 4 times smaller crop in each direction (a reasonable value) this would save a factor $4^2 = 16$ on memory use and a factor $4\log(4)$ on computation, with improved cache locality expected to provide further speed gains.

As an alternative for this two-pass algorithm, a mixed strategy could be considered, optimizing for matrices of relative shifts containing both k nearest neighbors and the quadratic, farther out combinations for every l-th image (striding through the full dataset), corresponding to a $\mathcal{O}\left(N \times k + \frac{N^2}{l^2}\right)$ algorithm.

3.7.3 IMPROVEMENTS TO ACCURACY

For any pair of images, there is an optimal (Gaussian) smoothing width σ to suppress the noise, depending on the signal-to-noise ratio (SNR). In spectroscopic LEEM imaging, the SNR strongly depends on the intensity of the image, and therefore on the landing energy, resulting in a widely varying SNR within a single spectroscopic dataset. In the currently described algorithm, a fixed σ is manually picked for the entire dataset and any matches between images with insufficient match quality, determined by the also manually set $w_{\min}$, are discarded.

This can be improved upon in several ways, as discussed below.

AUTOMATIC DETERMINATION OF w_{MIN}

The most straightforward way is to borrow once again from the stitching algorithm, and to determine $w_{\min}$ automatically as the *maximum value* for which the number of connected components of the graph of all valid matches is still equal to 1, i.e. such that all images can still be related to all the other images. An alternative, likely superior method is described and implemented by Savitzky et al. in Ref. [66]. They force all relative shifts to be consistent and drop any relative shifts that cause inconsistency in the optimization.

ADAPTIVE SMOOTHING

Two more elaborate ways to improve upon this involving an adaptive value for σ could be explored.

The first uses *image pyramids* to find the lowest value for σ for each match pair for which a sufficiently high w weight can be achieved. Image pyramids are a common way of dealing with high resolution images [76]. The pyramid is formed by multiple versions of the image, with each layer being a smoothed and down-scaled version of the layer below. By moving down the pyramid, more detail is obtained, at the cost of a larger image size. Starting at the top of the pyramid for each image, we have a high σ, low-resolution version of the image, for which the cross-correlations can be computed cheaply, with a high likelihood of a sufficient match between the images. Moving down the pyramid, the match can be updated to a higher precision version, until $w_{ij} < w_{\min}$, or until a larger than expected jump in the relative shift vector is detected. This algorithm is more involved than the original algorithm, but allows for optimal σ value selection for each match, ultimately increasing the information fed into the optimization and improving the result, without just adding σ as another full dimension to the computation.

SIGNAL-TO-NOISE DETERMINATION

The second way to select an optimal value of σ per image pair is based on the following. The smoothing is applied to suppress (amplification of) the noise in the image, which causes spurious peaks in the cross-correlation. For a higher SNR, a larger σ value is needed to average over enough pixels to suppress the noise to a level that the cross-correlation yields the correct result. Therefore, if the SNR of an image is approximately known, it would be possible to estimate the optimal value of σ to use on it.

The SNR can be determined by comparing two, subsequently taken, *nominally equal* images by computing the normalized weight $\bar{w}$ from the cross-correlations (equivalent to what is used in the drift correction algorithm). A higher $\bar{w}$ corresponds to more similar images and therefore less random noise.

Although taking twice as many images might sound like a high cost at first glance, in practice the cost is relatively low, as the images can be combined to a longer exposure image by averaging and treated as a single image afterwards. This combination could already be applied during imaging, and as an extension of that, one could adaptively average images until a sufficient SNR is achieved before moving on to the next parameter value in the measurement.

However, we can circumvent the need for extra images entirely by an approximation under the sufficient sampling assumption, i.e. in the case that subsequent images in the dataset are sufficiently close together in parameter space to sample all behavior as a function of the varied parameter (e.g. E_0). This assumption should be satisfied in almost any sensible experiment, so it is a reasonable one to make. In this case, we can replace the nominally equal image from the preceeding description by the next best option: a neighboring image in parameter space, which should only be marginally different. An even more sophisticated option would be to take an average of two neighboring images from opposing sides, although proper normalization of image intensities would be needed in that case.

REGULARIZATION OF POSITIONS

A final, relatively easy to implement, possible improvement to the accuracy of the algorithm is the addition of a regularization term in the optimization step. By penalizing the second derivative of the position, prior information is added: generally, the image is not expected to suddenly jump between subsequent images[2], for example, in the case of thermal drift, the mass of the stage and Newton's second law alone prevent sudden jumps of the sample stage, and therefore, of the image. This would also provide a better solution in the (undesired) case of multiple connected components in the matrix of used relative shifts, preventing the large jumps occurring without any regularization.

REFERENCES

25. R. Tromp, J. Hannon, A. Ellis, et al. A new aberration-corrected, energy-filtered LEEM/PEEM instrument. I. Principles and design. *Ultramicroscopy* **110,** 852–861. doi:10.1016/j.ultramic.2010.03.005 (2010).

26. S. Schramm, J. Kautz, A. Berghaus, et al. Low-energy electron microscopy and spectroscopy with ESCHER: Status and prospects. *IBM Journal of Research and Development* **55,** 1–1. doi:10.1147/JRD.2011.2150691 (2011).

27. R. M. Tromp, J. B. Hannon, W. Wan, A. Berghaus & O. Schaff. A new aberration-corrected, energy-filtered LEEM/PEEM instrument II. Operation and results. *Ultramicroscopy* **127,** 25–39. doi:10.1016/j.ultramic.2012.07.016 (2013).

29. J. I. Flege & E. E. Krasovskii. Intensity-voltage low-energy electron microscopy for functional materials characterization. *Physica Status Solidi - Rapid Research Letters* **8,** 463–477. doi:10.1002/pssr.201409102 (2014).

30. H. Hibino, H. Kageshima, F. Maeda, et al. Microscopic thickness determination of thin graphite films formed on SiC from quantized oscillation in reflectivity of low-energy electrons. *Physical Review B* **77,** 075413. doi:10.1103/PhysRevB.77.075413 (2008).

[2] In some tilted Dark Field alignments, focus tracking of the objective lens can actually cause apparent jumps between images. In this case using regularization would not be the best choice.

31. T. A. de Jong, J. Jobst, H. Yoo, et al. Measuring the Local Twist Angle and Layer Arrangement in Van der Waals Heterostructures. *physica status solidi (b)* **255,** 1800191. doi:10.1002/pssb.201800191 (2018).

34. J. Jobst, A. J. H. van der Torren, E. E. Krasovskii, et al. Quantifying electronic band interactions in van der Waals materials using angle-resolved reflected-electron spectroscopy. *Nature Communications* **7,** 13621. doi:10.1038/ncomms13621 (2016).

35. T. A. de Jong, D. N. L. Kok, A. J. H. van der Torren, et al. Quantitative analysis of spectroscopic low energy electron microscopy data: High-dynamic range imaging, drift correction and cluster analysis. *Ultramicroscopy* **213,** 112913. doi:10.1016/j.ultramic.2019.112913 (2020).

36. E. Bauer, M. Mundschau, W. Swiech & W. Telieps. Surface studies by low-energy electron microscopy (LEEM) and conventional UV photoemission electron microscopy (PEEM). *Ultramicroscopy* **31,** 49–57. doi:10.1016/0304-3991(89)90033-8 (1989).

37. T. A. de Jong, E. E. Krasovskii, C. Ott, et al. Intrinsic stacking domains in graphene on silicon carbide: A pathway for intercalation. *Physical Review Materials* **2,** 104005. doi:10.1103/PhysRevMaterials.2.104005 (2018).

38. J. Jobst, J. Kautz, D. Geelen, R. M. Tromp & S. J. van der Molen. Nanoscale measurements of unoccupied band dispersion in few-layer graphene. *Nature Communications* **6,** 8926. doi:10.1038/ncomms9926 (2015).

39. A. Schmid, W. Święch, C. Rastomjee, et al. The chemistry of reaction-diffusion fronts investigated by microscopic LEED I–V fingerprinting. *Surface Science* **331-333,** 225–230. doi:10.1016/0039-6028(95)00128-x (1995).

40. B. Frank, P. Kahl, D. Podbiel, et al. Short-range surface plasmonics: Localized electron emission dynamics from a 60-nm spot on an atomically flat single-crystalline gold surface. *Science Advances* **3,** e1700721. doi:10.1126/sciadv.1700721 (2017).

41. A. Locatelli, K. R. Knox, D. Cvetko, et al. Corrugation in Exfoliated Graphene: An Electron Microscopy and Diffraction Study. *ACS Nano* **4,** 4879–4889. doi:10.1021/nn101116n (2010).

42. T. A. de Jong. *Quantitative Data Analysis for spectroscopic LEEM* version v0.2.0. 2021. doi:10.5281/zenodo.3539538.

43. J. Röhrl, E. Rotenberg, J. Jobst, et al. Towards wafer-size graphene layers by atmospheric pressure graphitization of silicon carbide. *Nature Materials* **8,** 203–207. doi:10.1038/nmat2382 (2009).

44. F. Speck, J. Jobst, F. Fromm, et al. The quasi-free-standing nature of graphene on H-saturated SiC(0001). *Applied Physics Letters* **99,** 122106. doi:10.1063/1.3643034 (2011).

45. C. Riedl, C. Coletti, T. Iwasaki, A. A. Zakharov & U. Starke. Quasi-Free-Standing Epitaxial Graphene on SiC Obtained by Hydrogen Intercalation. *Physical Review Letters* **103,** 246804. doi:10.1103/PhysRevLett.103.246804 (2009).

46. R. M. Feenstra, N. Srivastava, Q. Gao, et al. Low-energy electron reflectivity from graphene. *Physical Review B* **87,** 041406. doi:10.1103/PhysRevB.87.041406 (2013).

47. H. Hibino, S. Mizuno, H. Kageshima, M. Nagase & H. Yamaguchi. Stacking domains of epitaxial few-layer graphene on SiC(0001). *Physical Review B* **80,** 085406. doi:10.1103/PhysRevB.80.085406 (2009).

48. T. A. de Jong & J. Jobst. *Data underlying the paper: Quantitative analysis of spectroscopic Low Energy Electron Microscopy Data* 2019. doi:10.4121/uuid:7f672638-66f6-4ec3-a16c-34181cc45202.

49. R. van Gastel, I. Sikharulidze, S. Schramm, et al. Medipix 2 detector applied to low energy electron microscopy. *Ultramicroscopy* **110,** 33–35. doi:10.1016/J.ULTRAMIC.2009.09.002 (2009).

50. R. Widenhorn, M. M. Blouke, A. Weber, A. Rest & E. Bodegom. *Temperature dependence of dark current in a CCD* in *Sensors and Camera Systems for Scientific, Industrial, and Digital Photography Applications III* (eds M. M. Blouke, J. Canosa & N. Sampat) **4669** (SPIE, 2002), 193–201. doi:10.1117/12.463446.

51. R. Widenhorn, J. C. Dunlap & E. Bodegom. Exposure Time Dependence of Dark Current in CCD Imagers. *IEEE Transactions on Electron Devices* **57,** 581–587. doi:10.1109/TED.2009.2038649 (2010).

52. R. Widenhorn, L. Mündermann, A. Rest & E. Bodegom. Meyer-Neldel rule for dark current in charge-coupled devices. *Journal of Applied Physics* **89,** 8179–8182. doi:10.1063/1.1372365 (2001).

53. J. A. Seibert, J. M. Boone & K. K. Lindfors. *Flat-field correction technique for digital detectors* in *Medical Imaging 1998: Physics of Medical Imaging* (eds J. T. Dobbins III & J. M. Boone) **3336** (SPIE, 1998), 348. doi:10.1117/12.317034.

54. R. Yu, S. Kennedy, D. Paganin & D. Jesson. Phase retrieval low energy electron microscopy. *Micron* **41,** 232–238. doi:10.1016/j.micron.2009.10.010 (2010).

55. S. M. Kennedy, C. X. Zheng, W. X. Tang, D. M. Paganin & D. E. Jesson. Laplacian image contrast in mirror electron microscopy. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* **466,** 2857–2874. doi:10.1098/rspa.2010.0093 (2010).

56. J. Jobst, L. M. Boers, C. Yin, et al. Quantifying work function differences using low-energy electron microscopy: The case of mixed-terminated strontium titanate. *Ultramicroscopy* **200,** 43–49. doi:10.1016/j.ultramic.2019.02.018 (2019).

57. S. M. Schramm, S. J. van der Molen & R. M. Tromp. Intrinsic Instability of Aberration-Corrected Electron Microscopes. *Physical Review Letters* **109,** 163901. doi:10.1103/PhysRevLett.109.163901 (2012).

58. *Catalog: MCP(MICROCHANNEL PLATE) AND MCP ASSEMBLY* Hamamatsu Photonics K.K., (), 1–2. https://www.hamamatsu.com/content/dam/hamamatsu-photonics/sites/documents/99_SALES_LIBRARY/etd/MCP_assembly_TMCP0003E.pdf.

59. H. Foroosh, J. Zerubia & M. Berthod. Extension of phase correlation to subpixel registration. *IEEE Transactions on Image Processing* **11,** 188–200. doi:10.1109/83.988953 (2002).

60. S. Klein, M. Staring, K. Murphy, M. Viergever & J. Pluim. elastix: A Toolbox for Intensity-Based Medical Image Registration. *IEEE Transactions on Medical Imaging* **29,** 196–205. doi:10.1109/TMI.2009.2035616 (2010).

61. D. Shamonin, E. E. Bron, B. P. Lelieveldt, et al. Fast parallel image registration on CPU and GPU for diagnostic classification of Alzheimer's disease. *Frontiers in Neuroinformatics* **7,** 50. doi:10.3389/fninf.2013.00050 (2013).

62. E. M. Ismaili Aalaoui & E. Ibn-Elhaj. Estimation of Subpixel Motion Using Bispectrum. *Research Letters in Signal Processing* **2008,** 1–5. doi:10.1155/2008/417915 (2008).

63. M. Guizar-Sicairos, S. T. Thurman & J. R. Fienup. Efficient subpixel image registration algorithms. *Optics Letters* **33,** 156. doi:10.1364/OL.33.000156 (2008).

64. G. Tzimiropoulos, V. Argyriou & T. Stathaki. Subpixel Registration With Gradient Correlation. *IEEE Transactions on Image Processing* **20,** 1761–1767. doi:10.1109/TIP.2010.2095867 (2011).

65. B. Schaffer, W. Grogger & G. Kothleitner. Automated spatial drift correction for EFTEM image series. *Ultramicroscopy* **102,** 27–36. doi:10.1016/j.ultramic.2004.08.003 (2004).

66. B. H. Savitzky, I. El Baggari, C. B. Clement, et al. Image registration of low signal-to-noise cryo-STEM data. *Ultramicroscopy* **191,** 56–65. doi:10.1016/j.ultramic.2018.04.008 (2018).

67. Dask Development Team. *Dask: Library for dynamic task scheduling* (2016). https://dask.org.

68. M. Rocklin. *Dask: Parallel Computation with Blocked algorithms and Task Scheduling* in *Proceedings of the 14th Python in Science Conference* (eds K. Huff & J. Bergstra) (2015), 126–132. doi:10.25080/Majora-7b98e3ed-013.

69. S. van der Walt, J. L. Schönberger, J. Nunez-Iglesias, et al. scikit-image: image processing in Python. *PeerJ* **2,** e453. doi:10.7717/peerj.453 (2014).

70. E. Jones, T. Oliphant, P. Peterson & E. Al. *SciPy: Open source scientific tools for Python, http://www.scipy.org* 2001. http://www.scipy.org.

71. M. A. Branch, T. F. Coleman & Y. Li. A Subspace, Interior, and Conjugate Gradient Method for Large-Scale Bound-Constrained Minimization Problems. *SIAM Journal on Scientific Computing* **21,** 1–23. doi:10.1137/S1064827595289108 (1999).

72. R. H. Byrd, R. B. Schnabel & G. A. Shultz. Approximate solution of the trust region problem by minimization over two-dimensional subspaces. *Mathematical Programming* **40-40,** 247–263. doi:10.1007/BF01580735 (1988).

73. F. Perez & B. E. Granger. IPython: A System for Interactive Scientific Computing. *Computing in Science & Engineering* **9,** 21–29. doi:10.1109/MCSE.2007.53 (2007).

74. F. Pedregosa, G. Varoquaux, A. Gramfort, et al. Scikit-learn: Machine Learning in Python. *Journal of machine learning research* **12,** 2825–2830. http://arxiv.org/abs/1201.0490 (2012).

75. D. P. Petersen & D. Middleton. Sampling and reconstruction of wave-number-limited functions in N-dimensional euclidean spaces. *Information and Control* **5,** 279–323. doi:10.1016/S0019-9958(62)90633-2 (1962).

76. E. H. Adelson, P. J. Burt, C. H. Anderson, J. M. Ogden & J. R. Bergen. Pyramid methods in image processing. *RCA engineer* **29,** 33–41 (1984).

77. P. Thevenaz, U. Ruttimann & M. Unser. A pyramid approach to subpixel registration based on intensity. *IEEE Transactions on Image Processing* **7,** 27–41. doi:10.1109/83.650848 (1998).

78. S. Wold, K. Esbensen & P. Geladi. Principal component analysis. *Chemometrics and Intelligent Laboratory Systems* **2,** 37–52. doi:10.1016/0169-7439(87)80084-9 (1987).

79. S. Jesse & S. V. Kalinin. Principal component and spatial correlation analysis of spectroscopic-imaging data in scanning probe microscopy. *Nanotechnology* **20,** 085714. doi:10.1088/0957-4484/20/8/085714 (2009).

80. H. Abdi & L. J. Williams. Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics* **2,** 433–459. doi:10.1002/wics.101 (2010).

81. N. Halko, P. G. Martinsson & J. A. Tropp. Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions. *SIAM Review* **53,** 217–288. doi:10.1137/090771806 (2011).

82. B. C. Geiger & G. Kubin. *Signal Enhancement as Minimization of Relevant Information Loss* in *SCC 2013; 9th International ITG Conference on Systems, Communication and Coding* (2013), 1–6.

83. P. Kovesi. Good Colour Maps: How to Design Them. *arXiv preprint.* http://arxiv.org/abs/1509.03700 (2015).

84. S. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory* **28,** 129–137. doi:10.1109/TIT.1982.1056489 (1982).

85. O. Renault, N. Barrett, A. Bailly, et al. Energy-filtered XPEEM with NanoESCA using synchrotron and laboratory X-ray sources: Principles and first demonstrated results. *Surface Science* **601,** 4727–4732. doi:10.1016/j.susc.2007.05.061 (2007).

86. J. Yamanishi, S. Iwase, N. Ishida & D. Fujita. Multivariate analysis for scanning tunneling spectroscopy data. *Applied Surface Science* **428,** 186–190. doi:10.1016/J.APSUSC.2017.09.124 (2018).

87. R. Okuta, Y. Unno, D. Nishino, S. Hido & C. Loomis. *CuPy: A NumPy-Compatible Library for NVIDIA GPU Calculations* in *Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Thirty-first Annual Conference on Neural Information Processing Systems (NIPS)* (2017), 7. http://learningsys.org/nips17/assets/papers/paper_16.pdf.

88. T. A. de Jong. *pyGPA* 2021. doi:10.5281/zenodo.5589555.