

Original Article

Fast Pixelated Detectors in Scanning Transmission Electron Microscopy. Part I: Data Acquisition, Live Processing, and Storage

Magnus Nord^{1,2*} , Robert W. H. Webster¹ , Kirsty A. Paton¹ , Stephen McVitie¹ , Damien McGrouther¹ , Ian MacLaren¹  and Gary W. Paterson^{1*} 

¹SUPA, School of Physics and Astronomy, University of Glasgow, Glasgow G12 8QQ, UK and ²EMAT, Department of Physics, University of Antwerp, Antwerp 2000, Belgium

Abstract

The use of fast pixelated detectors and direct electron detection technology is revolutionizing many aspects of scanning transmission electron microscopy (STEM). The widespread adoption of these new technologies is impeded by the technical challenges associated with them. These include issues related to hardware control, and the acquisition, real-time processing and visualization, and storage of data from such detectors. We discuss these problems and present software solutions for them, with a view to making the benefits of new detectors in the context of STEM more accessible. Throughout, we provide examples of the application of the technologies presented, using data from a Medipix3 direct electron detector. Most of our software are available under an open source licence, permitting transparency of the implemented algorithms, and allowing the community to freely use and further improve upon them.

Key words: 4D STEM, fast pixelated detector, file formats, live data processing, Medipix3

(Received 18 November 2019; revised 1 May 2020; accepted 4 June 2020)

Introduction

Several technological advances have been critical in the development of the scanning transmission electron microscope (STEM) from its inception (von Ardenne, 1938) to its current status as one of the most important techniques for high-resolution imaging of materials. Specifically, improved vacuum systems, field emission sources, aberration correction, and the introduction of annular dark field (ADF) detectors (Crewe, 1966; Crewe et al., 1968) were all crucial developments. ADF detectors are typically formed of one or more PN diode segments or scintillator photomultiplier tube arrangements. These are placed in the far field of the objective lens and sample an angular range of the diffraction pattern of the area of the sample illuminated by the electron beam. Such devices are well suited for use in STEM due to their fast readout, and imaging with pixel dwell times measured in microseconds is normal. ADF imaging was initially understood as being based on Z-contrast (Crewe, 1970a, 1970b), though understanding of the contrast mechanism evolved over time (Donald & Craven, 1979), in turn influencing the design of such detectors. In particular, later contributions demonstrated that the inner angle of ADF detectors had to be relatively high to exclude coherent diffraction

from dominating the signal (Pennycook & Jesson, 1991; Hartel et al., 1996). Other refinements of this arrangement have been introduced over the years, including the use of split detectors for differential phase contrast (Dekkers & de Lang, 1977; Chapman, 1978; Chapman et al., 1990; McGrouther et al., 2014), multiple annular detectors (Shibata et al., 2010, 2017), and the use of bright field or annular bright field imaging (Hammel & Rose, 1995; LeBeau et al., 2009; Findlay et al., 2010; MacLaren et al., 2015). However, all these detector configurations integrate over large angular ranges of the back focal plane, resulting in the loss of most of the information contained in the diffraction pattern. Furthermore, space constraints in the microscope's camera chamber can limit which detectors can be used simultaneously in an individual experiment, so that data acquisition may have to be repeated several times from the same area using different detectors to collect all the signals of interest. This can lead to difficulties in correlating the information contained in images acquired in successive experiments due to drift, and results in a higher overall dose to the sample, which is undesirable for beam-sensitive samples.

Recently, building upon technologies developed for particle physics (Turala, 2005; Wermes, 2005; Turchetta et al., 2007; Delpierre, 2014), pixelated detectors developed for X-ray imaging have been adopted for electron imaging (Clough et al., 2014; McMullan et al., 2014; McGrouther et al., 2015; Tate et al., 2016; Mir et al., 2017; Tinti et al., 2018). Compared with charge coupled device (CCD)-based detectors, these direct electron detectors (DEDs) typically offer much lower noise levels, improved detector quantum efficiency (DQE), and modulation

*Author for correspondence: Gary W. Paterson, E-mail: Dr.Gary.Paterson@gmail.com; gary.paterson@glasgow.ac.uk; Magnus Nord E-mail: magnus.nord@ntnu.no

†Current address: Department of Physics, Norwegian University of Science and Technology, Trondheim 7491, Norway.

Cite this article: Nord M, Webster RWH, Paton KA, McVitie S, McGrouther D, MacLaren I, Paterson GW (2020) Fast Pixelated Detectors in Scanning Transmission Electron Microscopy. Part I: Data Acquisition, Live Processing, and Storage. *Microsc Microanal* 26, 653–666. doi:10.1017/S1431927620001713

transfer function (MTF), some degree of radiation hardness, and crucially, fast readout of the images. This allows the efficient recording of the entire diffraction pattern at each scan position using millisecond or sub-millisecond dwell times, enabling either improvements in or the use of different imaging modes, such as nanobeam STEM diffraction (Mahr et al., 2019), position averaged convergent beam electron diffraction (LeBeau et al., 2010; Ophus et al., 2017), atomically resolved electrostatic field mapping (Hachtel et al., 2018; Fang et al., 2019), improved magnetic (Krajnak et al., 2016) induction characterization, determination of crystal periodicity along the beam direction (Nord et al., 2019a), fluctuation electron microscopy (Banerjee et al., 2017), and ptychography (Pennycook et al., 2015; Yang et al., 2016). Additionally, work is in progress to also improve scanning precession electron diffraction (SPED) (Rauch et al., 2010) using such DEDs, especially because of their better noise performance than optically coupled CCDs (MacLaren et al., 2020). Indeed, pixelated detectors are increasingly regarded as “universal” detectors (Tate et al., 2016; Yang et al., 2016; Hachtel et al., 2018; Fang et al., 2019; Ophus, 2019) capable of imaging under multiple modes.

Along with the many advantages that fast pixelated detectors bring, many practical limitations arise from their use, such as the ability to get real-time information from the data stream produced from a scan to enable navigation and identification of relevant sample features, and the storage and processing of very large datasets, often much larger than the available computer memory. In this paper (Part I), we present solutions for the hardware control, data acquisition, real-time processing and visualization, and storage of data from fast pixelated detectors. The majority of the software solutions presented in this work are made available under the free and open source GPLv3 licence, allowing transparency of the implemented algorithms, and the ability for anyone to use and to further improve upon them. The names of the software packages, modules, classes, and functions we present are given in typewriter font.

Most of the libraries reported here are implemented in Python. Python, being an open and free programming language, is rapidly becoming the standard language for many aspects of scientific computing (Oliphant, 2007; Gouillart et al., 2016). In addition to its comparative ease of use, which lowers the barrier for people to contribute and minimizes developer time, Python has an extensive standard library and a large ecosystem of external libraries, including ones for optimized numerical (Oliphant, 2006) and scientific (Jones et al., 2001) computing, image processing (van der Walt, 2014), data visualization (Hunter, 2007), and work flow documentation (Kluyver et al., 2016). Furthermore, it is straightforward to link Python to low-level C-code, allowing the development of optimized routines or use of external libraries (Behnel et al., 2011).

Within the electron microscopy community, a number of Python packages have also been developed. One example of this is HyperSpy (de la Peña et al., 2018), which contains functionality for processing data from a wide range of TEM techniques: electron energy loss spectroscopy, energy-dispersive X-ray spectroscopy, electron holography, and more standard imaging. It also serves as a base for several other packages, such as pyXem for analyzing SPED data (Johnstone et al., 2019), Atomap for processing atomic resolution STEM data (Nord et al., 2017), and pixStem for working with data from fast pixelated STEM detectors (pixStem devs, 2015). Several other packages exist, like rigidRegistration for doing rigid image registration of atomic resolution image stacks (Savitzky et al., 2018), and wrappers for doing STEM simulations, like PyPrismatic (Ophus, 2017). Other packages for

processing data from fast pixelated STEM detectors include py4DSTEM (Savitzky et al., 2019), LiberTEM (Clausen et al., 2019), pycroscopy (Somnath et al., 2019), and fpd (fpd devs, 2015).

The post-acquisition visualization and processing of data from fast pixelated detectors using the fpd and pixStem¹ libraries for the structural characterization of materials will be reported in Part II of this work (Paterson et al., 2020b). A third and final part (yet to be submitted) will cover aspects related to differential phase contrast analysis. Throughout all parts, we provide examples using data from a Medipix3 detector (Ballabriga et al., 2013). Although some sections of the codebase are specific to the use of this detector, such as aspects relating to data acquisition, many of the issues discussed and the techniques and tools presented in this part are applicable to a wide range of other detectors, while the data processing described in the forthcoming parts II and III of this paper series are applicable to data from any detector.

This paper is organized as follows. In the section “Medipix3 Detector”, the Medipix3 detector is briefly introduced. Methodologies for acquiring data from it are discussed in the section “Medipix3 Data Acquisition”. In the section “Live Data Processing”, an architecture developed to process a live data stream from a fast pixelated detector is outlined. In the section “Data Storage”, the issues around data storage are discussed and our implementation is presented. The source data and scripts to analyze the data and produce the results presented here are publicly available (Nord et al., 2019b).

Medipix3 Detector

All pixelated data reported in this work is from a 256×256 pixel Medipix3RX (henceforth referred to as Medipix3) detector (Ballabriga et al., 2013) affixed to a Merlin 1R retractable Medipix3 mount from Quantum Detectors (Harwell, Oxfordshire, UK). The Medipix3 detector is a radiation hard hybrid counting DED, where active analog and digital signal processing circuitry in each $55 \mu\text{m}$ pixel is bump-bonded to a relatively thick sensor layer. Si sensor layers of $500 \mu\text{m}$ are needed for operation at primary electron energies of 300 keV. In our case, a $300 \mu\text{m}$ silicon sensor layer was used for all data except that in Figure 1, where a $500 \mu\text{m}$ layer was used instead.

In electron microscopy applications, an incident electron produces electron-hole pairs in the sensor layer in sufficient numbers (Scholze et al., 1998) for the signal due to a primary electron to be clearly distinguishable from noise in the detector. This makes the detector capable of noiseless operation by the setting of an appropriate threshold for counting, and the detector is thus able to detect individual electrons. As a consequence, the Medipix3 detector is of potential use in time-resolved electron microscopy experiments, where sub-100 ns time resolution has been recently demonstrated (Paterson et al., 2020a).

Each pixel can operate independently, with its active circuitry processing only the signal induced in that pixel, in a mode of operation known as single pixel mode (SPM). Alternatively, in so-called charge summing mode (CSM), neighboring pixels can pool their circuitry and collectively process the signals induced in each pixel (Ballabriga et al., 2013). CSM attempts to account

¹After submitting this paper, it was decided to merge pixStem with pyXem (Johnstone et al., 2019). All of the features detailed in part I and II of this series of papers that are related to pixStem are in the processes of being added to pyXem and will continue to be available. The features of the fpd library remain unaffected.

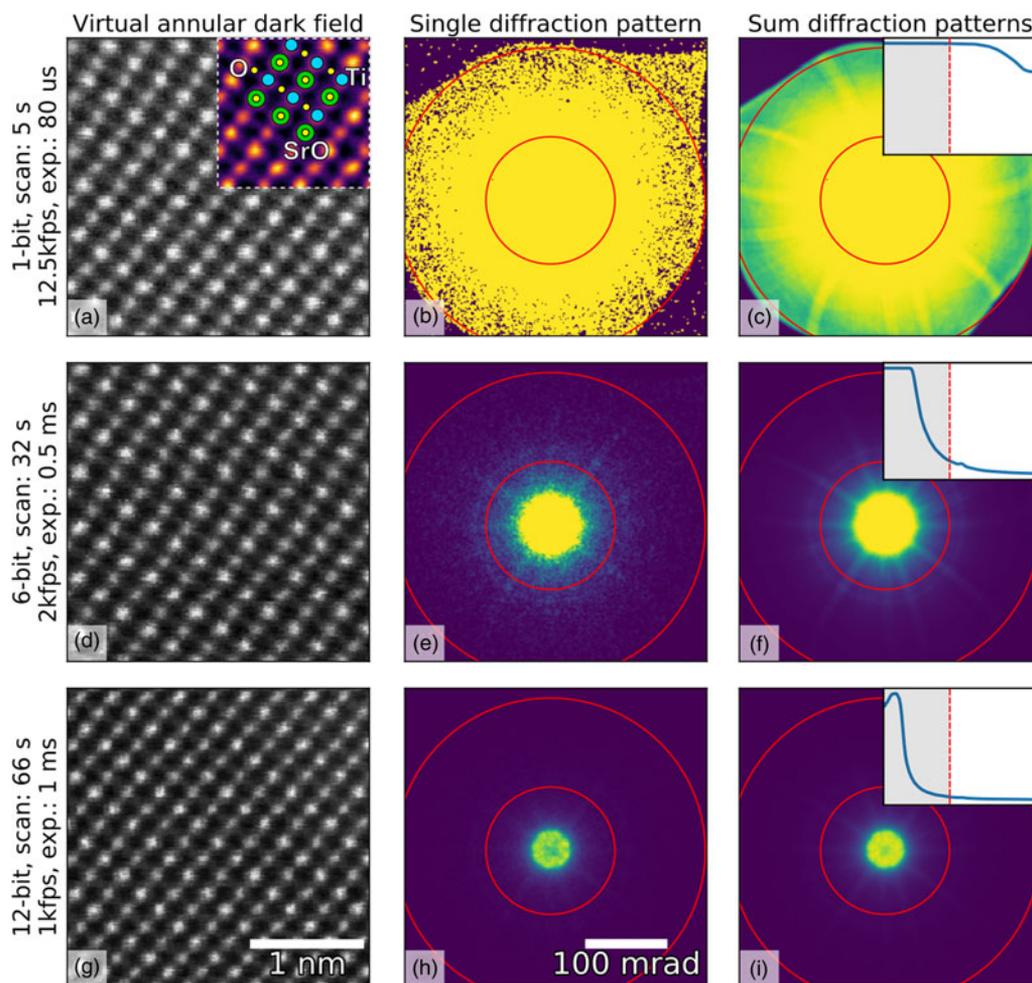


Fig. 1. Imaging of SrTiO₃ along the [110] direction using different bit depths and probe dwell times (in rows) with the Medipix3 detector in a continuous read-write mode. The bit depth, total scan time, and the detector frame acquisition rate and exposure are shown in the annotations. HAADF images (left column) were calculated by summing all counts inside a virtual aperture defined over the collection angles 80–192 mrad (assuming a linear mapping of pixel count to diffracted angle, which may not be entirely true in an image-corrected microscope), shown by the red lines in the diffraction images (middle column), using the pixStem library (pixStem devs, 2015). The colored section of the 1-bit HAADF image in (a) is Fourier filtered with a schematic overlay of the atomic columns imaged: green: Sr, yellow: O, and blue: Ti. The third column shows the summed diffraction patterns, with the insets displaying their radial distributions from 0 to 192 mrad. The dip in intensity at the centre of the direct spot in the 12-bit mode data in (h) and (i) is due to the higher bit depth, now allowing the details of the primary beam and low-order diffraction discs to be seen.

for charge spread between pixels due to electron-matter interactions in the thick sensor layer. At an acceleration voltage of up to 80 kV, the Medipix3 has a near-perfect DQE and MTF when imaging electrons (Mir et al., 2017). The use of alternative high-Z sensor-layer materials is expected to improve the performance at higher acceleration voltages (McMullan et al., 2007) and is currently being investigated.

Another notable feature of the Medipix3 detector is the ability to operate in a continuous read-write mode, where one of the two sets of counters in each pixel is used to readout the data while the other takes over counting. This gapless recording maximizes dose efficiency, which is important for beam-sensitive samples, and also enables faster acquisitions, which is important for minimizing artifacts due to microscope instabilities, particularly when imaging with atomic resolution.

The Medipix3 detector can be operated in 1-, 6-, 12-, and 24-bit depth modes, allowing the compromise between readout time, file size, and dynamic range to be varied. The clock on the Medipix3 was designed to be driven at frequencies up to

200 MHz but, with additional cooling, it can be overclocked to allow faster operation. With the 120 MHz clock rate of the Merlin readout system (Plackett et al., 2013) used here, the readout times are 70.8 μ s, 412 μ s, 822 μ s, and 1.64 ms, for 1, 6, 12, and 24-bit modes, respectively. While the 24-bit mode is ideal for very high dynamic range diffraction studies (Mir et al., 2017), the higher readout rates of the lower bit depth modes are more generally useful across a wide range of imaging conditions (it would take >4 ms to exceed 12 bits at 1 MHz count rates per pixel, so 24 bits are only needed for long counting times or high arrival rates on some pixels).

To demonstrate the use of different bit depths, atomic resolution data from SrTiO₃ imaged along the [110] direction were acquired on a Medipix3 detector at bit depths of 1, 6, and 12, giving maximum counts of 1, 63, and 4095, respectively. The data were acquired on a JEOL ARM 300CF using an acceleration voltage of 200 kV and a convergence angle of 22.4 mrad, with the Medipix3 operated in SPM with continuous read-write enabled. High-angle ADF (HAADF) images produced from these datasets

are shown in the left-hand column of [Figure 1](#), with the bit depth increasing from top (a) to bottom (g). The atomic resolution contrast in these images arises mostly from incoherent scattering of the electrons, similar to that in regular HAADF imaging with dedicated annular detectors. The middle- and right-hand columns show individual and summed diffraction patterns from each scan, respectively. The circular red lines in the diffraction patterns mark the edges of the virtual aperture used within which pixel counts were added up to give the intensity used for each pixel of the real-space images, while the insets in the third column show the radial distributions. The non-round “shadow” easily visible at the outer edges of the 1-bit diffraction pattern ([Fig. 1a](#)) is due to high-angle cutoff in the microscope due to the image corrector. Although the 1-bit diffraction patterns ([Figs. 1b, 1c](#)) do not seem to contain much information, the ADF data ([Fig. 1a](#)) show that high-quality atomic resolution imaging is possible, with the SrO, Ti, and O₂ columns (Abramov et al., 1995) all resolved, as shown in the inset schematic.

The very high frame rates of 12,500 frames per second with 1-bit data in a continuous read–write mode that are achievable makes this acquisition configuration particularly suitable for navigation during setup or in especially beam-sensitive materials. One image of 256 × 256 probe positions takes about 5 s to complete at this rate, but smaller scan sizes are often adequate for navigation. In the experiment, the beam current was maintained and this unavoidably resulted in the central portion of the diffraction pattern being saturated when recording at bit depths of 1 and 6. As shown by the radial distributions, we have selected the scattering angles where the detector is not saturated and contrast can be extracted. With shorter exposures or lower beam currents, regions closer to the central spot of the diffraction pattern will not be saturated and would produce useable image contrast. In 6-bit mode, more features of the diffraction pattern are visible than in the 1-bit mode, and the darkfield image ([Fig. 1d](#)) is better defined. This trend continues to the 12-bit mode ([Fig. 1g](#)) where the direct beam is no longer saturated, as shown in the inset in [Figure 1i](#). However, more atomic columns are present in the image as a result of larger spatial drift during the longer acquisition (9 or 10 Sr columns per row in the 12-bit data compared with 8 or 9 columns per row for the 1-bit data). The principal benefit of higher bit depth imaging in this context is that a greater range of scattering angles may be used for virtual aperture imaging post-acquisition, and the signal-to-noise ratio (SNR) is generally higher, even if there is a cost in acquisition time and consequent drift.

The selection of higher scattering angles by saturating the central spot in 1-bit and 6-bit modes is possible here because, unlike CCDs, the Medipix3 is not damaged by the very intense direct beam, and because the noise-free readout enables each single electron hit to be accurately recorded. With very intense beams (approximately 1 MHz count rate per pixel), the electron arrival rate can exceed the counting rate of the detector; this does no harm to the detector, but electrons are missed and the counts no longer represent an accurate reflection of arrival rates (and even a little below this level, counting linearity is lost). In this data, however, the beam current was not high enough to cause such an effect and the slight dips in intensity in the center of [Figures 1h](#) and [1i](#) are due to the real internal structure of the brightest portion of the diffraction pattern being resolved at the highest bit depth.

The fastest frame rate demonstrated above, corresponding to 80 μs per scan pixel, is still substantially slower than that of

commonly used scintillator or photomultiplier tube-based STEM detectors. While switching between these detectors is a common procedure with little overhead, pixelated STEM detectors with higher speeds would be beneficial for the efficiency of live-imaging and for low-dose imaging of beam-sensitive materials without having to reduce the beam current. The limiting factor in the Medipix3 setup we used is the 120 MHz clock of the Merlin readout system (Plackett et al., 2013). The Medipix3 chip itself can be clocked to 200–250 MHz, potential allowing a doubling of acquisition speed. Additionally, the Medipix3 chip allows readout of regions of interest (ROI), potentially allowing much higher sub-frame readout rates, but readout systems with this capability are not yet commonly available.

The pnCCD detector (Ryll et al., 2016) is a radiation hard 264 × 264 pixel CCD-based sensor with a full frame rate of 1000 fps. The chip supports binning along one axis, allowing speeds of up to 4,000 fps with 4 × binning (264 × 66 pixels). However, it was recently reported that these speeds may be doubled with optimization of the operation conditions and timing coordination of the readout ASICs (Huth et al., 2019). By windowing to a 24-pixel wide strip, the detector is reported to operate at 10,000 fps, which is approaching that of the 12,500 fps 256 × 256 1-bit data shown in this work. At the maximum SNR mode of operation and imaging 200 keV electrons, the maximum number of primary electrons that may be measured per pixel is about 1 in the pnCCD detector. This number rises to about 7 in the highest capacity mode of operation, but with a reduction in the SNR. This detector does not benefit from the gapless or noise-free readout of the Medipix detector used here, but can still produce excellent results at low doses (Huth et al., 2019).

Alternative modes of operation can yield even faster data readouts from current generation detectors. The Timepix3 chip (Poikela et al., 2014) operates with a 640 MHz clock, giving a timestamp resolution of 1.56 ns, and supports a data-driven mode of acquisition. In this mode, only data from events are read out rather than the full array of pixel counts. The data packet itself includes additional information such as time over threshold and time of arrival and, consequently, additional or alternative strategies are required to process this type of data (with the potential benefit of more advanced signal processing). The maximum hit rate is 40 M hits/s/cm² (approximately 80 M hits/s for a single 1.98 cm² chip) and provides faster readout in an event driven mode than in a frame mode for less than 50% occupancy. While this approach allows very short effective exposures, due to the increased size of the packet, the overall counting rate is reduced compared with the Medipix3 detector. However, unlike in the 1-bit Medipix3 data, the intensity distribution of the image signal would be accurately recorded without saturation in event driven modes of acquisition.

Beyond this, the collaboration behind the next-generation Medipix4 detector is targeting imaging rates which are compatible with human CT imaging (Campbell et al., 2016). The typical detector dose varies in CT imaging, but can be of the order of 10³–10⁴ M hits/s/cm² (Taguchi & Iwanczyk, 2013), significantly higher than is possible in current-generation Medipix detectors.

Regardless of the imaging mode used for the collection of the source data, smaller bit depths also make both the file storage and the data processing more efficient; a 6-bit dataset is about four times smaller than a 24-bit one of the same scan area, making it much more convenient to store and transfer. This advantage also extends to the data processing, since loading and processing data files which are four times smaller will be much quicker.

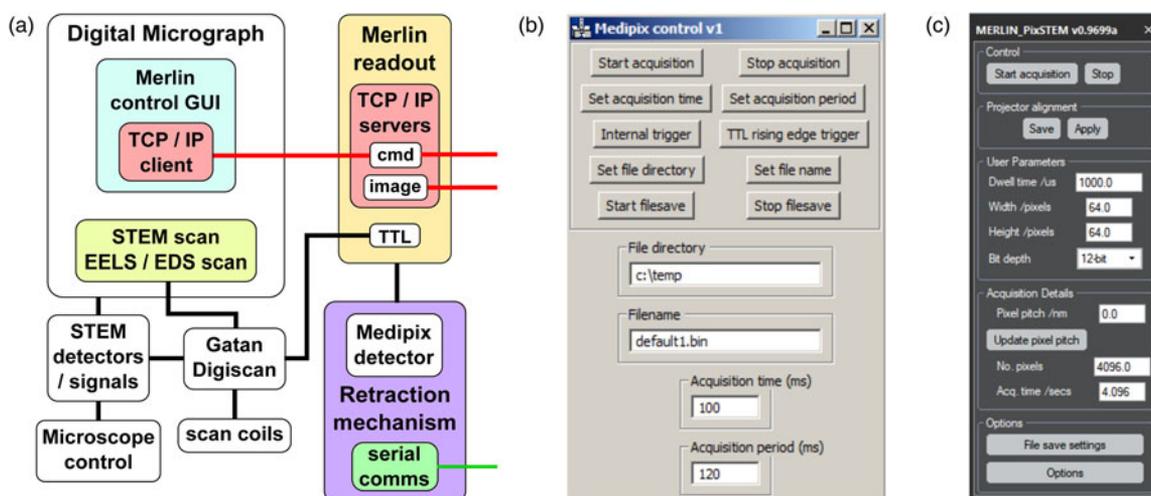


Fig. 2. (a) Schematic of the Merlin Medipix3 readout control architecture and screenshots of (b) low-level (Merlin Control) and (c) optimized continuous read-write mode (MERLIN_PixSTEM) plugins for Gatan DM. Note that the DM plugin in (c) is not related to the similarly named Python `pixStem` library.

Medipix3 Data Acquisition

Data from the Medipix3 detector were acquired through the Merlin readout system (Plackett et al., 2013). This allows setting of the acquisition parameters, either through a graphical user interface (GUI) or over TCP/IP, and reads and processes the raw data through a field-programmable gate array (FPGA), returning the data to the acquisition computer. The FPGA processing can be bypassed to some extent by operating the system in a “raw” mode. This enables larger scan sizes at high frame rates, with the requirement that the data must be reshaped post-acquisition, and with no live visualization of the acquired images directly in the Merlin software. However, the Merlin TCP/IP data API remains functional, so it is possible to get live imaging through other means (see the section “Live Data Processing”).

The Merlin system can be triggered by software over TCP/IP or by hardware (TTL) input. We typically use the latter approach and couple to the TTL signals produced by a Gatan DigiScan system, as shown schematically in Figure 2a. This produces extra acquisitions due to triggers sent during the flyback time, and the handling of these is discussed in the section “Data Storage”. The main advantage of this approach is that Gatan Digital Micrograph (DM), in addition to allowing access to microscope control, can be used for setting scan parameters in one of several ways discussed below, and additional STEM detector signals may be acquired simultaneously.

The simultaneously acquired DM datasets also serve to document the microscope and scan parameters in the data tags, which can then be used in data conversion (discussed in the section “Data Storage”), abstracting away the differences in how various microscope manufacturers provide microscope configuration information.

When regular STEM detectors can be used to navigate, the images produced from them may be used to set ROI scans using an image produced by a prior “survey” scan, following the spectrum imaging methodology, or regular STEM scans may be used to maximize read rates. When these approaches are used, a scripted DM plugin may be used for setting low-level Merlin parameters (Merlin DM Plugin devs, 2017), as shown in Figure 2b. Alternatively, real-space pixel sizes and scan ranges may be set and low-level DM commands used to configure and

enable the scan. A scripted DM GUI has been developed, MERLIN_PixSTEM, to coordinate this with configuring the Merlin system to acquire data in the optimized continuous read–write mode and is shown in Figure 2c. Among the other features implemented, this plugin also allows different projection system settings to be saved and restored, enabling efficient switching between different detectors. The Merlin system includes two TCP/IP servers, shown in red in Figure 2a, one for setting and reading acquisition parameters and the second for image data transfer. The DM-scripted GUIs, indicated in cyan in Figure 2a, interface with the Merlin communication server through a separate TCP/IP C++ plugin (Merlin DM Plugin devs, 2017), shown in red. The TCP/IP plugin may be installed alone, allowing it to be used for many other communication purposes. For more advanced control of the Merlin system over TCP/IP, a Python implementation of Merlin TCP/IP commands has been developed (Merlin Interface devs, 2016).

An example of an additional STEM signal we collect is the noise correction (NC) signal, which may be used for gun-noise correction of the 4D dataset. The NC signal is produced by a current pickup attached to the condenser aperture and gives a measure of the gun emission. A similar approach to gun signal measurement was recently reported and shown to have good linearity to the probe current (House et al., 2018). Correction of gun noise is particularly useful in intensity-based low contrast imaging modes, as shown in the bright field (BF) images of a mouse liver microtomed thin section in Figure 3. The image in Figure 3a is produced by summing the entire diffraction pattern (an example is shown in the inset) at each scan position. Sample contrast primarily arises due to incoherent Rutherford scattering of electrons to angles beyond the detector. The large circular feature in the bottom left corner is part of a mitochondrion organelle, while the darker spotted stripe structures are endoplasmic reticula studded with ribosomes.

The horizontal stripes in the as-measured image in Figure 3a are from short-period variations in the cold-FEG emission. The gun signal measured by the NC detector is shown in the inset of Figure 3b. Figure 3b itself shows the corrected image produced by minimizing the contrast introduced by these gun emission current variations using a linear gun-noise model. This reveals much

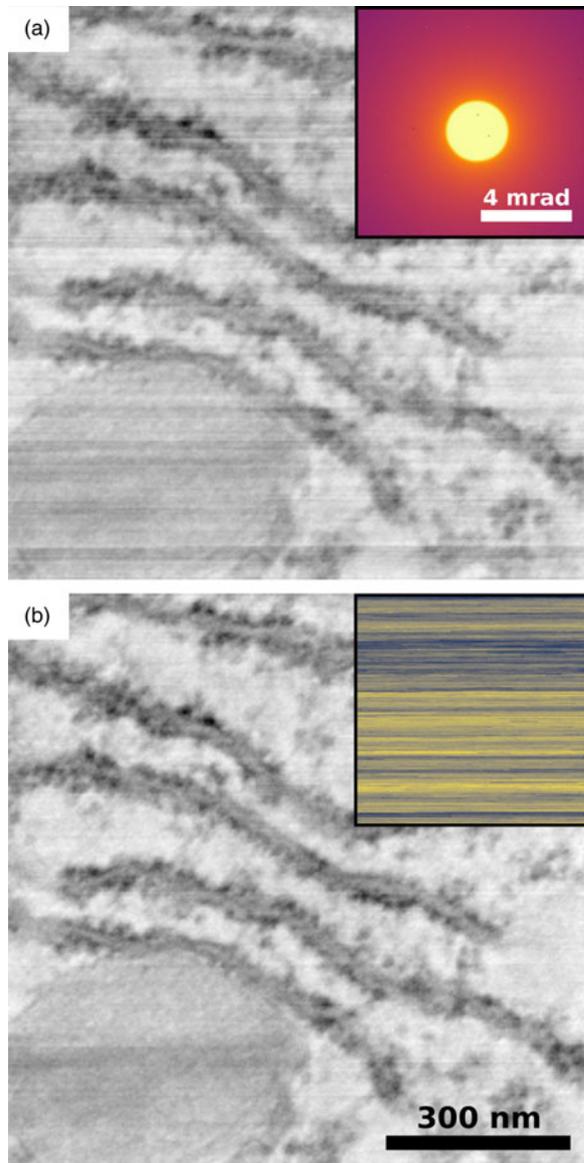


Fig. 3. Gun noise correction in a BF STEM image produced from a 4D dataset from a thin fixed but unstained section of mouse liver, showing part of one cell including the end of a mitochondrion organelle and part of the rough endoplasmic reticulum. Measured (a) BF image and (b) the same image after gun noise correction using the `fpd.tem_tools.nc_correct` function. The inset in (a) shows the summed diffraction pattern on a logarithmic scale, while that in (b) shows the recorded gun noise. The acceleration voltage was 200 kV, the objective lens was off, the condenser aperture was $30\ \mu\text{m}$, the camera length was 600 cm, the convergence semi-angle was 13.1 mrad, and the pixel spacing was 3.7 nm.

more detail in the BF image than previously seen in Figure 3a. The residual horizontal features in the corrected image are most likely a result of variations in linearity of the NC detector signal from things like amplifier drift or external noise. Taking the corrected image as a reference, the power SNR of the uncorrected image, calculated using the implementation of the two-image method (Frank, 1980) in the `fpd.utils` module, is 11 dB, giving a measure of the improvement in the image quality by applying gun-noise correction.

The Medipix3 data acquired following the above methodologies can be saved to disk on the acquisition computer in a flat binary format and may also be sent over the network using

TCP/IP. The conversion of the binary data to more appropriate formats is discussed in the section "Data Storage". The network transfer of data has many potential uses, and we discuss these in the context of live data processing in the next section.

Live Data Processing

Live feedback from the data collected by a fast pixelated detector in a STEM acquisition is crucial for both optimizing imaging conditions and navigating to ROI in a sample. This is especially true for some modes of imaging where traditional STEM detectors may not produce useful contrast, such as when imaging magnetic features which are typically not visible in STEM without a custom segmented detector and a readout system (McGroutner et al., 2014). To facilitate real-time feedback, we developed the Python library `fpd_live_imaging` (FPD Live Imaging devs, 2015), which implements multiple common analysis routines and wraps processing routines from other libraries (fpd devs, 2015). Although the `fpd_live_imaging` package was developed for use with the Medipix3 detector and the Merlin readout system, its design is modular and can easily be extended to work with any detector. Our implementation takes advantage of the many cores available in modern CPUs by employing Python's multiprocessing library. Shared parameters and data are passed between the separate processes through "queue" objects or other shared memory. This approach enables good performance, even at very high data rates.

The internal workings of the package are outlined in Figure 4. The Medipix3 1R insertion and retraction mechanism (shown colored in purple, *i*) is controllable through a serial interface (shown in green) and is made possible through library function calls. As discussed in the previous section, the Merlin system (drawn in yellow, *ii*) can be interfaced with via two TCP/IP servers (shown in red), which are utilized by the `fpd_live_imaging` package (white, *iii*) to get data from the detector and to control the acquisition of data. The first step in the visualization is receiving the raw binary data from the Merlin TCP/IP data interface using the `receive_data_medipix` function (*iv*). This function runs a TCP/IP socket which gets the raw binary data and passes it along to a parser function. The function has its own CPU process to be able to handle the very high framerate of the Medipix3 detector. Due to the nature of the TCP/IP protocol, the raw binary images can be split into different fragments. These fragments are pieced together in the parse function, which results in the image in the form of a NumPy array (Oliphant, 2006). The function also handles the bit depth of the data and the number of pixels in the detector, and also runs in its own separate CPU process.

After having constructed the image in the form of the NumPy array, a copy is sent to any number of data processing classes. These data processing classes are shown in blue (*v*) in Figure 4 and can be separated into two categories based on the imaging mode: scanning and parallel. The scanning data classes include things like virtual bright field and annular darkfield, where the input detector image is reduced to a single output value. In the parallel data classes, the output image is the same size as the input one, and the processing methods include passing through the input image, a thresholded version of the input image, or a Fourier transformed image. All these run in separate CPU processes. In addition to the aforementioned processing classes, are ones for single pixel extraction, center of mass, and phase-correlation for electro- or magneto-static field imaging, and routines for HOLZ analysis. Virtual detector imaging and HOLZ data

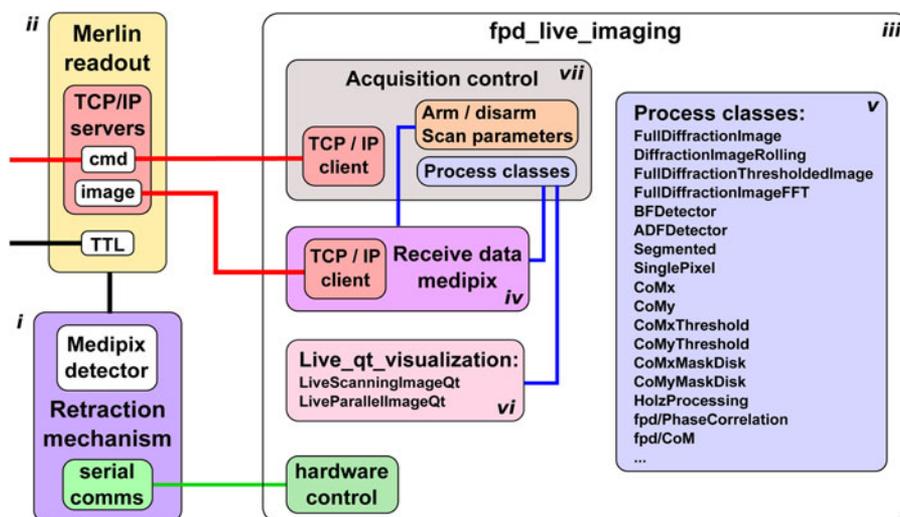


Fig. 4. Schematic of the fast pixelated detector live visualization library architecture, showing the relationship between the Medipix3 detector and the retraction mechanism (purple, *i*), the Merlin readout system (yellow, *ii*), and the `fpd_live_imaging` library (white, *iii*). Multiple processing classes (blue, *v*) are implemented for scanning and imaging modes.

processing are covered in Part II of this work (Paterson et al., 2020b), while field mapping will be covered in Part III.

The processing time varies greatly, depending on the computational complexity of the routine (Nord et al., 2016), and the choice of routine depends upon the nature of the sample. For example, in magnetic imaging, the integrated induction components perpendicular to the electron path can be determined from deflections in the position of the bright field disc (Chapman & Scheinfein, 1999). The center of mass calculation provides good contrast in many cases but can be affected by the crystallinity of the sample due to intensity diffracted from the bright field disc to angles either outside or inside the detector collection angle (Chapman et al., 1990). Phase- or cross-correlation (Krajnak et al., 2016) approaches can greatly improve upon this at the expense of computation time and can be crucial to detecting magnetic contrast in highly diffracting samples. On the other hand, single pixel extraction, where a single pixel on the edge of the disc is used as a measure of up to around pixel-level disc shifts, requires the minimum of processing and is orders of magnitude faster, taking approximately $2 \mu\text{s}$ when the 256×256 scan position 12-bit dataset is in memory (Nord et al., 2016). As each selected pixel gives a measure of a component of the integrated induction in a direction tangential to the disc, the use of only two pixels out of each diffraction image is sufficient to form a qualitative 2D vector map, which allows the user to at least navigate to an appropriate position, and set magnification and focus. Multiple processes may be run sequentially or simultaneously, allowing the tradeoff between runtime and sensitivity to be seen in real time.

The output data from any kind of processing is sent to a visualization class, which shows the result of the processing on the computer running the `fpd_live_imaging` package. Due to rescaling of the intensity to optimize the contrast, this visualization is qualitative, while the calculations themselves can be quantitative. This computer may be anywhere on the network. The visualization is separated into parallel and scanning modes, as shown in pink (*vi*) in Figure 4, and they also run in separate CPU processes. An example of the visualization GUI is shown in Figure 5b. In this case, the image is from thresholded center of mass analysis of data

from a patterned DC sputtered 8 nm permalloy film capped with 4 nm of copper. The $2 \mu\text{m}$ discs were patterned with a Ga focused ion beam, and the contrast in the resulting structures shows they support magnetic vortices. A detailed study of the sample will be published elsewhere. The GUI has buttons for setting the brightness and contrast during the acquisition, and the analysis parameters can be tuned during imaging, allowing for live optimization of the required contrast. Alternatively, the processed data can be sent over TCP/IP to any computer on the network, for example, directly into Digital Micrograph.

All the above processes are orchestrated from the “Acquisition Control” class (shown in brown, *vii* in Figure 4), which handles the initialization and connection of all of these separate functions. For ease of use, the Acquisition Control class can be accessed through a GUI, as shown in Figure 5a. This allows for starting and stopping of the acquisition, modification of the scan parameters, the addition and removal of processing classes, modification of their parameters, and insertion and retraction of the detector itself.

The three separate stages described above, reading data from the detector, processing the images, and visualizing or sending the result over TCP/IP, are implemented in modular design, making simple the addition of new detector data sources, image processing classes, and visualization.

Data Storage

The principal issues when choosing a file format for fast pixelated detector data are common across data from all detectors: the ability to store the data with the dimensionality of the scan, store metadata along with the detector data, allow access to subsets of the data without reading the entire and often very large dataset into memory, support compression, and be an open format with read and write support across a variety of programming languages. An HDF5 (The HDF Group, 1997–2018)-based format was chosen for our use since it meets all of the above requirements.

The HDF format has long been widely used in the synchrotron community and is increasingly being used in electron microscopy

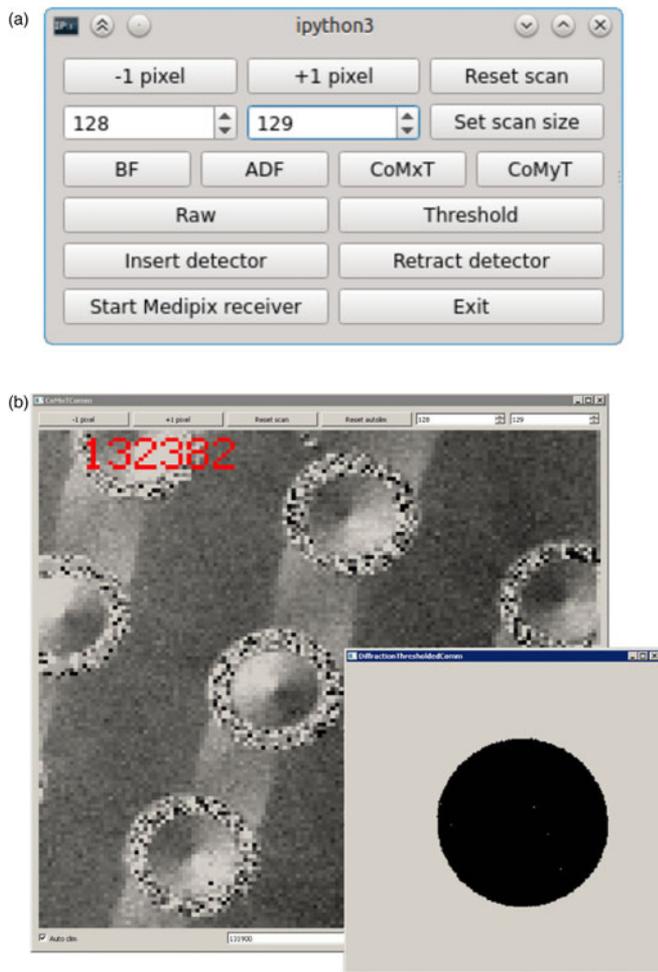


Fig. 5. *fpd_live_imaging*'s graphical user interface showing (a) the control window for the visualization and (b) a thresholded center of mass contrast of a patterned 8-nm permalloy film capped with 4 nm of copper. The contrast in the $2\ \mu\text{m}$ discs represents the beam deflection along a single axis and shows that the discs support magnetic vortices. The inset in (b) shows the thresholded detector image.

(de la Peña et al., 2018; EMD authors, 2019; Somnath et al., 2019). It can be both read and written in a number of programming languages, including MatLab, C++, Python, Java, R, and Gatan Digital Micrograph through a third party plugin (Niermann, 2016). The HDF5 format consists of an arbitrary structure of hierarchies of groups containing further groups or datasets, enabling the relationship between data to be indicated by the file structure. For datasets, the data-type definitions are stored with the data, making it self-describing and ensuring maximum portability. Additionally, all groups and datasets can have attributes, allowing user and acquisition metadata to be stored along with the detector data in appropriate locations. The datasets may be of any number of dimensions, and so, it is ideal for multidimensional data from fast pixelated detectors when used in STEM or other acquisition modes.

HDF5 has in-built support for a variety of compression algorithms and other so-called “filters,” all providing transparent read and write access to the data. To allow access to subsets of the data without having to decompress the entire dataset, the dataset can be divided into smaller pieces and stored in a B-tree, a balanced hierarchical data structure, by enabling “chunking.” Figure 6a shows an example of the potential

chunking of a one-dimensional (1D) scan dataset. The stack of images (shown on the left) occupy a 3D data “cube” (middle), with one axis being the scan dimension. On the right of panel (a), we show the same dataset with two chunks along each dimension, with each chunk in a different color. The dataset access sequence is summarized in Figure 6b. When indexing a chunked dataset, the B-tree is navigated, each chunk containing the required data is decompressed, and only the selected components are returned. For example, when reading the image slice shown by the blue dashed line in the right of Figure 6a, each of the top four chunks must be read.

Chunk Size

When choosing a chunk size, a compromise is made between the cost of B-tree navigation, compression level, and data reading speed, with the optimum choice ultimately depending on the intended data access pattern. For STEM data, the diffraction pattern can be sparse and compression can be optimized by chunking in both the scan and image dimensions. Although most of the issues discussed in this section will be common to all detectors, we note that the Medipix detector has three relevant features which separate it from most others. The first two are that the detector can unambiguously detect single electron impacts, and that the data are read out free of addition noise, meaning that dark regions of images can be truly filled with zeros rather than noise. The third point is that data from the Medipix3 detector are zero padded to align it with common data types (e.g. 12-bit data are stored as 16-bit), allowing compression to achieve significant reductions in file size. For example, with a chunking of 16 along each axis of a 4D dataset, the in-built lossless GZIP compression at level 4 typically reduces the data size of a scanning acquisition of 256×256 probe positions in a 12-bit mode from 8.6 to 2.7 GB.

Many chunking strategies are possible, and here, we explore two of them: a hypercube with equal chunk sizes across all axes; and a hyperrectangle with equal chunk sizes along the scan axes, and detector axes chunk sizes matching the detector dimensions. The first approach gives the most uniform data access properties across different axes and, importantly, can improve data processing efficiency by allowing reduction of the volumes of data that must be read for some analyses. For example, with a chunk size of 16 along each axis, getting the direct beam in a dataset where it resides in four chunks would require loading into memory only those chunks, corresponding to only 1.6% of the total file. The second approach of hyperrectangles is more suited to applications which only ever access full images, since requesting even a single pixel from an image would require the entire image to be read by the HDF5 library, and, similarly, indexing a 2D slice perpendicular to the detector axis would require the entire dataset to be read, resulting in significant overheads.

Figures 6c–6f show three HDF5 performance metrics as a function of the chunk edge length for the two chunking approaches (hypercube: solid lines; hyperrectangle: dashed lines) using the liver sample data in Figure 3. The three metrics are normalized write time (Fig. 6c), compression ratio (Fig. 6d), and normalized read time (Figs. 6e, 6f). The write and read times were normalized with respect to the values for the hypercube chunk length of 16 (marked by symbols), while the compression ratio is with respect to the size of the raw Merlin data file. Hypercube chunks of length 16 are used as the reference because

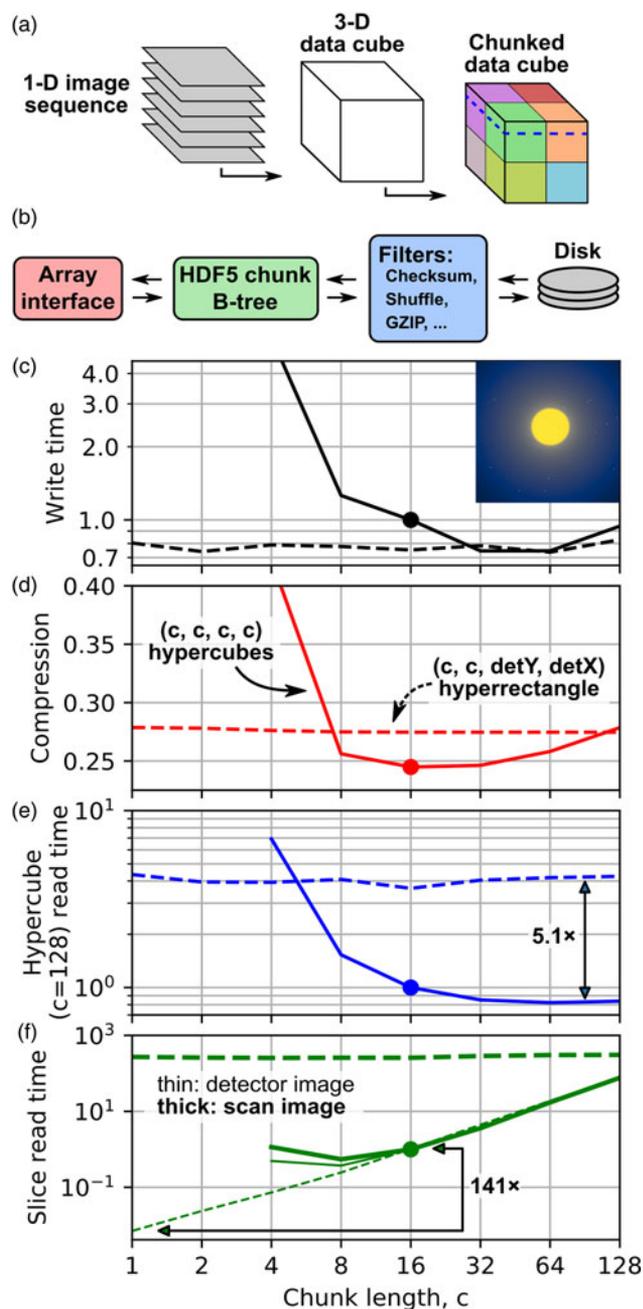


Fig. 6. (a) Example of potential dataset chunking for data from a 1D scan stored in an HDF5 file. (b) Data indexing sequence for chunked data. (c–f) HDF5 chunk performance metrics for the 256×256 probe position STEM dataset from Figure 3 with a 256×256 Medipix3 detector in a 12-bit mode. Level 4 GZIP compression was used. All panels show metrics for hypercube chunks (solid lines) and hyperrectangle chunks (dashed lines) with dimensions matching those of the detector (detY, detX). The inset in (c) shows a diffraction image on a logarithmic scale. The (e,f) read and (c) write times are the ratios of the values to those for the hypercube chunk length of 16 (marked by symbols). The compression ratios (d) are of the entire HDF5 file relative to only the raw Merlin binary file. The read times in (e) and (f) are those required to load a 128-sided hypercube or single slices of the dataset, respectively, into an in-memory NumPy array using h5py.

they are the default values in our implementation, as a result of it being a reasonable compromise for most applications.

For hyperrectangle chunks, the write time (Fig. 6c), compression ratio (Fig. 6d), and read time for a 128-side length hypercube

(Fig. 6e) are mostly independent of chunk size, whereas there are significant variations for hypercubes. As the chunk size is increased, all three metrics start very high (poor) due to the very large number of chunks, and then either go through a minimum at a chunk length around 16–32 (write time and compression) before increasing slightly, or plateau (hypercube read time). With all else being equal, the hypercube read time for the largest chunk sizes in Figure 6e should be $4\times$ worse for the hyperrectangle than for the hypercube, due to the overhead from the HDF5 library having to read the entire dataset, and the actual value of 5.1 is close to this. Similarly, reading a subset of any of the chunk sizes shown in Figure 6e will result in performance reduction, with smaller chunk sizes being less affected.

In Figure 6f we show the read times for indexing a single slice of the dataset, with the slice creating a detector image (thick lines) or a scan image (thin lines). As explained above, when indexing across a non-detector axis, the entire dataset must be read and this is the reason for the high and chunk size independent read times for the hyperrectangle approach (thick dashed line). Above a chunk length of around 16, all other datasets lie on top of one another and follow a linear relationship. Below this point, the hypercube read times begin to plateau and then increase, while the hyperrectangle chunked read times maintain the linear trend all the way to chunk lengths of 1. At this chunk size, reading a single image is $141\times$ faster than for our default setting of length 16 hyperchunks, which is not as fast as the $256\times$ smaller data would predict, due to additional overheads. Thus, for reading single images, having chunks of single images is a simple strategy to maximize performance, at the expense of flexibility in reading the data in other ways. However, a very simple method that often allows for similar read speed while maintaining flexibility is to read images (when they are needed as full images) from a hypercube chunked dataset in numbers that are aligned to a chunk size. This is the approach of data processing using the fpd package, which is discussed in more detail in Part II (Paterson et al., 2020b). For the example in Figure 6f, taking length 16 hypercubes as the reference, reading 16×16 images with hyperrectangles of length 1 takes $0.95\times$ the reference time, while reading the same with hyperrectangles of length 16 takes $1.12\times$ the reference time. For markedly different datasets or where the data access pattern is known in advance, the optimum chunking may be somewhat different, and this can be set by the user at the point of conversion.

Merlin Data

The Merlin readout software stores the detector and readout system parameters in a separate header file, and the detector data as a stream of uncompressed binary data, with each image containing a variable length header of acquisition parameters specific to that image. The MerlinBinary class from the fpd_file module of the fpd library (fpd devs, 2015; fpd demos devs, 2018) allows parsing of data files and array access to raw data using memory mapping and conversion to the HDF5 format. The scan parameters and metadata can be extracted from Digital Micrograph files acquired simultaneously with the diffraction patterns, or may be supplied separately. For the former case, the DM files are accessed through the HyperSpy library (de la Peña et al., 2018) and are also embedded in the HDF5 file as raw binary blobs for reuse in the proprietary DM software. All DM files are also stored in the HDF5 file in the open EMD format (EMD authors, 2019) (discussed in the next section). Examples of Merlin data converted

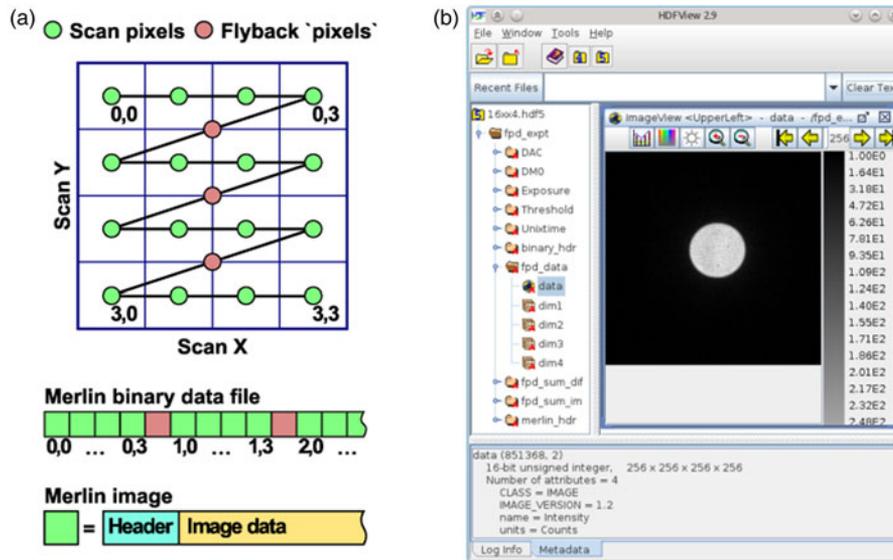


Fig. 7. (a) Example of additional “flyback” pixels in a 2D STEM scan and the on-disk Merlin binary file structure. (b) Overview of our HDF5 file structure, shown in HDFView (The HDF Group, 2017), using the same data as used for Figure 3.

to HDF5 format using the MerlinBinary class are available in the open data deposit for this work (Nord et al., 2019b).

Figure 7a shows an example of a 2D scan when the acquisition is being triggered by the microscope scanning system. Images, indicated in green, are acquired on a regular scan grid. As discussed in the section “Medipix3 Data Acquisition”, during the time when the beam is being moved from the end of one row to the start of the next, the “flyback” time, the DigiScan system continues to send triggers, causing additional images to be acquired. These are shown in red and may be excluded during data access and conversion to the HDF5 format with appropriate parameter settings.

The image data in the Merlin binary file is in C-order, that is, with the fastest moving index being in the last dimension, as depicted in Figure 7a. The HDF5 library is a self-describing one and returns datasets stored within it in the form appropriate for the library being used, but stores the data internally in C-order. C-order is also the default ordering in NumPy, and thus, we naturally store the multidimensional pixelated STEM datasets in C-order, with the first axes being the scan ones and the last two being the detector ones. Most pixelated STEM datasets are 4D and of the type described. However, the Medipix3 detector can be operated in color mode, where an additional axis representing multiple thresholds exists between the scan and detector axes. This axis, while not generally used in STEM acquisitions at present, is used for spectroscopic X-ray imaging, and is useful for characterizing the detector performance using X-rays, and is supported by the fpd library.

While the HDF5 conversion is most appropriate for data archival and later processing of data acquired under all modes of operation, the MerlinBinary class also provides a memory mapped array interface to the data on disk for most, but not all, acquisitions. For example, 1-bit data acquired in raw mode is stored as 1-bit by the Merlin system and has the image segments out of order, and cannot currently be easily memory mapped. However, in most cases, this mode of access allows the dataset to be visualized or processed without conversion of the data on disk to the HDF5 format, which is

particularly useful for checking datasets immediately after acquisition.

HDF5 File Structure

Figure 7b shows an overview of one of the HDF5 files read in HDFView (The HDF Group, 2017), a Java GUI program that allows, among many other things, quick inspection of HDF file contents. Information from the binary headers such as the DAC values, the image exposure, the comparator threshold values, and the acquisition time are automatically extracted and included as datasets in the HDF5 file. During conversion to the HDF5 format, a sum of both the image and diffraction dimensions are generated and stored in the HDF5 file as separate datasets, resulting in bright field and diffraction sum images. These images may be used for data inspection and navigation without having to process the entire dataset in order to re-render them every time the file is loaded. Also during conversion, any bad pixels in the detector (hot, noisy, or dead) may be replaced by interpolated values when the user supplies a mask image, which can be important for optimization of some forms of data analysis. Additionally, an arbitrary function may be applied to each image during conversion. This has many possible uses, including to correct for unwanted image shifts due to descans and to apply image transforms to correct image distortion.

The detector data, including images created from it, is stored in the EMD format, a simple open subset of the HDF5 format, created by a specific collection of datasets and attributes (EMD authors, 2019). The EMD datasets may be read in software such as EMD viewer (EMDViewer devs, 2015), HyperSpy (de la Peña et al., 2018) and, of course, any HDF5 reader. Many utility functions are also provided in the fpd.fpd_file module, allowing conversion to other formats and extraction of data. These include ones to access datasets as a namedtuple (fpd_to_tuple) which allows for indexing or tab completion and one for accessing the same as HyperSpy objects (fpd_to_hyperspy).

Other projects have also adopted or are compatible with the EMD format (de la Peña et al., 2018; Savitzky et al., 2019), and

we note that work is currently underway by the LiberTEM project (LiberTEM devs, 2018) to include (transmission) electron microscopy data in the NeXus data format (Könnecke et al., 2015). The Nexus format is another open subset of the HDF5 file format, originally developed to improve the data exchange within the fields of neutron, X-ray, and muon experiments. Having a common data format across all of these fields would be beneficial, since it would make the sharing of data and of processing routines that rely on metadata easier than is currently the case. This format could clearly be used within the suite of tools described in this paper.

Merlin Equipped Scanning Precession Electron Diffraction Systems

A recent development in precession electron diffraction (PED) is the use of fast pixelated detectors (Midgley & Eggeman, 2015). One such example is the work by NanoMEGAS to incorporate a Medipix3 DED into their DigiSTAR precession system in order to enable high fidelity recording of diffraction patterns in scanning PED (SPED) applications, as has been demonstrated in recent work (MacLaren et al., 2020). Additional benefits brought about by the use of DEDs in SPED will be discussed in Part II of this work (Paterson et al., 2020b). We note here, however, that the properties of the 4D dataset obtained by such a system are in many ways equivalent to those of 4D non-SPED datasets, and so many of the same issues of data access, storage, and processing apply here too. To enable these datasets to be more easily used, the `topspin_app5_to_hdf5` function of the `fpd.fpd_io` module allows the conversion of data originally recorded in the native NanoMEGAS TopSpin app5 format to one almost identical to the HDF5 format outlined above. Alternatively, the Merlin acquisition software can be programmed to output the data directly to a raw file, whilst the acquisition is being performed and controlled by the TopSpin software. The main differences between the converted files is the inclusion of precession metadata instead of Medipix3 metadata in SPED datasets, and the absence of simultaneously acquired DM datasets.

Summary

The use of fast pixelated detectors for electron imaging is a burgeoning field with the prospect of revolutionizing many aspects of TEM and, in particular, scanning TEM. We have presented many of the key tools needed to (i) acquire data from fast pixelated detectors, (ii) analyze in real time the data from one and visualize the results, and (iii) store data from them in an optimized way.

The software packages presented are hosted in public repositories (fpd devs, 2015; FPD Live Imaging devs, 2015; pixStem devs, 2015; Merlin Interface devs, 2016; Merlin DM Plugin devs, 2017; fpd demos devs, 2018), are under active development, and contain many more features than are covered in this short publication. Many of the data analysis algorithms in these libraries are applicable to data from any detector. Most of these packages are provided under an open source licence, allowing transparency of the algorithms implemented and for them to be continually improved upon by the community.

Part II of this paper will cover post-acquisition processing and visualization of data from fast pixelated detectors, with examples of their application to the study of the structure of materials studied using scanning TEM (Paterson et al., 2020b). A final part III will cover aspects related to differential phase contrast analysis.

Availability of data and materials

The source data and scripts to analyze the data and produce the results presented here are publicly available at <https://doi.org/10.5281/zenodo.3479124> (Nord et al., 2019b). The software presented in this work is available in public Git repositories (fpd devs, 2015; FPD Live Imaging devs, 2015; pixStem devs, 2015; Merlin Interface devs, 2016; Merlin DM Plugin devs, 2017; fpd demos devs, 2018), which also document the source of contributions to the code. Enquiries about the MERLIN_PixSTEM DM plugin should be directed to Damien.McGrouther@glasgow.ac.uk.

Acknowledgments. We thank Diamond Light Source, especially Dr. Chris Allen and Prof. Angus Kirkland, for access and support in the use of the electron Physical Science Imaging Centre E02 microscope (proposal no. EM16952) that provided the atomic resolution SrTiO₃ data presented here; Quantum Detectors Ltd. for Medipix3 detector support; Dr. Bruno Humbel from Okinawa Institute of Science and Technology and Dr. Caroline Kizilyaprak from the University of Lausanne for providing the liver sample; Dr. Kayla Fallon from the University of Glasgow for the sample data shown in Figure 5b, and Dr. Sinan Azzawi and Prof. Del Atkinson from University of Durham for providing the sample; Dr. Ingrid Hallsteinsen and Prof. Thomas Tybell from the Norwegian University of Science and Technology (NTNU) for providing the SrTiO₃ sample shown in Figure 3; and NanoMEGAS for the loan of the DigiSTAR precession system and TopSpin acquisition software.

Author contributions statement

G.W.P. and M.N. were the principal authors of the libraries reported herein (details of all contributions are documented in the repositories), and have made almost all of these available under open source licence GPLv3 for the benefit of the community. R.W.H.W., S.McV., I.M., K.A.P., and D.McG. have all contributed through acquisition and analysis of data and through participation in the revision of the manuscript. R.W.H.W. and K.A.P. have also made contributions to the source codes in these libraries. G.W.P. and M.N. have led the drafting of this manuscript.

Financial support. The performance of this work was mainly supported by the Engineering and Physical Sciences Research Council (EPSRC) of the UK via the project “Fast Pixel Detectors: a paradigm shift in STEM imaging” (grant no. EP/M009963/1). G.W.P. received additional support from the EPSRC under grant no. EP/M024423/1. M.N. received additional support for this work from the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement no. 838001. The studentship of R.W.H.W. was supported by the EPSRC Doctoral Training Partnership grant no. EP/N509668/1. S.McV. was supported by EPSRC grant no. EP/M024423/1. I.M. was supported by EPSRC grant no. EP/M009963/1. The studentship of K.A.P. was funded entirely by the UK Science and Technology Facilities Council (STFC) Industrial CASE studentship “Next2 TEM Detection” (no. ST/P002471/1) with Quantum Detectors Ltd. as the industrial partner. D.McG. was also supported by EPSRC grant no. EP/M009963/1. As an inventor of intellectual property related to the MERLIN detector hardware, he is a beneficiary of the license agreement between the University of Glasgow and Quantum Detectors Ltd. The development of the integration of TopSpin with the Merlin readout of the Medipix3 camera has been performed with the aid of financial assistance from the EPSRC under grant no. EP/R511705/1 and through direct collaboration between NanoMEGAS and Quantum Detectors Ltd.

References

- Abramov YA, Tsirelson VG, Zavodnik VE, Ivanov SA & Brown ID (1995). The chemical bond and atomic displacements in SrTiO₃ from X-ray diffraction analysis. *Acta Crystallogr. B* 51, 942–951. <http://dx.doi.org/10.1107/S0108768195003752>

- Ballabriga R, Aloyz J, Blaj G, Campbell M, Fiederle M, Frojdh E, Heijne EHM, Llopart X, Pichotka M, Procz S, Tlustos L & Wong W (2013). The Medipix3RX: A high resolution, zero dead-time pixel detector readout chip allowing spectroscopic imaging. *J Instrum* **8**, C02016. <http://dx.doi.org/10.1088/1748-0221/8/02/c02016>
- Banerjee A, Baker LJ, Doye A, Nord M, Heath RM, Erotokritou K, Bosworth D, Barber ZH, MacLaren I & Hadfield RH (2017). Characterisation of amorphous molybdenum silicide (MoSi) superconducting thin films and nanowires. *Supercond Sci Technol* **30**, 084010. <http://dx.doi.org/10.1088/1361-6668/aa76d8>
- Behnel S, Bradshaw R, Citro C, Dalcin L, Seljebotn DS & Smith K (2011). Cython: The best of both worlds. *Comput Sci Eng* **13**, 31–39. <http://dx.doi.org/10.1109/MCSE.2010.118>
- Campbell M, Aloyz J, Ballabriga R, Frojdh E, Heijne E, Llopart X, Poikela T, Tlustos L, Valerio P & Wong W (2016). Towards a new generation of pixel detector readout chips. *J Instrum* **11**, C01007. <http://dx.doi.org/10.1088/1748-0221/11/01/c01007>
- Chapman J, Batson P, Waddell E & Ferrier R (1978). The direct determination of magnetic domain wall profiles by differential phase contrast electron microscopy. *Ultramicroscopy* **3**, 203–214. [https://doi.org/10.1016/S0304-3991\(78\)80027-8](https://doi.org/10.1016/S0304-3991(78)80027-8)
- Chapman JN, McFadyen IR & McVitie S (1990). Modified differential phase contrast lorentz microscopy for improved imaging of magnetic structures. *IEEE Trans Magn* **26**, 1506–1511. <http://dx.doi.org/10.1109/20.104427>
- Chapman JN & Scheinfein MR (1999). Transmission electron microscopies of magnetic microstructures. *J Magn Magn Mater* **200**, 729–740. [https://doi.org/10.1016/S0304-8853\(99\)00317-0](https://doi.org/10.1016/S0304-8853(99)00317-0)
- Clausen A, Weber D, @probonopd, Caron J, Nord M, Müller-Caspary K, Ophus C, Dunin-Borkowski R, Ruzaeva K, Chandra R, Shin J & van Schyndel J (2019) Libertem/libertem: 0.2.2. <https://doi.org/10.5281/zenodo.3489385>
- Clough RN, Moldovan G & Kirkland AI (2014). Direct detectors for electron microscopy. *J Phys Conf Ser* **1** 522, 012046. <http://dx.doi.org/10.1088/1742-6596/522/1/012046>
- Crewe AV (1966). Scanning electron microscopes: Is high resolution possible? *Science* **154**, 729–738. <http://dx.doi.org/10.1126/science.154.3750.729>
- Crewe AV (1970a). High resolution scanning microscopy of biological specimens. *Ber Bunsenges Phys Chem* **74**, 1181–1187. <https://doi.org/10.1098/rstb.1971.0037>
- Crewe AV (1970b). The current state of high resolution scanning electron microscopy. *Q Rev Biophys* **3**, 137–175. <http://dx.doi.org/10.1017/S0033583500004431>
- Crewe AV, Wall J & Welter LM (1968). A high-resolution scanning transmission electron microscope. *J Appl Phys* **39**, 5861–5868. <http://dx.doi.org/10.1063/1.1656079>
- Dekkers N & de Lang H (1977). A detection method for producing phase and amplitude images simultaneously in a scanning transmission electron microscope. *Philips Tech Rev* **37**, 1–9.
- de la Peña F, Ostasevicius T, Fauske VT, Burdett P, Prestat E, Jokubauskas P, Nord M, Sarahan M, MacArthur KE, Johnstone DN, Taillon J, Caron J, Migunov V, Furnival T, Eljarrat A, Mazzucco S, Aarholt T, Walls M, Slater T, Winkler F, Martineau B, Donval G, McLeod R, Høglund ER, Alxneit I, Hjorth I, Henninen T, Zagonel LF, Garmannslund A. & 5ht2 (2018) hyperspy/hyperspy: HyperSpy 1.3.1. <https://doi.org/10.5281/zenodo.1221347>
- Delpierre P (2014). A history of hybrid pixel detectors, from high energy physics to medical imaging. *J Instrum* **9**, C05059. <http://dx.doi.org/10.1088/1748-0221/9/05/c05059>
- Donald AM & Craven AJ (1979). A study of grain boundary segregation in Cu-Bi alloys using STEM. *Philos Mag A* **39**, 1–11. <http://dx.doi.org/10.1080/01418617908239271>
- EMD authors (2019) Electron Microscopy Datasets: An HDF5-based interchange file format for electron microscopy data and metadata. Available at <https://emdatasets.com/format> (retrieved June 3, 2018).
- EMDViewer devs (2015) EMD Viewer. Available at <https://emdatasets.com/viewer> (retrieved June 3, 2018)
- Fang S, Wen Y, Allen CS, Ophus C, Han GGD, Kirkland AI, Kaxiras E & Warner JH (2019). Atomic electrostatic maps of 1D channels in 2D semiconductors using 4D scanning transmission electron microscopy. *Nat Commun* **10**, 1127. <http://dx.doi.org/10.1038/s41467-019-08904-9>
- Findlay SD, Shibata N, Sawada H, Okunishi E, Kondo Y & Ikuhara Y (2010). Dynamics of annular bright field imaging in scanning transmission electron microscopy. *Ultramicroscopy* **110**, 903–923. <http://dx.doi.org/10.1016/j.ultramic.2010.04.004>
- fpd demos devs (2018) Notebook examples for the fpd package. Available at <https://gitlab.com/fpdpy/fpd-demos> (retrieved June 3, 2018).
- fpd devs (2015) FPD: Fast pixelated detector data storage, analysis and visualisation. Available at <https://gitlab.com/fpdpy/fpd> (retrieved February 6, 2018).
- FPD Live Imaging devs (2015) Python library for live visualization of data acquired on a fast pixelated detector in a scanning transmission electron microscope. Available at https://gitlab.com/fast_pixelated_detectors/fpd_live_imaging (retrieved June 3, 2018).
- Frank J (1980) *The Role of Correlation Techniques in Computer Image Processing*, pp. 187–222. Berlin, Heidelberg: Springer. https://doi.org/10.1007/978-3-642-81381-8_5
- Gouillart E, Nunez-Iglesias J & van der Walt S (2016). Analyzing microtomography data with Python and the scikit-image library. *Adv Struct Chem Imaging* **2**, 18. <http://dx.doi.org/10.1186/s40679-016-0031-0>
- Hachtel JA, Idrobo JC & Chi M (2018). Sub-ångstrom electric field measurements on a universal detector in a scanning transmission electron microscope. *Adv Struct Chem Imaging* **4**, 10. <http://dx.doi.org/10.1186/s40679-018-0059-4>
- Hammel M & Rose H (1995). Optimum rotationally symmetric detector configurations for phase-contrast imaging in scanning transmission electron microscopy. *Ultramicroscopy* **58**, 403–415. [http://dx.doi.org/10.1016/0304-3991\(95\)00007-N](http://dx.doi.org/10.1016/0304-3991(95)00007-N)
- Hartel P, Rose H & Dinges C (1996). Conditions and reasons for incoherent imaging in STEM. *Ultramicroscopy* **63**, 93–114. [http://dx.doi.org/10.1016/0304-3991\(96\)00020-4](http://dx.doi.org/10.1016/0304-3991(96)00020-4)
- House SD, Tom Schamp C & Yang JC (2018). Real time acquisition and calibration of S/TEM probe current measurement simultaneously with any imaging or spectroscopic signal. *Microsc Microanal* **24**, 126–127. <http://dx.doi.org/10.1017/S1431927618001125>
- Hunter JD (2007). Matplotlib: A 2D graphics environment. *Comput Sci Eng* **9**, 90–95. <http://dx.doi.org/10.1109/MCSE.2007.55>
- Huth M, Ritz R, O’Leary CM, Griffiths I, Nellist P & Soltau H (2019). Ultrafast ptychography with 7500 frames per second. *Microsc Microanal* **25**, 40–41. <http://dx.doi.org/10.1017/S143192761900093X>
- Johnstone DN, Crout P, Högås S, Martineau B, Smeets S, Laulainen J, Collins S, Morzy J, Prestat E, Ånes H, philippcrout, Doherty T, Ostasevicius T & Bergh T (2019) pyxem/pyxem: pyxem 0.9.2. <https://doi.org/10.5281/zenodo.3407316>
- Jones E, Oliphant T, Peterson P, et al. (2001) SciPy: Open source scientific tools for Python. Available at <http://www.scipy.org> (retrieved October 30, 2018).
- Kluyver T, Ragan-Kelley B, Pérez F, Granger B, Bussonnier M, Frederic J, Kelley K, Hamrick J, Grout J, Corlay S, Ivanov P, Avila D, Abdalla S & Willing C (2016). Jupyter notebooks—A publishing format for reproducible computational workflows. In *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, Loizides F & Schmidt B (Eds.), pp. 87–90. Amsterdam, Netherlands: IOS Press. <http://dx.doi.org/10.3233/978-1-61499-649-1-87>
- Könnecke M, Akeroyd FA, Bernstein HJ, Brewster AS, Campbell SI, Clausen B, Cottrell S, Hoffmann JU, Jemian PR, Männicke D, Osborn R, Peterson PF, Richter T, Suzuki J, Watts B, Wintersberger E & Wuttke J (2015). The NeXus data format. *J Appl Crystallogr* **48**, 301–305. <http://dx.doi.org/10.1107/S1600576714027575>
- Krajnak M, McGrouther D, Maneuski D, O’Shea V & McVitie S (2016). Pixelated detectors and improved efficiency for magnetic imaging in STEM differential phase contrast. *Ultramicroscopy* **165**, 42–50. <https://doi.org/10.1016/j.ultramic.2016.03.006>
- LeBeau JM, D’Alfonso AJ, Findlay SD, Stemmer S & Allen LJ (2009). Quantitative comparisons of contrast in experimental and simulated bright-field scanning transmission electron microscopy images. *Phys Rev B* **80**, 174106. <http://dx.doi.org/10.1103/PhysRevB.80.174106>

- LeBeau JM, Findlay SD, Allen LJ & Stemmer S (2010). Position averaged convergent beam electron diffraction: Theory and applications. *Ultramicroscopy* **110**, 118–125. <http://dx.doi.org/10.1016/j.ultramic.2009.10.001>
- LiberTEM devs (2018) LiberTEM/nexus-4dstem. Available at <https://github.com/LiberTEM/nexus-4dstem>.
- MacLaren I, Frutos-Myro E, McGrouther D, McFadzean S, Weiss JK, Cosart D, Portillo J, Robins A, Nicolopoulos S, del Busto EN & Skogebj R (2020) Orientation mapping using scanned precession electron diffraction with a direct electron detector. *Microsc Microanal*
- MacLaren I, Wang L, Morris O, Craven AJ, Stamps RL, Schaffer B, Ramasse QM, Miao S, Kalantari K, Sterianou I & Reaney IM (2013). Local stabilisation of polar order at charged antiphase boundaries in antiferroelectric $(\text{Bi}_{0.85}\text{Nd}_{0.15})(\text{Ti}_{0.1}\text{Fe}_{0.9})\text{O}_3$. *APL Mater* **1**, 021102. <http://dx.doi.org/10.1063/1.4818002>
- Mahr C, Müller-Caspary K, Ritz R, Simson M, Grieb T, Schowalter M, Krause FF, Lackmann A, Soltau H, Wittstock A & Rosenauer A (2019). Influence of distortions of recorded diffraction patterns on strain analysis by nano-beam electron diffraction. *Ultramicroscopy* **196**, 74–82. <https://doi.org/10.1016/j.ultramic.2018.09.010>
- McGrouther D, Benitez Romero MJ, McFadzean S & McVitie S (2014) Development of aberration corrected differential phase contrast (DPC) STEM. *JEOL News* **49**(1), 2–10.
- McGrouther D, Krajnak M, MacLaren I, Maneuski D, O'Shea V & Nellist PD (2015). Use of a hybrid silicon pixel (Medipix) detector as a STEM detector. *Microsc Microanal* **21**, 1595. <http://dx.doi.org/10.1017/S1431927615008752>
- McMullan G, Cattermole DM, Chen S, Henderson R, Llopart X, Summerfield C, Tlustos L & Faruqi AR (2007). Electron imaging with Medipix2 hybrid pixel detector. *Ultramicroscopy* **107**, 401–413. <http://dx.doi.org/10.1016/j.ultramic.2006.10.005>
- McMullan G, Faruqi AR, Clare D & Henderson R (2014) Comparison of optimal performance at 300 keV of three direct electron detectors for use in low dose electron microscopy. *Ultramicroscopy* **147**, 156–163. <https://doi.org/10.1016/j.ultramic.2014.08.002>
- Merlin DM Plugin devs (2017) Digital micrograph scripts and plugin for communicating between DM and the Merlin Medipix readout and control software, potentially allowing full control of the Medipix detector. Available at https://gitlab.com/fast_pixelated_detectors/merlin_dm_plugin (retrieved October 14, 2019).
- Merlin Interface devs (2016) Python library for interfacing with a Medipix3 detector through the Merlin software through the TCP/IP command interface. Available at https://gitlab.com/fast_pixelated_detectors/merlin_interface (retrieved June 3, 2018).
- Midgley PA & Eggegan AS (2015). Precession electron diffraction—A topical review. *IUCr* **2**, 126–136. <http://dx.doi.org/10.1107/S2052252514022283>
- Mir JA, Clough R, MacInnes R, Gough C, Plackett R, Shipsey I, Sawada H, MacLaren I, Ballabruga R, Maneuski D, O'Shea V, McGrouther D & Kirkland AI (2017). Characterisation of the Medipix3 detector for 60 and 80 keV electrons. *Ultramicroscopy* **182**, 44–53. <https://doi.org/10.1016/j.ultramic.2017.06.010>
- Niermann T (2016) Digital Micrograph plugin to read/write HDF5 files. Available at https://github.com/niermann/gms_plugin_hdf5 (retrieved June 3, 2018).
- Nord M, Krajnak M, Bali R, Hlawacek G, Liersch V, Fassbender J, McVitie S, Paterson GW, MacLaren I & McGrouther D (2016). Developing rapid and advanced visualisation of magnetic structures using 2-D pixelated STEM detectors. *Microsc Microanal* **22**, 530–531. <http://dx.doi.org/10.1017/S1431927616003500>
- Nord M, Ross A, McGrouther D, Barthel J, Moreau M, Hallsteinsen I, Tybell T & MacLaren I (2019a). Three-dimensional subnanoscale imaging of unit cell doubling due to octahedral tilting and cation modulation in strained perovskite thin films. *Phys Rev Mater* **3**, 063605. <http://dx.doi.org/10.1103/PhysRevMaterials.3.063605>
- Nord M, Vullum PE, MacLaren I, Tybell T & Holmestad R (2017). Atomap: A new software tool for the automated analysis of atomic resolution images using two-dimensional Gaussian fitting. *Adv Struct Chem Imaging* **3**, 9. <http://dx.doi.org/10.1186/s40679-017-0042-5>
- Nord M, Webster RWH, Paton KA, McVitie S, McGrouther D, MacLaren I & Paterson GW (2019b) Dataset. <https://doi.org/10.5281/zenodo.3479124>
- Oliphant TE (2006) *A Guide to NumPy*. UT, USA: Trelgol Publishing.
- Oliphant TE (2007). Python for scientific computing. *Comput Sci Eng* **9**, 10–20. <http://dx.doi.org/10.1109/MCSE.2007.58>
- Ophus C (2017). A fast image simulation algorithm for scanning transmission electron microscopy. *Adv Struct Chem Imaging* **3**, 13. <http://dx.doi.org/10.1186/s40679-017-0046-1>
- Ophus C (2019). Four-dimensional scanning transmission electron microscopy (4D-STEM): From scanning nanodiffraction to ptychography and beyond. *Microsc Microanal* **25**, 563–582. <http://dx.doi.org/10.1017/S1431927619000497>
- Ophus C, Ercius P, Huijben M & Ciston J (2017). Non-spectroscopic composition measurements of $\text{SrTiO}_3\text{-La}_{0.7}\text{Sr}_{0.3}\text{MnO}_3$ multilayers using scanning convergent beam electron diffraction. *Appl Phys Lett* **110**, 063102. <http://dx.doi.org/10.1063/1.4975932>
- Paterson GW, Lamb RJ, Ballabruga R, Maneuski D, O'Shea V & McGrouther D (2020a). Sub-100 nanosecond temporally resolved imaging with the Medipix3 direct electron detector. *Ultramicroscopy* **210**, 112917. <http://dx.doi.org/10.1016/j.ultramic.2019.112917>
- Paterson GW, Webster RWH, Ross A, Paton KA, Macgregor TA, McGrouther D, MacLaren I & Nord M (2020b) Fast pixelated detectors in scanning transmission electron microscopy. Part II: Post acquisition data processing, visualisation, and structural characterisation. <https://arxiv.org/abs/2004.02777>.
- Pennycook S & Jesson D (1991). High-resolution Z-contrast imaging of crystals. *Ultramicroscopy* **37**, 14–38. [http://dx.doi.org/10.1016/0304-3991\(91\)90004-P](http://dx.doi.org/10.1016/0304-3991(91)90004-P)
- Pennycook TJ, Lupini AR, Yang H, Murfitt MF, Jones L & Nellist PD (2015). Efficient phase contrast imaging in STEM using a pixelated detector. Part I: Experimental demonstration at atomic resolution. *Ultramicroscopy* **151**, 160–167. <https://doi.org/10.1016/j.ultramic.2014.09.013>
- pixStem devs (2015) pixStem: Analysis of pixelated STEM data. Available at <https://gitlab.com/pixstem/pixstem> (retrieved October 3, 2018).
- Plackett R, Horswell I, Gimenez EN, Marchal J, Omar D & Tartoni N (2013). Merlin: A fast versatile readout system for Medipix3. *J Instrum* **8**, C01038. <http://dx.doi.org/10.1088/1748-0221/8/01/C01038>
- Poikela T, Plosila J, Westerlund T, Campbell M, Gaspari MD, Llopart X, Gromov V, Kluit R, van Beuzekom M, Zappone F, Zivkovic V, Brezina C, Desch K, Fu Y & Kruth A (2014). Timepix3: A 65K channel hybrid pixel readout chip with simultaneous ToA/ToT and sparse readout. *J Instrum* **9**, C05013. <http://dx.doi.org/10.1088/1748-0221/9/05/c05013>
- Rauch EF, Portillo J, Nicolopoulos S, Bultreys D, Rouvimov S & Moeck P (2010). Automated nanocrystal orientation and phase mapping in the transmission electron microscope on the basis of precession electron diffraction. *Z Kristallogr* **225**, 103–109. <https://doi.org/10.1524/zkri.2010.1205>
- Ryll H, Simson M, Hartmann R, Holl P, Huth M, Ihle S, Kondo Y, Kotula P, Liebel A, Müller-Caspary K, Rosenauer A, Sagawa R, Schmidt J, Soltau H & Strüder L (2016). A pnCCD-based, fast direct single electron imaging camera for TEM and STEM. *J Instrum* **11**, P04006. <http://dx.doi.org/10.1088/1748-0221/11/04/p04006>
- Savitzky BH, Baggari IE, Clement CB, Waite E, Goodge BH, Baek DJ, Sheckelton JP, Pasco C, Nair H, Schreiber NJ, Hoffman J, Admasu AS, Kim J, Cheong S-W, Bhattacharya A, Schlom DG, McQueen TM, Hovden R & Kourkoutis LF (2018). Image registration of low signal-to-noise cryo-STEM data. *Ultramicroscopy* **191**, 56–65. <https://doi.org/10.1016/j.ultramic.2018.04.008>
- Savitzky BH, Zeltmann S, Barnard E, Ierandc, Brown H. G., Henderson M & Ginsburg D (2019) py4dstem/py4DSTEM: DOI release. <https://doi.org/10.5281/zenodo.3333960>.
- Scholze F, Rabus H & Ulm G (1998). Mean energy required to produce an electron-hole pair in silicon for photons of energies between 50 and 1500 eV. *J Appl Phys* **84**, 2926–2939. <http://dx.doi.org/10.1063/1.368398>
- Shibata N, Kohno Y, Findlay SD, Sawada H, Kondo Y & Ikuhara Y (2010). New area detector for atomic-resolution scanning transmission electron microscopy. *Microscopy* **59**, 473–479. <http://dx.doi.org/10.1093/jmicro/dfq014>
- Shibata N, Seki T, Sánchez-Santolino G, Findlay SD, Kohno Y, Matsumoto T, Ishikawa R & Ikuhara Y (2017). Electric field imaging of single atoms. *Nat Commun* **8**, 15631. <http://dx.doi.org/10.1038/ncomms15631>

- Somnath S, Smith CR, Laanait N, Vasudevan RK, Ievlev A, Belianinov A, Lupini AR, Shankar M, Kalinin SV & Jesse S (2019) USID and Pycroscopy—Open frameworks for storing and analyzing spectroscopic and imaging data. [arXiv:1903.09515](https://arxiv.org/abs/1903.09515)
- Taguchi K & Iwaczyk JS (2013). Vision 20/20: Single photon counting x-ray detectors in medical imaging. *Med Phys* **40**, 100901. <http://dx.doi.org/10.1118/1.4820371>
- Tate MW, Purohit P, Chamberlain D, Nguyen KX, Hovden R, Chang CS, Deb P, Turgut E, Heron JT, Schlom DG, Ralph DC, Fuchs GD, Shanks KS, Philipp HT, Muller DA & Gruner SM (2016). High dynamic range pixel array detector for scanning transmission electron microscopy. *Microsc Microanal* **22**, 237–249. <http://dx.doi.org/10.1017/S1431927615015664>
- The HDF Group (1997–2018) Hierarchical Data Format, Version 5. Available at <http://www.hdfgroup.org/HDF5>.
- The HDF Group (2017) HDFView is a visual tool for browsing and editing HDF4 and HDF5 files. Available at <https://support.hdfgroup.org/products/java/hdfview> (retrieved June 3, 2018).
- Tinti G, Fröjdh E, van Genderen E, Gruene T, Schmitt B, de Winter DAM, Weckhuysen BM & Abrahams JP (2018). Electron crystallography with the EIGER detector. *IUCrJ* **5**, 190–199. <http://dx.doi.org/10.1107/S2052252518000945>
- Turala M (2005) Silicon tracking detectors—Historical overview. *Nucl Instrum Methods Phys Res A* **541**, 1–14. <http://dx.doi.org/10.1016/j.nima.2005.01.032>
- Turchetta R, Fant A, Gasiorok P, Esbrand C, Griffiths J, Metaxas M, Royle G, Speller R, Venanzi C, van der Stelt P, Verheij H, Li G, Theodoridis S, Georgiou H, Cavouras D, Hall G, Noy M, Jones J, Leaver J, Machin D, Greenwood S, Khaleeq M, Schulerud H, Østby J, Triantis F, Asimidis A, Bolanakis D, Manthos N, Longo R & Bergamaschi A (2007). CMOS monolithic active pixel sensors (MAPS): Developments and future outlook. *Nucl Instrum Methods Phys Res A* **582**, 866–870. <http://dx.doi.org/10.1016/j.nima.2007.07.112>
- van der Walt S, Schönberger JL, Nunez-Iglesias J, Boulogne F, Warner JD, Yager N, Gouillart E, Yu T & the scikit-image contributors (2014) scikit-image: image processing in Python. *PeerJ* **2**, e453. <http://dx.doi.org/10.7717/peerj.453>
- von Ardenne M (1938). The scanning electron microscope: Theoretical fundamentals. *Z Phys* **109**, 553–572. <http://dx.doi.org/10.1007/BF01341584>
- Wermes N (2005). Pixel detectors for tracking and their spin-off in imaging applications. *Nucl Instrum Methods Phys Res A* **541**, 150–165. <http://dx.doi.org/10.1016/j.nima.2005.01.052>
- Yang H, Rutte RN, Jones L, Simson M, Sagawa R, Ryll H, Huth M, Pennycook TJ, Green MLH, Soltau H, Kondo Y, Davis BG & Nellist PD (2016). Simultaneous atomic-resolution electron ptychography and Z-contrast imaging of light and heavy elements in complex nanostructures. *Nat Commun* **7**, 12532. <http://dx.doi.org/10.1038/ncomms12532>