Simulations of the physics and electronics in 2D semiconductor pixel detectors

Ashley Joy

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

of

University College London.

Physics and Astronomy

University College London

July 6, 2017

Abstract

In this thesis, simulations have been developed of entire 2D Semiconductor pixel detectors, including the physics and electronic responses. The most important concept of this is the separation of simulation into three distinct physical regimes, which should each be simulated in a different manner.

The first stage is a particle level simulation of X-Ray photons and their secondary particles interacting with the detector, and recording where interactions depositing energy in the sensitive layer occur. This uses Geant4, as the energy range in use lies just within Geant4's minimum energy bound.

The second stage is a charge cloud simulation of the interaction energy conversion to electron hole pairs, and then the movement of these electrons or holes in the sensitive layer towards the collection area or pixels. This uses an analytic solution to the problem of a charge cloud moving and interacting with itself within a semiconductor, this work is based upon the earlier HORUS simulation and more detailed lattice simulations by R. F. Fowler, et al.

The third stage is an electronics level simulation of how the collected charge is manipulated by the detector into a digital read out signal. This is done with a collection of functions representing the response of electronic components to a signal, that can be used in turn to represent the electronic response to an electrical signal.

Work was also conducted to develop a model of simulating the charge plasma effects expected to be seen in European XFEL detectors at high charge densities, however a lack of suitable data to test or tune the model renders this a theoretical exercise.

Simulations using the software have been compared to data taken with real detectors to test the accuracy of the simulation, with a focus on the charge cloud simulation.

The software has been written to be adaptable so that it can simulate all three main detectors at European XFEL, this adaptability then means that it can in principle be used for any other pixel detector.

Acknowledgements

I would like to thank my supervisor, Matthew Wing, for his support and guidance over the course of my studies, as well as Mark Lancaster for inspiring me to study for a PhD.

I would like to thank my colleagues at UCL who I have worked with, Erdem Motuk and Sam Cook, as well as all my colleagues at European XFEL, particularly Tonn Rüter. I would also like to thank members of the LPD group who I worked with, Marcus French, Matthew Hart and Paul Seller.

I would like to thank all my friends who suffered with me in C17: Andy, Andy, Becca, Ben, Sally and Steven, as well as those with better and more distant working environments: Alex, Becky and Xin.

I would also like to thank my friends around the world who believed in and supported me: Joey, Jay, Lee, Cyle, Josh, Ben and Elliot. I would like to single out particular thanks to Lisa and Johan, who were there for me when I needed it the most.

Finally, I would also like to thank my family, for their utterly invaluable support over the years.

Contents

List of Figures

List of Tables

Glossary

ADC Analogue to Digital Converter. 9, 41, 44–48, 55, 65, 118, 155

AGIPD Adaptive Gain Integrating Pixel Detector. 9, 22, 41, 52, 54, 55, 58–60, 63, 65–68, 79, 116, 154

APS Advanced Photon Source. 14, 146–148, 152, 153, 155, 180, 187

ASIC Aplication Specific Integrated Circuit. 9, 15, 16, 23, 53, 55, 56, 58, 59, 65, 66, 68, 120, 152, 154, 155, 161–163, 182

CCD Charged Couple Device. 38, 50, 52–55, 61, 87, 108, 118, 119

CDF Cumulative Distribution Function. 147

CdZnTe Cadmium zinc telluride. 23

CERN European Organization for Nuclear Research. 23

CLI Command Line Interface. 81

CTI Charge Transfer Inefficiency. 54, 87, 118, 119, 141

DAC Digital to Analogue Converter. 9, 44, 46, 47

DSSC DEPFET Sensor with Signal Compression. 10, 52, 54, 55, 58, 60–62, 114

European XFEL European X-ray Free Electron Laser. 8, 18, 22–27, 41, 48, 50, 52–56, 58, 60, 61, 80–82, 86, 91, 95, 97, 99, 100, 103, 107–109, 111, 114, 118, 119, 123–125, 129, 132, 137, 153–155, 181–187

FEL Free Electron Laser. 24

FEM Front End Module. 56, 58, 59, 65, 66, 87, 156

FXE Femtosecond X-ray Experiments instrument. 25

Geant4 for GEometry ANd Tracking, a toolkit for simulating particles passing through matter.. 11, 12, 23, 92, 93, 97, 100–103, 105, 181, 185

GPU Graphical Processing Unit. 155

GUI Graphical User Interface. 81

HED High Energy Density Science instrument. 25, 27

HORUS the HPAD Output Response fUnction Simulator. 22, 63–68, 72, 76, 79–81, 86, 95, 112, 114

HPAD the original name for the detector that would become AGIPD.. 63

IDL Interactive Data Language. 63, 64, 66, 72, 80

Karabo the European XFEL control and data management software framework .. 10, 11, 81–83, 88, 91, 97, 118, 119, 154, 155, 181

LPD Large Pixel Detector. 9, 10, 22, 41, 50–60, 63, 66–68, 76, 79, 80, 93, 115, 130, 146, 147, 152–156, 162, 166, 172, 179–181, 186, 187

MAPS Monolithic Active Pixel Sensor. 38, 53

medipix a family of photon counting hybrid pixel detector chips designed by an international consortium lead by CERN.. 23

MID Materials Imaging and Dynamics instrument. 25, 27

pnCCD a variant of Charge Coupled Device manufactured by PNDetector GmbH.. 9, 50, 52, 53, 55, 61, 108, 119, 138, 139, 181, 183, 186

PyDetLib a python based analysis framework written for European XFEL detectors.. 15, 154–156, 158, 160, 163

RAL Rutherford Appleton Laboratory. 55

SASE Self-amplified spontaneous emission. 24

SCS Spectroscopy & Coherent Scattering instrument. 25, 27

SPB/SFX Single Particles, Clusters, and Biomolecules & Serial Femtosecond Crystallography instrument. 25, 27

SPICE Simulation Program with Integrated Circuit Emphasis. 182

SQS Small Quantum Systems instrument. 25, 27

TCAD Technology Computer Aided Design. 23, 182

X-CSIT X-ray Camera SImulation Toolkit. 11, 12, 15–18, 22, 23, 63, 76, 81, 82, 84, 85, 87–97, 99, 100, 102–105, 107–109, 112, 114–116, 118–120, 129, 144, 146, 149, 150, 155, 166, 167, 169, 170, 172–186

XFEL X-ray Free Electron Laser. 24, 128

Chapter 1

Introduction

Semi-conductor sensor technology is used in a wide variety of applications, including particle trackers at collider experiments, heavy ion detectors, commercial digital cameras and X-ray cameras. They use a layer of doped, reverse-biased semiconductor to absorb the energy of particles and convert it to mobile charge that can be measured. The theory behind the physics effects used by or seen in semiconductor detectors is described in Chapter 2. This thesis concerns the simulation of semiconductor pixel detectors, where charge is collected with a pixel grid to form a 2D image. The 2D image explains why such semiconductor pixel detectors are often called X-ray cameras. A description of semiconductor pixel detectors, their design, and examples of those intended for use at the European XFEL can be found in Chapter 3.

This thesis presents the development of a generic simulation of these semiconductor pixel detectors. This software, the X-CSIT, has been designed for universal use across all pixel detectors to be used at the European XFEL. Chapter 4 describes HORUS_LPD, a predecessor to X-CSIT that simulated the LPD. It was adapted from HORUS, a simulation of the similar AGIPD. Chapter 5 describes the modular design of X-CSIT that enables it to be adapted to the many different detectors that will be used at European XFEL, and which enable individual components of it to be upgraded. Chapter 6 describes the physical simulations used by X-CSIT, including the simulation of particle interactions, diffusion of charge in the semiconductor layer and the detector electronics. Chapter 7 details the analysis of an

experiment conducted to test and validate the charge sharing algorithms of X-CSIT, and the comparison of the result to simulations conducted with X-CSIT. Chapter 8 summarises the current status of X-CSIT, the physical effects it simulates, the preparation already conducted to expand it later, and the status of testing and validation. Finally the thesis is concluded in Chapter 9.

Other work has been conducted in simulation of similar semiconductor detectors. M. Benoit [1] and M.E. Myronakis [2] separately used Geant4 [3, 4] to simulate Cadmium zinc telluride (CdZnTe) detectors for γ and X-ray imaging. F. Marinho [5], D. Wu [6] and D. Krapohl [7, 8, 9] also made use of Geant4 to simulate generic semiconductor detectors, most notably the medipix [10, 11, 12, 13] medical hybrid pixel detector from European Organization for Nuclear Research (CERN). The focus here on using Geant4 naturally places the focus of these simulations on the particle interactions. Other simulations have focused more on the charge transport aspect of simulation (see Chapter 5 for details of how X-CSIT splits these regimes). K. Mathieson [14], J. Kalliopuska [15], J. Schwandt [16], A. Ellakany [17] and M.M. Vignetti [18] all used various Technology Computer Aided Design (TCAD) software, most notably Silvaco TCAD [19] and Synopsys TCAD [20], to simulate charge transport. X-CSIT is principally focused around its modular design the split of physical regimes into different, but linked, simulations. Bespoke simulations were chosen for the charge and electronics simulations to maintain control over their features and capabilities, particularly with a view to including plasma simulations in the charge simulation and the on-ASIC storage used by European XFEL detectors. This simplified the process of achieving a fully functional version of X-CSIT and, therefore, proving the core concept of simulation separation. The separation of simulations would then allow the modification of sub-simulations or complete replacement with alternatives should that be desired. The choice of Geant4 for the particle simulation was made because it was felt that a bespoke particle simulation of sufficient accuracy, and which modelled all the desired interactions, would not be an effective use of time given that Geant4 provided all the functionality envisioned.

1.1 European XFEL

European XFEL is a 3.4 km long X-ray Free Electron Laser (XFEL), being built in the Hamburg area, Germany, shown in Fig. 1.1. Like other laser light sources it is a multipurpose experiment that produces X-ray laser flashes for experimental scientists. Free Electron Lasers (FELs) achieve this by accelerating electron bunches to high energies, up to 17.5 GeV for European XFEL, then passing them through magnetic undulators to emit X-ray light. A unique feature of FELs is the exponential growth in the amount of radiation emitted by the electrons caused by the microbunching of electrons. Microbunching is a longitudinal density modulation of the electron bunch into smaller bunches separated by a distance equal to the wavelength of the emitted radiation. This microbunching is caused by the interaction of the electron bunch with the electromagnetic field of photon radiation and leads to stimulated emission by the electrons in phase with the electromagnetic field, thus amplifying the initial radiation. FELs thus have more in common with a conventional laser, where here the medium is a bunch of free electrons, than traditional synchrotrons. However, the amplification of laser light still requires an initial source of light. In XFELs the Self-amplified spontaneous emission (SASE) principle is used. Here the initial radiation is provided by spontaneous emission of synchrotron light in the early stages of the undulator, and the XFEL does not require an external source. This does however mean that the emission of laser light in an XFEL is started by a random process, and the intensity of laser light is not consistent from pulse to pulse.

Compared to conventional synchrotron X-ray light, the laser light from a FEL has significantly higher brilliance. Brilliance has units of photons per mm^2, per second, per $mrad^2$, per 0.1% bandwidth. This combines all the important metrics of a laser light source: number of photons produced, concentration into a small spot size, delivery over a short period of time, low angular divergence and low energy divergence. Compared to other XFELs, which produce X-ray pulses at 30 Hz to 120 Hz [21, 22, 23, 24], European XFEL can provide up too 27000 laser light flashes per second, at a peak rate of 4.5 MHz.

Figure 1.1: Satellite view of the European XFEL, showing the start at DESY and the experimental hall location in Schleswig-Holstein. Renders showing European XFEL tunnels and building have been added [25].

European XFEL will have three beamlines at startup, SASE 1 and 2 will produce X-ray laser light at energies of 3 keV to 25 keV, while SASE 3 will re-use the electron bunches from SASE 1 to produce 0.26 keV to 3 keV soft X-ray light. Each of the three beamlines will have two experimental stations as shown in Fig. 1.2: the Materials Imaging and Dynamics instrument (MID) [26, 27] and the High Energy Density Science instrument (HED) [28, 29] on SASE 2, the Single Particles, Clusters, and Biomolecules & Serial Femtosecond Crystallography instrument (SPB/SFX) [30, 31] and the Femtosecond X-ray Experiments instrument (FXE) [32, 33] on SASE 1, the Small Quantum Systems instrument (SQS) [34, 35] and the Spectroscopy & Coherent Scattering instrument (SCS) [36] on SASE 3.

As mentioned earlier, European XFEL has a pulsed timing structure. The linear accelerator delivers 10 trains of electron bunches to the SASE 1/2 beam splitter every second at 10 Hz. Each train of bunches can contain up to 2700 electron bunches over 600 µs, which the beam switch distributes between SASE 1 and 2 [37, 38]. Electron bunches are separated by 222 ns within each train, therefore laser light

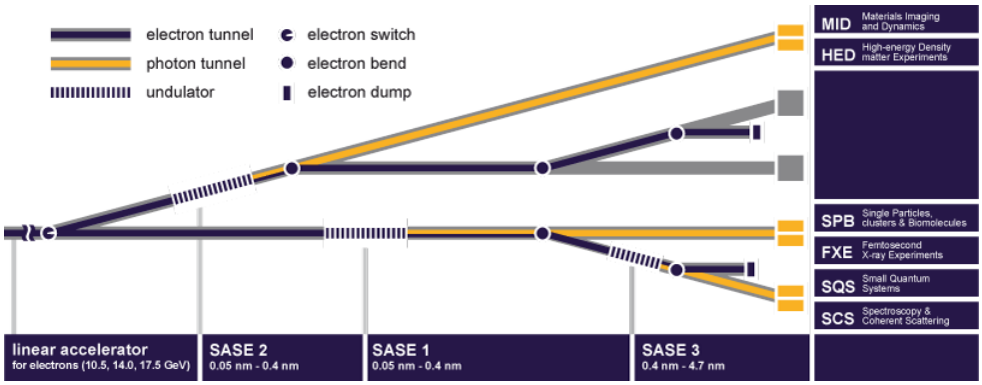


Figure 1.2: Schematic of the three current European XFEL beamlines and the six instrument stations [25].

flashes appear at a peak rate of 4.5 MHz. Up to 27,000[1] laser light flashes can be delivered by European XFEL per second. The high frequency of flashes poses a unique problem for detectors at European XFEL, where specialised bespoke detectors are needed to operate at the peak rate of 4.5 MHz by storing images in analogue memory, then conducting digitization and readout in the 99.4 ms gap between each bunch train.

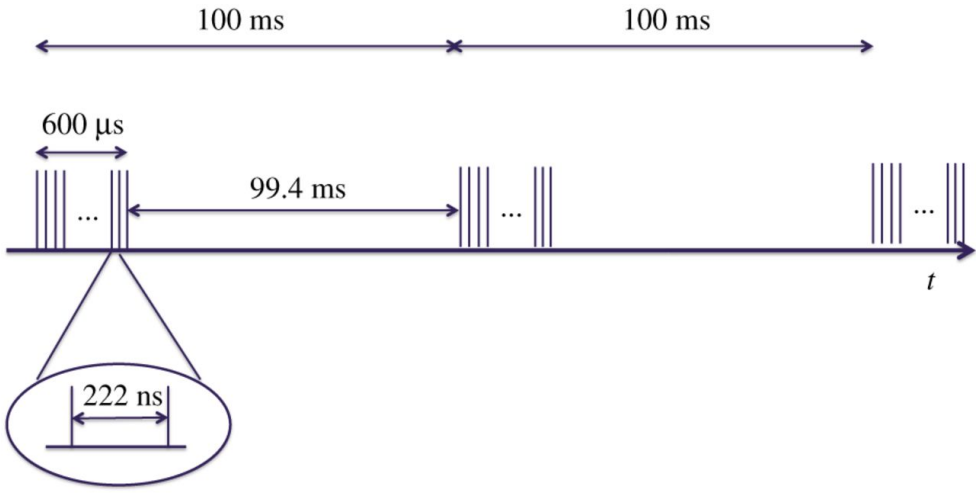


Figure 1.3: The European XFEL electron bunch timing structure [39].

As has been mentioned, European XFEL is a multipurpose detector, providing laser light for a variety of experiments. European XFEL categorises the most com-

[1] In practice some bunches need to be dumped while the beam is switched between each beamline [37].

mon experiments it expects to service into three fields. These are the investigation of tiny structures (at the SPB/SFX, SCS and MID instrument stations), filming ultrafast processes (at all six instruments) and investigation of extreme states of matter (the HED and SQS instruments).

Tiny structures refers to the ability of X-ray laser light to image very small structures, particularly biomolecules. Taking X-ray diffraction images of molecules usually requires them to be crystalised [39] so that many molecules can overlap their diffraction patterns, as the power required to image a single molecule is more than sufficient to destroy it. Free electron lasers offer the ability to image individual molecules because of their very short, $< 100\,\text{fs}$, pulse times. This allows them to perform an X-ray diffraction image of a molecule before the damage caused by the photon beam has time to affect the image [40, 41, 42, 43]. This is known as "diffraction-before-destruction".

The short pulse time is also of use when filming very fast processes, such as chemical reactions. Here the short pulse time allows scientists to take snapshots of processes as they are occurring with exceptional time resolution [44]. The unique timing system of European XFEL will also allow taking multiple consecutive images of the same sample over a very short period of time[32], up to $600\,\mu\text{s}$ after the initial pulse.

Finally European XFEL will be able to study extreme states of matter, high pressure, temperature or energy density, including plasmas [45, 46, 47]. Here European XFEL can be used to provide the large amounts of energy required to excite the sample, take the image of the resulting high energy density state, or even both [28].

Simulations of the detectors to be used at European XFEL will enable research groups working at European XFEL to test and optimise their experiments in simulations before arriving at European XFEL to conduct their experiment, as well as helping them understand their results later.

Chapter 2

Theory

2.1 Photon interactions with matter at X-ray energy

When travelling through silicon, X-ray photons can interact with matter via three different processes. Fig. 2.1 shows the cross sections of these three processes in the 1 keV to 30 keV range, which includes the energy ranges most frequently used by X-ray laser light experiments. Discussion of these effects, and other notable or relevant processes, are described below.

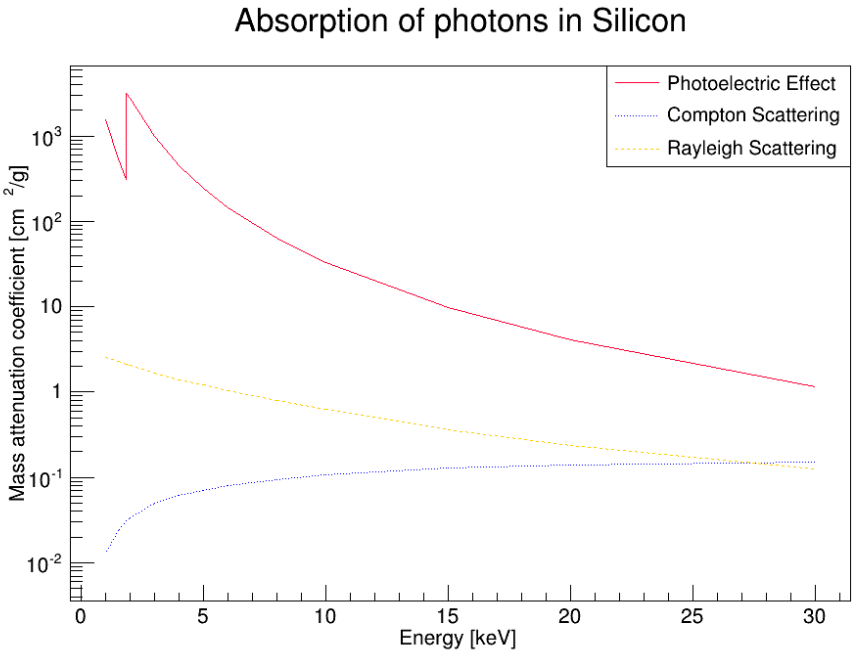


Figure 2.1: Photon cross sections for the photoelectric effect, Rayleigh scattering and Compton scattering, between 1 keV *to* 30 keV. *Data from [48].*

2.1.1 The photoelectric effect

The photoelectric effect occurs when a photon interacts with an atomic electron and is fully absorbed. The atomic electron is then ejected from the atom with energy

$$E_e = h\nu - B_e \tag{2.1}$$

where $h\nu$ is the energy of the initial photon and B_e is the binding energy of the atomic orbital the electron was in, and therefore the energy required to remove it from the atom. Classically this binding energy was known as the work function, this is the energy required to remove an electron from the surface of a metal plate by illuminating the plate with light. In this case it is the energy required to remove it only from the atom, not including any other losses from Coulomb interactions.

Photoelectric effect interactions are more likely to occur with the lower orbital, more tightly bound electrons in an atom [49]. This can be seen in Fig. 2.1, where a large drop in the photoelectric cross section can be seen at 1.84 keV. This is the binding energy of a K-shell[1] electron. Electrons absorbing photons with less energy than this will not have enough energy to escape the atom, and therefore the photoelectric effect does not occur.

After an electron has been ejected from an atom it leaves behind a vacancy. This vacancy will be filled either by the emission of a characteristic X-ray emission, or by an Auger emission. Both effects occur when an electron in a higher energy band drops down to fill the vacancy left by the photoelectric effect. Because the new band requires less energy than the original, the difference in energy is emitted as a photon, often still within the X-ray band. The name characteristic emission comes from the fact that the energy released in the photon is determined solely by the difference in energy bands of the atom, and therefore photons are released at frequencies specific to each atom.

The Auger effect occurs when a characteristic X-ray does not escape the atom, and is instead absorbed by another electron in a higher orbit. This electron is then, much like the original electron that underwent the photoelectric effect, ejected from

[1] K-shell is the atomic physics term for atomic electron orbitals with quantum number $n = 1$. It is therefore the first and most tightly bound electron shell.

the atom with an energy described by Equation 2.1, where the binding energy B_e is a lower value than with the original photoelectric emission, and $h\nu$ is equal to the previous binding energy. Partly because the Auger electron must come from a less tightly bound orbital, the Auger effect is much less likely than a characteristic emission.

2.1.2 Rayleigh scattering

Rayleigh scattering is the second most dominant physical process photons under go when interacting with silicon in the 1 keV to 30 keV range, as can be seen in Fig. 2.1. Rayleigh scattering is also known as coherent scattering because of the effect it has on the photon, which is to scatter it without affecting its energy or frequency. Rayleigh scattering is an elastic collision between a photon and an atomic electron, however insufficient energy is transferred to the electron for ionisation or excitation of the atom to occur. This leads to low permissible scattering angles [50].

It is trivial to show that scattering a photon without changing its energy does not conserve momentum, however when the scattering particle, in this case a silicon atom, has a mass much greater than the energy of the photon the recoil energy is negligible. Therefore Rayleigh scattering, while the second most dominant process behind the photoelectric effect, does not contribute to photon energy absorption.

2.1.3 Compton scattering

Compton scattering, as can be seen in Fig. 2.2, is the dominant form of photon energy deposition in silicon between approximately 100 keV and 10 MeV. This is above the 10 keV to 25 keV range in use by X-ray laser experiments. Between 10 keV to 25 keV Compton scattering only comprises 0.3% to 4.9% of photon interactions. Over this range the Compton scattering cross section only increases by 50%, the rest of the fractional increase is made up from the drop in photoelectric cross section.

Compton scattering is an inelastic collision between a photon and a charged particle, in the case of semiconductor detectors this is a ‘free’ or loosely bound electron. The Klein–Nishina formula for the Compton scattering differential cross

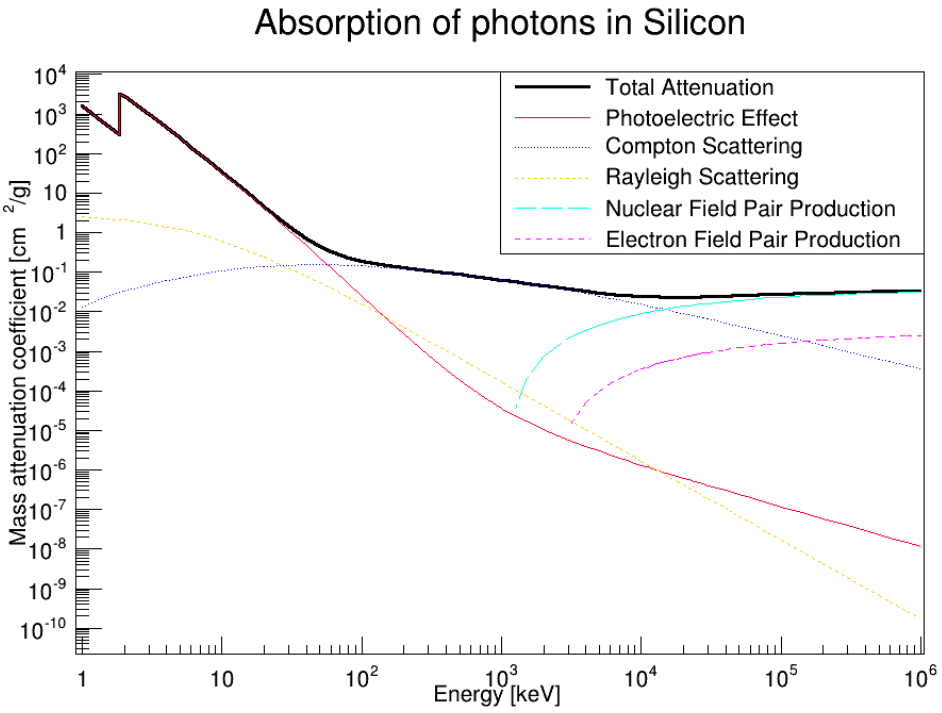


Figure 2.2: Photon cross sections for the photoelectric effect, Rayleigh scattering, Compton scattering and pair production, between 1 keV *and* 1 GeV. *Data from [48].*

section of unpolarised light, Equation 2.2 [51], is for photon collisions with a free electron. However, loosely bound electrons with a binding energy that is small compared to the photon energy are sufficiently similar to free electrons for Compton scattering to occur and the Klein–Nishina formula

$$\frac{d\sigma}{d\Omega} = \alpha^2 r_c^2 \left(\frac{\nu'}{\nu_0}\right)^2 \left[\frac{\nu'}{\nu_0} + \frac{\nu_0}{\nu'} - 1 + \cos^2\theta\right]/2 \tag{2.2}$$

$$\frac{\nu'}{\nu_0} = \frac{1}{1 + \left(\frac{h\nu_0}{m_e c^2}\right)(1 - \cos\theta)} \tag{2.3}$$

to be valid. Here $d\Omega$ is an infinitesimal solid angle element, α is the fine structure constant, r_c is the reduced Compton wavelength of the electron, θ the scattering angle, $\frac{\nu'}{\nu_0}$ the ratio of photon energy after and before the collision and m_e the mass of the electron. Note that $\alpha r_c = r_e$ is the classical electron radius and the Klein–Nishina formula is sometimes given in this form.

As a result of Compton scattering the electron is ejected from the atom and the photon is scattered by θ with a new energy $h\nu'$. This drop in photon energy is why Compton scattering is referred to as incoherent scattering, as opposed to the coherent Rayleigh scattering that does not alter the photon energy.

2.1.4 Pair production

Pair production occurs when, within the magnetic field of an atom, a photon splits into an electron and a positron. This process is the dominant form of photon energy absorption above approximately 10 MeV, see Fig. 2.2, however it requires a photon energy of at least twice the rest mass of the electron, which is 0.511 MeV. Pair production is therefore impossible in the X-ray region.

2.2 Charge spreading in semiconductors

Silicon is a semiconductor and therefore has a band gap between its valence band (the highest energy atomic orbitals that are filled under normal, non-ionised, conditions) and its conduction band (higher energy atomic orbitals that allow conduction when filled). Therefore energy can be absorbed by silicon and other semiconductors such that it is used to raise electrons into the conduction band and create an electron–hole pair. The electron and hole are then mobile within the semiconductor and can move under the effect of an applied electric field. This movement of charge can then be measured by an electric circuit, allowing detection or measurement of the initial energy deposit.

Both the photoelectric effect and Compton scattering produce electrons as free particles in the detector material. This can produce electrons with a sufficiently high energy, and therefore lifespan, that they should be tracked as free particles by a simulation and not be considered energy that has been deposited. These free electrons are charged, unlike the original incident photons, and therefore undergo Coulomb interactions with atomic electrons. These Coulomb interactions are a scattering of the free electron off an atomic electron, with a transfer of energy to the atomic electron which is then ejected from the atom. This creates two electrons, with less energy than originally, which continue to interact in the same way with further

atoms.

Each subsequent interaction results in particles with less and less energy until interactions no longer produce free particles but instead excite electrons to the conduction band of the semiconductor, creating electron–hole pairs. It would seem obvious that the amount of energy required to create an electron–hole pair, the mean ionisation energy, would be equal to the band gap of the semiconductor. However the mean ionisation energy exceeds the band gap, because conservation of momentum requires that lattice vibrations, or phonons, must also be created. As a result not all energy is used for electron–hole creation and the mean ionisation energy exceeds the band gap [52]. It is observed that the relationship between the mean ionisation energy, ε_i, is related to the band gap energy, E_g, by

$$\varepsilon_i = C_1 + C_2 E_g \tag{2.4}$$

regardless of the material or the ionising particle [53]. Here C_1 and C_2 are constants found by plotting ε_i and E_g as seen in Fig. 2.3. In silicon the mean ionisation energy is around 3.65 eV, larger than the band gap of 1.12 eV.

A consequence of the mean ionisation energy exceeding the band gap is the variation in the number of electron–hole pairs produced. Because energy can be lost to phonons instead of electron–hole pairs, the mean ionisation energy is the average of the energy that is used to create electron–hole pairs, not the exact energy required to free an electron–hole pair. If all energy were used to create electron–hole pairs then conservation of energy would demand no variation in the number of electron–hole pairs produced.

At the other extreme, where each electron–hole pair is created independently of each other, Poisson statistics describe the variance, σ^2, on the mean with

$$\sigma^2 = \overline{N} \tag{2.5}$$

where $\overline{N}$ is the mean number of electron–hole pairs generated. Since each electron–hole pair produced as a result of a single photon share an origin they are obviously not created independently. The variation therefore falls between the two extremes

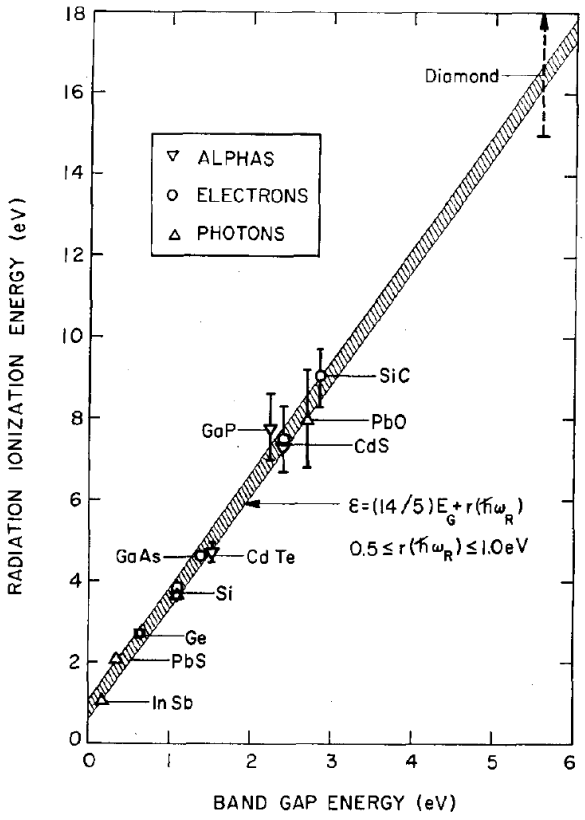


Figure 2.3: The relationship of mean ionisation energy to band gap, for different materials and radiation. From [53].

and is described by the Fano factor,

$$\sigma^2 = F\overline{N}, \tag{2.6}$$

which is defined as

$$F = \sigma^2/\mu \tag{2.7}$$

where μ is the mean of a random process [54, 55, 56, 57]. The Fano factor is a number between 0 and 1, where 1 describes Poisson like statistics and 0 has no variation at all. Experimentally the Fano factor of silicon is found to be about 0.118 [58], leading to a significantly better accuracy when measuring the energy deposit of a single photon than would be the case with Poisson statistics.

Once electron–hole pairs have been created the next step is to collect the charge

carriers from the semiconductor and measure them with a circuit. Conductivity in semiconductors is achieved by the process of doping [59]. This is where impurities are intentionally added to the semiconductor crystal. If a dopant is introduced to the semiconductor with an atomic number one higher than that of the semiconductor material it will have an outer shell electron that is only lightly bound. If this electron is thermally excited it will produce a negative charge that is mobile in the semiconductors and a non-mobile positively charged ion in the crystal. Likewise a dopant with an atomic number one lower than the semiconductor will have an atomic bond free. Like with the other kind of dopant only a small amount of energy is required to ionise the dopant. In this case an electron from a neighbouring atom is borrowed to produce a negatively charged ion, but also a positively charged mobile hole. This is a vacancy where an electron is not present in the otherwise neutrally uncharged semiconductor and behaves like a positive mobile charge. These dopants are referred to as n–type and p–type for the negative and positive mobile charges they introduce into the semiconductor.

A p–n junction is formed by creating a semiconductor with a region of n-type semiconductor abutting a region of p-type semiconductor, this is usually achieved with a bulk n-type doping, then injecting p-type dopants into one surface in higher concentrations. The point where they meet forms the p–n junction [60, 61, 62]. Here thermal diffusion will cause positive charges from the p–type and negative charges from the n–type to combine at the boundary. This creates a depletion zone where no free charges are present. The size of the depletion zone is limited by the electric field that develops across it. The removal of positive charge from the p–type gives a negative charge to the p–type region, and likewise a positive charge to the n–type. The depletion zone therefore has a built in potential difference V_{bi} across its width and becomes a capacitor with an electric field that limits its own size.

To increase the size of the depletion zone a reverse bias is applied to the semiconductor. This is an external voltage applied in the same direction as the built in voltage. This strips more positive charge from the p–type and negative charge from the n–type, resulting in an expansion of the depletion zone, see Fig. 2.4. The width

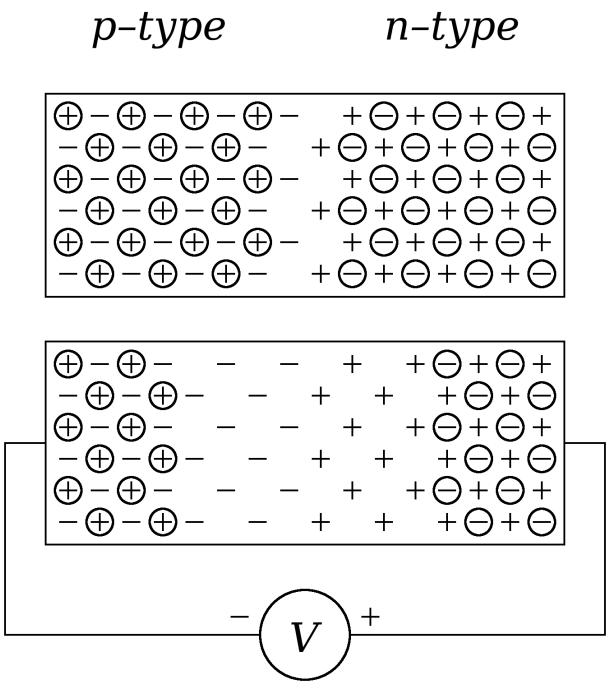


Figure 2.4: Regions of adjacent p–type and n–type semiconductor, with mobile positive and neutral charges (circled) and immobile negative and positive charges. Where the two regions adjoin a depletion zone forms without mobile charges. When an external reverse bias is applied free charges are drawn from the semiconductor and the depletion zone increases in size. Based on a figure in [63].

of the depletion zone is then usually given as

$$w_d = \sqrt{\frac{2\varepsilon(V_b + V_{bi})}{e}\left(\frac{1}{N_n} + \frac{1}{N_p}\right)} \tag{2.8}$$

where ε is the dielectric constant, V_b the bias voltage, V_{bi} the built in voltage from thermal diffusion, e the charge on the electron, N_n the dopant concentration of the bulk n–type annd N_p the dopant concentration of the p–type. Because semiconductor junctions are usually created with a highly doped implant[2] in a lightly doped bulk, the doping concentration term for the implant, which is usually N_p, can be ignored. The bulk doping concentration is then simply known as N. Likewise the built in voltage V_{bi} is usually of the order of 0.5 V [64] with applied voltages exceeding 50 V, the built in voltage is therefore usually ignored or combined with the

[2] $N_p > 10^{18}\,\mathrm{cm}^{-3}$, $N_n \approx 10^{12}\,\mathrm{cm}^{-3}$[64]

applied voltage into the total potential difference V.

At reverse bias voltages such that the depletion width is less than the size of the semiconductor the electric field has a triangular profile given by

$$|E(x)| = \frac{eN_n}{\varepsilon}(w_d - x) = \sqrt{\frac{2N_n eV}{\varepsilon}}\left(1 - \frac{x}{w_d}\right). \tag{2.9}$$

As the reverse bias is increased the depletion width as well as the maximum electric field with also increase as seen in the left half of Fig. 2.5. The depletion voltage is the reverse bias necessary to extend the depletion zone to the entire volume of the detector and is given as

$$V_d = \frac{Ned^2}{2\varepsilon}. \tag{2.10}$$

Where d is the depth of the whole sensor, and in this case also equal to the width of the depletion zone w_d. Once the depletion width encompasses the whole of the sensor it is said to be fully depleted. Any further voltage added to the reverse bias adds to the electric field linearly such that

$$|E(x)| = \frac{2V_d}{d}\left(1 - \frac{x}{d}\right) + \frac{V - V_d}{d}. \tag{2.11}$$

This is demonstrated in the right half of Fig. 2.5. Once a sensor is fully depleted and additional reverse bias is applied it is said to be overbiased.

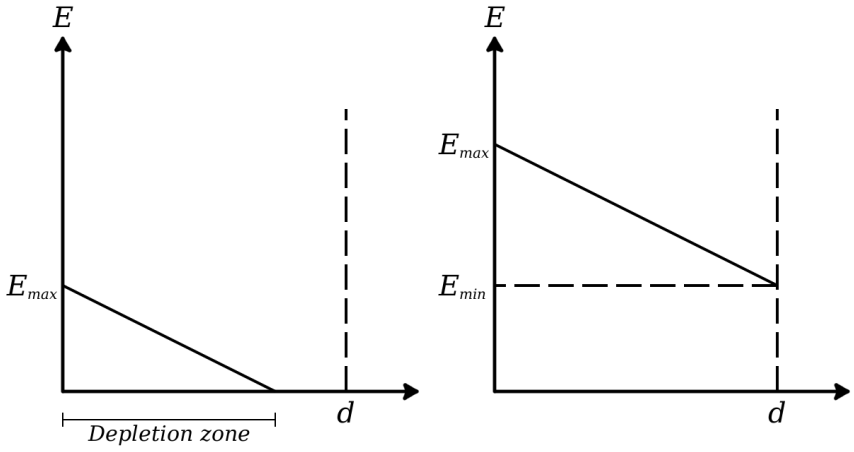


Figure 2.5: Diagram of the electric field in a partially depleted sensor (left) and a fully depleted and overbiased sensor (right). Based on plots in [63].

Any electron–hole pairs created in the depletion zone by particles depositing energy will be affected by the electric field present across the depletion zone. Electrons will thus be drawn one way across the width of the sensor and holes the other. In this way the charge created by energy deposition can be drawn from the semiconductor so that it can be measured.

Pixel detectors collect either electrons or holes in a pixel array to form a position sensitive image of energy deposition. Charged Couple Devices (CCDs) capture charges with an array of potential wells, with charge being moved to a common collection strip along one edge of the detector. Hybrid pixel detectors use bump bonds to the semiconductor so that each pixel in the array collects charge separately. Monolithic Active Pixel Sensor (MAPS) use a radically different method of construction in which the same block of semiconductor is used both for the pn–junction and the integrated circuit. This requires substantially higher doping than in regular detectors and subsequently lower resistivity that limits depletion widths to only a few μm [65]. For the short interaction lengths of visible light this is still usable (it is the digital camera market where MAPS are most frequently seen), however for high energy physics this means most of the charge will be deposited outside the depletion zone and diffusion must be relied upon to collect the charge. This and recombination of electron–holes during diffusion drastically lowers the amount of charge recovered and subsequently affects energy resolution. It also means that the following discussion of physics in contiguous, fully depleted sensors is not relevant to MAPS.

While ideally each photon would deposit all of its charge in a single pixel, it is observed that charge created as a result of a photon interaction can deposit charge over multiple pixels. This effect is known as charge sharing, as it appears to involve a pixel with an event sharing the charge with other neighbouring pixels. This effect is a result of the electron or hole movement in the semiconductor. While an applied electric field will move electrons or holes towards a collection area, the electron or hole is still free to undergo a random walk in the plane normal to the direction of the electric field. Therefore a cloud of electrons or holes originating from a single

source is seen to spread out radially to the electric field. Other factors can also affect the spread of a charge cloud, such as inter-charge interaction, impurities in the semiconductor and external magnetic fields. If this occurs towards the 'edge' of a pixel[3] then some of the charge may cross the pixel boundary and subsequently be collected in a different pixel to that the original interaction was in.

R.F. Fowler [66] did theoretical and computational work to determine the size of the spread of charge diffusing through a semiconductor. He theorised an equation for the spread

$$\sigma^2(d) = \frac{2k_BT}{q\beta}\log\left(1+\frac{\beta}{\alpha}d\right) \tag{2.12}$$

where d is the depth of interaction[4], k_B the Boltzmann constant, T the temperature, q the electron charge and the electric field of the semiconductor, $E(x)$, is given as

$$E(x) = \alpha + \beta x. \tag{2.13}$$

Here α and β are positive constants and x is the depth in the semiconductor. Given a fully depleted, reverse biased sensor it is common for $\beta \ll \alpha$, and therefore Equation 2.12 simplifies to

$$\sigma_t^2(d) = \frac{2k_BT}{q\alpha}d. \tag{2.14}$$

Fowler concluded that his equation gave the correct shape of the spread, but for depths lower than 100 µm the value for the size of the spread was not a good fit.

2.3 Plasma effects

The plasma effect was first noticed in 1960 by Miller et al. [67] as an increase in the collection time of the charge pulse when using semiconductor detectors to detect

[3] In reality there is no edge within the detector, rather it is a mid way point between the edges of two collection points.

[4] Depth here is the distance from the origin of the charge to the collection side. The paper was written from the point of view of a front side illuminated detector, where the charge is collected on the side of the semiconductor facing the illumination.

^{252}Cf fission fragments. The collection time of charge from the semiconductor is normally independent of charge deposition. When each electron and hole travels independently, the only influences on travel time are the detector material itself, the electric field and the start point in the detector. Plasma effects induce a charge-deposit-dependent increase in the charge collection time [68, 69]. Additionally the charge spread, which like the charge collection time and rise time is usually independent of charge deposit, also increases in a charge-deposit-dependent manner when plasma effects occur.

These plasma effects occur when the electron–hole charge density in a region of the semiconductor becomes high enough to form a charge plasma. The region of charge then begins to screen itself against the detector's external electric field. When this occurs, charge then only moves as a result of random diffusion and inter-particle Coulomb forces. The charge plasma will not resume normal charge collection until it has diffused sufficiently to no longer screen itself against the electric field. This diffusion increases the spread of charge, and the time it takes delays charge collection. The increase in charge collection time also leads to recombination of electrons and holes, resulting in a lower charge signal collected than what was present initially.

Because semiconductor plasma effects occur when the plasma begins to screen itself against external charges, the distinction between when plasma effects occur and when they do not occur should be related to the Debye length. The Debye length λ_D, is a parameter defining the scale at which individual particle behaviour gives way to a collective plasma like behaviour [70] and therefore when charge screening will take place. The Debye length is given as

$$\lambda_D = \sqrt{\frac{\varepsilon k_B T}{nq^2}} \tag{2.15}$$

where n is the number density of mobile charges, T their temperature, ε the permittivity, k_B the Boltzmann constant and q the charge of the electron. In many plasmas only the electrons are counted and the effect of ions is ignored due to their lower mobility, however in the case of an electron–hole plasma, holes are as mobile as

electrons and therefore should also be counted in n.

J. Becker [71] demonstrated that such plasma effects could occur at X-ray energies and intensities expected to be seen at the European XFEL. He used laser light of 660 nm and 1015 nm wavelength to simulate X-rays of 1 keV and 12 keV respectively. 660 nm laser light and 1 keV X-rays have the same attenuation length in silicon of 3 µm, likewise both 1015 nm laser light and 12 keV X-rays have an attenuation length of 250 µm. Charge carrier densities of up to $10^{16}\,\mathrm{cm}^{-3}$ were created, corresponding to 10^5 photons of 12 keV in a small tube, comparable to the maximum per pixel recordable signals of the Adaptive Gain Integrating Pixel Detector (AGIPD) and the Large Pixel Detector (LPD). He also showed that the spread of the charge would be comparable to the pixel sizes of the European XFEL detectors, and therefore would have a significant impact on charge sharing were they to occur at European XFEL.

2.4 Electronics

The electronic circuitry used to measure the charge signal collected from the sensor material and read it out are largely bespoke to each detector. However two specific components will be mentioned and discussed here for how almost universal they are to semiconductor detectors. These are the charge sensitive amplifier and the Analogue to Digital Converter (ADC). The charge sensitive amplifier is used in semiconductor detectors to convert a charge signal collected from the sensor layer into a proportional voltage output that can then be used by subsequent components. The ADC is used to convert an analog voltage signal into a machine readable digital one.

2.4.1 Charge sensitive amplifier

Most semiconductor detectors use a charge sensitive amplifier as the first electronic component of the pixel circuit on the ASIC [72, 73]. This is because the charge sensitive amplifier outputs a voltage signal that is proportional in size to an amount of charge collected onto a capacitor. It is therefore a useful method of converting the signal charge collected from the sensor into a voltage signal that can be used

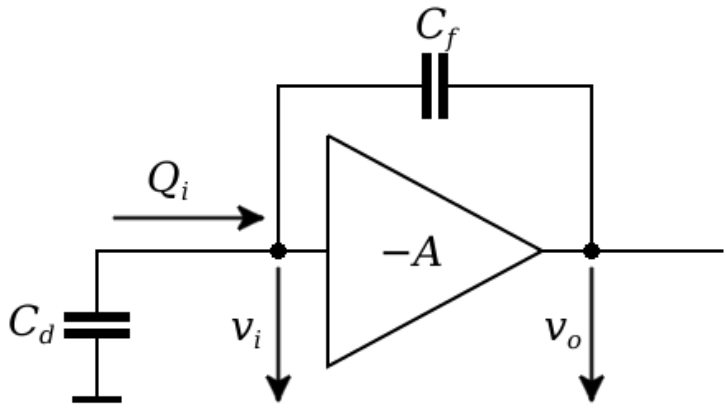


Figure 2.6: Schematic of a charge sensitive amplifier.

by subsequent components. Additionally to this the voltage output does not need to draw current from the initial charge signal[5] that is collected and it is thus preserved. Because the charge sensitive amplifier is often the first component of a detectors electronics it is frequently called a preamplifier, particularly if other amplifiers are used later in the electronics.

A schematic of the setup for a charge sensitive amplifier is shown in Fig. 2.6. Here the sensor material is shown as a capacitor C_d from which charge Q_i is fed into the input of the amplifier. We assume for simplicity that the amplifier has infinite resistance so that we can ignore any discharge of signal charge through the amplifier itself. The inverting amplifier has a voltage gain of $-A$, an input voltage of V_i and an output voltage of V_o. Therefore

$$V_o = -AV_i. \tag{2.16}$$

A feedback capacitor C_f is placed over the amplifier. The voltage across the capacitor is therefore

$$V_f = V_i - V_o = V_i + AV_i = (A+1)V_i. \tag{2.17}$$

We also know that the charge on the capacitor C_f is

$$Q_f = C_f V_f = C_f(A+1)V_i. \tag{2.18}$$

[5]Between images the charge signal needs to be cleared from the capacitor and some designs do so by allowing it to discharge over the amplifier.

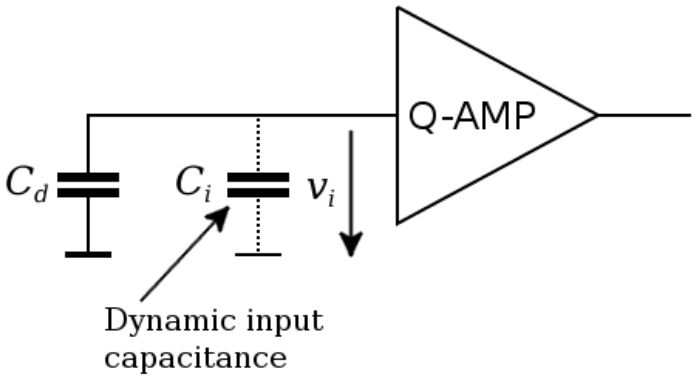


Figure 2.7: The "dynamic" input capacitance that appears to be driving the charge sensitive amplifier in parallel to the sensor.

Since we are assuming an infinite resistance on the amplifier the entire input charge Q_i will be deposited in the feedback capacitor C_f. Therefore $Q_f = Q_i$. The charge input to the amplifier therefore appears as a "dynamic" input capacitance C_i across the input voltage V_i as shown in Fig. 2.7. This dynamic capacitance is therefore

$$C_i = \frac{Q_i}{V_i} = C_f(A+1). \tag{2.19}$$

We can therefore calculate the ratio between the output voltage V_o and the input charge Q_i as

$$A_Q = \frac{V_o}{Q_i} = \frac{AV_i}{C_iV_i} = \frac{A}{C_i} = \frac{A}{C_f(A+1)} \tag{2.20}$$

by substituting equations 2.16 and 2.19. It can therefore be seen that if A is very much larger than 1 that equation 2.20 becomes

$$A_Q = \frac{V_o}{Q_i} \approx \frac{1}{C_f}. \tag{2.21}$$

The charge sensitive amplifier therefore converts an input charge to an output voltage with a ratio set exclusively by the capacitance of the feedback capacitor.

However it can be seen from Fig. 2.7 that the input charge Q_i is not equal to the signal charge Q_s that is initially deposited on the sensor C_d, since Q_s will be distributed between the dynamic input capacitor C_i and the detector C_d such that

$$Q_s = Q_i + Q_d \tag{2.22}$$

where Q_d is the charge that remains on the sensor. The ratio between the measured input charge Q_i and the signal charge Q_s is therefore

$$\frac{Q_i}{Q_s} = \frac{Q_i}{Q_d + Q_i} = \frac{C_i}{C_d + C_i} = \frac{1}{1 + \frac{C_d}{C_i}}. \tag{2.23}$$

Therefore a large dynamic input capacitance compared to the detector capacitance is desired to maximise the efficiency of charge collection.

2.4.2 Analogue to Digital Converters

Analogue to Digital Converters (ADCs), also known as digitizers, are a near universal component of a semiconductor detector that convert the analogue voltage signal delivered by the pixel electronics to a machine readable digital signal. This conversion requires the quantisation of the continuous analogue signal into a discrete digital one, and therefore some information will be lost. ADCs are usually measured by the size of their output signal, so an n-bit ADC will have 2^n output channels.

Conceptually the simplest possible ADC is the flash or parallel ADC [74, 75]. This consists of a voltage ladder with the reference of a comparator connected between each resistor, as seen in Fig. 2.8. The voltage ladder is connected to ground at one end and a maximum voltage signal at the other. This provides a continuous, linear scale of reference voltages. The signal voltage is then fed into the input of the comparators, that then output to an encoder. All the comparators with a reference voltage lower than the signal will activate, and the encoder creates a digital output signal corresponding to the highest reference voltage comparator that activates. This kind of amplifier is fast, simple and accurate. So long as all the resistors are identical, linearity of the ADC is maintained. However, the requirement to drive a large number of comparators, one for each possible output channel, leads to a large power requirement. As an example, an 8-bit flash ADC can draw 5 W [76], often too much to fit onto a small integrated circuit.

The most common type of ADC is the successive approximation ADC. Here an input signal is compared to a reference voltage from a Digital to Analogue Converter (DAC), shown in Fig. 2.9. The DAC successively runs through the potential output

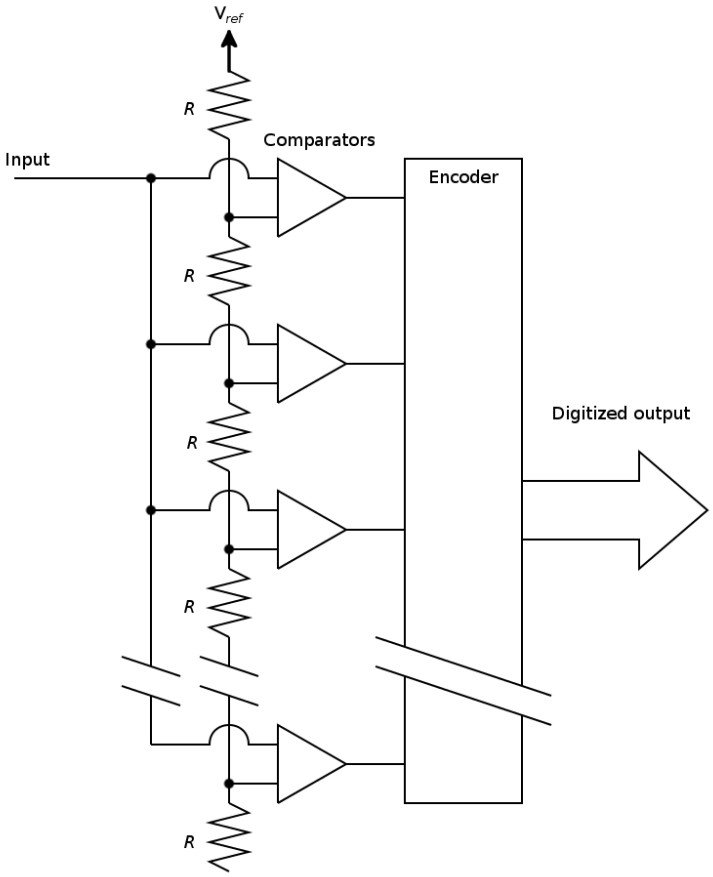


Figure 2.8: Block diagram of a flash ADC, showing how the voltage ladder connects to the parallel comparators.

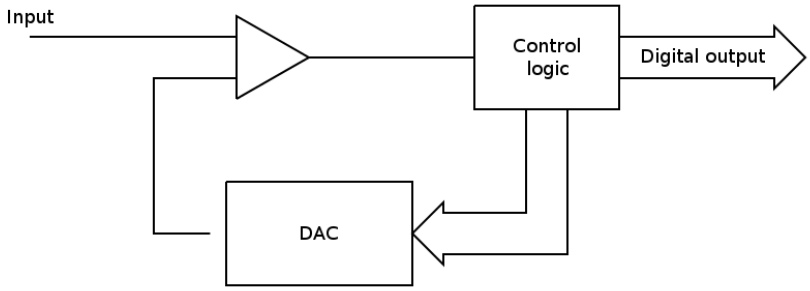


Figure 2.9: Block diagram of a successive approximation ADC. The DAC output is repeatedly compared to the input signal until the control logic has digitized the signal.

signal one bit at a time, starting at the most significant bit, to compare to the input signal. If the input signal is larger than the reference voltage from the DAC that bit remains set and the next bit is tested. If the reference voltage becomes larger than the input signal the corresponding bit is turned back off and the next bit is tested. For example a 4-bit successive approximation ADC with a 0 V to 15 V range is digitizing an input signal of 11.5 V. The first bit of the 4-bit output is switched on, and the DAC produces a reference voltage of 8 V. This is smaller than the input signal so the next bit is added and the reference voltage becomes 12 V. This is larger than the input signal so the second bit is switched back off and the third bit is switched on. This progression is shown in Fig. 2.10, and the ADC outputs a final signal of 11 V. This type of ADC is relatively fast, taking n steps where n is the number of bits in the output signal. It is also space efficient on a circuit and uses little power. 16 to 20 bit devices are possible with a 1 µs sample speed and a power use of 100 mW [77]. However, the successive approximation ADC suffers particularly badly from differential nonlinearity, where inconsistencies occur in the voltage width of individual digital channels [78]. This occurs because the resistors in the DAC need to set the reference voltage precisely, any deviation from the intended reference signal will directly affect the output. For differential nonlinearity of $< 1\%$ the resistor setting the 2^{12} channel in a $13-$ bit ADC must be accurate to $< 2.4 \cdot 10^{-6}$ [77].

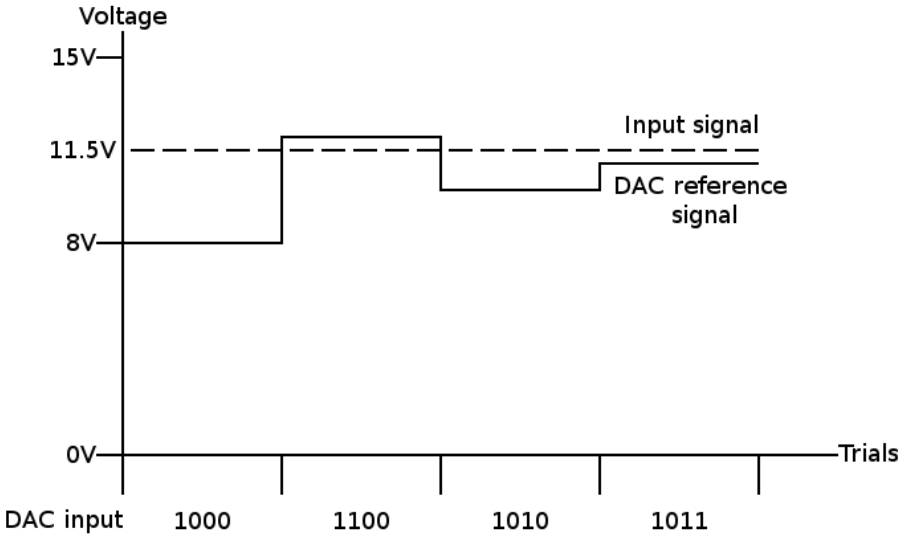


Figure 2.10: Plot showing the voltage input to the comparator in Fig. 2.9. The input voltage from the electronics, shown dashed, remains constant as the DAC varies its input to the comparator to try and match without going over.

Another common type ADC is the Wilkinson ADC, shown in Fig. 2.11. Here the input signal is used to charge a capacitor. The capacitor is then disconnected from the input signal and discharged at a constant rate I_R by a current source. The time taken for the voltage across the capacitor to drop below a baseline value V_{BL} is measured by a counter, counting pulses from a clock. The number of clock pulses counted between the start of the discharge and when the voltage drops below V_{BL} is related to the signal voltage V_S by

$$T_C = C\frac{V_s - V_{BL}}{I_R} \tag{2.24}$$

where C is the capacitance of the capacitor. It can be seen that if V_{BL} is sufficiently close to zero that $T_C \propto V_S$ and correct choice of C and I_R can allow the clock pulse count to be used as the output. The clock pulses, either from an internal crystal oscillator or an external source, are expected to be very consistent, avoiding most of the differential nonlinearity problems with successive approximation ADCs. However, the Wilkinson ADC has a timing profile dependent on the clock speed and the signal size. For example, a 13-bit ADC with a 100 MHz clock would

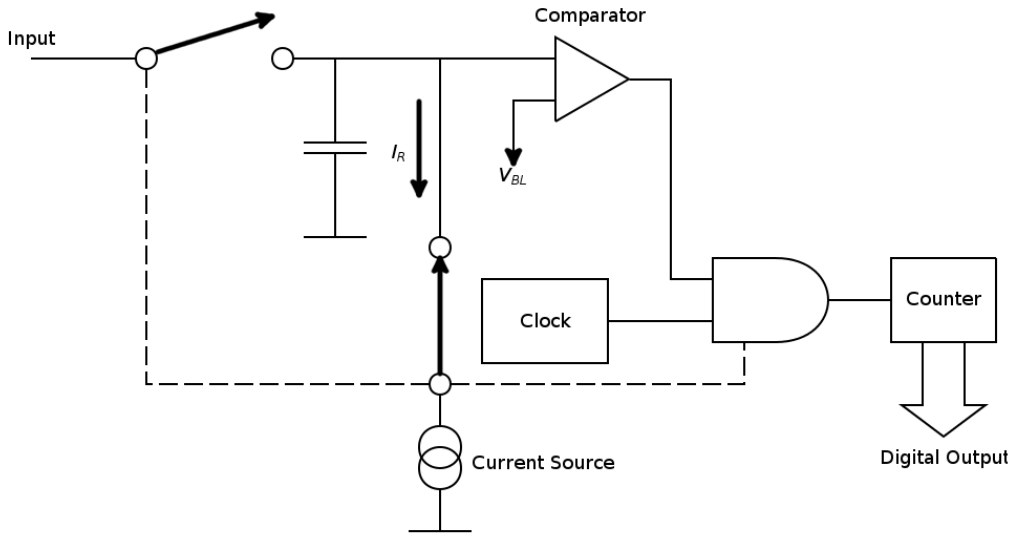


Figure 2.11: Block diagram of a Wilkinson ADC, showing the capacitor, current source, comparison to V_{BL}, the clock, the counter and the control logic back to the two switches.

require up to 82 µs to complete [79].

Another possible solution is to use a hybrid of two different ADC types. Here one type of ADC would be use to provide coarse conversion (e.g. 6 out of 10 bits) with another type providing fine accuracy with the remaining bits. This therefore represents an attempt to find a compromise between speed, accuracy and power usage.

2.5 Conclusion

This chapter has included discussion of the physical processes that are relevant to a simulation of an X-ray semiconductor detector. This has included all of the processes that an incident X-ray photon can undergo in matter; the process by which energy from an incident X-ray frees electron–hole pairs in matter; and the way these electrons and holes then move through the semiconductor. The implementation of this knowledge as a practical simulation is given in Chapter 6.

Plasma effects are a special case of X-ray semiconductor detectors, only occurring at very high charge densities that have been theorised to occur at the European

XFEL. This chapter has included discussions of the effects that charge plasmas have on the collection of charge carriers, and details of research that has been conducted into X-ray charge plasmas. Details of the Debye length are given in this chapter, as it was used in the simulation of plasma effects to determine when they would occur. Details of the proposed simulation can also be found in Chapter 6.

Chapter 3

Semiconductor detectors

3.1 Pixel detectors

Semiconductor detectors are constructed in a variety of designs for a variety of different applications. The 2D semiconductor pixel detector uses an array of pixels to provide a two dimensional position sensitive image of energy deposition. That they produce a pixel image of photon intensity is why this kind of detector is often known as an X-ray camera when used for detecting X-ray photons. This is because the same technology is used in some visible-light commercial digital cameras, as well as being used in astronomy for detection of many different wavelengths.

X-ray semiconductor detectors vary in size by orders of magnitude. At European XFEL, detectors currently being tested include the Large Pixel Detector (LPD), shown in Fig. 3.1, with 500 µm pixels and a pnCCD[1] with 75 µm pixels, although the pnCCD is also available with pixels as small as 36 µm. The Large Pixel Detector (LPD) in total has over a million pixels and is half a metre wide and tall, while the pnCCD has $25,000$ pixels and has about $1.5\,\mathrm{cm}^2$ of active area. Specifications of several detectors designed for or intended to be used at European XFEL are shown in Table 3.1.

The semiconductor X-ray pixel detectors simulated in this thesis use two principal types of technology to extract the charge generated in a semiconductor layer by an incident particle. These are Charged Couple Devices (CCDs) and hybrid

[1] pnCCD is a variant of CCD manufactured by PNDetector GmbH.

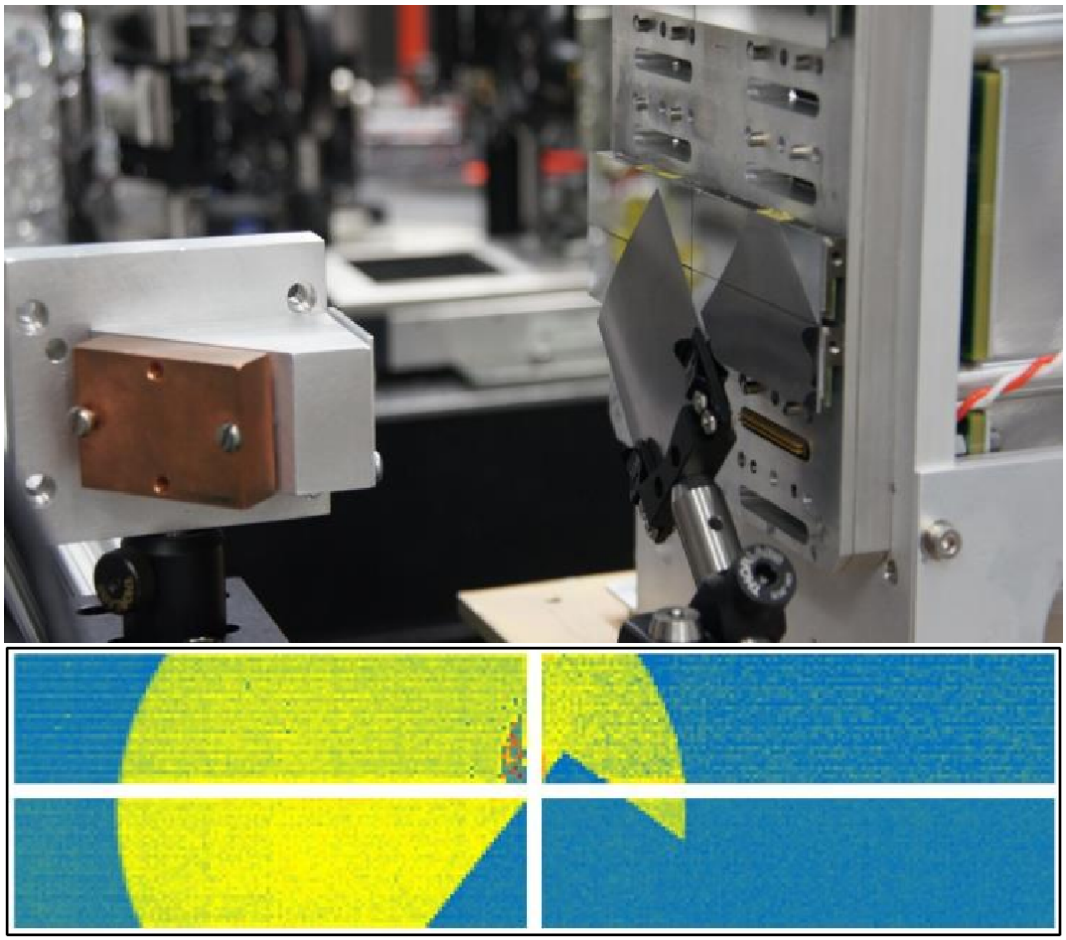


Figure 3.1: An image of an LPD super-module prototype being setup with six attached modules, and the image taken by the lower four modules [80].

Table 3.1: Selected specifications for some of the detectors to be used at European XFEL

	LPD [81, 82]	AGIPD [83, 84, 85]	DSSC [86, 87, 88]	pnCCD [89]
Technology	Hybrid pixel detector	Hybrid pixel detector	Hybrid pixel detector	CCD
Pixels	500 µm square	200 µm square	204 µm × 236 µm hexagonal	75 µm square
Depth	500 µm	500 µm	450 µm	300 µm
Dynamic range	1×10^5 at 12 keV	1×10^4 at 12 keV	6000 at 1 keV	Spectroscopic mode
Dynamic range technique	Triple gain profile	Pre-amplifier chosen gain	DEPFET non-linear gain	Linear
Total pixel count	1024×1024 pixels	1024×1024 pixels	1024×1024 pixels	128×200 pixels
Number of sensors	8×32	2×8	$2 \times 4 \times 4$	1
Sensor size	32×128 pixels	512×128 pixels	256×128 pixels	128×200 pixels
Sensitive area	262 144 mm^2	41 943 mm^2	37 932 mm^2	144 mm^2
Photon energy range	1 keV to 24 keV	3 keV to 13 keV	0.5 keV to 24 keV	0.1 keV to 15 keV
Optimal energy range	12 keV	3 keV to 13 keV	0.5 keV to 6 keV	0.1 keV to 15 keV

pixel detectors[2][3]. The hybrid pixel detector retrieves charge from the semiconductor with an electrical connection from each sensor pixel or strip to a readout circuit. In the case of the hybrid pixel detector each pixel is connected to its own circuit via bump bonding between the semiconductor layer and an Aplication Specific Integrated Circuit (ASIC) chip with the readout circuits. The requirement that each pixel in a hybrid pixel detector has its own readout circuit and bump bond to the semiconductor constrains the lower limit of pixel size in a hybrid pixel detector more than an equivalent CCD. An example of this is shown in Fig. 3.2, which compares the size of the pnCCD used at European XFEL and a single module of LPD. Additionally each bump bond, a small ball of solder pressed between the readout chip and sensor contacts, is a potential point of failure. A failure in a bump bond can leave an irreparable dead pixel on the sensor, with expected rates of failure of up to 1% [90].

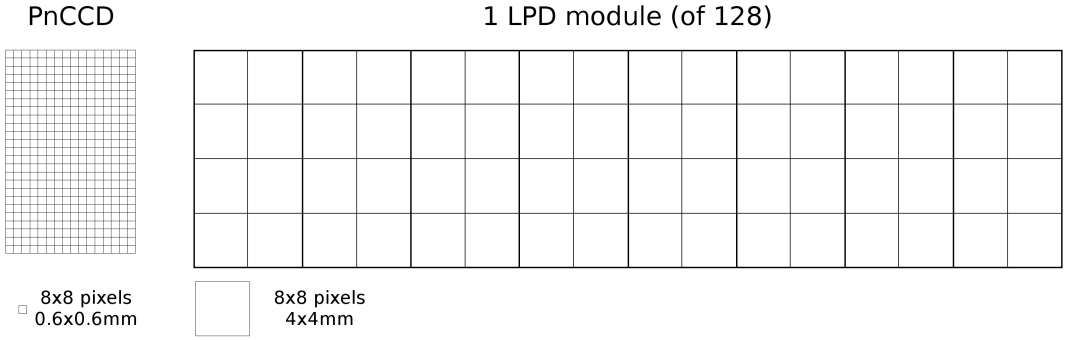


Figure 3.2: Scale comparison of the sensitive areas of pnCCD and a single module of LPD. Each square represents an 8×8 *square of pixels.*

Charged Couple Devices (CCDs) create a pixel grid with potential wells in the semiconductor. These potential wells collect and hold the charge, which is not immediately extracted. Charge is moved horizontally towards the readout point through the successive activation of adjacent potential wells. By activating a potential well adjacent to one that is already active and contains charge, the charge

[2]Both CCD and hybrid pixel detector refer to the whole unit or the whole detector, not a component of them.

[3]Monolithic Active Pixel Sensor (MAPS) are also used for X-ray imaging, but are not planned for use at European X-ray Free Electron Laser (European XFEL)

will spread out across the two potential wells. Turning off the initial potential well will then cause all remaining charge to move into the still-active neighbouring well. Charge is transferred to a horizontal register at the edge of the sensor, one column of pixels at a time, and then from the register to the on-chip amplifier. This is repeated with each cell to move charge towards the readout strip [91]. The movement of charge between potential wells cause CCDs to suffer from Charge Transfer Inefficiency (CTI), where a small portion of charge in a potential well does not move along to the next well, but is then collected by later charge bunches moving through. This is particularly evident in radiation damaged detectors as imperfections in the semiconductor contribute to CTI. CCDs can also suffer from blooming, which occurs when more charge is deposited in a potential well than it can hold, leading to charge overflowing into neighbouring pixels. Additionally trapped charges in CCDs can modify the electric field of the pixel in the CCD. The addition of charge to the potential wells in a CCD neutralises an equal number of opposite sign ions, lowering the potential difference and electric field across the depletion zone of the semiconductor. This leads to a subsequent increase in charge spreading[4] at high signal levels [92], however trapped charges would contribute to this effect permanently.

Because detectors are usually square, it is relatively easy to create a larger detector by tiling together multiple smaller detectors. Each individual detector is then known as a module, or tile, and the collection of modules is referred to as the detector, even if the individual elements are capable of independent operation. This allows a larger detector to be created without creating physically larger semiconductor strips, at the cost of dead space between the modules. All three of the bespoke detectors being constructed for European XFEL use this technique, with the Adaptive Gain Integrating Pixel Detector (AGIPD) and the DEPFET Sensor with Signal Compression (DSSC) using 16 modules and LPD 256 modules. Likewise, for a bump bonded hybrid pixel detector it is possible to bond multiple readout chips to the same piece of semiconductor. However, this can lead to the same problem as

[4] In this context, often called the point spread function.

with multiple modules, with a region of insensitivity between the pixels of each readout chip. AGIPD solves this problem by having double sized pixels at the edge of each readout chip, extending the edge of chip pixels to close the gap, while LPD uses an interposer (an integrated circuit die with no active components, that connects the sensor to the ASIC at different pitches) between the semiconductor and the chip to entirely remove dead space between chips.

The purpose of the readout circuit, whether it be connected to a row in a CCD or a single pixel in a hybrid pixel detector, is to measure the amount of charge the detector has collected and convert it to a readable digital signal. For X-ray imaging applications this is usually achieved by collecting the charge on a capacitor to convert it to a voltage, then amplifying the voltage and digitizing it. At the conclusion of each measurement the capacitor needs to be grounded, or cleared, to reset it ready for the next measurement. It is not uncommon for the digitizer[5] in particular to be shared by multiple readout circuits, such as in LPD where each pixel in a column of a module will use the same digitizer.

3.2 European XFEL detectors

The preceding section has explained in general terms how a pixel detector collects charge from the semiconductor layer and converts it to a signal. The rest of this chapter gives examples of what such detectors look like, as well as describing in greater detail their specific construction. These examples are focused on the three principal European XFEL detectors, those for which the simulation written in this thesis was intended, as well as pnCCD, which was also simulated as part of testing.

3.2.1 LPD

The Large Pixel Detector (LPD), shown in Fig. 3.3, is one of the three megapixel X-ray cameras being developed specifically for use at European XFEL. LPD has been developed and built by the Rutherford Appleton Laboratory (RAL), producing an image of 1024×1024 pixels for a real pixel count of $1,048,576$. As with the other two European XFEL detectors, AGIPD and DSSC, LPD is comprised of sixteen

[5] Also frequently known as an Analogue to Digital Converter (ADC).

smaller detectors arranged together, each with their own Front End Module (FEM) and its connections to the common clock & control system [93] and train builder [94]. Each of these sixteen super-modules mounts sixteen modules on the front. Each module is a 128 pixel wide by 32 pixel tall semiconductor strip, mounted on the super-module 2 across and 8 high for a 256×256 grid. Each module mounts 8 ASIC chips to the back of the semiconductor layer, each handling 16×32 pixels. A schematic of the LPD ASIC channel is shown in Fig. 3.4. The ASICs are connected to the semiconductor sensor by an interposer layer, allowing the ASICs to be physically smaller than the pixel area they read from, and therefore eliminates all dead space between ASIC chips. Of the three detectors, LPD has the largest pixel size, 500 µm square, and is therefore also physically the largest of the detectors. The modules and super-modules are arranged so that the gaps between each sensitive region, regardless of whether the gap is between modules or super-modules, is always 4 pixels, or 2 mm.

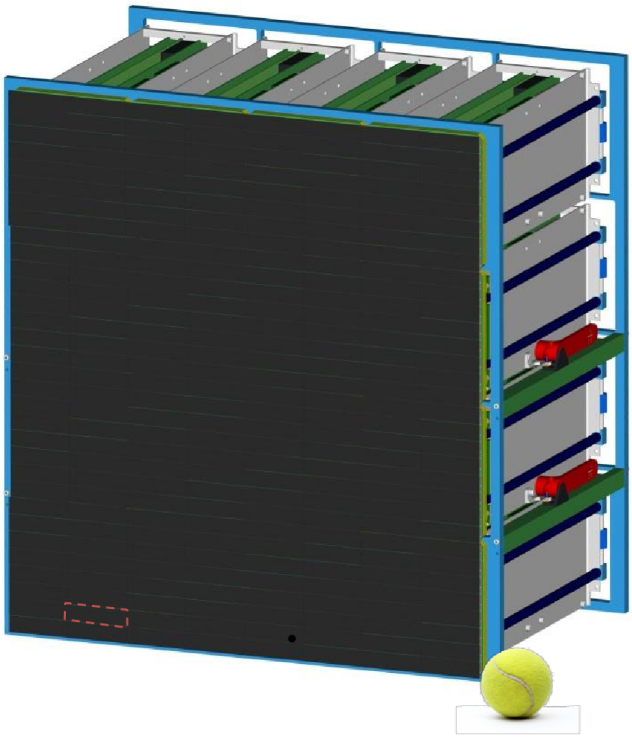


Figure 3.3: A computer generated render of LPD.

As with the other European XFEL detectors, LPD is designed to work with

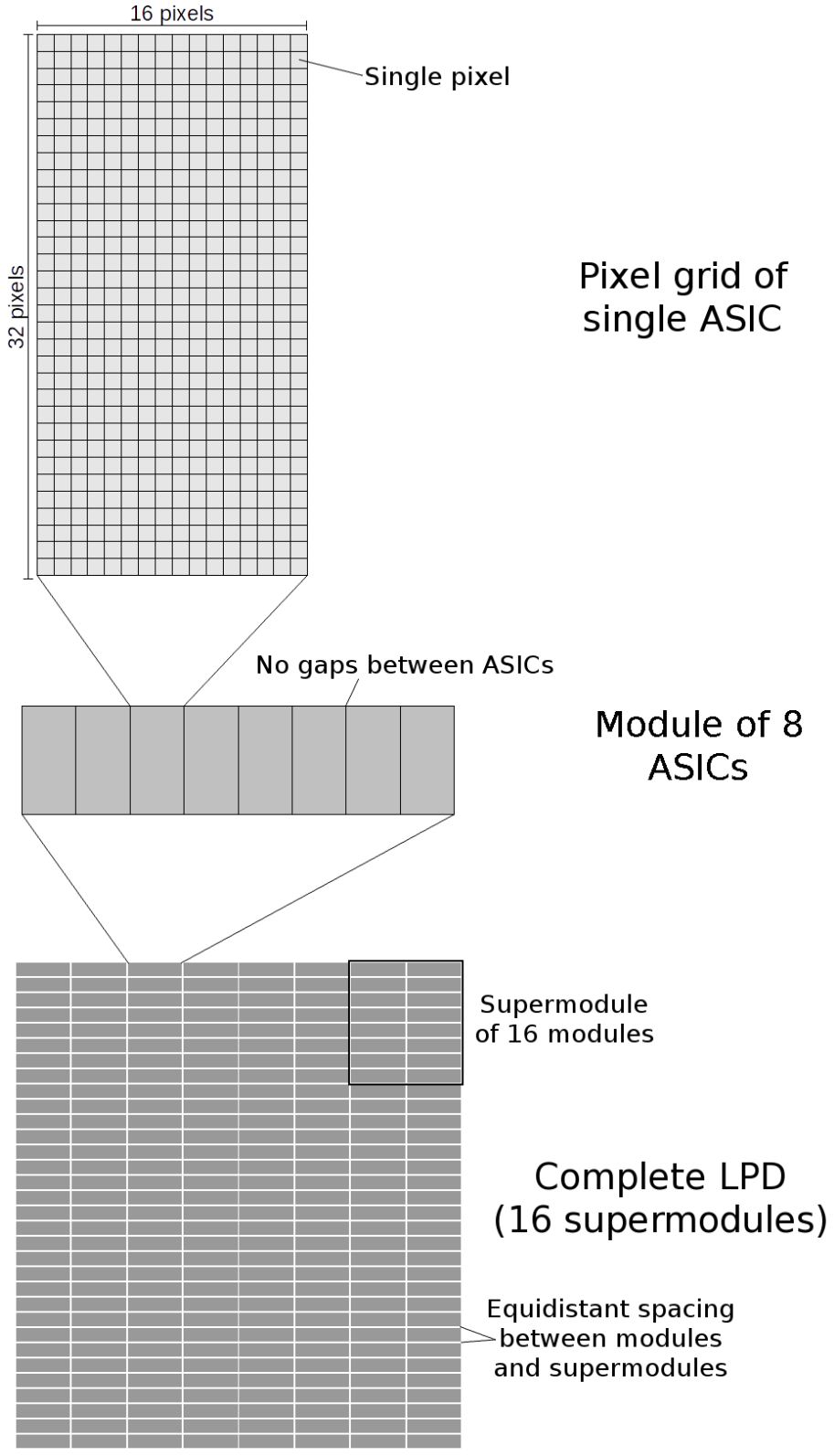


Figure 3.4: Multiple scale diagram showing the components of LPD from the pixel level to the whole detector.

the European XFEL timing structure. This involves up to 2700 $<$ 100 fs flashes at 4.5 GHz, followed by a 99.4 ms gap. LPD can capture up to 512 flashes per bunch train, which are stored in analogue form on capacitors in the ASIC. Between bunch trains, LPD then digitises the images, processes them on the FEM and conducts readout ready for the next pulse.

For each image taken by LPD, three images are recorded. The input signal is split down three channels, each of which multiply the signal by a different amount (referred to as the 1×, 10× and 100× amplifiers) and then store it on the ASIC to await digitisation. A schematic of the LPD ASIC channel is shown in Fig. 3.5. Between bunch trains all three signals are digitized and passed to the FEM, which then picks the best signal, converts it to a photon value and conducts readout. The multiple gain system allows LPD to have both the high dynamic range offered by lower amplifiers, and the high signal to noise ratio offered by larger amplifiers at lower intensities, on the same image. In addition to this, each pixel of LPD can be independently set to use either a 50 pF or 5 pF capacitor with the pre-amplifier. Compared to the 50 pF capacitor, the 5 pF capacitor cuts the dynamic range by 10, but, like the larger amplifiers, increases signal to noise ratio.

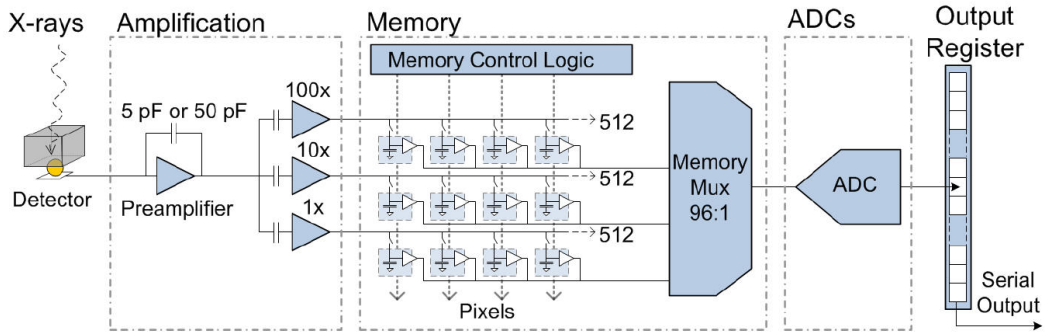


Figure 3.5: A simplified schematic of the single pixel ASIC structure [81].

3.2.2 AGIPD

The Adaptive Gain Integrating Pixel Detector (AGIPD), shown in Fig. 3.6, is one of the three European XFEL detectors and, along with DSSC, is being constructed by DESY. Like LPD, AGIPD is a 1024 × 1024 pixel detector which consists of sixteen smaller detectors and FEMs connecting to the same clock & control and train

builders. Each sub-detector has a single semiconductor sensor of 128 pixels high by 512 wide, comprised of 16 ASICs which each provide 64×64 pixels. Four such modules form each quadrant of AGIPD, and each quadrant is designed to tessellate in concert with the other quadrants in a windmill fashion to create the central beam hole. Each pixel of AGIPD is smaller than that of LPD at 200 µm square, making the total surface area of AGIPD smaller than that of LPD. To eliminate dead space between ASIC chips, the edge pixels of each ASIC chip are twice as large, with corner pixels being four times normal size. This extends the pixel grid over the edge of the ASIC chips.

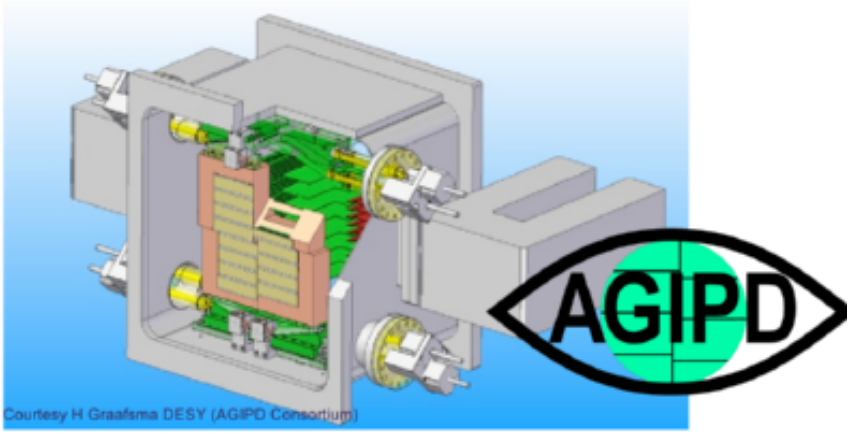


Figure 3.6: A computer generated render of AGIPD.

AGIPD uses an adaptive gain profile to obtain the same benefits as the multiple gain profiles of LPD, simultaneous high dynamic range and high signal-to-noise ratio at low intensities. AGIPD has three gain profiles, seen in Fig. 3.7, and depending on the size of the signal recorded by the pre-amplifier selects one of them by switching in resistors parallel to the main amplifier. AGIPD therefore only amplifies and stores a single signal per pixel per image, along with a 2 bit identifier so the signal can be reconstructed into a photon count by the FEM. In a similar fashion to LPD, AGIPD can store 312 individual images per bunch train, which are then readout between trains.

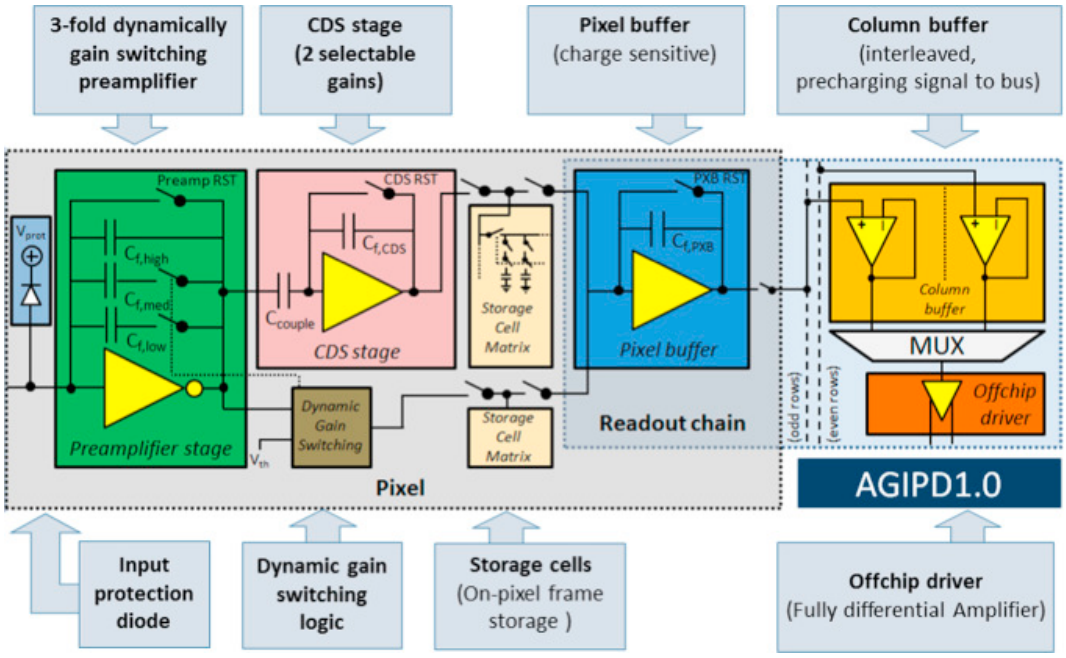


Figure 3.7: Schematic of the AGIPD pixel electronics and readout chain [85].

3.2.3 DSSC

The DEPFET Sensor with Signal Compression (DSSC), shown in Fig. 3.8, is the third detector being constructed for European XFEL. Like LPD and AGIPD it aims to achieve both low photon count sensitivity and high dynamic range. Unlike LPD and AGIPD it does not do this with electronics, instead using a DEPFET based sensor with a non-linear gain profile. Unusually for a pixel detector, DSSC uses hexagonal pixels. These are arranged into 128×256 sensors, two of which are mounted lengthwise to create a 128×512 pixel strip. 4 such strips are then mounted together to form each quarter of the detector. As with AGIPD these quadrants are designed to tile together to adjust the central hole size. DSSC has also been designed for a lower optimal energy range than LPD and AGIPD, being intended for use on the SASE 3 soft X-ray beamline. Like LPD and AGIPD, DSSC is designed to operate with the European XFEL timing structure, and is planned to store at least 512 images per bunch train. A schematic of the DSSC pixel circuitry is show in Fig. 3.9. At the time of writing, DSSC is still in the development stages and is not intended for operation at the opening of European XFEL. As such any simulation software written for use at European XFEL needs to be adaptable for DSSC, but

does not immediately need to simulate it.

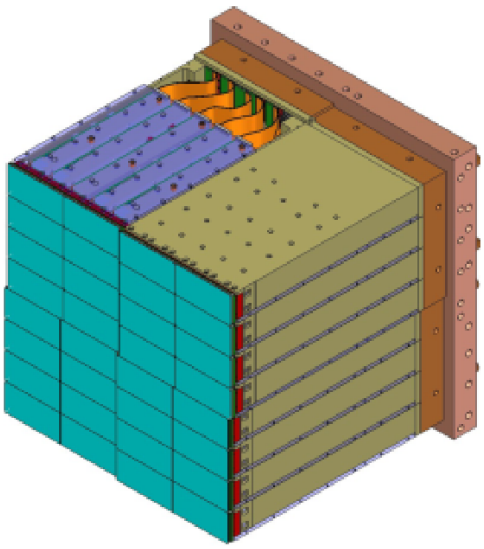


Figure 3.8: A computer generated render of DSSC.

3.2.4 PnCCD

PnCCD is a commercially available CCD semiconductor detector available in a variety of configurations. The model used in this thesis is a 300 µm thick sensor, with a 200×128 grid of 75 µm square pixels. All pnCCDs are fully depleted, backside-illuminated detectors. In particular the backside illumination maximises the distance from interaction to collection, this makes pnCCD ideal for testing the charge spreading simulation. While pnCCD is being considered for use at European XFEL it would not capture more than one image per bunch train as the three bespoke detectors will.

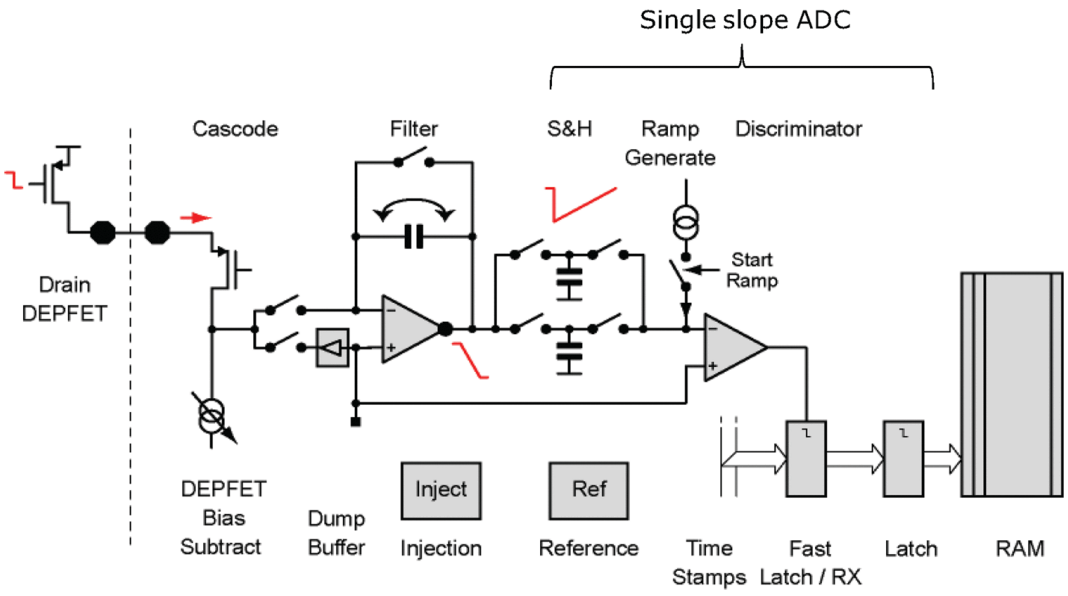


Figure 3.9: Schematic of the DSSC pixel electronics and readout chain [86].

Chapter 4

HORUS_LPD

the HPAD Output Response fUnction Simulator (HORUS)[1] [95, 96] is a simulation of the Adaptive Gain Integrating Pixel Detector (AGIPD) detector, written in the Interactive Data Language (IDL) programming language by G. Potdevin and J. Becker of the Adaptive Gain Integrating Pixel Detector (AGIPD) group. In this chapter work to adapt HORUS from a simulation of AGIPD, to a simulation of its sister detector Large Pixel Detector (LPD) is described. Initially this simulation did not warrant a new name, the intention had been that the modification would run simulations of both detectors. Shortly before publication this modification was named HORUS_LPD [97] to differentiate it from the original, as by this point compatibility between the two had been abandoned. By the time it was named, development of HORUS_LPD had already finished and a replacement, X-ray Camera SImulation Toolkit (X-CSIT), was in the planning stage.

4.1 HORUS

HORUS was initially written as a simulation of the novel, adaptive gain electronics system planned for use with the then HPAD, before it was renamed to AGIPD. J. Becker later added a physics simulation to HORUS, simulating photon interaction and charge sharing.

HORUS is written in IDL programming language. IDL is designed to handle and process images in an untyped[2] form, quickly and with very efficient manip-

[1] HPAD was the original name of the detector that would become AGIPD.

[2] In an untyped language there are no variable types for a programmer to define how data is stored.

ulation commands. This makes it an excellent tool for image manipulation and HORUS makes use of this. The electronics component of HORUS are considered as an image of pixels to which mathematical operations simulating components are applied. This approach uses the best functionality of IDL and is very efficient. However the physics simulation uses loops over a list of particles. As noted in the IDL documentation, loops in IDL are extremely inefficient and should avoided.

The physics simulation starts with an image input of photons to be generated at the face of the detector, with each pixel storing the number of photons to be generated in that pixel. By default each pixel of the image matches to one pixel of the detector, however the image can be over sampled by up to a factor of five. Each pixel of the detector would then receive photons from twenty five pixels of the input image. HORUS then generates photons, randomly positioning them within the input image pixel. Once positioned an interaction length for the photon is randomly generated from the mean lifetime of the photon in silicon. The incident angle of the photon is calculated from the pixel's position in the detector and the target location, which is positioned a user-set distance in front of the centre of the detector. An interaction point is then calculated from the interaction length and the incident angle. If this position is outside the sensitive layer, because the photon either passed entirely through the semiconductor or because it passed out the side of the module, the photon is discarded. If the interaction location is within the sensitive layer then a charge cloud simulation is run.

The charge cloud simulation distributes charge statistically using a two dimensional Gaussian distribution, with a width determined according to the work by R.F. Fowler [66]. See Section 2.2 for details. The energy of each photon is converted directly to charge at the calculated interaction point, no scattering or secondary particles are simulated. This energy is used, via the Fano factor, to determine the amount of charge in the cloud. Charge sharing is determined between the pixel the photon lands in and its eight neighbours[3] using the cumulative distribution of the

[3]The eight neighbours are the four pixels directly adjacent to the four sides of the pixel and the four pixels diagonally adjacent to the corners.

Gaussian distribution in the x and y axes. Because the multivariate distribution is circular, the cross terms of the distribution become zero and the equation becomes a sum of a purely x equation and a purely y equation. The cumulative distribution up to a line is then the cumulative distribution in the one dimensional form, using the distance from the spread origin to the line. Therefore the amount of charge crossing a pixel boundary or pixel grid line can be calculated using the cumulative distribution function. The proportion of charge in each pixel is then calculated by multiplying the proportions in its associated column and row. The results are then multiplied by the amount of charge and saved to the charge image. HORUS then moves onto the next photon in the list.

The rest of HORUS is concerned with the simulation of the electronics of the detector. Each stage of the electronics simulation is conducted by a different class. The main program feeds each class the image of the detector from the previous class and receives back the result. It also passes to the class any run specific triggers, but not any detector set-up information. Variables relating to the shape of the detector or the working of any electronic components are stored as global variables, accessible by any class and set at the start of the program.

The Aplication Specific Integrated Circuit (ASIC) simulation starts by amplifying the signals by one of the three gain profiles, switching between them based on the input signal as in AGIPD, and outputs a parallel image containing data on which gain was used. It also adds random Gaussian noise to the image, with a standard deviation specific to the gain profile used. Loss and flat rate noise from the charge being stored in a capacitor is then added. The Analogue to Digital Converter (ADC) then digitises the signal before the ASIC simulation finishes and the Front End Module (FEM) simulation quantises the image into photon count, including using the parallel gain profile map. The simulation is specific to AGIPD, the unique quirks of AGIPD, such as the triple gain profile and the double sized pixels, are hard coded into the relevant functions. Because the simulation was never intended to simulate anything but AGIPD no allowance was made to disable AGIPD specific features or room given to add others.

4.2 Conversion to LPD

HORUS begins by mapping the input image, which may be over sampled, to a table of modules. A pixel image of a single module is then created by adding ASIC chips together. HORUS then copies this module image into each entry of the table of modules. All variables such as ASIC size, number per module and number of modules are set by global variables (variables accessible anywhere in the IDL program). All HORUS classes use an input of this form and all but the last create their output the same way, by creating an image of a module (which may have more than one item per pixel), creating a table of modules using the module image as a template then storing the result in the table.

The first step in adapting HORUS into HORUS_LPD was to modify this image to match LPD. HORUS had no allowance for the rectangular ASICs used by LPD as AGIPD uses square ASICs. Simulating this required modification to the file that set the global variables and every class in HORUS, changing a size parameter into a size x and y parameter. Additionally HORUS has no support for super-modules. LPD mounts modules together in super-modules with common FEM electronics and readout. To anything interfacing with the detectors the LPD super-module is equivalent to the AGIPD module, however on the front side of the detector the super-module has sixteen separate sensitive areas (each a module) to the single sensitive area on AGIPD. While each module in AGIPD operates independently, each module in LPD is fixed in position to a super-module. Support for this functionality involved further modification similar to adding rectangular ASICs; modification to the global variable file and every class that handled data throughout the program.

The edge pixels of each ASIC in AGIPD are twice the size of regular pixels, with corner pixels four times as large, so as to cover the border of ASICs where they touch. HORUS simulates this by making ASICs two pixels larger in each direction, then at the amplification stage it takes the charge from the outermost pixels and adds it to the next outermost pixel. This aspect of HORUS was not a separate function or class, instead being a component of the amplification class. For HORUS_LPD this needed to be disabled as the ASICs in LPD are edgeless, this involved analysis of

the entire class to identify and disable the lines of code responsible.

AGIPD and LPD use two different methods to achieve the same goal of high dynamic range, and high sensitivity at low photon counts. Both detectors use three different gain, or amplification, amounts. A larger gain will have worse actual noise than a lower gain, but the increase in noise is not proportional to the increase in signal. Therefore a higher gain will have a better signal to noise ratio and greater sensitivity to lower photon counts, including single photons. However the large amplification limits the size of initial signal which will not exceed the maximum capacity of the circuit once amplified; this limits the maximum signal that can be seen. Therefore both detectors use two additional amplifiers with lower gains. While having a worse signal to noise ratio these have a higher maximum signal. Therefore the dynamic range of the detector is extended with one gain profile, accuracy is retained for smaller input signals with another, and a third gain offers a mid way point between the other two.

The two detectors differ in how they implement this multiple gain scheme. AGIPD uses an adaptive gain, choosing one of its three available gain profiles depending on the input signal. It then records which of the gain profiles was used along with the signal. LPD splits the signal down three paths, each using one of the three gain profiles simultaneously. All three signals are then stored and digitized before the front end electronics determine which of the three to use. HORUS simulates AGIPD by storing both signal and a 2-bit gain profile number for each pixel. HORUS_LPD needed to pass all three signals for each pixel instead, necessitating another change to the image creation code in most classes. Additionally all of the electronics needed to be altered to run three times on the three images and then store all three results to the output image. The image reconstruction class required re-writing to simulate LPD's method of recombining the three images into a single photon normalised output.

Finally the method of applying noise was altered. HORUS applied electronics noise at each component level, with methods specific to each component. However, component specific noise values were not available for LPD. Only total noise av-

erages for each of the three gain channels was available. So for HORUS_LPD all noise is applied at image reconstruction. While it would be unphysical for noise to occur here it enabled use of the LPD calibration data that was available.

LPD has an additional feature over AGIPD, which is the ability to set each individual pixel to use a 5 pF instead of a 50 pF capacitor when collecting charge. This has the effect of lowering the signal, and therefore the dynamic range of each gain profile, by a factor of ten and creating less noise. While on the real detector this can be set for each pixel, testing with HORUS_LPD required a different settings file with the alternate gain and noise profiles. The simulation was then run twice using the same seed value, but with different settings files. This produced an identical output from the physics simulation for both runs. While not ideal this was sufficient for initial testing and HORUS_LPD was replaced before per-pixel capacitor settings were added.

At the end of the modifications to HORUS, only the physics function of HORUS_LPD was still compatible with AGIPD. All of the electronics simulations had to be changed so significantly from the original AGIPD simulations that they had to be made into entirely new functions with incompatible data input and outputs. Even though the vast majority of the mathematics and theory was identical the two simulations were almost entirely separate.

4.3 Testing done with HORUS_LPD

To test HORUS_LPD, a test image was created by creating a radial profile in Mathemaica from multiple Gaussian profiles, shown in Fig. 4.1. This profile includes a central peak, a higher non-symetric ring around it and a long tail. Pixel photon counts were determined using this profile, which was applied radially around the centre of the detectors, seen in Fig. 4.2.

This new input was run through HORUS_LPD and the output can be seen in Fig. 4.3. Values for the electronics noise of LPD were taken from an internal document on the expected performance of the version 2 LPD ASIC [98]. The output image uses the same scale as the input image and here the drop in intensity can be

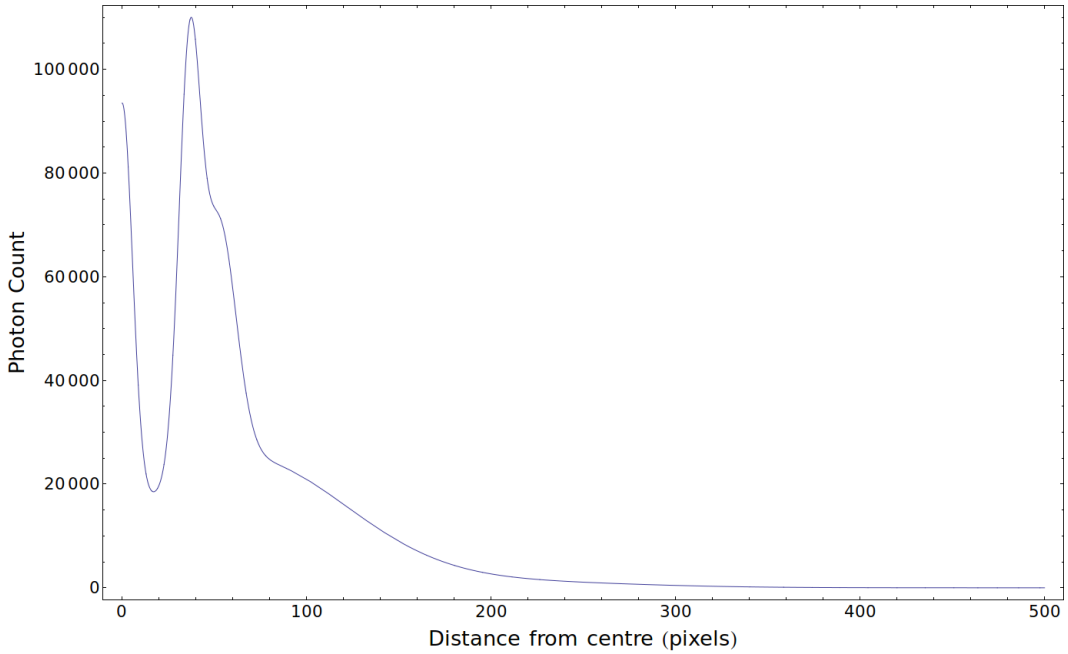


Figure 4.1: Line plot of the input image taken along the radial axis.

seen. The grid of zero output seen across the image are the gaps between modules where no sensor was present, and thus no signal was recorded. Comparing the two images it can be see that the shape and location of the ring is unchanged, therefore showing that image reconstruction is working correctly.

Fig. 4.4 shows a heat map of noise intensity. This is the magnitude of difference between the output image and the input image, the latter scaled down in intensity by the average proportion of photons not absorbed by the detector, 12.21%[4]. This image therefore is a comparison between the actual output, and an idealised perfect output. Notable features are the two overlapping circles, these are the areas where the intensity was high enough to use the $10\times$ and $1\times$ gain profiles, each with a higher average noise than the $100\times$ gain profile. Because the intensity decreases with distance from the centre, the cut-off points between gain areas are seen as circles. Also visible is a ring of high noise in the centre of the image, corresponding to the location of the input ring. This is where the error on photon counting is highest

[4]This was calculated by running the HORUS_LPD physics simulation with a high intensity beam of 100 million photons into one pixel away from the edge of a module and comparing the incident photon count to the signal. The input image also has the areas between modules masked off.

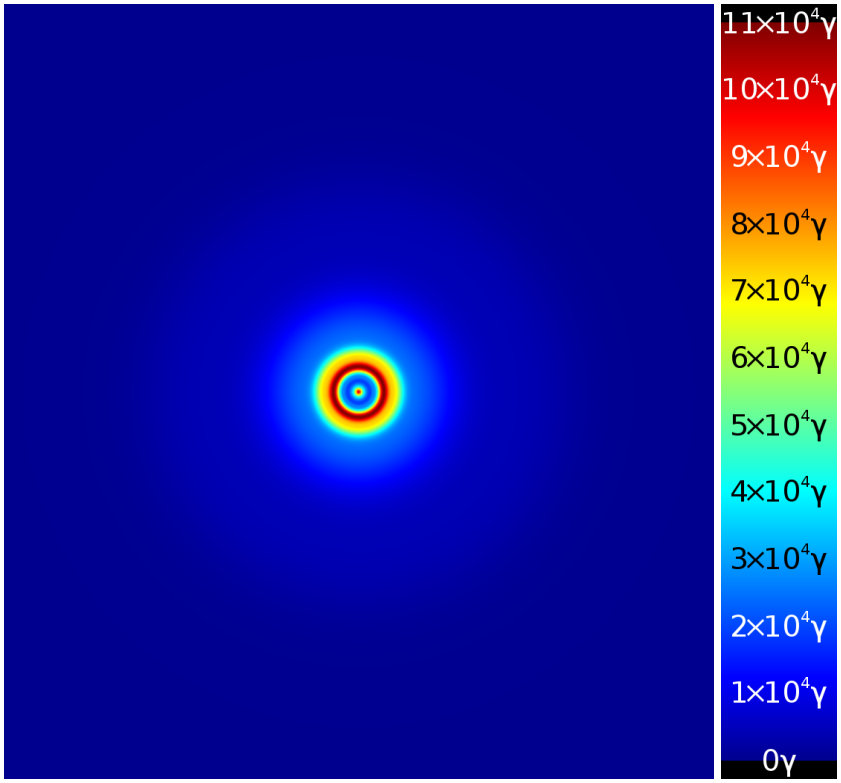


Figure 4.2: Heat map of input image and legend.

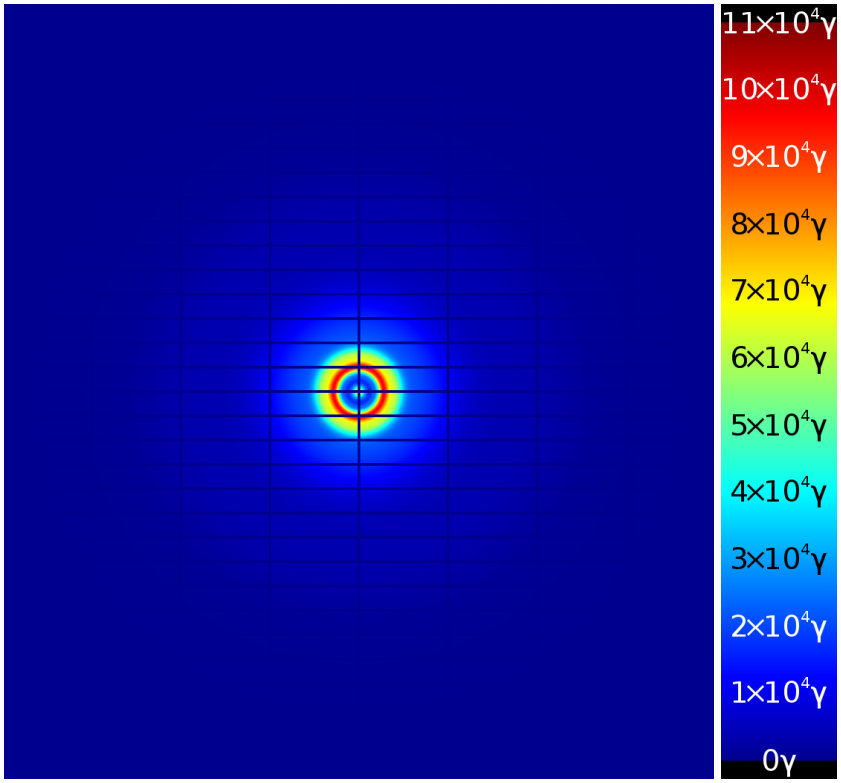


*Figure 4.3: Heat map of output image (*50 pF *capacitor setting) and legend.*

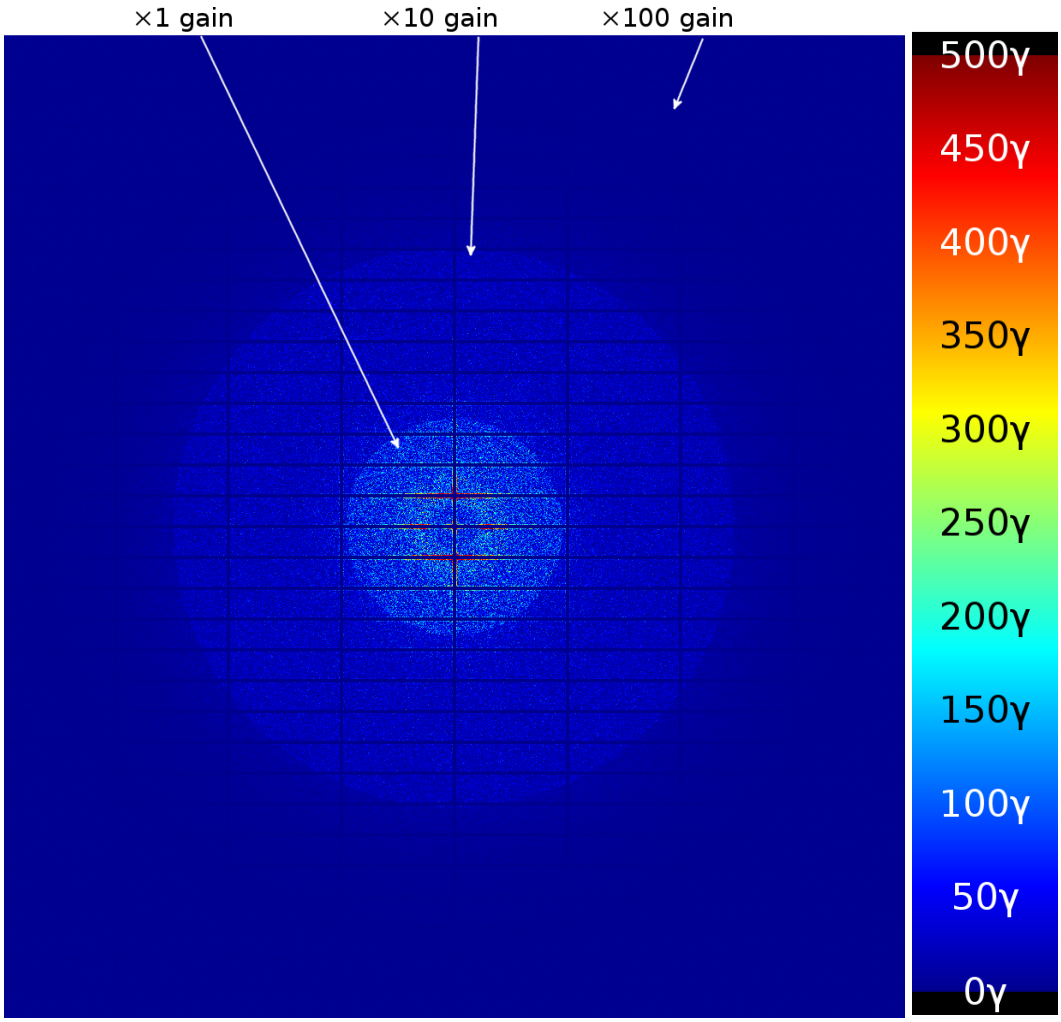


*Figure 4.4: Heat map of noise image (*50 pF *capacitor setting) and legend.*

due to the high photon flux.

Most notable on this image however are lines of very high noise along several module edges, this being most clearly visible where the high intensity ring crosses a module edge. Fig. 4.5 shows an enlarged view of the centre of Fig. 4.4 where the detail can be seen more clearly. While the lines of high noise are most clearly seen when they overlap the central ring, excesses on edges are noticeable elsewhere. Fig 4.6 shows a non-magnitude image of the noise. The scale for this image is twice as large so as to include negative numbers as well, with green now representing zero noise, red an excess and blue a deficiency. Here it can be seen that the lines of high intensity noise are all deficits, where fewer photons are seen than would be

expected. The remainder of the noise, including that from counting statistics in the high intensity ring, is randomly distributed.

The HORUS main program is an IDL function, and all intermediate detector state data is held within it. Only one output can be extracted from HORUS_LPD at the end of the function, intermediate data cannot be accessed. Comparing noise resulting from the physics simulation and noise from the electronics readout simulation requires the input image[5], the output image and the signal, or detector image, after the physics simulation has completed but before the readout electronics. Subtraction of these images provides the total noise of the whole detector, noise due to the physics simulation and noise due to the readout electronics. The distinction between the latter two is important, in the HORUS physics simulation some 12.21%[6] of photons incident on the detector penetrate the sensitive layer and are not recorded. This creates a statistical uncertainty on the number of photons absorbed in a pixel. This noise is physically limited, whereas the electronics noise is a detector quantity.

Fig. 4.7 and 4.8 shows the noise produced by HORUS_LPD due to readout electronics only. The central region of Fig. 4.8 has been blanked out where the input signal exceeded the dynamic range of the detector in the 5 pF setting. The regions where the higher dynamic range, higher noise profiles are used are clearly visible as circles on the 50 pF plot and as rings on the 5 pF plot. No other intensity related noise is seen on either plot. In particular there is no sign of the edge pixel effects see in Fig. 4.4, Fig. 4.5 and Fig. 4.6.

Fig. 4.9 shows a heat map of noise magnitude due to the physics simulation. The image appears roughly proportional to the input image, notably the input ring, as is to be expected. The image also shows the unexpected edge pixel noise effects seen previously. The zoomed in views in Fig. 4.10 and 4.11 confirms this, showing all of the edge pixel noise seen previously.

Investigation determined that the edge pixel photon deficiencies were caused by the charge sharing code used by HORUS and HORUS_LPD. Photons landing near the edge of a pixel may split their charge, or charge share, between the pixel

[5] Scaled down by 12.21% as before.

[6] As calculated by HORUS.

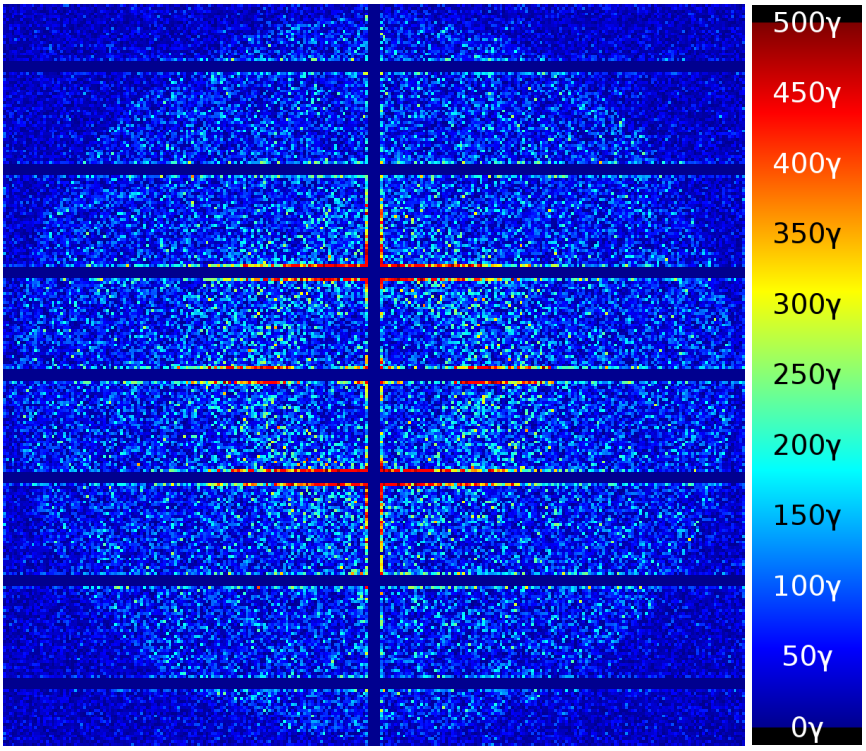


Figure 4.5: Zoomed view of Fig. 4.4.

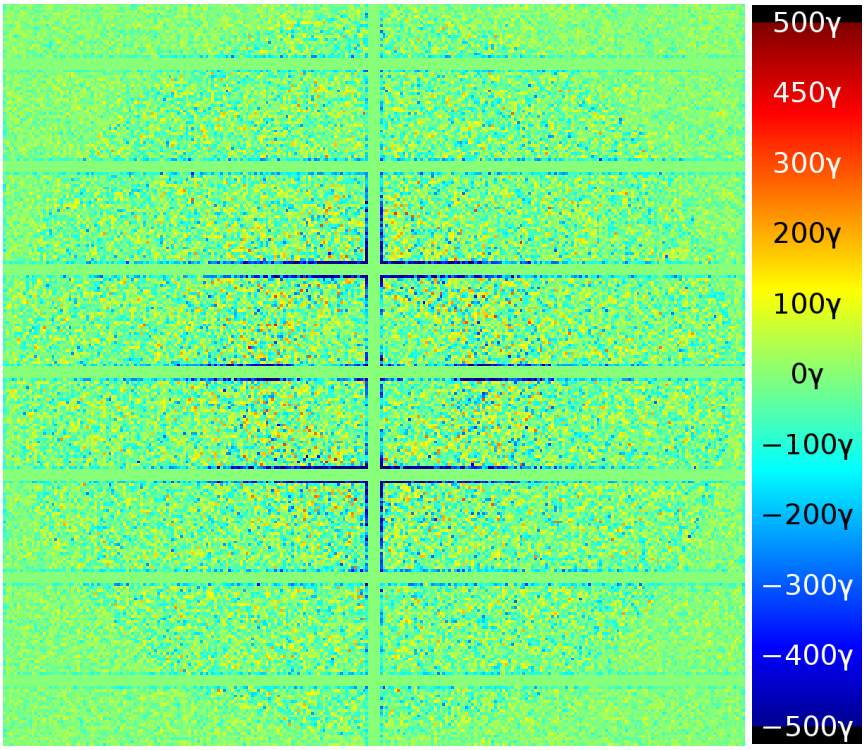


Figure 4.6: Zoomed view of Fig. 4.4 rescaled to show true deviation, not magnitude of noise.

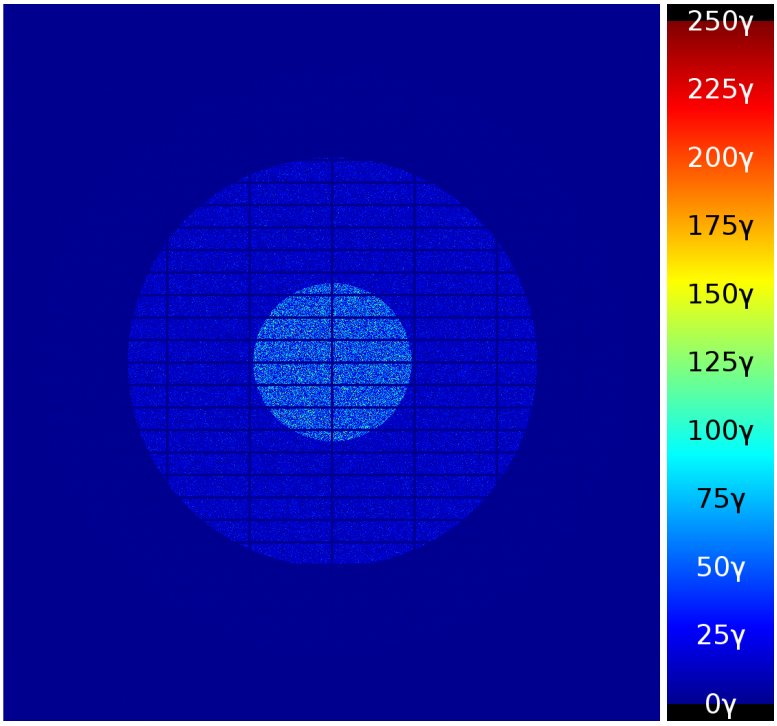


Figure 4.7: Heat map of electronics noise images for the 50 pF *setting.*

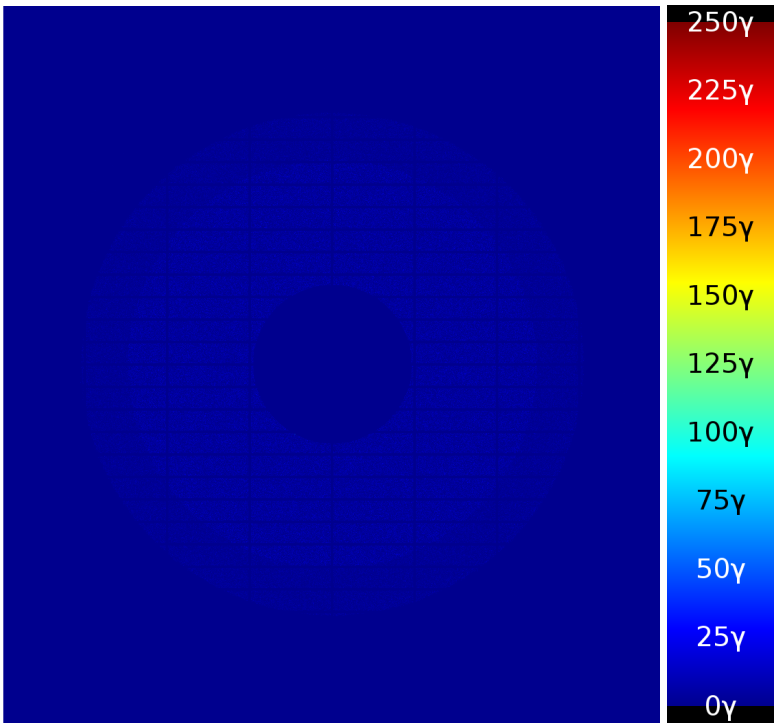


Figure 4.8: Heat map of electronics noise images for the 5 pF *setting. The centre circle of the image where the dynamic range of the* 5 pF *setting was exceeded has been blanked out.*

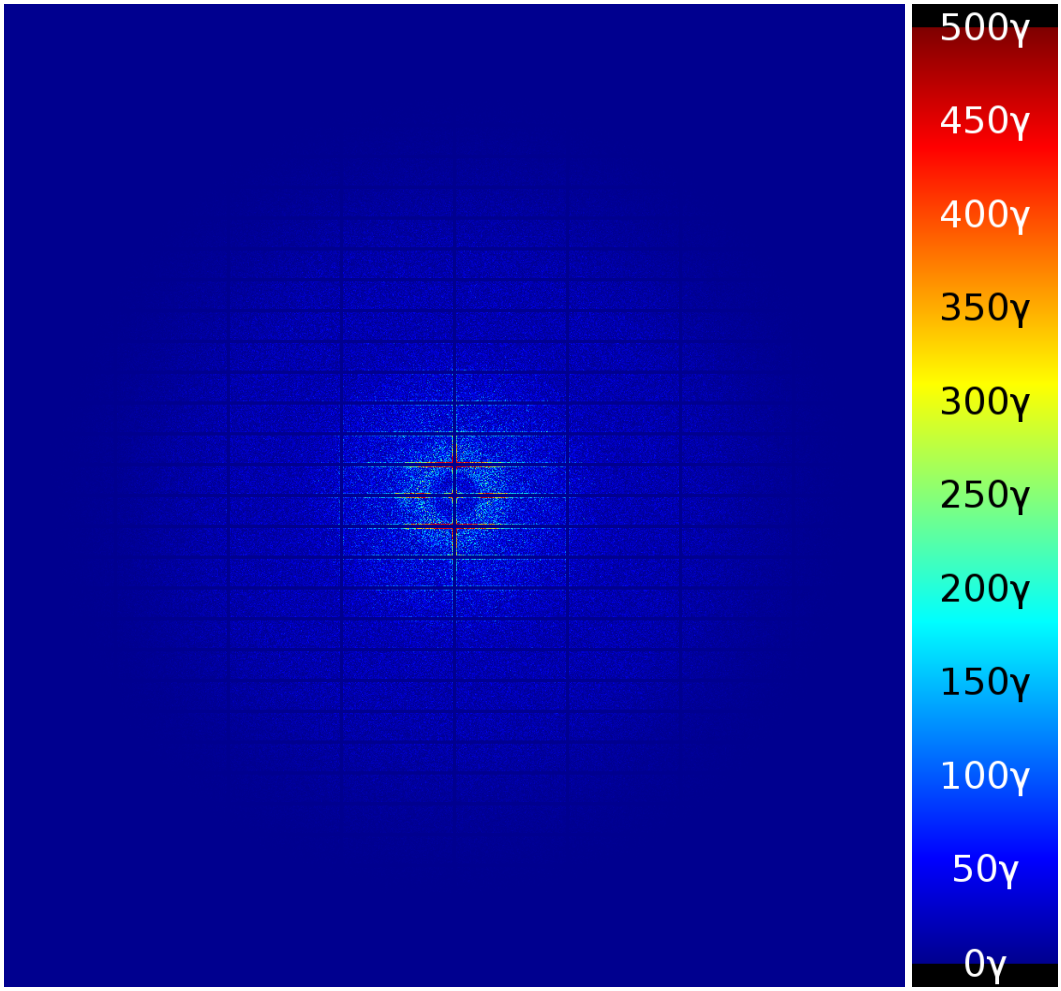


*Figure 4.9: Heat map of counting noise image (*50 pF *capacitor setting) and legend.*

they land in and neighbouring pixels, determined by the position of the impact[7]. Away from the edges of a module, pixels will share charge with their neighbour and the neighbour will share charge back, in roughly similar amounts because the input image does not have structure on a pixel scale (such as sub-pixel sized high intensity spots). At the edges of the module HORUS was still sharing charge even though there was no pixel to receive[8]. However while charge sharing happens in all pixels, it is because there was not a neighbouring pixel to share charge back into the module that we see a deficit of charge. Discussion of this effect with the LPD group determined that this was unlikely to be a real physical effect, and that the charge would be constrained to the semiconductor.

Fig. 4.12 shows the magnitude of the physics noise subtracted from the magnitude of the electronics noise. The red pixels show where electronics noise exceeded physical noise, and blue where physical exceeded electronics. The edge pixel effects are evident, but also the central ring, where high intensity had led to physical noise dominating over electronics. Towards the outer edge of the middle gain profile region and most of the lowest gain profile region, it can be seen that electronics noise steadily becomes the dominant source. These regions are where the signal is at its lowest but still requires a greater dynamic range gain profile. The reverse is also true, with physical noise slightly overcoming electronics at the inner edges of the highest and middle gain profile regions.

To improve upon Fig. 4.12, Fig. 4.13 was created. This is a histogram of pixel values from the HORUS_LPD runs, plotting physical noise (in the plot this is referred to as counting statistics noise) and electronics noise against the number of photons incident upon the detector in each pixel. Data from both the 50 pF and 5 pF capacitor setting runs are shown. Because both runs used the same seed value, and did not differ until after the physics simulation completed, their physics simulation output was identical. Therefore, only one set of data was used for the counting

[7] See section 2.2 for details and 6.2 for how this was simulated in X-CSIT.

[8] In the code HORUS calculates the amount of charge shared to neighbouring pixels and the amount of charge not shared, then adds this charge to the pixels. For edge pixels it continued to do this, but charge that was to be shared over the module edge was discarded.

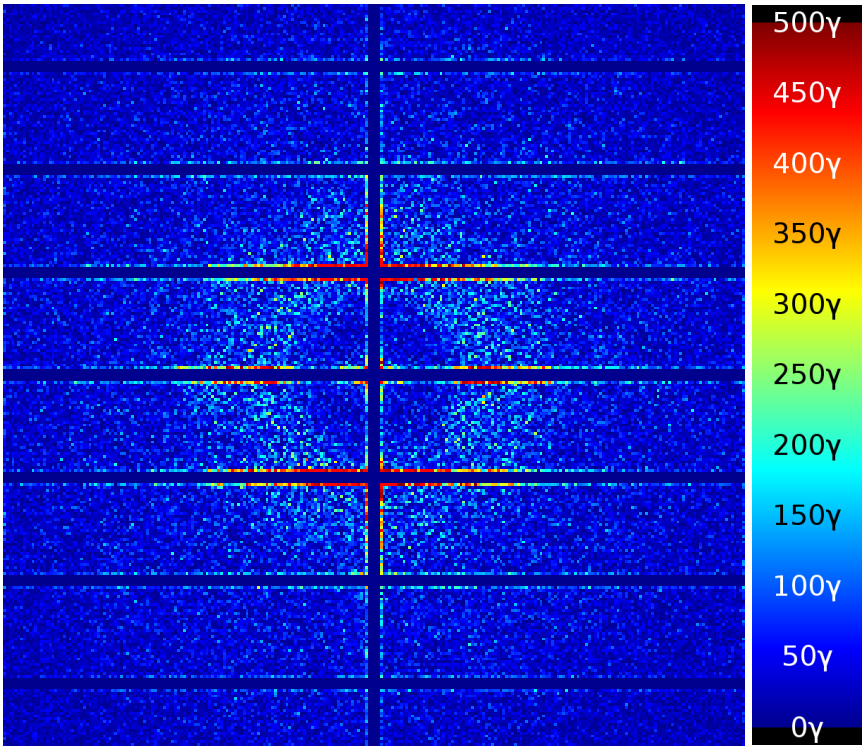


Figure 4.10: Zoomed of Fig. 4.9, heat map of noise due to counting error.

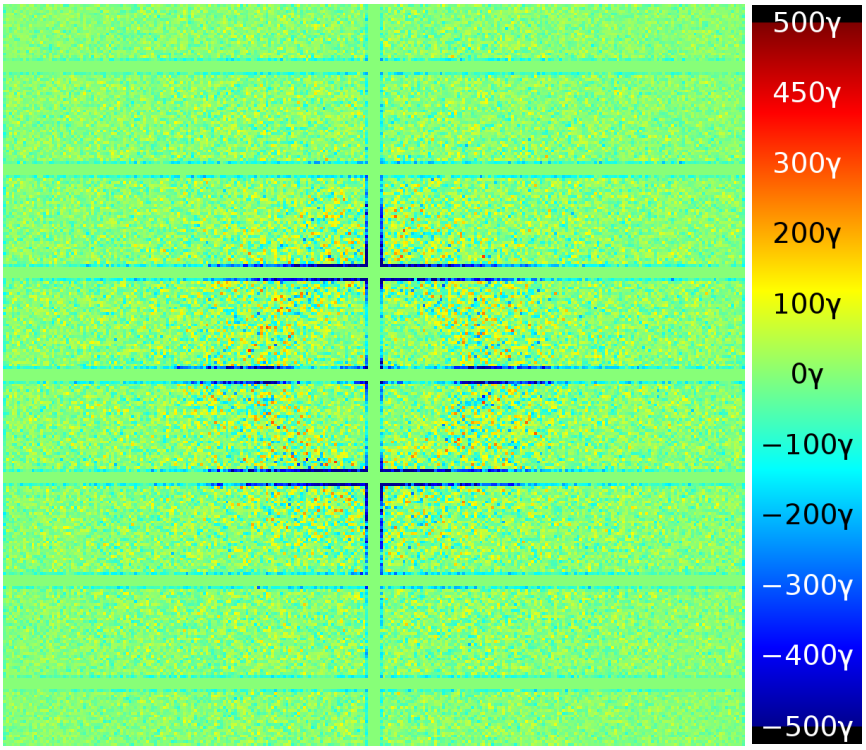


Figure 4.11: Zoomed view of Fig. 4.9 showing noise due to counting, rescaled to show true deviation.

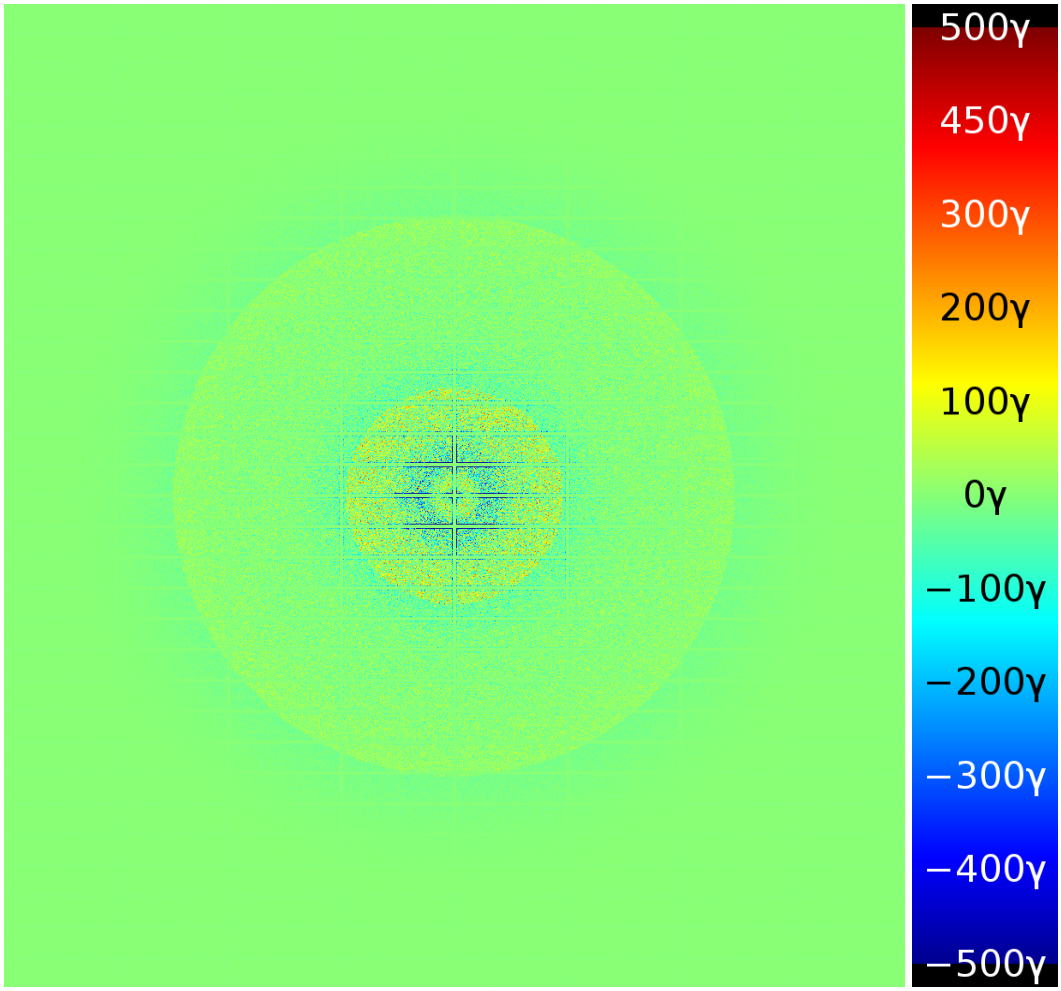


Figure 4.12: Heat map of electronics noise minus counting noise (green shows equal noise, yellow–red shows electronics dominating, cyan–blue shows counting dominating).

statistics noise, all three data sets shown have the same total data points and each bin has the same error on all signals.

Looking at the plots for electronics noise in Fig. 4.13 the three gain stages of LPD can be seen. The left most, lower horizontal lines of both the 50 pF and 5 pF capacitor setting runs show bins with pixels that used the 100× gain amplifier. Likewise the lines in the middle have pixels that used the 10× gain amplifier and the highest noise lines are pixels that used the 1× amplifier. Although the higher amplifiers will have higher noise in output voltage than the lower gain amplifiers, this increase in noise is not as large as the increase in amplification. Thus when Fig. 4.13 shows the output normalised to photons the higher gain stages have a lower per-photon noise. The bins of the histogram are not matched up to the cut-off points between gain profiles, as a result there are several data points between the horizontal lines that have contributions from both gain profiles. The most significant result shows that for almost the entire LPD dynamic range the electronics noise is lower than the physical noise. The physical noise is consistently proportional to the input photon count for almost the entire LPD dynamic range. There is a small peak at the very top end of the dynamic range visible. This may be related to the edge pixel effects which, while occurring throughout the detector and therefore the whole dynamic range, may consist of a higher proportion of pixels at the top intensities. The higher intensity region is concentrated in a ring very close to the centre of the detector which crosses many module boundaries for its size, in particular it crosses over a corner between four modules twice, at the top and bottom of the ring. This may make the very high noise edge pixels form a larger proportion of pixels at the very top of the chart. However it should also be noted that lower statistics also make the errors here larger as well, so the peak may be purely statistical.

4.4 Lessons learnt

HORUS was designed to exclusively simulate AGIPD. It was never intended to be modified to simulate any other detector and as a result the structure of the program made it difficult to do so. At the conclusion of work on HORUS_LPD, plans for a

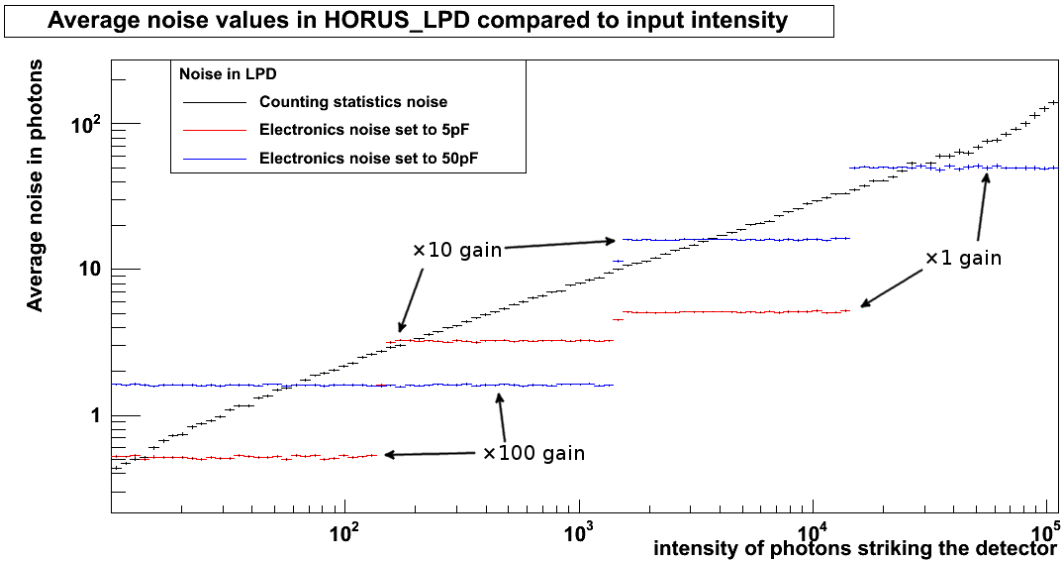


Figure 4.13: Average noise values due to counting statistics and electronics noise across the LPD dynamic range.

replacement were drawn up to address the lessons learnt working on HORUS and to provide common simulation of the three detectors designed for European XFEL.

The IDL programming language, while very powerful in its specific field, is ill suited to the numerical mathematics required of a physics simulation. Additionally use of it requires a license that is not always desirable in academic research involving multiple institutes. A switch to an open language would make the simulation far more distributable and easier for users to work with. Another improvement would be a more accurate photon input than the HORUS pixel based system. Even with oversampling, HORUS had a lower limit on the structure it could simulate. Thin rings or small spots of intensity could not be simulated, especially with the larger pixels of LPD.

A modular and more easily adaptable structure was also required. The replacement simulation would be required to simulate at least three detectors with differing layouts and electronics. This would be achieved by making a modular simulation of the physics to which the quirks and unique methods of each detector could be added. This would also lead to a simulation in which physics simulations are also modular and could be replaced in the future should an alternative model become

available. Adaptability of the data structure and a change in the way the simulation interacts with it would also be required for the various shapes and sizes of detector, as well as the differences in what data needed to be stored.

Finally a large change to the structure was decided upon. In HORUS a physics simulation first calculates a photon interaction, then the spread of the charge that photon releases, and then carries on to the next photon. The rest of HORUS then simulates the electronics response. The future replacement for HORUS needed to separate these areas of physics into three simulations of particles, charge and electronics. Doing so would allow separate development, improvement or modification of each component. With correct modularisation each sub-simulation could then be set up for or adapted to a new detector independently.

The replacement for HORUS and HORUS_LPD, later named X-CSIT, would then be used by European XFEL and incorporated into Karabo [99], the distributed control and analysis system used by European X-ray Free Electron Laser (European XFEL).

Karabo is designed to unify control of the four aspects of an experiment shown in Fig. 4.14: Control of the experiment itself, Data acquisition, management of the data once acquired and application of scientific computing to analyse the data. In this framework, simulations come under the view of scientific computing, even though they may not use raw data.

Fig. 4.15 shows the layout of a Karabo system. A central message broker passes communications between the various device servers and control systems of Karabo. This forms a single point of failure, but facilitates remote control of any part of a Karabo network from any control point. Users connect to Karabo and send commands either via an IPython [100] Command Line Interface (CLI) or with a Graphical User Interface (GUI) written in python using the PyQT [101, 102] framework. The GUI does not connect directly to the message broker. Instead it connects via TCP to a GUI Server device that then connects to the broker, enabling server-side preprocessing to reduce client CPU time and network load. Devices are the controllable objects in Karabo that then control physical items or software. They

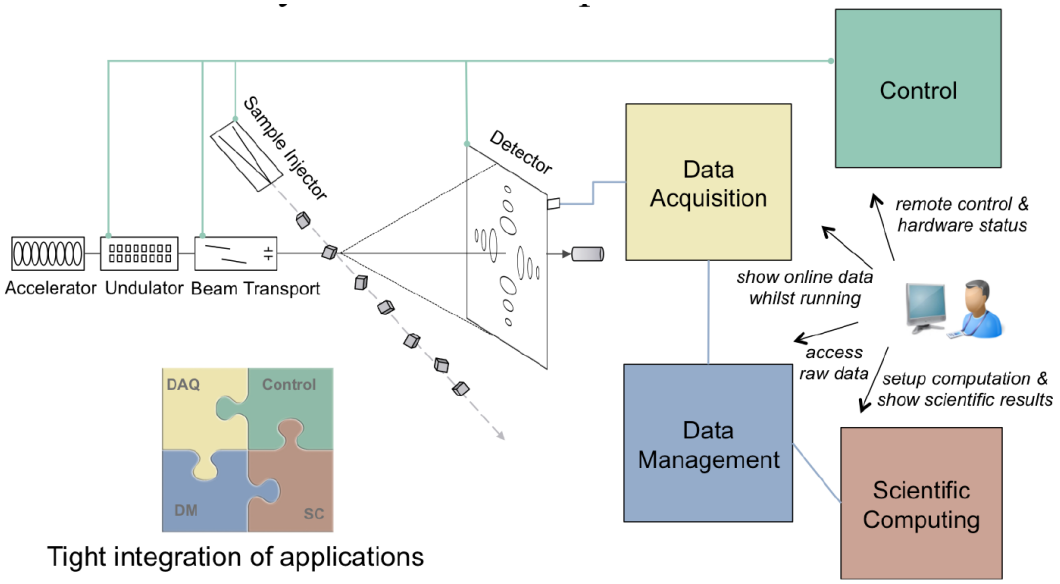


Figure 4.14: Karabo as a homogenous software framework. From [99].

can be written in either C++ or python. These are managed by device servers, which initialize an run the devices. Some devices, such as the HV and pump devices seen in Fig. 4.15, directly control physical components of an experiment. Others control data acquisition systems, such as the diode and camera devices. These devices have direct connections to the data storage devices to output data. These connections are part of Karabo and use Karabo defined data structures. Devices sharing data via these nodes that are both on the same device server will exchange data transparently via memory pointers Devices are also used to control software, such as simulations or data analysis.

The integration if X-CSIT into Karabo will enable it to run on European XFEL servers and take the place of real detectors in analysis pipelines, helping users plan or analyse their experiments, or as part of complete end-to-end simulations of European XFEL [103].

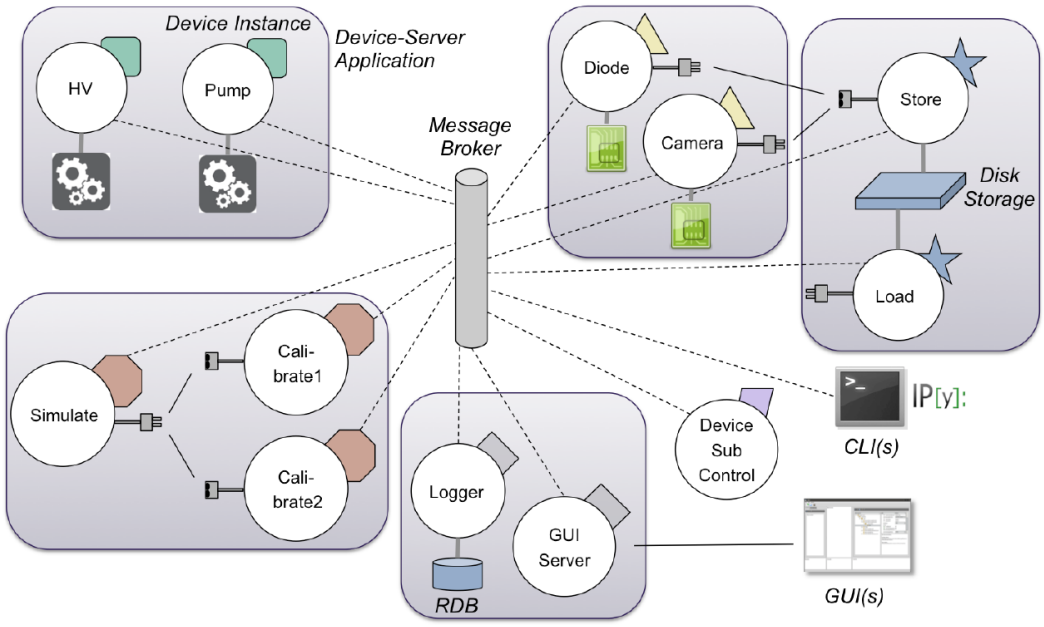


Figure 4.15: A diagram of the basic architecture of Karabo. Gray rectangles indicate device servers, circles show devices of different category (green: control, yellow: data acquisition, blue: data management, red: scientific computing). Dashed lines indicate message-based control whilst solid ones indicate direct point-to-point communication. From [99].

Chapter 5

X-CSIT technical description

5.1 Layout of X-CSIT sub-simulations

5.1.1 Layout

The X-ray Camera SImulation Toolkit (X-ray Camera SImulation Toolkit (X-CSIT)) consists of three sequential sub-simulations and the connections between them, seen in Fig. 5.1. These sub-simulations, the particle, charge and electronics, each simulate a different physical regime in the detector. They simulate respectively: the interactions of particles with the detector, the spread of charge clouds between pixels and the response of the detector electronics including output.

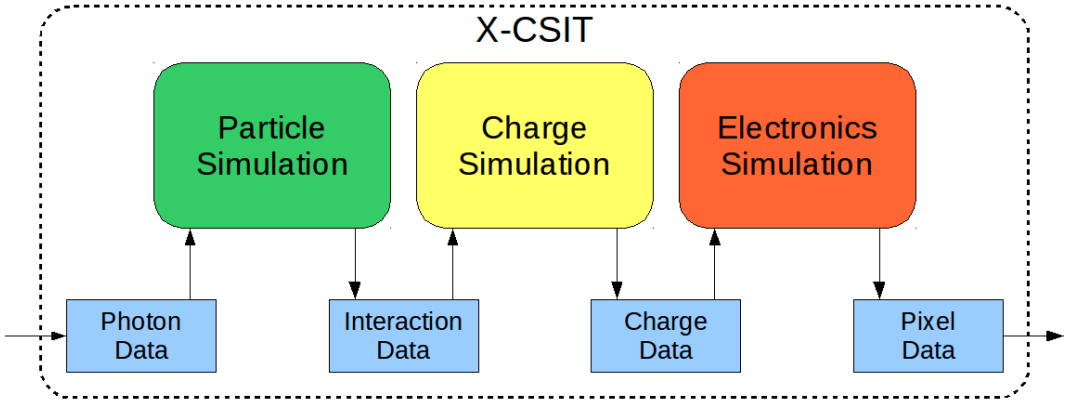


Figure 5.1: The layout of the X-CSIT sub-simulations and data classes, and the flow of information through them

These sub-simulations run independently of each other allowing them to be run at different times or on separate machines. It also allows separate development

and modification of the modular sub-simulations. Entire sub-simulations could be swapped out and replaced without affecting the other two.

The sub-simulations are connected by the data passed from one sub-simulation to the next as they run sequentially, listed in Table 5.1. The data classes define the information about the state of the whole simulation that has been calculated by one sub-simulation and is required as input by the next. These data classes also include the input and output of the whole simulation.

Table 5.1: The four data classes used by X-CSIT

Data name	Storage format	Number of entries	Data per entry
Photon data class	Sequential list	Entries added as required	Energy position (vector) direction (vector)
Interaction data class	Sequential list	Entries added as required	Energy position (vector) time
Charge data class	2D array	Defined at creation	Charge (floating point)
Pixel data class	2D array	Defined at creation	Charge (integer)

5.1.2 Motivation

Three different physical regimes take place in an X-ray camera. First the incident X-ray photons and child particles interact with the detector and deposit energy. Second the deposited energy frees electron–hole pairs which move under the effect of the electric field. Finally the detector's electronics respond to the collected charge. The principle development in X-CSIT is the recognition that a simulation of an X-ray camera can and should simulate these regimes differently and separately in three sub-simulations: a particle simulation, a charge simulation and an electronics simulation. A complete simulation can then be created by linking these sub-simulations

together.

The particle sub-simulation

The X-ray photons incident upon the detector should be considered and simulated as particles interacting with a medium (the detector). Each should be quantified by its position and momentum relative to the stationary detector and not by quantities considered by the end user, such as the pixel it is considered to have entered or landed in, as with the HPAD Output Response fUnction Simulator (HORUS) (see Chapter 4). The important result of these X-ray interactions is where in the sensitive layer of the detector they deposit energy in a physical process that frees electron–hole pairs. The particle sub-simulation must therefore simulate the physical processes particles under go in the semiconductor material as described in Section 2.1, particularly the photoelectric effect, Compton scattering and Rayleigh scattering.

The charge sub-simulation

The deposited energy frees electron–hole pairs from the valence to the conduction band of the semiconductor. The number it frees is proportional to the energy deposited plus Fano noise, see Section 2.2. The subsequent spreading of charge as it travels through the detector is a difficult problem to solve at the particle level. In silicon, a 12 keV photon at European XFEL would create approximately 3500 electron–hole pairs. A problem of scale occurs as the number of particles that need simulating increases by three orders of magnitude. Additionally all of the particles now interact with each other as well as the local electric field.

However, the magnitude of the charge created opens up a new opportunity. Because there are so many electron–hole pairs created, their final state can be simulated as a statistical distribution. The important result of the charge simulation is how much charge from an event is recorded and in which pixels. This means that if the probability distribution of the final state is known, in the simplest view it is only necessary to calculate how much of that spread falls on either side of a boundary between pixels.

The electronics sub-simulation

The charge collected by each pixel is the output from the charge simulation and the input to the third sub-simulation, the electronics simulation. Here the state of the detector is no longer represented by particles in a medium, but by charge flowing through a circuit. However, a continuous, simultaneous simulation of the whole circuit is not required. The readout circuits used in pixel detectors are not continuously flowing, instead they pass a single signal through each component in turn. A simulation of this can be built on this same principle. The physical effect of each component is applied to the signal in sequence. Common components that would be simulated include amplification and digitisation.

It also falls to the electronics simulation to simulate the readout of charge from the sensitive layer to the electronics. In a detector using bump bonding to perform readout this is a simple matter of applying the charge from the charge simulation across a capacitor. In the case of a Charged Couple Device (CCD), where charge is moved in the medium pixel by pixel to a common readout along one edge, the distinction between electronics and charge simulation becomes blurred.

While writing a simulation of CCDs it was found most convenient to define the boundary between the two simulations as being where the state of the detector can best be described as a matrix of pixels. This leaves the electronics simulation to simulate the movement of charge through the CCD and the transfer inefficiency, or blooming, that occurs. In effect this treats the CCD as the first electronic component simulated. This method of simulating the Charge Transfer Inefficiency (CTI) in the electronics simulation was first added to X-CSIT by T. Rüter, as described in his masters thesis [104].

The point of final readout from the electronics simulation should be the same as the output from the detector being simulated. This means that the electronics simulation may be required to simulate any Front End Modules (FEMs) a detector may use in addition to the electronics present on the module. The principle here is that X-CSIT should simulate the whole detector, such that a simulation in X-CSIT can be used in an analysis chain instead of a real detector. This includes providing

output at the same point and in the same format as the real detector.

5.2 Data handling structure

5.2.1 Connections between simulations

X-CSIT is built upon the principle that the different physical regimes in an X-ray detector should be simulated sequentially by three different sub-simulations. It is therefore important to connect these sub-simulations together in a well defined manner. These connections are the data input and the output of the sub-simulations, with each sub-simulation producing an output that contains all the data required for the next sub-simulation to use as input.

This data connection must describe the whole state of the physics in the detector at the point when the state changes from being best described by one sub-simulation, to being best described by the next. It should be noted that this is not necessarily a single point in time in the reference frame of the detector. The particle simulation stores the state of a particle event whenever it determines the charge simulation should take over, along with the time this occurs.

It is then the purpose of each sub-simulation to calculate the input required by the next sub-simulation. This principle binds the separate components of X-CSIT together into a single simulation of a detector, from the input of photons to the output of detector images.

These connections are achieved through each sub-simulation accepting a smart pointer to the data classes it will use as input and output. This use of smart pointers avoids copying data between or into and out of the sub-simulations. It also means that whatever function or program is running the X-CSIT sub-simulation(s) maintains control of and access to the data class, whether this is a user created program, or a control and distribution system such as Karabo. Finally it also allows the use of inheritance in the data classes. The sub-simulations then access the data in the data classes through the data class functions, without seeing how the data is stored or where it comes from. The use of this functionality will be expanded upon in Section 5.2.3.

5.2.2 Descriptions of data classes

X-CSIT connects the three sub-simulations together by defining what data is passed from one sub-simulation to the next to transfer the state of the detector. To transfer this data, four data classes are defined. These define what data is necessary to input into X-CSIT, transfer the detector state from one simulation to the next, and finally output to the user. These four data classes are: the photon data class, the interaction data class, the charge data class and the pixel data class.

The photon data class

The photon data class serves as an input to X-CSIT as the particle simulation. It is a sequential list, with each entry describing a single photon that is incident upon the detector. To do this the entry must store the photon's energy, position and direction of travel in the detector reference frame. This method aims to make the input to X-CSIT as accurate as possible and fully described. Other methods of describing the system such as a pattern, image or probability distribution would need to be converted to a photon list as the particle simulation works at the individual particle list. Specifying a particle input gives the user the greatest control over their input possible. Because X-CSIT is a simulation of the detector, not of the whole experiment, the particle list should describe the system after diffraction from the target.

The interaction data class

The second data class, the interaction data, is similar in structure to the first in that it is a one dimensional list. Each entry describes an event where energy has been deposited by a particle, that should now be simulated as a cloud of electron–hole pairs. It is a role of the particle simulation to determine when this energy should be simulated by the charge simulation, and not as a particle. To do this it must specify the amount of energy deposited, the position within the sensitive material it occurs in and the time at which it happens in the detector frame.

Each of these events is likely to be simulated as a charge cloud in the charge simulation, although that is up to the charge simulation. Simulations of charge plasma effects, which occur when very large amounts of energy is deposited in a

small space, would probably combine events together into a larger, single event.

The charge data class

The charge data class is a two dimensional data structure that stores how much charge has been collected by each pixel of the detector. This therefore makes it of fixed size in comparison to the variable length of the first two data classes. Describing the state of the system at this point, the connection between charge and electronics simulations, requires only the amount of charge collected by each pixel. However it is possible to extend this and store further data, as section 5.3 will explain.

The pixel data class

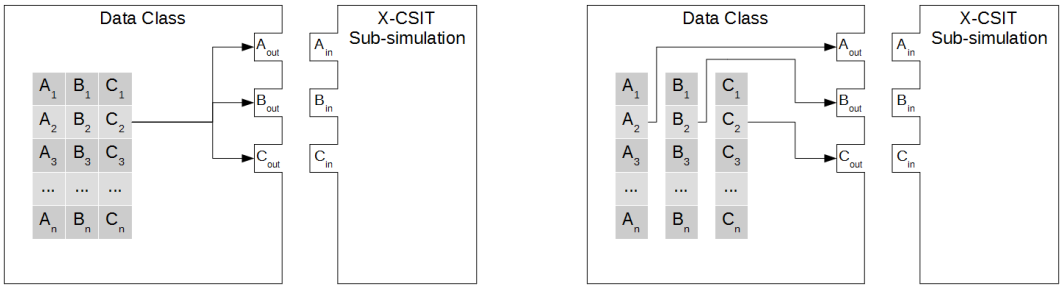
The pixel data class serves as output from both the electronics simulation and X-CSIT. It is a matrix, or image as it should be interpreted by the user, of integer values. The only difference between the charge and pixel data classes is that the former uses floating point storage, and the latter integer. While the charge data class could store the number of electrons collected by each pixel, the charge simulation does not simulate each electron or hole individually, but as part of a larger collection as a charge cloud.

The units of the pixel data output are not specified as they should match the output units of the detector, which can vary. The simulation and detector output may represent a signal normalised to the energy of the input photons, or it may simply be the output of the detector's analogue to digital converters. With a pixel data output in the same integer units as used by the real detector, the data can be converted to the same output format as the detector or, as described in Section 5.2.3, a pixel data class can be written that stores the data in the detector output format which is then available to the user.

5.2.3 Design and expandability of data classes

X-CSIT has been designed to hide how the data classes store data from the X-CSIT simulations. This enables two features. The first is that X-CSIT can use data in any format the user wishes (see Fig. 5.2). The second is that more data can be stored

in a data class without X-CSIT sub-simulations knowing and without affecting the simulation.



(a) Data class storing all data together. *(b) Data class storing data seperately.*

Figure 5.2: Diagrams showing two implementations of a data class interacting with a sub-simulation. The sub-simulation uses the data classes in the same way regardless of how the data is stored.

The desire for data storage indifference was inspired by the need for integration into Karabo at European XFEL. Hiding the method of data storage from X-CSIT sub-simulations meant that when X-CSIT was run by Karabo, data could be stored in data formats compatible with Karabo data transmission[1]. This enabled use of Karabo data transmission without converting data types. When used with Karabo, data is controlled by Karabo and data classes are given access to it (e.g. Fig. 5.3). No data is copied from Karabo into X-CSIT or vice versa. Likewise data stored in or accessed by the data classes is not hidden from the user. The user is free to store or access the data however and whenever they want.

The second added piece of functionality is the ability to expand the data stored without affecting the access of the original data by classes, code or simulations that do not know that more data is stored (see Fig. 5.4). This is part of the design of X-CSIT that allows components to be extended or modified without affecting the rest of the simulation. Additional functions are added to a child abstract class which control access to the new data. Classes which want to use the new functions cast their data class pointer, from the abstract base class to the child abstract class with

[1] At the time of X-CSIT's design and writing of the data class structure, Karabo was still in early development and it was not certain what data formats Karabo would use.

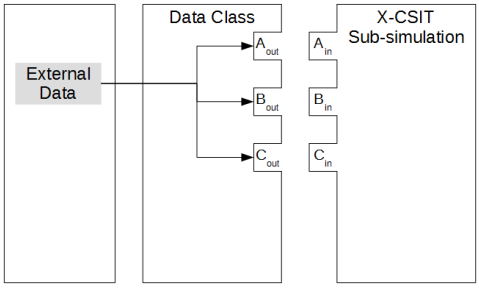
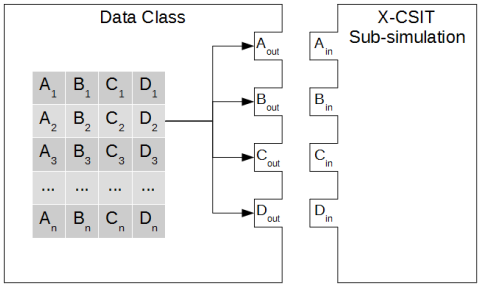
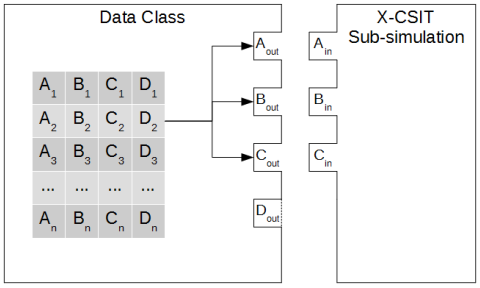


Figure 5.3: Diagram showing a data class interfacing X-CSIT with a external data storage, without storing data itself.

the additional function. Because the method of data storage is hidden from X-CSIT the actual contents are unknown and irrelevant to it. The only connection to the data is through the interface functions in the data class. So long as the modified data class still returns sensible information through the base class functions any other component of X-CSIT will continue to function, albeit without the use of the additional data.



(a) Data class with expanded data.



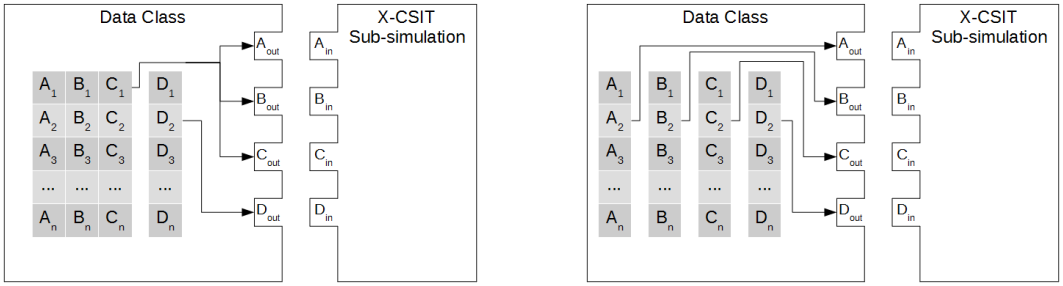
(b) Data class with unused expanded data.

Figure 5.4: Diagrams showing a data class with expanded data, interacting with a sub-simulation which can and a sub-simulation which cannot see the expanded data. The latter sub-simulation interacts with the data class as though it only had three pieces of data.

Three examples of this functionality will be given.

X-CSIT has only been designed for photon interactions in matter. However Geant4, which is currently the only option for the particle simulation, is capable of simulating more, such as electrons and neutrons. Expanding X-CSIT to simulate

other particles would require a new X-CSIT data class and PrimaryGeneratorAction[2] class in the particle simulation. The data class should inherit from the particle data base class, provide access functions to a particle identifier, and store a particle identifier with the rest of the data, such as in Fig. 5.5. Ideally this should be done with an abstract class adding functions and a concrete class inheriting from that. The new PrimaryGeneratorAction class needs to cast the pointer it uses to access the data to the new type, then use the new features. The rest of X-CSIT, including the majority of the particle simulation, is unaware of this change and functions as before.



(a) Data class with additional data table for extra data.

(b) Data class with separate tables for each item of data.

Figure 5.5: Diagrams showing data classes with different methods of expanding the data they hold.

The second example is a method of outputting extra data from the Large Pixel Detector (LPD) simulation. As discussed in Section 3.2.1, LPD uses three parallel output circuits for each pixel, then consolidates them into a single image. It would be possible to write an output data class which stores all three parallel circuits data in addition to the final output image. Storage of these additional images could be achieved with an additional dimension to the image table, or by the data class storing additional two-dimensional tables or other data classes. The latter would allow the user at the end of the simulation to view or manipulate all the images in the same

[2]The PrimaryGeneratorAction class is a user created class in any implementation of Geant4 [106]. An implementation of this class is provided as part of the Geant4 integration in the X-CSIT particle simulation.

format.

Finally what is perhaps the most useful application of data class expandability is the multiple module feature. Many detectors feature more than a single contiguous sensor, or module, and it is possible that the modules of a detector will not all be identical. This means that for future expandability these detectors cannot use an additional index or dimension in an array to identify the module the data relates to. Instead the charge and electronics simulations of X-CSIT run for a single module, with multiple modules added principally by expansions to the data classes. Each interaction output from the particle simulation for each module is stored in its own data class which is in turn held within the main data class output by the particle simulation, as seen in Fig. 5.6. The whole dataset is available from the main data class, but the datasets for each module are available as separate classes which can each be used as an input to the charge simulation. This allows the charge simulation for each module, which cannot interact with each other at this point, to be run independently. Each module can then be setup differently, or even run concurrently on multiple threads or machines. Alternatively each module can be run sequentially with the same setup and the same simulation instance without loss of runtime. In both cases the same functionality is available for the electronics simulation, and like with the output of the particle simulation, data classes containing other data classes for each module are available for the output from the charge and electronics simulations.

These final two examples demonstrate the control the user has over data storage. In both cases information (the module an entry or matrix relates to) is stored by the data class by where an item is accessed from, not as a data point in memory. This is analogous to how the 2D table data classes store the x and y pixel values for each pixel entry. They store this by indexing the table, not by storing additional integer values. This saves memory space and offers functional advantages when access to a specific dataset is desired, as it can be easily separated from the rest.

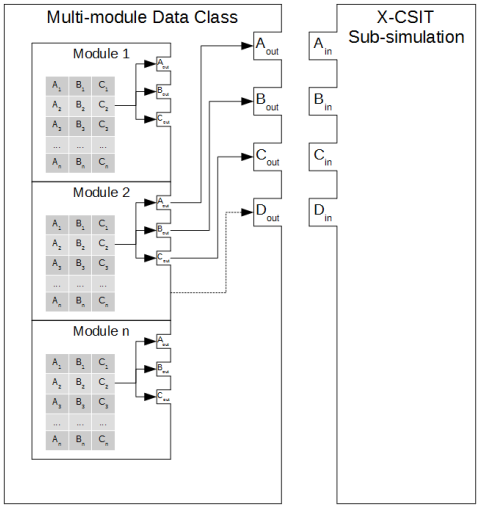


Figure 5.6: Diagram showing a multi-module data class storing multiple other data classes.

5.3 Expansion and adaptability of structure

The core design principle of X-CSIT has been the adaptability of it for different designs of detector. It has always been intended to use X-CSIT to simulate the three principle detectors at European XFEL. Experience with HORUS demonstrated how adapting a bespoke simulation of one detector to simulate another can involve deconstructing and re-writing huge sections of code. The intent with X-CSIT was therefore to create a simulation that was highly expandable and adaptable.

Each X-CSIT sub-simulation has an abstract base class which defines what data class types should be used for the input and output of the simulation. Concrete classes implementing the simulations inherit from one of these base classes. The input and output definitions are smart pointers that have a type of a data abstract base class. The data abstract base class defines the functions necessary for and available to the simulation to access the data. The concrete simulation classes are given a smart pointer to a concrete data class and use the functions defined in the data abstract base class to access the data. Because the pointer type is still that of the data abstract base class the sub-simulation does not know what concrete data class actually implements the functions or how it stores data.

The use of inheritance based programming allows for multiple child classes,

but at the moment only one class exists for the particle and charge sub-simulations, that inherits directly from the abstract base class. Each detector simulated by X-CSIT will use its own class inheriting from the electronics sub-simulation base class. This class then accesses the mathematical simulation component of the electronics sub-simulation from a collection of classes that each simulate an electronic component. The use of abstract base classes is also a design decision that is common with the data classes where this feature is more frequently used.

Using abstract base classes for all sub-simulations maintains consistency with the electronics sub-simulation and data classes, but is also part of X-CSIT's design for expandability. While it does not make sense at the moment to change or add additional choices of simulation for each of the particle and charge simulations, this may change later in the lifetime of X-CSIT. It is much easier to design X-CSIT to have this feature now and to not use it, than to desire it later and retrofit it.

Clearly defining the base form of each simulation also serves to keep them separate. By keeping the three sub-simulations separate and independent of each other they can all be modified, changed or replaced without affecting the rest of the simulation. Only the components modified and their output need to be tested, or indeed understood by the user modifying the code, the rest of the simulation can be left untouched.

This focus on expandability and ease of modification continues within the simulation classes. Within the charge simulation, effort has been made to keep aspects of the simulation in separate classes, such as the charge plasma and charge diffusion sub-sub-simulations. Additionally each of these simulation classes is given to the concrete simulation class as a smart pointer, much like the data classes. Therefore simulation classes can be modified or replaced by a new class, as long as it inherits from the smart pointer type. The charge diffusion simulation also accepts a definition class provided by the user. This class serves to define the size of the sensitive area the charge simulation is simulating, and how this space maps to the output pixels. This allows the user to set the simulation for most of the variations that occur between detectors, including size, pixel grid dimensions and differently sized pixels

in the same module, so long as they still form a grid.

The particle simulation achieves expandability and ease of modification through its use of Geant4 which works in a similar way, where pointers are given to a run manager. Here X-CSIT serves mostly to wrap Geant4, providing input and output functionality to and from it, controlling the run manager and tying it into the rest of the simulation while keeping features already part of Geant4. For simple, single module detectors users may only need to provide a DetectorConstruction class as per normal Geant4, connecting this to an X-CSIT provided sensitive detector. For more complicated multi-module detectors more programming is required, but examples of how to do this are provided and may only require filling in details of the number and order of modules. For even more complicated detectors the full power of Geant4 is available.

In the electronics simulation the expandability is inherent. An electronics simulation is created by collating the necessary classes and methods, assembling them in order and setting them to behave as the components of the detector. Adding additional methods to a simulation is done in exactly the same way as adding one of the many methods provided as part of X-CSIT. It is also possible to treat the particle and charge simulation in this way by creating a detector specific simulation class that inherits from the X-CSIT default class and differs only in providing all of the necessary setup classes on initialisation, instead of requiring the user or the controlling program to do this. This allows a single class to be designated as the only required component for each sub-simulation of a detector, particularly useful in a situation where multiple detector simulations are in use, such as Karabo at European XFEL.

The ultimate goal of X-CSIT is to have a set of three sub-simulations which have robust and tested physics simulations, but which allow the user to define the shape and layout of the detector without pre-defined notions of how the detector is constructed. By creating a simulation that can be adapted to and used for the different designs of detector at European XFEL, X-CSIT is also capable of simulating other detectors that have similar design characteristics. However, attempting to create a truly universal simulation would be impossible, instead X-CSIT has been

designed to be modular and adaptable, so that when additional functionality beyond what it was initially designed for is required, that functionality can be added without needing to understand or re-write the entire simulation.

Chapter 6

X-CSIT simulation description

This chapter describes how the physical effects that occur within a semiconductor detector are simulated. As described before in Chapter 5, X-ray Camera SImulation Toolkit (X-CSIT) is divided into three sub-simulations that each simulate a different stage of the detector and the different physics that occur. Sections 6.1, 6.2 and 6.3 describe these sub-simulations, what they need to simulate and how this is implemented in X-CSIT. They cover, respectively, the particle simulation, the charge simulation and the electronics simulation.

Section 6.4 describes the charge plasma simulation. This is intended to be a component of the charge simulation, simulating the plasma effects that occur at high charge densities. However while a simulation has been devised and developed, it was not possible to perform the experiments required to verify the simulation and therefore it remains a purely theoretical exercise for now. A description of the work done and theory behind it is provided; and future work that will validate the simulation is anticipated.

6.1 Particle simulation

A simulation of a semiconductor detector must begin with the simulation of the incident particles the detector is intended to observe. At European XFEL these incident particles will be X-ray photons and in X-CSIT these are simulated by the particle simulation. The particle simulation has the end goal of determining where energy from the incident particles is deposited in the sensitive semiconductor layers, via

an interaction that results in the energy freeing electron–hole pairs. This involves simulating the transport of each incident photon, their interaction with the detector and where they deposit energy, as well as the same for any secondary particles produced.

The particle simulation uses Geant4 [3, 4], a Monte Carlo particle simulation program originally designed for use in high energy physics. It simulates the passage of particles and photons through matter and interactions with a detector environment, including the non-sensitive components of a detector[1]. Geant4 simulates the interactions they undergo and any secondary particles produced. It is used in a wide variety of applications, including high energy, nuclear and accelerator physics, medical and space science. Using it in the hard X-ray energy region requires use of the Livermore [105] low energy electromagnetic physics process package; the $\sim$10 keV range is below what the regular high energy physics packages are tuned to simulate. Even then, the energy range used by X-CSIT is on the lower limit of where the Livermore physics package is accurate. However Geant4 simulates all of the physical effects, described in Section 2.1, that photons can undergo in silicon, as well as the interactions of secondary photons and electrons.

Geant4 simulates each particle individually in turn, and therefore only simulates the interactions of the particles with the detector environment, not other particles. Geant4 simulates the movement and interactions of particles via steps. A simplified version of this step method is that for each physics process that the particle could undergo, a function calculates a step length based upon the mean interaction lengths for the particle in consideration. This step length is the distance that the particle would travel before undergoing that process, if the process is selected. Whichever process calculates the shortest step length is then invoked for the step, the particle is transported the proposed step length and undergoes any relevant continuous process, then the physical process is simulated. This is shown in Fig. 6.1,

[1] This is important mostly for the upper energy range of photons expected to be used at European XFEL and other FELs ($\sim$20 keV), which will frequently penetrate the sensitive semiconductor layer, and may then interact with the rest of the detector in a manner that produces particle backscatter onto the semiconductor.

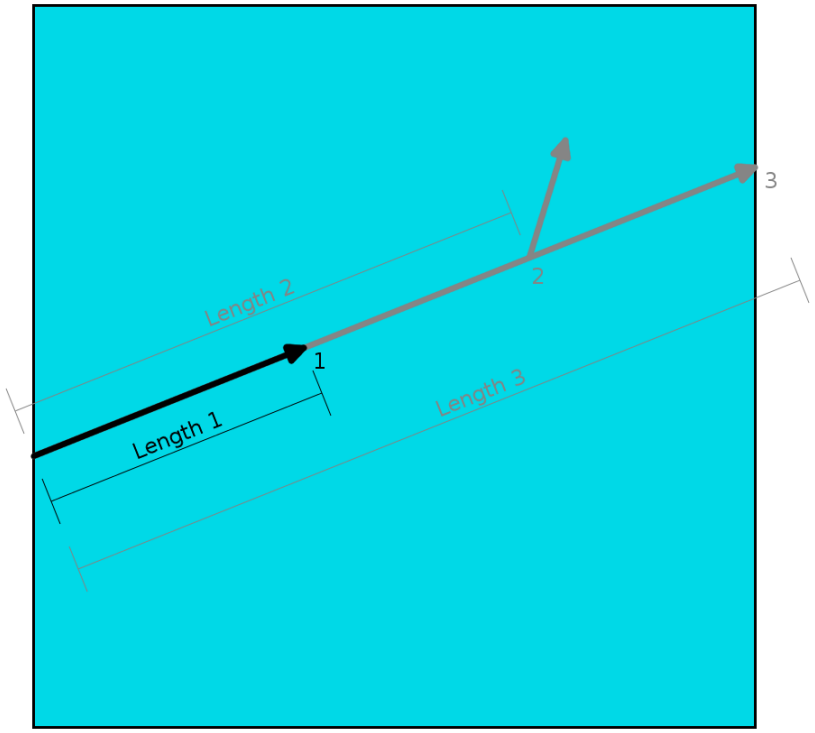


Figure 6.1: Diagram showing the Geant4 step decision process. Here three processes are proposed for a photon: absorption (1), scattering (2) and movement to the edge of the current volume (3). The absorption process generated the shortest step distance, so was chosen.

where three processes propose a step length and the process with the shortest step length is chosen. Any secondary particles are added to the top of the particle stack[2], unless Geant4 deems them to be below the production threshold.

Geant4 does not have low energy cutoffs, all particles that are injected or created are tracked down to zero range or kinetic energy. Instead Geant4 uses production thresholds, measured in a range. Secondary particles are not produced if their range is calculated to be less than the production threshold, and are instead integrated into the energy deposit. This places the accuracy of the simulation on position over energy, ensuring the accuracy of the location a particle stops at or

[2]This means that Geant4 will simulate all secondary particles created by a primary particle, in order of last created to first, before moving on to the next primary particle. The same rule applies for any tertiary particles, and so on.

deposits energy. This focus on the accuracy of location is useful for many applications and X-CSIT is no exception, as Geant4 is used here to determine the locations, down to the micrometre, where energy is deposited.

Geant4 builds a description of the physical simulation environment by placing volumes in a world environment. These volumes are defined in a three step process: a solid volume, a logical volume and a physical volume [106]. The first step, defining a solid volume, creates a three dimensional shape from a list of available Geant4 shapes, with defined exterior dimensions. The available shapes available in Geant4 includes boxes, trapezoids, cylinders, tubes, cones and spheres. Note that the name solid is a technical term relating to the solid modeling of the detector elements, the solid volume does not have any material or state associated with it. The next step is to define a logical volume. This manages the properties associated with a detector element, most importantly the material it is made from and a solid volume to define its shape. Solid volumes have a one-to-many relationship to logical volumes, any number of logical volumes can use the same solid volume to define their shape and size. Logical volumes therefore define the logic of how a detector element interacts with the rest of the simulation. The final step in creating an element of a detector is the physical volume. This places an instance of a logical volume inside the detector environment, defining what other, already existing logical volume the new volume will be placed within, along with its position and rotation. The only physical volume without a mother volume (which must entirely contain the child volume) is the world volume, within which all other volumes are contained. A logical volume can be placed by any number of physical volumes.

To extract energy deposition data from Geant4, X-CSIT makes use of sensitive detectors. Sensitive detectors are Geant4 classes that allow the user to perform code when particles interact in logical volumes the sensitive detectors are attached to, and can be thought of as more powerful and adaptable versions of a monitor. Because they have access to the particle and interaction data they can be used to perform output of interaction information. X-CSIT uses a sensitive detector to extract data when energy is deposited in the semiconductor layer, specifically the energy, loca-

tion and time of the energy deposit. In setups where multiple modules are simulated, the copy number of the volume is also recorded[3]. This data is recorded in Geant4 hits classes, at the end of the run the hits are read and added to X-CSIT data classes so they can be handed to the charge simulation. However, not all of the physical volumes created in Geant4 will have sensitive detectors attached to them. Particularly with the higher energy X-ray photons expected at European XFEL and other FELs ($\sim$20 keV), photons will frequently penetrate the sensitive layer, and may then interact with the rest of the detector in a manner that produces particle backscatter onto the semiconductor layer. Many detectors also make use of an entrance window on front of the semiconductor. This is intended to reduce background events at different energies to what the detector is designed for, and is therefore usually transparent to event photons, but a full simulation of the detector will often include this for completeness.

Fig. 6.2 shows a visualisation of the first five photons of a test run using Geant4 in the particle simulation, and Table 6.1 shows the output from the particle simulation corresponding to those photons. Photons were fired into the detector in a line at $x = 0$, spaced 1 mm apart from each other in the y axis. Only four of the five photons shown in Fig. 6.2 are absorbed by the detector, and energy deposits from them can be seen in Table 6.1. The coordinates given in Table 6.1 are relative to the centre point of the detector, therefore here the z axis covers 250 µm to -250 µm. From the lack of energy deposit it can be determined that photon three scattered elastically before it was absorbed. Each of the four photon events are seen to first deposit energy of either 1828.5 eV or 151.55 eV, although X-CSIT does not record the nature of the event that causes the deposit, these energies are the K shell and L shell binding energies. The four photon absorptions are therefore most likely photoelectric effect absorptions, and the subsequent deposits the result of secondary electrons.

[3]Where detectors use multiple identical modules, the modules are only defined once in Geant4. They are then placed in the world multiple times with each copy identified by the copy number. The sensitive detector is attached to the volume definition and therefore only a single sensitive detector is required for the entire detector.

Table 6.1: X-CSIT particle simulation output corresponding to the photons seen in Fig. 6.2. The original photon number is not part of the data, and is added only for clarity.

Photon No.	Energy (eV)	X-coord (μm)	Y-coord (μm)	Z-coord (μm)
1	1828.5	0	11750	6.60387
	2993.1	−0.403379	11749.7	6.15236
	3852.02	−0.728103	11749.7	6.4521
	2275.11	−0.702772	11749.7	6.46825
	1051.27	−0.408379	11749.7	6.16281
2	151.55	0	10750	165.234
	1838.33	−0.271045	10749.8	164.994
	8929.52	−0.436439	10749.8	164.642
	1080.6	−0.276278	10749.8	164.995
3	1828.5	18.7938	9807.07	71.2727
	10171.5	19.0373	9807.26	71.6543
4	1828.5	0	8750	−140.12
	10171.5	0.441462	8749.86	−140.284

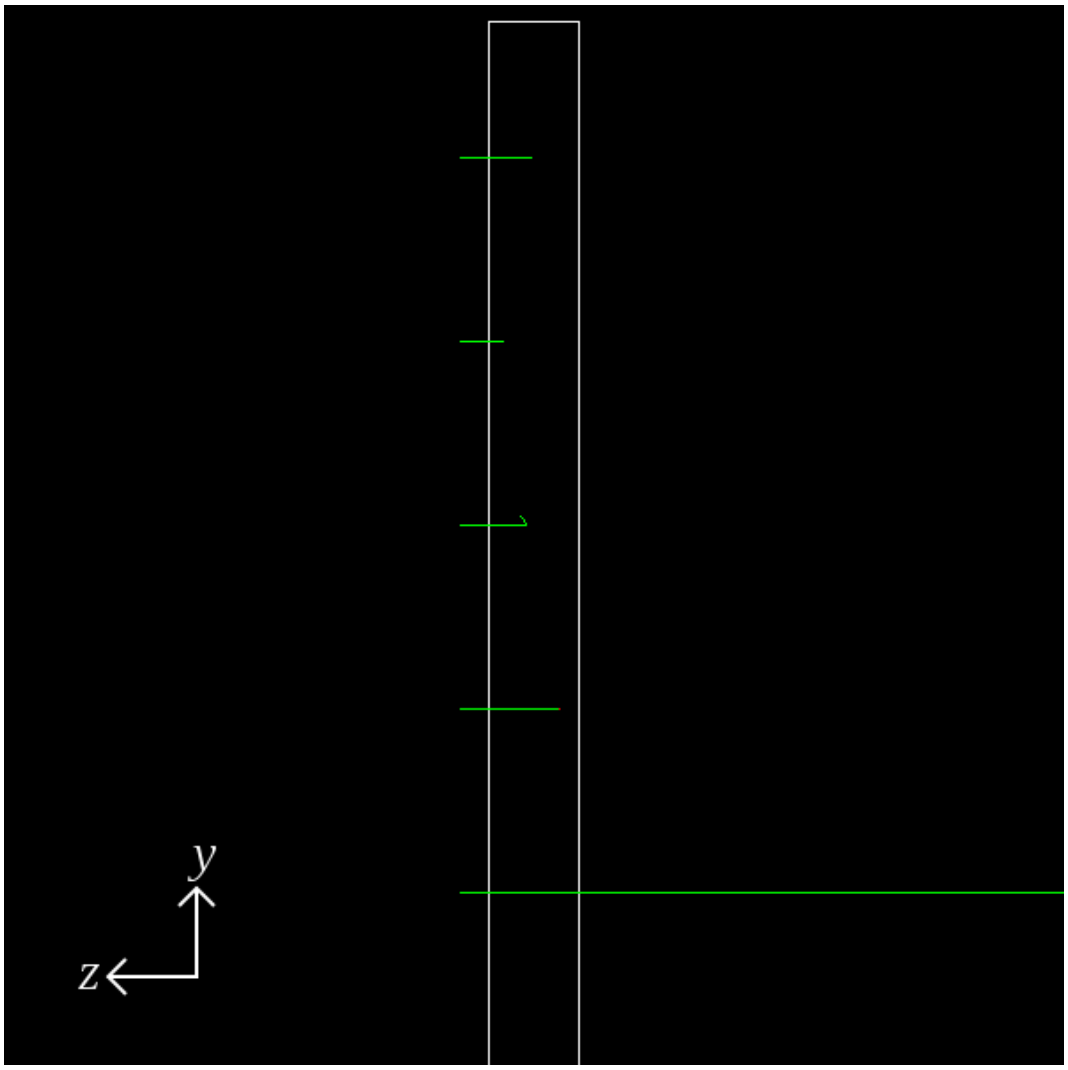


Figure 6.2: Geant4 visualisation of the X-CSIT particle simulation. The first five photons of a test are shown in green, fired into the detector from left to right, spaced 1 mm *apart from each other. The first four photons are absorbed, while the fifth penetrates the detector. The third photon is deflected elastically before it is absorbed. The red dot at the end of the fourth photon is the charged secondary electron that only travels far enough to colour one pixel of this image.*

6.2 Charge simulation

The charge simulation conducts simulation of the mobile charge clouds in the semiconductor layer. This process starts with charge cloud creation, based upon the energy deposit data provided by the particle simulation. The charge simulation then simulates the passage and expansion of the charge clouds as they are moved to the collection side of the detector by the electric field. The culmination determines how much charge will be collected by each pixel of the detector. The charge simulation does not attempt to simultaneously simulate each mobile charge and the interactions between them as the charge cloud moves. Instead the simulation characterises the spread of the charge cloud as a distribution function. Then charge can be distributed between pixels based upon how much of the distribution overlays each pixel. The whole simulation is conducted in three stages, each of which are modularised so that improved models can be inserted easily. These stages are: charge generation, spread size calculation and charge collection.

The charge generation stage simulates the conversion of an energy deposit into a large number of electron–hole pairs. However not all of the energy will be used to free electron–hole pairs and as a result the number of electron–hole pairs produced can vary from the average. The first step of the charge simulation is to calculate how many electron–hole pairs are produced by an energy deposit given this variation. In truth this process does not occur instantaneously, nor at a single point in space. Instead it is more of a shower as ever larger numbers of lower energy particles are produced. However it is assumed by the charge simulation that both the time this shower takes and the range of it are small when compared to the spread of charge that then takes place, and therefore can be considered point like.

In silicon an average of one electron–hole pair is generated per 3.65 eV of deposited photon energy. This is larger than the 1.12 eV band gap of silicon because energy is ‘lost’ to phonon creation. Because not all of the energy will be used to generate electron–hole pairs, there will be a statistical variation on the number of electron–hole pairs produced by a given amount of deposited energy. If each electron–hole generating event were independent then the distribution would be

described by Poisson statistics. Here the variance σ_p^2 would be equal to the mean μ

$$\sigma_p^2 = \mu, \tag{6.1}$$

where the mean here is described by the energy deposit E_D and the average pair creation energy $\overline{E}_{pair} = 3.65\,\mathrm{eV}$:

$$\mu = \frac{E_D}{\overline{E}_{pair}}. \tag{6.2}$$

Because each electron–hole pair created from a single deposit are obviously not uncorrelated, they are also not independent. The variance, σ^2, is therefore lower than Poisson statistics would predict, but is still related to the mean. Here we introduce the Fano factor [54], F,

$$F = \frac{\sigma^2}{\mu}. \tag{6.3}$$

The Fano factor is a number between 0 and 1 describing the variation relative to Poisson like statistics, 0 being no variation and 1 being Poisson like. Experimentally the Fano factor is found to be about 0.118 [58] in silicon. X-CSIT therefore calculates the number of electron–hole pairs generated by an event, N_{pair}, randomly according to a normal distribution with standard deviation σ,

$$\sigma = \sqrt{\frac{FE_D}{\overline{E}_{pair}}}. \tag{6.4}$$

The number of electron–hole pairs generated by each energy deposit is calculated sequentially by a random number generator.

After the number of electron–hole pairs in a spread has been calculated, the charge spreading simulation switches to a statistical simulation instead of a Monte-Carlo simulation. This is because there are a large number of particles ($\sim$1000s) compared to the potential number of final states (collection on 1 to 4 pixels with European XFEL detectors, with collection on 3 and 4 pixels being fairly uncommon). Therefore the simulation calculates the average amount of charge that will be collected by each pixel and uses that for the output.

The charge cloud is assumed to have a Gaussian profile, as is to be expected from a diffusion dominated spread. The next step of the simulation is then to calculate the size of this Gaussian distribution of charge.

The calculation used is based on work done by R.F. Fowler [66], who started with a description of the electric field in a semiconductor detector, $E(x)$, as a function of depth, x, in the sensor

$$E(x) = \alpha + \beta x \tag{6.5}$$

where α and β are positive constants. By deriving the drift velocity from this, integrating over time and then multiplying by the diffusion constant, he gets

$$\sigma^2(d) = \frac{2k_B T}{q\beta} \log\left(1 + \frac{\beta}{\alpha} d\right) \tag{6.6}$$

where k_B is the Boltzmann constant, T the temperature and q the electric charge. The quantity d is the depth of the interaction, which is defined for a front side illuminated detector where electron collection is on the side of the semiconductor facing the incoming photons. A better definition including rear side illuminated detectors is that d is the distance in the direction of the electric field that charge must travel to be collected.

In the case where $\beta \ll \alpha$, where the applied field is much stronger than that due to doping, we get

$$\sigma(d) = \sqrt{\frac{2k_B T}{q\alpha} d}. \tag{6.7}$$

This case is true for almost all fully depleted sensors, where the applied voltage is more than exactly what is required to deplete the sensor.

This is the equation that X-CSIT uses to calculate the size of the charge spread. However as pointed out by Fowler, it fails to account for any diffusion along the direction of travel or the effect of the charge clouds on the field or themselves. Fowler found that for depths lower than 100 µm, Equation 6.7 did not give a good fit, but gave a reasonable approximation beyond 100 µm. This poses a problem for simulations of front side illuminated detectors, such as most Charged Couple Devices (CCDs), however the hybrid pixel detectors at European XFEL and pnCCD are rear side illuminated. With a rear side illuminated detector, interactions are more likely to occur with large depths. Therefore the majority of interactions will fall into the > 100 µm region of acceptable approximation. This is especially the case with

thicker detectors and when used with lower energies (in the hard X-ray range, lower energy photons have shorter interaction lengths in silicon).

Still, even though the model used may have a good approximation for the majority of interactions, this area of the simulation is a candidate for improvement. Initially this will consist of tweaking normalisation parameters to match data. This will, particularly with the high depth interactions most commonly seen in European XFEL detectors, give an improved spread equation that is on average largely correct. This should be sufficient for whole detector simulations at European XFEL and other similar experiments, but will still lack accuracy for edge cases. Low depth interactions will likely remain inaccurate and therefore the simulation will probably be less accurate for front side illuminated detectors. Additionally the assumption of $\beta \ll \alpha$ is only true for fully depleted sensors. Even a small region of the sensor remaining undepleted will render α negative in the depleted region. Because α is approximately proportional to the applied voltage[4] it can be approximated with only the bias voltage. At lower bias voltages and in the non-fully depleted case a full understanding of the sensor doping and electric field is required.

The third stage of the charge diffusion simulation is to take the charge spread and electron–hole pairs generated beforehand and distribute them to pixels. Having calculated the size of the spread once it reaches the collection side of the silicon, the charge collection stage overlays this on a pixel grid. X-CSIT does not simulate the real positions of collection bonds, or potential wells. Instead it approximates them by drawing an imaginary line midway between pixel bonds that forms the pixel boundary. Pixels are then assigned charge according to how much of the calculated charge spread lands within the line drawn by the pixel boundaries. The approximation is therefore that charge landing within the pixel boundary will be collected by that pixel.

To calculate the charge landing on the opposite side of a pixel boundary to the interaction point, and therefore the amount of charge that is spread, we start with a two dimensional Gaussian probability distribution in space, seen in Fig. 6.3. It

[4]It is proportional to the applied voltage minus the voltage required to deplete the sensor.

has a centre point at (x_0, y_0) and a standard deviation of σ_x and σ_y. In the case of a symmetrical spread, such as occurs when charge spreads, these are equal. The spread therefore has the form

$$f(x,y) = \frac{1}{2\pi\sigma^2} \mathrm{e}^{-\left(\frac{(x-x_0)^2}{\sigma^2}+\frac{(y-y_0)^2}{\sigma^2}\right)} \tag{6.8}$$

which can also be written as

$$f(x,y) = \frac{1}{\sqrt{2\pi}\sigma} \mathrm{e}^{-\frac{(x-x_0)^2}{\sigma^2}} \frac{1}{\sqrt{2\pi}\sigma} \mathrm{e}^{-\frac{(y-y_0)^2}{\sigma^2}} \tag{6.9}$$

and can therefore be seen to be a normalised distribution. To calculate the proportion of this spread, X_1, that is on the left hand side of a line at x_1, shown in Fig. 6.3, we must integrate the distribution from $-\infty$ to x_1 in x and from $-\infty$ to ∞ in y:

$$X_1 = \int_{-\infty}^{x_1} \int_{-\infty}^{\infty} f(x,y)\,\mathrm{d}x\,\mathrm{d}y \tag{6.10}$$

and substituting in Equation 6.9

$$X_1 = \int_{-\infty}^{x_1} \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma} \mathrm{e}^{-\frac{(x-x_0)^2}{\sigma^2}} \frac{1}{\sqrt{2\pi}\sigma} \mathrm{e}^{-\frac{(y-y_0)^2}{\sigma^2}}\,\mathrm{d}x\,\mathrm{d}y \tag{6.11}$$

$$X_1 = \int_{-\infty}^{x_1} \frac{1}{\sqrt{2\pi}\sigma} \mathrm{e}^{-\frac{(x-x_0)^2}{\sigma^2}}\,\mathrm{d}x \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma} \mathrm{e}^{-\frac{(y-y_0)^2}{\sigma^2}}\,\mathrm{d}y. \tag{6.12}$$

In Equation 6.12 it can be seen that the second half of the equation is equal to 1, since it is a normalised function being integrated over all space. The first half of the equation is evidently the cumulative distribution of the single dimensional case of the normal distribution. This can be calculated using an error function and the quantity $x_1 - x_0$. This function does not rely on $\sigma_x = \sigma_y$, however when it is true the spread is rotationally symmetric and the choice of axes becomes irrelevant, therefore the use of the single dimensional cumulative distribution is true for any line.

This use of the single dimension cumulative distribution function works with the assumption of pixel boundaries. Where all of the charge can be assumed to land in either of two adjacent pixels, the distribution can be calculated using a single error function. This method can also be used to calculate spread into a pixel diagonally adjacent to the pixel a spread originates in. This is shown in Fig. 6.4, where we

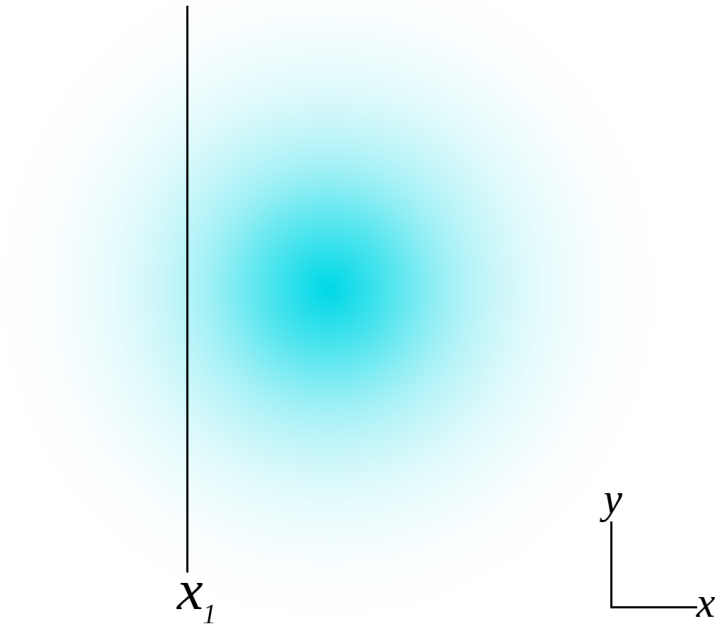


Figure 6.3: Diagram of a spread function centred at (x_0, y_0) with a line at x_1.

want to calculate the amount of the spread that lands in the region bordered by lines at x_1 and y_1, here the desired integration is

$$\int_{-\infty}^{x_1}\int_{-\infty}^{y_1} f(x,y)\,\mathrm{d}x\,\mathrm{d}y \tag{6.13}$$

which is equal to

$$\int_{-\infty}^{x_1} \frac{1}{\sqrt{2\pi}\sigma} \mathrm{e}^{-\frac{(x-x_0)}{\sigma^2}}\,\mathrm{d}x \int_{-\infty}^{y_1} \frac{1}{\sqrt{2\pi}\sigma} \mathrm{e}^{-\frac{(y-y_0)}{\sigma^2}}\,\mathrm{d}y. \tag{6.14}$$

Equation 6.14 is a now the multiplication of two separate, single dimensional cumulative distributions. Therefore charge spreading from a pixel into its eight neighbouring pixels can be calculated using four cumulative distributions, shown in Fig. 6.5 as X_1, X_2, Y_1 and Y_2, and subtraction of these values. This assumes that the charge spread will always be smaller than the pixel grid size, as any charge spreading over the pixel boundary is automatically assumed to be collected in the next neighbouring pixel. It would be possible to expand this method to a larger grid size, but it is not expected to be necessary when simulating the large pixels of the European X-ray Free Electron Laser (European XFEL) detectors.

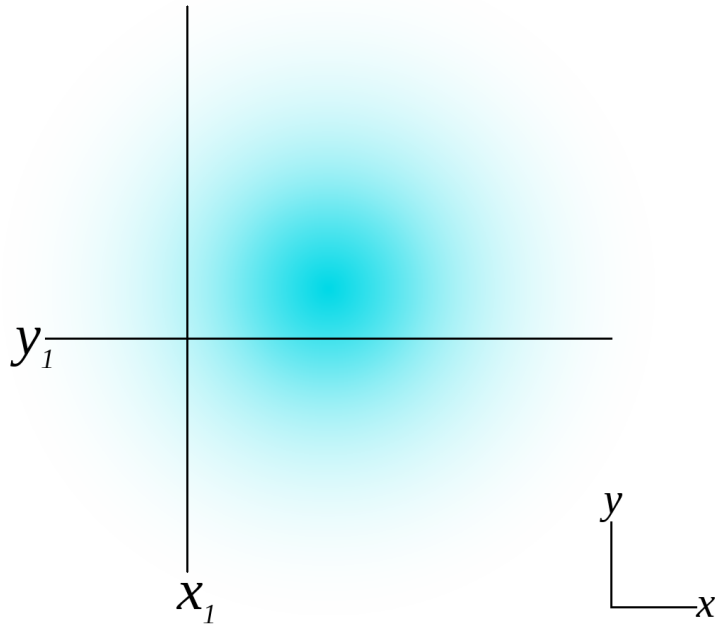


Figure 6.4: Diagram of a spread function centred at (x_0, y_0) with corner at the intersection of lines at x_1 and y_1.

A demonstration that the charge simulation is correctly applying Equation 6.7 and Equation 6.12 can be seen in Fig. 6.6. Here 1000 charge events have been inserted into a semiconductor detector 250 µm from the collection side, at varying distances from a pixel boundary. The proportion of charge that was not shared into the neighbour is shown by the black points on the plot. The red line on the plot is the manually calculated plot of Equations 6.7 and 6.12 that the spread pattern should follow.

The above method is based upon that used by X-CSIT's predecessor the HPAD Output Response fUnction Simulator (HORUS), although in code it has been expanded to include catch functions so charge is not spread into non-existent pixels. An alternative method using a lookup table and pre-calculated spread was developed by R. Ayers [107]. This lookup table replaces the error function, and contains the proportion of charge in one of the quadrants of a pixel corner. This value is stored for corner positions relative to the interaction in units of standard deviation. By using units of standard deviation the spread can be scaled to charge spreads of

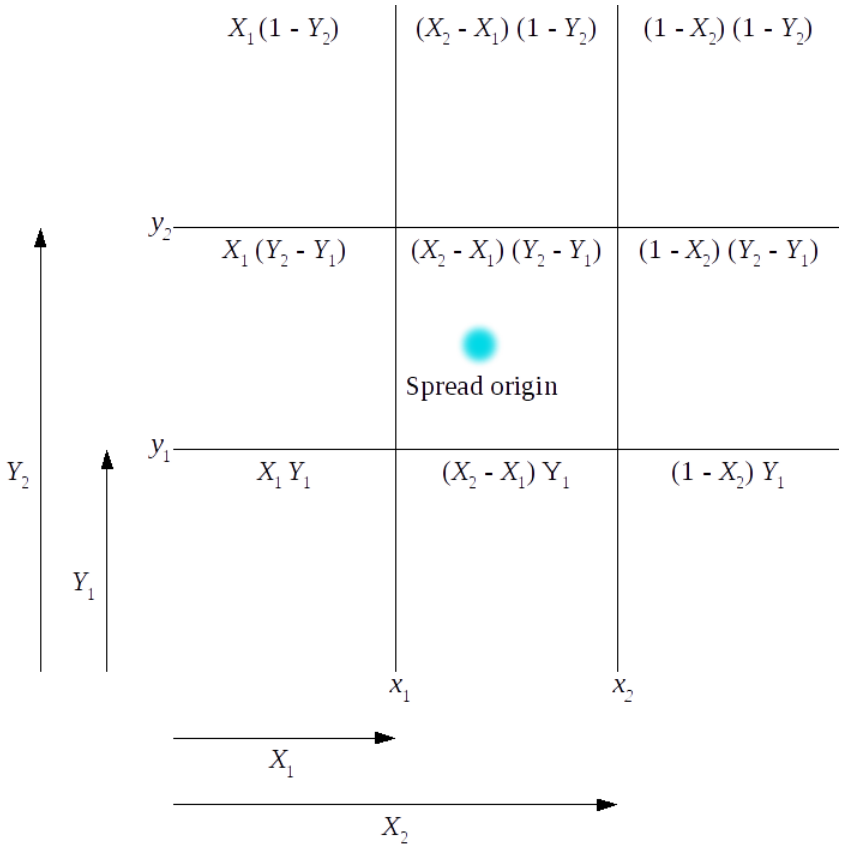


Figure 6.5: Diagram showing a nine pixel grid, four cumulative distribution functions up to the lines at X_1, X_2, Y_1 as Y_2, and the multiplication and subtraction of these results that give the proportion of charge in each pixel.

any size. To go with the lookup table method, Ayers also developed an algorithm to use this lookup table to calculate spread over any number of pixels. Because the lookup table contains proportions for a quadrant, it calculates charge for a pixel as the charge in a quadrant that includes that pixel, minus the charge in all other pixels that quadrant contains. By working from the furthest charge receiving pixel (the lookup table has a cutoff at 5σ) inwards, this method can calculate charge deposits for all affected pixels[5]. Using the symmetry of a square it is possible to rotate and flip a distribution such that only one eighth of it is required, the method implemented by Ayers only flips the distribution in one axis, and therefore requires half

[5]This includes the pixel the interaction lands in, which is then the remaining charge not in other pixels.

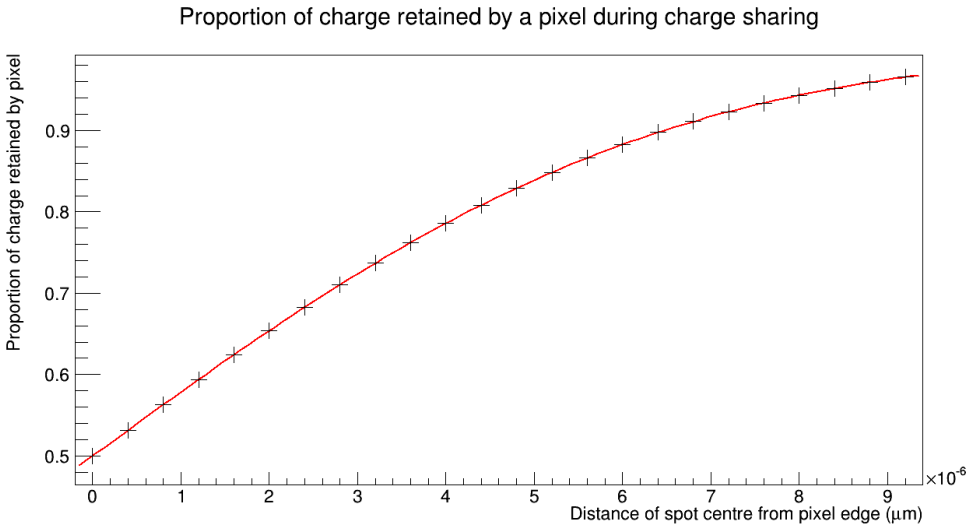


Figure 6.6: A plot of charge not shared into a neighbouring pixel, by an energy deposit at various distances from the pixel boundary. Crosses show results from X-CSIT, and the red line shows the spread pattern X-CSIT should follow. The energy deposit was placed 250 µm *from the collection surface of the detector, which was at room temperature with a* 250 V *bias voltage.*

of the distribution. This algorithm is of particular interest for detectors with small pixels and simulation of plasma effects. For the detectors at European XFEL it is not expected that individual charge events will ever spread over their large pixels sizes. Additional to this is that the lookup table, while implemented with the same equation as the HORUS like method, can contain any shape of distribution. It is therefore more readily expanded when improved spread functions are developed. It will also be the only option when developed for the hexagonal pixels of DSSC, as without perpendicular pixel boundaries the cumulative distribution cannot be simplified to the single dimensional equation.

6.3 Electronics simulation

The readout electronics of a detector, that takes the charge collected from the semiconductor layer and converts it to a readable signal, vary between detectors. The electronics simulation of X-CSIT attempts to provide a simulation of all possible detectors by applying the actions of circuit components or effects sequentially. By

splitting a readout circuit into a sequence of components and functions, such as amplification, digitisation and noise, X-CSIT can provide simulations of each of these components. These components can then be assembled together to simulate the full circuit. This method works because the detector circuits X-CSIT is designed to simulate do not operate continuously. They instead collect charge over a period of time, then count that charge. Circuits operating in such a counting mode can be simulated one component at a time. This allows for a much simpler electronics simulation than would otherwise be required.

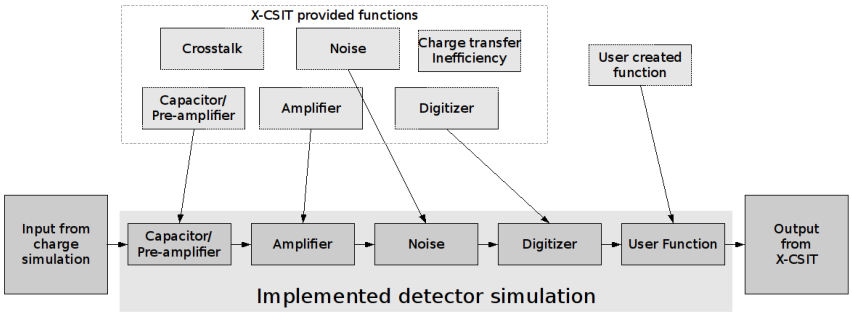


Figure 6.7: Diagram showing how an example electronics simulation of a detector is created by linking together simulations of electronic components. Only the functions required for the specific detector are used.

The electronics simulation provided in X-CSIT is therefore comprised of modules that each simulate an electronics component, or effect, in as much detail as required. A simulation of a circuit is then constructed by linking together modules, show in Fig. 6.7, each acting on a pixel matrix representing the status of the circuit. This module based approach gives the maximum freedom possible for simulating the wide variety of detector circuits in existence. Existing modules can be copied, modified or expanded upon, and new modules can be added without interfering with existing functionality. Examples of extending functionality include adding per-pixel electronic response, such as pixel or column/row specific noise response, or user set circuit variation. An example of the latter is Large Pixel Detector (LPD), where each pixel can be set to use either a 5 pF or 50 pF capacitor with the pre-amplifier.

The electronics simulation is therefore mathematically less complicated than the previous sub-simulations, but it is by necessity the most open to modification

and expansion. The electronics simulation is also where the X-CSIT data structures are most apparent in their use. Each of the modules within the electronics simulation directly acts upon data held in these classes. This also allows use of the data class expandability, such as with Adaptive Gain Integrating Pixel Detector (AGIPD) where after amplification a 2 bit number is stored alongside a signal to denote which gain rate was used. Other classes, such as noise and digitizers, can still be used with the signal without modification, as they are not affected by the additional data.

6.3.1 Amplification

Amplification is one of the critical components of a semiconductor detector. The signals produced by photon interactions are generally, with the exception of the most intense images, too small to use with a digitizer and must first be amplified. An additional factor is that while amplification will introduce noise to a signal, a well designed amplifier will keep this noise to a minimum, while increasing the signal relative to any other noise. Therefore while introducing noise, an amplifier can increase the signal to noise ratio.

In X-CSIT an amplifier is simulated as a multiplication of charge, plus the application of noise which is covered separately. The simple modules included with X-CSIT apply the same amplification to all pixels in a detector, although the nature of the electronics simulation is such that the user can modify this to apply gain on a per-pixel basis, such as directly from a calibration file.

6.3.2 Noise

The calculation of noise generated by the electronics of a detector and the simulation of the effect this has on the final image is an important function of X-CSIT. X-CSIT comes with modules for simulating noise as either an independent background process, or as one dependent on the signal itself. All methods use a normal distribution for generating noise values, with the user providing standard deviation values that are either flat or relative to the signal.

A demonstration of the noise module, as well as the electronics simulation, is seen in Fig. 6.8. Here a 4×50 pixel detector was given 30000 collected electrons,

equivalent to approximately ten 12 keV photons, in each pixel artificially so that only the electronics simulation had to be run. The electronics simulation for this detector was set to consist of a 50 pF capacitor to collect the charge, a 10× amplifier, and the noise on that amplifier. The noise was a random Gaussian addition to the pixel value, with a standard deviation of 4.8×10^{-5} mV, equivalent to 5% of a photon. The voltages of each pixel following the noise was then output to file, and the whole simulation was repeated 1000 times to increase statistics. The voltages of each pixel were then histogrammed in Fig. 6.8, showing that the electronics simulation is correctly performing actions on the pixel grid, as well as showing that the random Gaussian noise module is applying the correct amount of noise.

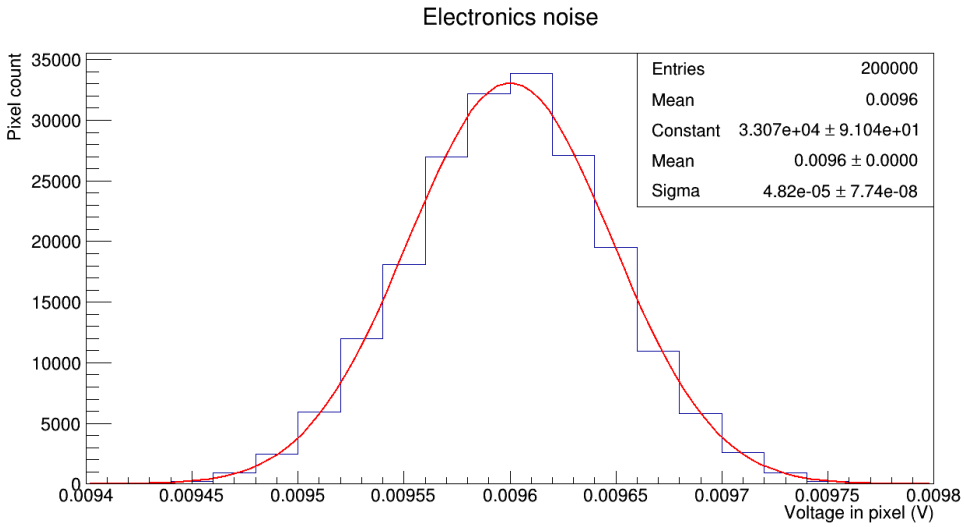


Figure 6.8: Histogram of the pixel voltage in a detector after noise has been applied. The first two entries in the statistics box relate to the histogram in blue, while the next three are the parameters of the fit to the histogram, shown in red.

Modules are also provided for both analogue signals and digital signals. Noise on the latter is of course unphysical as a digital signal should not suffer noise. However it can be useful to apply noise at the end of an electronics simulation, after a signal has been digitised, if a detector has been measured or calibrated with noise measured as an effect on the final signal. In this case the application of noise does not represent a detector process, but a simulation of how calibration data describes the detector performance. The modular nature of the electronics simulation allows

simulation both in this manner, and simulation of noise at the individual component level.

6.3.3 Digitization

The digitizer in X-CSIT performs a conventional analogue to integer conversion. It represents an ideal digitizer, in which each digital step in signal is equally spaced, from a zero value up to the maximum readout. The module takes parameters of the number of digitizer output steps and the maximum input signal. The number of digitizer output steps, or the output range, would be expected to be a power of two, but can be set to any positive integer. The maximum input signal is the smallest input signal that should result in the maximum input signal, as any input signal in excess of the input range would also be expected to result in a maximal output signal. The digitizer function therefore calculates the output signal as

$$Out = In * (OutputSteps - 1)/InputRange \tag{6.15}$$

where Out is the output signal, In is the input signal, $OutputSteps$ is the number of digitizer output steps and $InputRange$ is the maximum input signal the digitizer can read.

The modular nature of the electronics simulation would make the creation and addition of non-linear digitizers, such as the differential nonlinearity suffered by successive approximation Analogue to Digital Converters (ADCs), relatively simple. This is not included in the current simulations as no calibration data from the real detectors is currently available. European XFEL intends to integrate X-CSIT with the Karabo calibration database [99] for the detectors, which should include data on digitizer nonlinearity.

6.3.4 Charge transfer inefficiency

Charge Transfer Inefficiency (CTI) is a complicated effect occurring in CCDs, wherein a portion of charge collected by a potential well in the CCD is not moved to the next cell as would be expected. This charge is instead collected by subsequent pixels as they move through the potential wells of the detector. The process is complicated because this effect occurs principally because the charge becomes

trapped in lattice imperfections, caused either at manufacture or by radiation damage. Charge transfer inefficiency is therefore an effect that occurs while charge is still inside the semiconductor layer. However, because it affects the charge when it has already been distributed between pixels it was simpler to model it using the electronics simulation than to modify the charge simulation.

The Charge Transfer Inefficiency model used by X-CSIT was originally written by T. Rüter for a Karabo/Python based simulation of the pnCCD electronics, written before the electronics simulation of X-CSIT was complete. It was then re-written in C++ as a component for the X-CSIT electronics simulation. This model assumes that a fixed proportion of charge in any pixel will be lost when that pixel is moved from one potential well to the next. The simulation operates by moving each charge pixel through a potential well grid, to the output line, as it would do in the detector. At each step the charge pixel has any residue charge already in the potential well added to it, then leaves behind a fixed portion of its own charge when it is moved. Once a pixel has been moved through the potential well grid of all the pixels read out before it, the next pixel in line is moved.

The Charge Transfer Inefficiency (CTI) function of the electronics simulation is currently the only CCD specific simulation of X-CSIT. Although CCDs can also suffer from blooming, where a pixel is so filled with charge that it spills over into adjacent potential wells, as well as reduced electric fields as pixels fill with charge, the simulations of CCDs thus far have been at low enough flux that simulation was not deemed necessary.

6.3.5 Capacitor leakage

The detectors at European XFEL are required to capture images at a rate of 4.5 MHz during the beam train, then conduct output in the down time between trains. 4.5 MHz is too fast for the detectors to digitise the signals between frames, so the detectors instead store their amplified signals on capacitors. These capacitors are susceptible to capacitor leakage between frames, and therefore X-CSIT implements a simple capacitor leakage function. This takes a parameter of what proportion of charge will be lost from each memory cell, and applies it to the entire frame. Be-

cause this leakage is circuit and time dependent, calculation of the leakage is left to the user or calibration data.

6.3.6 Crosstalk

Crosstalk between pixels can occur in semiconductors, resulting in the signal from one pixel being transferred to neighbouring pixels. The electronics simulation of crosstalk described here is intended to simulate only crosstalk originating from electronic interference after charge has been collected. Crosstalk in detectors is heavily dependent on the design of the Aplication Specific Integrated Circuit (ASIC) chip of the detector, so the X-CSIT simulation was designed so that crosstalk between neighbouring pixels in the vertical, horizontal and diagonal directions can be applied differently, but simultaneously. The module applies crosstalk by running a line scan across the pixel grid, then calculating the amount of charge sharing that will occur between the considered pixel, and the pixels ahead in the line scan, below, and diagonally below. This method ensures that all pairs of pixels are only considered once, and charge transfer between the pixels can occur in either direction. The charge sharing is then applied to the grid after all pairs of neighbouring pixels have been considered. The example crosstalk algorithm included with X-CSIT shares a proportion of charge between neighbours relative to the difference in their signal, but this is intended to serve only as an example of how to set up the crosstalk function as no detector simulated in X-CSIT has required creation of a real crosstalk function.

A demonstration of the crosstalk function is shown in Table 6.2, which shows a unit test of an example crosstalk class, set to share 1% of the difference between neighbouring pixels into the pixel with the lowest value. Blue cells show where 100.0 units were added to the pixel prior to crosstalk, red cells show where 1000.0 units were added. Because this is a unit test unconnected to any other physical simulation the signals shown are unitless. Five tests are included on the grid: the first shows a pixel engaging in crosstalk with eight neighbouring pixels, the other four tests show two adjacent pixels holding signal engaging in crosstalk in the four possible types of adjacency, as well as pixels experiencing cross talk from multiple

adjacent pixels. Pixels with no initial charge have 1% of the charge each neighbouring pixel started with after crosstalk is complete. Red cells show pixels with 921.0 units, having transferred 70.0 units to neighbouring pixels with no signal, and 9.0 units to the adjacent pixel that started with 100.0 units. The first blue cell shows the expected 92.0 units, while the other cells show 102.0 units, having transferred 7.0 units to neighbouring cells with no charge, and received 9.0 units from the neighbouring pixel. This therefore demonstrates that the method of conducting crosstalk between pixels will only operate once for each pair of pixels, and that all pixels are considered before the effects of crosstalk, not after engaging in crosstalk with other neighbouring pixels.

Table 6.2: Output from a crosstalk unit test. Pixels in blue had 100 *units injected, pixels in red had* 1000 *units injected. The crosstalk simulation was set to move* 1% *of the difference between any two neighbouring pixel from the higher signal pixel to the lower signal pixel.*

0.0	0	0.0	0.0
1.0	1.0	1.0	0.0
1.0	92.0	1.0	0.0
1.0	1.0	1.0	0.0
0.0	0.0	0.0	0.0
10.0	11.0	11.0	1.0
10.0	921.0	102.0	1.0
10.0	11.0	11.0	1.0
0.0	0.0	0.0	0.0
1.0	1.0	1.0	0.0
11.0	102.0	11.0	0.0
11.0	921.0	11.0	0.0
10.0	10.0	10.0	0.0
0.0	0.0	0.0	0.0
10.0	10.0	10.0	0.0
10.0	921.0	11.0	1.0
10.0	11.0	102.0	1.0
0.0	1.0	1.0	1.0
0.0	0.0	0.0	0.0
0.0	1.0	1.0	1.0
10.0	11.0	102.0	1.0
10.0	921.0	11.0	1.0
10.0	10.0	10.0	0.0
0.0	0.0	0.0	0.0

6.4 Charge plasma simulation

6.4.1 Charge plasmas

Plasma effects occur in semiconductor detectors when the density of charge carriers becomes large in a small volume of the semiconductor. When the charge density becomes large enough the charge begins to screen itself against external fields, such as the electric field applied across the semiconductor to move the charge. This results in a charge plasma. While the outer edges of the charge plasma will continue to experience the electric field and gradually be stripped away, the centre of the charge plasma will not move under the effect of the external field. The charge cloud will however diffuse. This diffusion is initially thermal dominated, but as electrons and holes are separated from each other internal electrostatic repulsion will also contribute [108]. Because charge plasmas are high charge density events they are expected to be caused by high photon flux; the high count statistics should then lead the depth profile of the resulting energy deposit to be close to the statistical average. This will lead to the highest charge deposit densities occurring at the event facing surface of the semiconductor. In rear-facing detectors such as those at European XFEL this can lead to holes being collected faster than electrons, leading to the electrons experiencing repulsion-driven diffusion.

Plasma effects seen by a user of the detector result in an increase in the charge collection time [68, 109], an increase in the size of the charge spread [71, 110], and a decrease in the amount of charge collected due to recombination of electron–hole pairs [69]. The increase in charge collection time and recombination of charge occurs because of the time taken for the charge plasma to disperse due to diffusion and the stripping away of the outer edges. The increase in charge collection time also contributes to the increase in charge spread via thermal diffusion, but electrostatic repulsion due to the large charge density is also a contributor [71, 108]. All of these effects are relevant to detectors at European XFEL. The charge spread is evident as it will cause greater amounts of charge sharing between pixels, and recombination because of the decrease in signal collected. The increase in charge collection time is also relevant to European XFEL because the bespoke detectors will operate

at 4.5 MHz. The 222 ns gap between frames can be less than the diffusion time of charge plasmas and not be enough for full charge collection. This would lead to charge sharing not only between pixels, but also between frames taken by a detector.

The plasma effect simulation is an extension to the charge cloud simulation described in Section 6.2. The charge cloud simulation assumes that each charge cloud event acts independently and does not interact with other charges in the semiconductor. Plasma effects occur when charge densities become large enough for intercharge interaction to become the dominant factor. Therefore the plasma effect simulation needs to be a component of the charge simulation, taking over from the diffusion based simulation where plasma effects occur. This therefore leads to a two stage simulation of plasma effects. The first is to identify what regions of the semiconductor need to be simulated by the separate plasma effects simulation, and the second is to then run the two parallel simulations, plasma and diffusion, in their separate regions. The diffusion simulation must also be fully functional without the presence of a plasma simulation. If a user does not expect to see plasma then they should be able to run without a simulation of it and default to the diffusion model. This is achieved with a plasma search function that identifies regions of the detector where plasma effects will occur and dominate. This then produces a list of the locations where plasma effects occur that the diffusion simulation should skip when running. In the absence of the plasma location list the diffusion simulation does not ignore any charge and runs as usual.

6.4.2 Data and experimental tuning

Experimental requirements

The plasma effects hypothesised to occur at European XFEL require a high number of photons, applied over a small area and in a short time. Estimates based on Becker's work [71] indicate the required photon count per plasma event to be

in the several thousands region[6] This high photon count then needs to be applied onto an area of approximately 1 µm square. In the absence of a charge plasma, the charge produced in a semiconductor will diffuse and be collected in the order of 10 ns[7]. Producing a charge plasma requires applying the high photon count before the diffusion can occur significantly. The plasma generating photon beam therefore requires a sub nanosecond time frame, and any sort of continuous source would be unsuitable.

No source able to match all three requirements and produce charge plasmas was available to use over the course of this PhD. It was recognised that this would be the case while work on the plasma simulation was underway, but before it was finished. The charge plasma simulation therefore currently remains at a prototype stage, untested on data.

However, plasma effects are still expected to be seen at European XFEL, a plasma simulation is still desired and it is hoped that after this PhD it will be possible to take the required data. As part of the commissioning process of European XFEL itself, beam time will be made available to European XFEL work groups and European XFEL is a source capable of producing charge plasma effects. Therefore this section contains details of the plasma simulation prototype. This consists of the workings and theoretical basis of a model to determine where and when charge plasmas occur, as well as an experimental design to take the data required to tune the charge plasma simulation. A simulation of the effects of charge plasmas, including charge spread, capacitive coupling crosstalk [111], recombination and time delay, was intended to be derived from the results of the proposed experiment, and is thus not presented.

[6] With 660 nm light, an equivalent of 4.68×10^4 1 keV photons in a spot with ≈ 3 µm produced noticable deviation from lower intensities. This equates to about 1.28×10^7 electron hole pairs. 660 nm light and 1 keV photons have an attenuation length of ≈ 3 µm. This gives an electron–hole density estimate of $\approx 6.46 \times 10^4$ µm^{-3} within the first σ of the spot and the first attenuation length.

[7] 12 ns to 30 ns for holes in a 450 µm thick detector and bias voltages of 500 V to 200 V [71].

Proposed experiment

Data is required to tune two aspects of the charge plasma simulation: The minimum charge plasma and the plasma spread function. The first quantity, the minimum charge plasma, is the quantity required to tune the plasma search module. This is the smallest charge plasma, by total energy deposit, that is noticeable from regular diffusion. This is the cutoff value that the plasma search function should use to determine where to simulate plasma effects. The second quantity is the plasma spread function. This function quantifies the size of the charge spread produced by a charge plasma, once the plasma has dissipated and the charge is collected. This is expected to be a function of charge, or deposited energy. The two quantities are of course linked, as the minimum charge plasma should coincide with the point where the plasma spread function merges with the diffusion prediction[8] (see Fig. 6.9). It should therefore be possible to predict the minimum charge plasma from the plasma spread function, however it would be best verified experimentally.

Using a suitable source, capable of generating a charge plasma, an experiment can be conducted with a detector to try and determine both the minimum charge plasma and the plasma spread function. The assumption of a Gaussian like spread shape will make the experiment significantly simpler (as was found in [71]).

In the experiment, a photon source is used to generate a charge plasma in the semiconductor a known distance from a boundary between two pixels. When a charge plasma is generated and collected, the proportion of charge in the two pixels is observed. A plot is then made of the distance from plasma origin to boundary, versus the charge proportion (Fig. 6.10). At the spot origin[9] on this graph, charge proportion will obviously be 50% (so if a boundary were placed there, 50% of the charge would land in each pixel), and at the known distance of the spot from the

[8]Concentrations of charge should experience some repulsion-driven spread increase, even if a charge plasma is not formed. Unsymmetrical collection of charge from the semiconductor would enhance this effect by giving the concentration a net charge. An experiment to tune the plasma simulation would also be able to test whether this effect can produce noticeable changes.

[9]This thought experiment considers a spot size that is small compared to the spread size. Chapter 7 discusses a similar experiment in the non-plasma regime which does account for beam size.

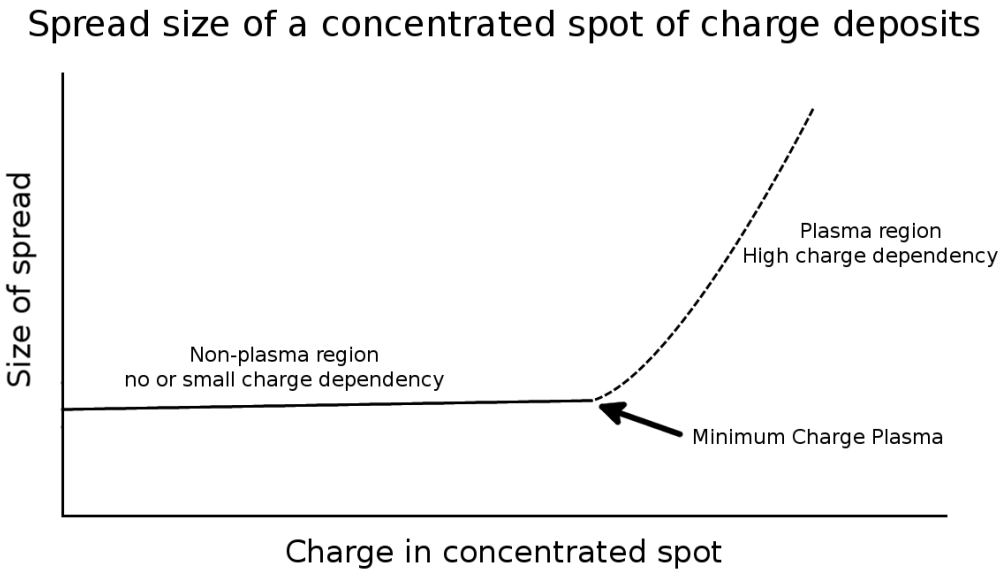


Figure 6.9: The expected relation between charge sharing and charge amount in a concentrated charge deposit, e.g. as a result of a diffraction spot. Below charge plasma levels the spread size will increase slowly with charge, possibly too slowly to notice. The relationship under plasma conditions is unknown, as is the boundary between the two.

pixel boundary the proportion observed in the pixels is plotted. From these two points, the graph of the cumulative distribution of a Gaussian function can be fitted. From this fit, finally the size of the initial spread can be determined. In this case it would also be possible to skip the fitting on a graph and re-arrange the equation of the cumulative distribution function with a known x and solve for σ. This value of σ will be for a specific value of C, the total charge deposited into the charge plasma. It should be noted that due to recombination of electron–hole pairs, C will not be the charge collected by the sensor, and should more accurately be calculated from the photon flux. These values can then be plotted to find the relationship between the charge spread and the amount of charge in a plasma. The minimum charge plasma should be found at the lower bounds of intensity. This experiment therefore requires a source that can be varied in its intensity. If using an X-ray source with

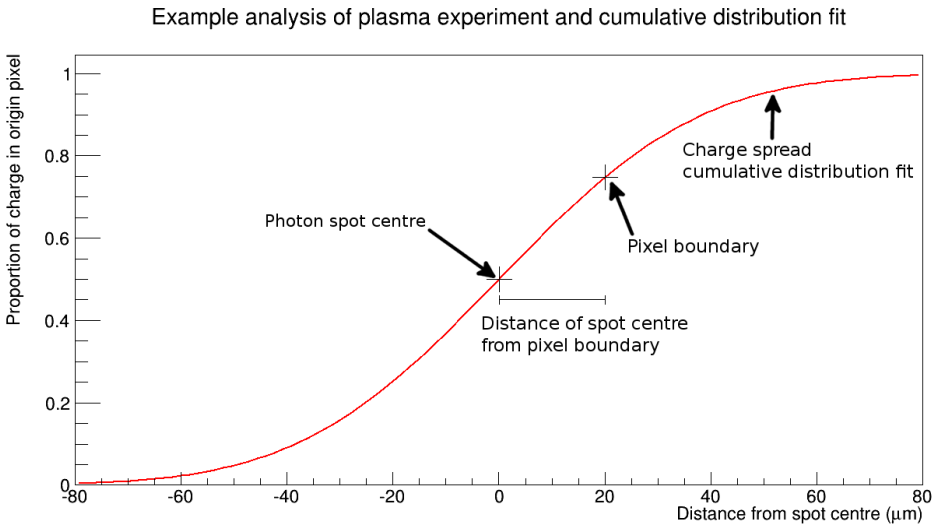


Figure 6.10: Cumulative charge distribution from a plasma. The graph is centred at the photon spot centre which is known to be the centre of the distribution. The proportion of charge not shared to the neighbouring pixel is plotted the known distance away and the cumulative distribution is fitted to these points. From the fit the size of the spread is derived.

consistent photon output[10] a range of intensities can be selected to find the plasma spread function and bring down statistical uncertainties on each data point. If using an XFEL then each event will have a different intensity and value of C that will need to be derived from the sum of charges in the two pixels. This experimental setup has the weak point that the exact distance between the spot centre and the pixel edge must be known. This is difficult to measure, as it usually requires positioning the spot directly over a pixel boundary and observing a 50% split in intensity on the output, then moving the spot a measured distance.

The experiment can also be modified to increase the accuracy of measured results, instead of creating a plasma close to a pixel edge, a plasma can be created close to a pixel corner. If the distance from the spot to the two pixel boundaries is different, then for the same plasma event two data points will be created on the cumulative distribution graph (Fig. 6.11). This does involve a more complicated

[10]Plasma effects have not been observed at current X-ray synchrotron brilliances, thus non-XFEL sources capable of generating such charge plasmas are at the time of writing hypothetical.

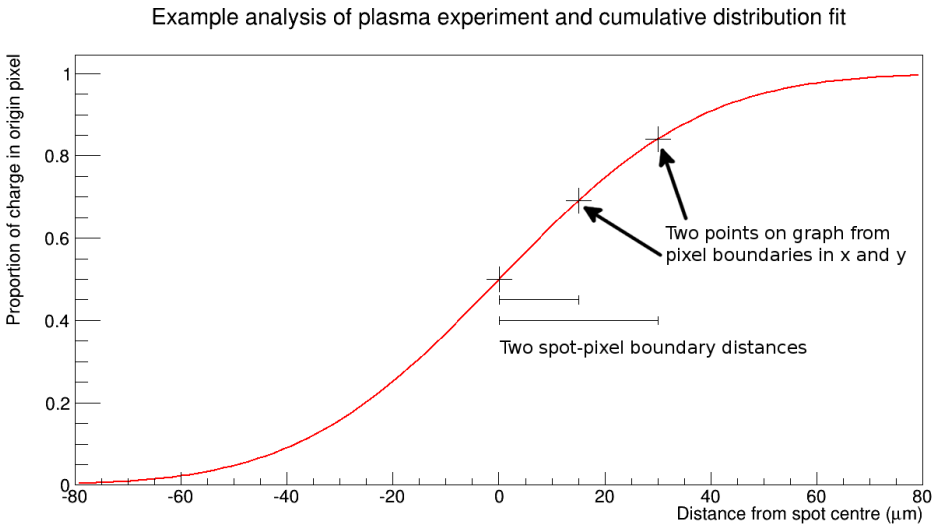


Figure 6.11: Cumulative charge distribution from a plasma with two data points and the spot centre.

analysis, as the proportions crossing the pixel boundaries in x and y will mix together in the pixel on the other side of the corner to the spot. This modification is particularly relevant when operating at an European XFEL, where the inconsistent output means that each charge deposit cannot be repeated with exactly the same energy. An alternate use of the second spot would be to extend the size of plasmas that can be accurately observed. This is demonstrated in Fig. 6.12, as boundary distances that are too far or too close to the spot (relative to spread size) will have poor resolution.

6.4.3 Simulation of charge plasmas

Plasma search

Charge plasmas form as a result of highly localised charge deposits, and high charge density. Identifying where charge events and clouds stop acting independent of each other and from charge plasmas[11] involves identifying regions of high charge density.

In X-CSIT, charge data is provided as a list of interactions, including their

[11] It is possible that scenarios arise where charge density is high enough for clouds to not act independently, but also not form charge plasmas. Such effects may be better simulated by expanding the plasma simulation to include near-plasma events.

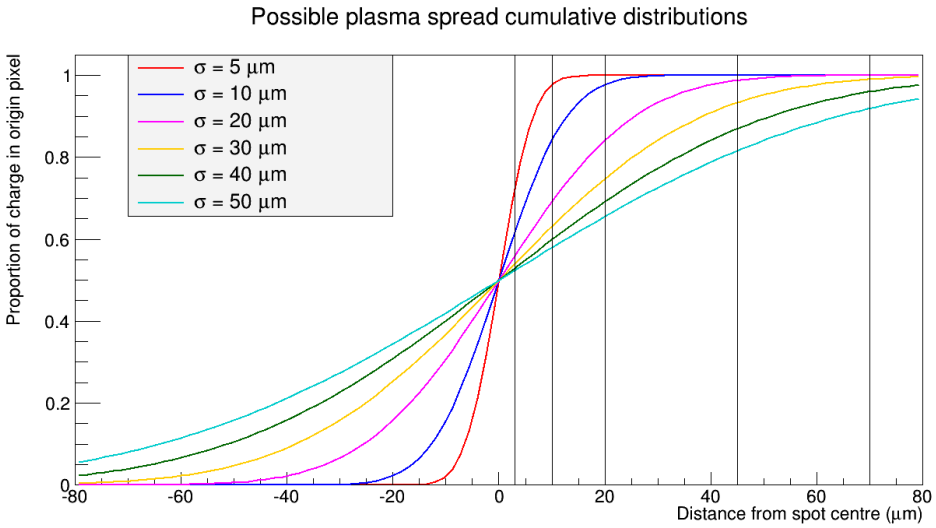


Figure 6.12: The plot of six cumulative distributions of different sizes. Vertical lines have been added at 3 µm, 10 µm, 20 µm, 45 µm *and* 70 µm *to help visualise the differences in detector response from each spread.*

positions. Identifying regions of density involves histogramming the data points. This poses a technical problem, as searching for µm scale plasma effects involves histogramming at the µm scale. 1 µm sized histogram bins in all three axes for a single LPD pixel would be 500^3 bins, and for a 32×128 pixel module would mean some 5.12×10^{11} bins. Storing a double floating point number (8 bytes) for each bin to count the charge deposit would require a histogram of approximately 4 TB size. RAM amounts of desktop computers are measured in a few GB at the time of writing. Storing this data would only be possible at the present time by saving it to a hard drive.

To overcome this, the plasma search function is designed with two steps. The first step is simply to only perform histogramming in the $x-y$ axes, ignoring the vertical component of the charge distribution. This is viable because incident photons arrive at an angle close to the normal of the detector face. Also, because charge plasmas are high flux events, the vertical distribution of energy deposits resulting from a photon spot will be similar to the average absorption profile of photons. Following a standard absorption curve, most of the energy will be deposited closer to the illuminated face of the detector than deeper into it. This last point means that

even photons arriving at an angle will concentrate their charge towards the surface, and also concentrate it in the $x-y$ axes. This however is of course an inaccurate method, and should only be taken as part of the first step of finding locations where charge plasmas might be found.

The second step is to stage histograms. Large sections of a detector, even one in use, are likely to have low photon counts or no incident photons (Fig. 6.13). Staging the histograms involves histogramming the detector at a much lower resolution than is required to identify charge plasma locations. Any large regions that do not have enough deposited charge to form a charge plasma, even if that charge is concentrated into a small space, can be discarded (Fig. 6.14). Regions that do have enough charge to form plasmas can then be histogramed at a higher resolution, i.e. smaller bin size (Fig. 6.15). The process continues for as many steps as required until the desired resolution is reached and regions of plasma are identified (Fig. 6.16). Regions which do have sufficient charge to form plasma, but which upon closer histogramming are revealed to be too spread out to form plasma, are also discarded. This histogramming process therefore has a variable size in bytes, but by only histogramming regions of interest, memory will be saved compared to a full histogram. The only exception to this would be if almost the entire detector is sufficiently saturated with charge for a plasma to form across the entire volume of the detector. If this occurs the interaction data will vastly outsize the histogram, as the interaction data has multiple floating point numbers per interaction, while the histogram will have multiple interactions per floating point number.

This is demonstrated in Fig. 6.17, where photons have been fired into a model detector. This detector has 4×2 pixels each 400 µm cubed, and 11000 photons were fired into it all at the same spot. The plasma search first searched using bins the same size as the pixels, then bins one quarter the size at 100 µm square, then 25 µm square. These histogram bins are larger than would be used in practice so they can be seen on the plot. Fig. 6.17 shows a histogram of energy deposit in eV. Overlaid on top of this are the plasma search histogram cells. A thick border of a cell indicates that the content charge exceeded the cutoff value and therefore was

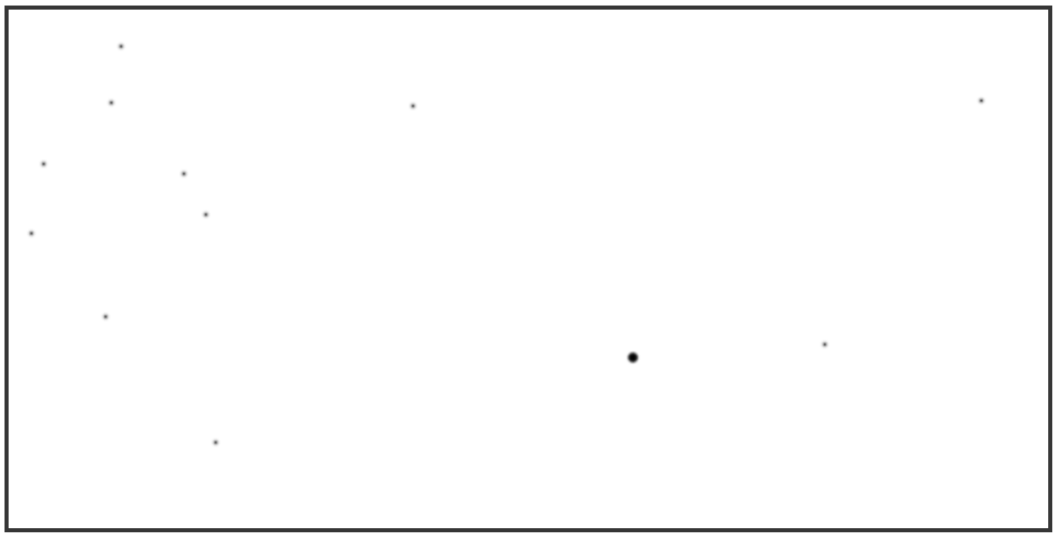


Figure 6.13: Multiple charge deposits from photon spots are recorded in a simulation of a detector, several smaller deposits are scattered across the detector along with one large deposit large enough to form a charge plasma.

investigated further by the plasma simulation.

In practice, charge deposits are unlikely to line up perfectly with histogram bins. Therefore either the histogramming needs to be of a resolution smaller than the size of charge deposits to identify their centre, or the histogramming should be used only as a starting point for a more detailed analysis of the charge deposit. This second option is preferable because most photon spots will be incident on the detector at an angle that while close too, is not exactly normal to the surface. The European XFEL detectors are 450 µm to 500 µm in depth, meaning that for histogram bins of 1 µm a pencil beam with an incident angle[12] over 0.1621° will cross more than one $x-y$ bin.

This plasma search algorithm is not fully capable of identifying the exact centre, or distribution, of a charge plasma generating charge deposit. The described plasma search function is a two dimensional search, therefore photon spots hitting the detector at an angle are likely to distribute their charge over multiple histogram bins. The histogram also does not contain information about the depth of charge

[12]This considers an infinitesimal beam crossing diagonally across a pixel, so uses a width of $\sqrt{2}$µm and a height of 500 µm.

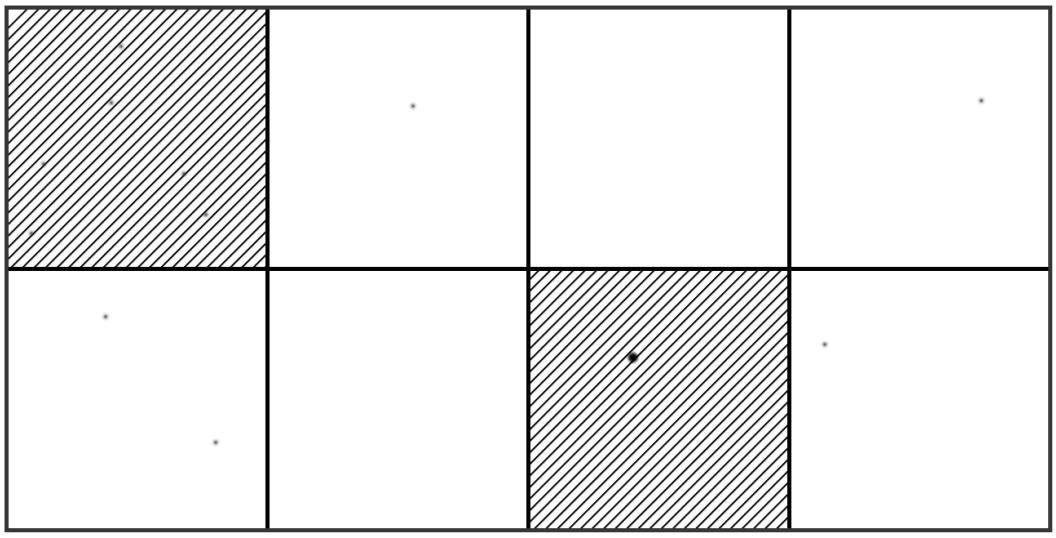


Figure 6.14: The detector is split up into large histogram bins and two bins (hatched) have enough charge to form a charge plasma. One of them is the bin with the charge plasma, the other merely has enough combined charge spread over the bin from multiple smaller spots.

deposits, something necessary to describe the charge plasma. Therefore the plasma search algorithm described cannot be used to adequately describe the charge deposits. It does however give a good indication of what regions will be dominated by the subsequent charge plasma event. The addition of a grouping algorithm would be required to ensure that neighbouring histogram bins that all contain charge from the same photon spot are jointly considered to be dominated by charge plasmas. However after this, the plasma search algorithm can be used as an input to both the charge diffusion and charge plasma simulations, indicating to both what regions of the detector each is responsible for simulating in parallel with each other.

As has been stated in this section, the charge plasma simulation is not complete. As of writing, a prototype demonstrating the functionality of the plasma search histogramming algorithm is the only completed element of the plasma simulation. This prototype lacks the ability for the user to set the size of each layer of histogram, as well as the number of histogram layers, these are instead fixed, hard coded values. It is however integrated into the charge simulation, including lookup functions the charge diffusion simulation can use to check if an event should be sim-

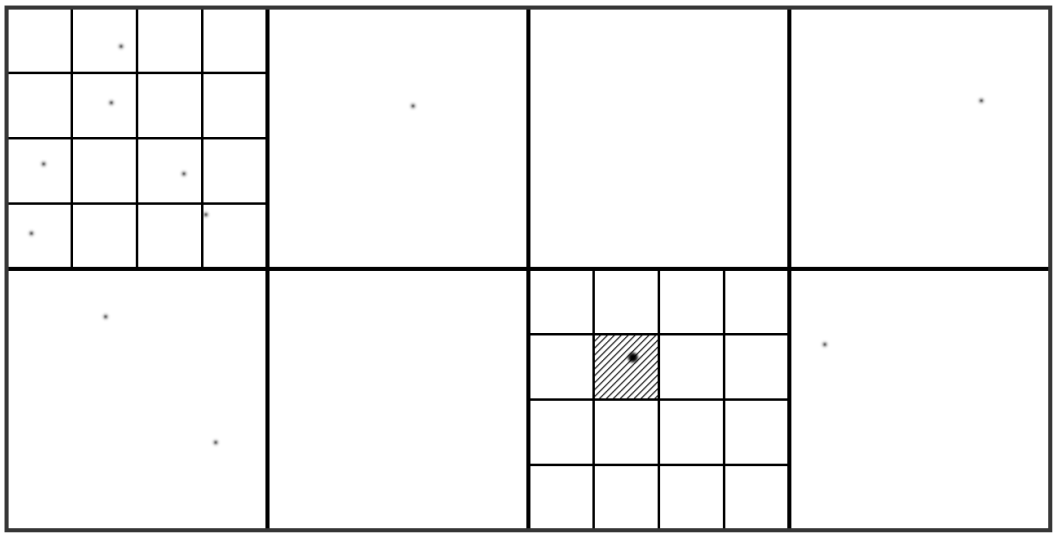


Figure 6.15: A second round of histogramming is conducted only in the regions identified by the first round. One region does not have enough concentrated charge in any one bin, while in the other region, the charge plasma is identified.

ulated or left to the plasma simulation. These functions have direct access to and knowledge of the histogramming layers to provide the fastest possible response. This means that if an event that is being checked is in a histogram region that does not contain any plasma cells it is rejected immediately without checking at a smaller resolution.

In the absence of any data to describe the minimum charge plasma, the plasma search function currently uses an estimate of the minimum charge for each cell based upon the Debye length. The Debye length in the context of a plasma is the range over which the electric field of a charged particle is felt by other particles [70]. This leads to it also being the range over which electrostatic fields are screened by the plasma. A Debye sphere is a sphere inside a plasma with the radius of the Debye length. A Debye sphere therefore screens its centre against any external electrostatic fields. This forms a condition for conventional plasmas that the Debye sphere be smaller than the region being considered. The Debye length also has relevance in semiconductors, where it again describes the distance over which external electric fields or fluctuations are screened in the semiconductor. The Debye length

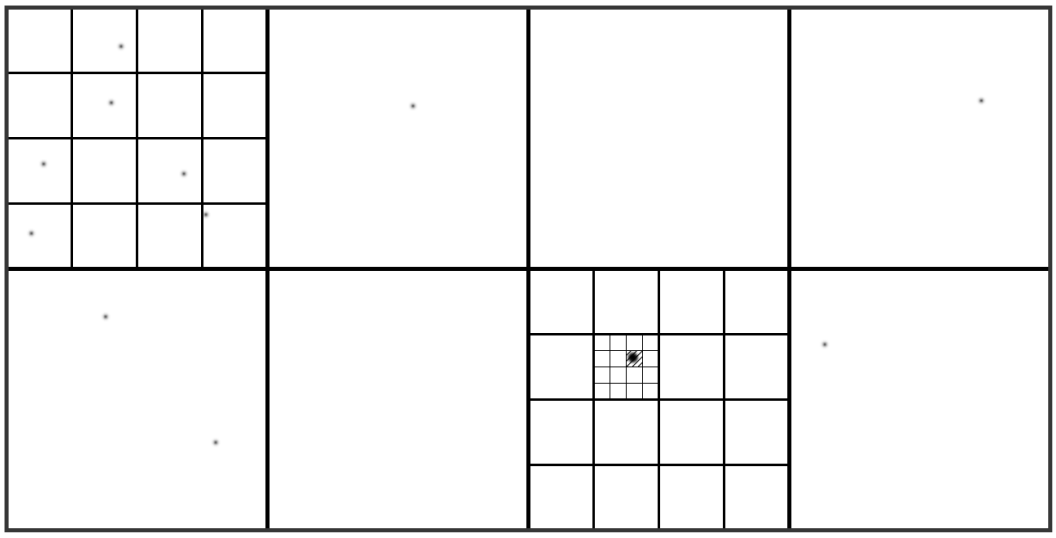


Figure 6.16: A third round of histogramming is conducted in the only location still being considered for charge plasma.

for semiconductors is given as

$$L_D = \sqrt{\frac{\varepsilon k_B T}{q^2 N_d}} \tag{6.16}$$

where ε is the permittivity, k_B is the Boltzmann constant, T is the temperature, q the elemental charge and N_d the density of dopants. In plasmas the Debye length is given as

$$\lambda_D = \sqrt{\frac{\varepsilon_0 k_B / q^2}{n_e/T_e + \sum_j z_j^2 n_j / T_j}} \tag{6.17}$$

where ε_0 is the permittivity of free space because in a plasma the background medium is treated as a vacuum with $\varepsilon_r = 1$, n_e the density of electrons, T_e the temperature of electrons, $z_j q$ the charge, T_j the temperature and n_j the density of ion species j. In the context of plasmas the ion term is often dropped when the mobility of the ions is negligible compared to the process time scale. The two equations are recognisable as being very similar, with the semiconductor Debye length depending on a single species of particle and the plasma Debye length accounting for electrons as well as varying species of ions. Therefore a Debye length for a charge plasma should be

$$\lambda_D = \sqrt{\frac{\varepsilon k_B T}{2 n_p q^2}} \tag{6.18}$$

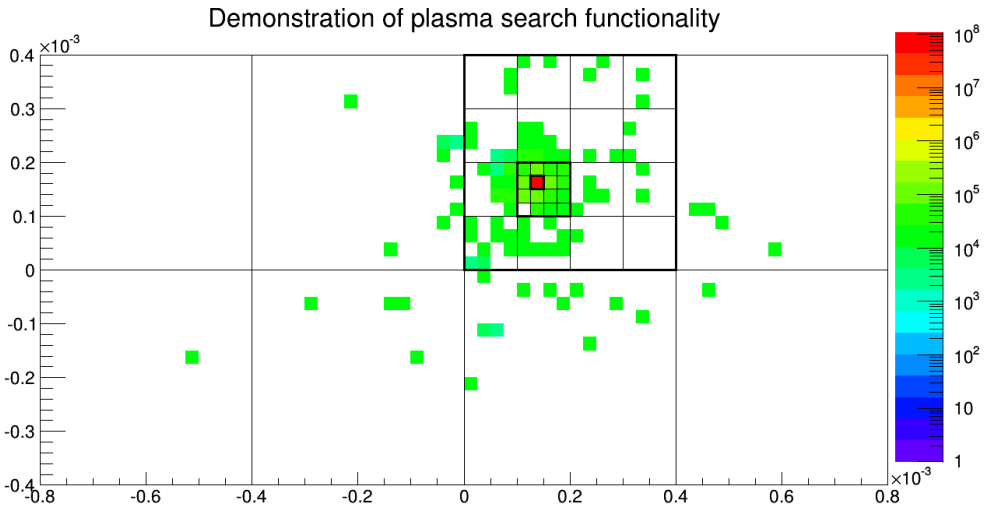


Figure 6.17: A histogram of energy deposit in eV of a photon spot incident on a model detector of dimensions $1.6\,\text{mm} \times 0.8\,\text{mm}$. *Overlaid is the result of a plasma search, with thin lines indicating histogram bins, and thick lines histogram bins identified as potentially containing plasma.*

where n_p is the density of electron–hole pairs and T is their temperature. Both electrons and holes contribute to the charge plasma and therefore should be counted for the Debye length, however in all other contexts they are considered either as electron–hole pairs, or only one species is considered and the other discarded. Therefore $2n_p$ is really $n_e + n_h$ the sum of electron and hole densities, but because electrons and holes are always created as pairs $n_e = n_h = n_p$ at the time of their creation.

Plasma effect simulation

The second step of charge plasma simulation is the simulation of the final effects of the plasma seen by the detector. This needs to include a numerical calculation of a charge spread with the addition of a charge dependent spread size, as well as collected charge dropping due to recombination and any timing effects that exceed the collection time of the detector. This is the principal bottleneck in development of the plasma simulation. As noted earlier in Section 6.4.2, this needs to be derived from experimental data that is not available.

Currently the prototype plasma simulation lacks histogram bin, or cell, grouping at the plasma search stage. This is the identification of neighbouring cells that are likely to contain charge from the same photon spot, and should therefore all be considered for plasma simulation even if some cells (or all if charge is evenly split between them) do not meet the criteria to form plasma on their own. A plasma simulation will need to identify these groups and should, before simulating the charge spread, determine a better understanding of the charge deposits causing it. This is principally related to identifying the centre of a plasma event, as the two dimensional plasma search is not sufficient to determine this. Therefore in any collected group of cells, the plasma simulation would ideally identify the centre of the deposit independently of the positions of the cells, probably with a weighted average of deposit location. Identifying the direction of photon incidence, and the line around which the deposit is centred, may also prove useful. Then, particularly with marginal plasma cases, it should perform a more detailed check that a plasma will form. Finally it can then perform the calculation of charge spread based upon test data.

An additional factor is that charge plasmas are a slow process relative to regular diffusion and comparable to the timing of detector images. A serious concern over plasmas that may occur at European XFEL is that they may take longer to dissipate than the collection time of each image frame. This would lead to the plasma spilling over into the next frame, and likely also lose charge during the pixel resets some European XFEL detectors perform between flashes. The effect of charge plasmas on the collection time of charge is better understood than the effect on charge spreading. Charge plasmas were originally discovered as a result of a charge dependent increase in charge collection time with heavy ion detectors [67, 112]. Literature from this field should be used to program the time component of the simulation, which for an imaging detector would be a cutoff, wherein only a portion of the charge in a plasma is collected.

One final problem that may arise is that the spread and timing components of the charge plasma effect have been considered separately, and it is possible that the

Table 6.3: Specifications of the pnCCD tested and simulated in X-CSIT.

Pixel size	75μm square
Sensor size	200$\times$128 pixels
Depth	300μm
Bias Voltage	200V

two are linked. It is known that charge continues to be collected from a detector while a charge plasma is in existence. It is possible that the charge trickle from the plasma does not follow the same spread pattern as the whole of the plasma (as would be determined by experiment). It may instead be smaller having been stripped from the outer edges of the plasma while it is still expanding. Therefore if the detector frame ends before the plasma has been collected it is possible that the spread seen would be smaller than the data of a full collection would predict.

6.5 Initial testing

Initial testing of X-CSIT was conducted and published as part of a collaborative work [113], using data taken by pnSensor GmbH using a pnCCD [114, 115] (see Table 6.3) and a Fe-55 source. A simulation of the pnCCD was implemented in X-CSIT, see Fig. 6.18, and the data from the real and simulated detectors were run through the same analysis pipeline previously validated with the CERN Axion Solar Telescope CCD [116].

Fig. 6.19 shows the histograms produced from the uncorrected measured and simulated data. The residuals between the two data sets shown are given by $(\text{measured} - \text{simulated})/\sqrt{\text{measured}}$. These spectra are not normalised with respect to each other. Instead, using the known source characteristics, the simulations were run to produce a number of primary photons that is equal to the number of photons incident on the detector in the measurements. The visible peaks from left to right are: noise peak, Al fluorescence, K and K 55Mn decay lines at 5.9 keV and 6.5 keV. Good agreement can be seen between the measured and simulated data above the noise peak (3000 ADU). The data in the regions just below the 55Mn peaks are

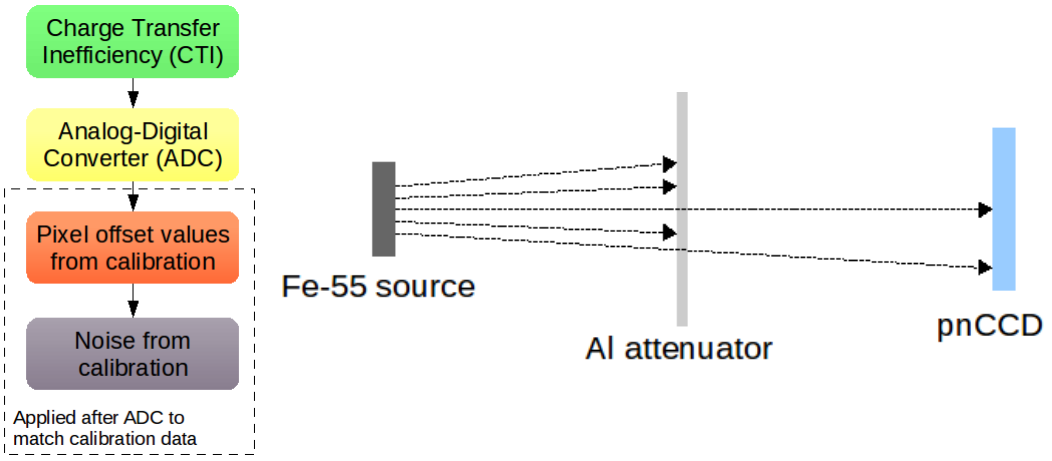


Figure 6.18: Components of the pnCCD electronic simulation and diagram of the experimental setup.

underestimated by the simulation, while the region below this, 8000 ADU to 9500 ADU, and the shoulder of the noise peak are overestimated. This could indicate that the simulation creates an excess of charge sharing events with the majority of the charge in the primary pixel of the split event. The low energy cutoff in the pnCCD is not simulated.

In Figs. 6.21, 6.22 and 6.23 spectra are shown for the different event types, characterised by the shape of the spread on the pixel grid as shown in Fig. 6.20. Additionally, first singles, i.e. singles that are also the first event in a readout column per transfer and thus has not been affected by the residual of a charge transfer of another event, are shown. The data has been offset and common mode corrected and each pixel of an event contributes to the histograms. These show qualitatively similar features on both the simulated and measured results in all four event types, however quantitatively the ratios of event types do not match. Events in three regions were compared for each event type: the low energy background region, ranging up to energies of 2000 ADU 1.58 keV, the continuum region, in which split partners will largely be found (2000 to 6000 ADU 4.75 keV) and the peak region, containing the 55 Mn photo-peaks ($>$ 6000 ADU). For each region mean deviations in terms of sigma uncertainties of the measured data have been calculated. These values are given in Table 6.4.

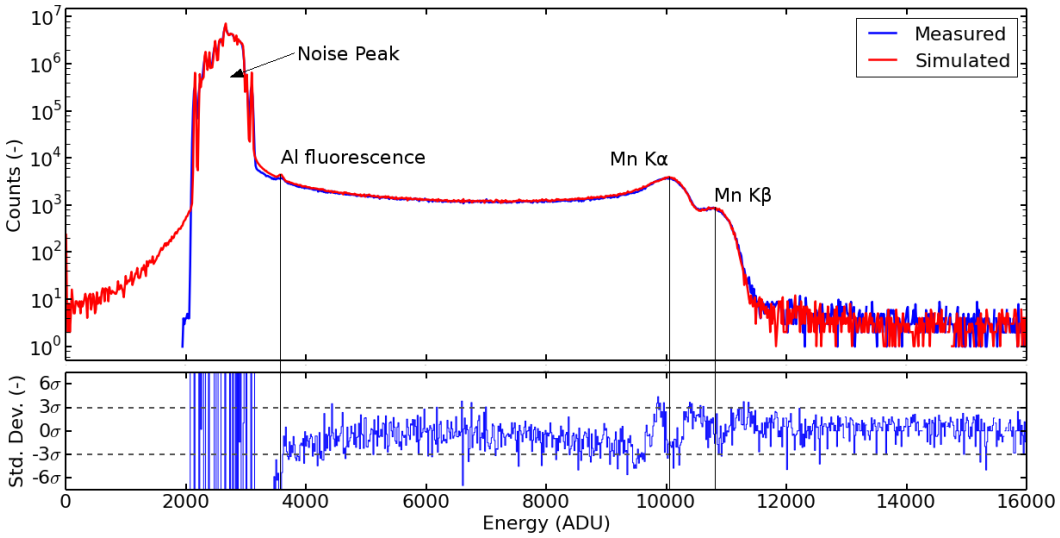


Figure 6.19: Top panel: the uncorrected energy spectra for the measured and simulated data given in detector units. Lower panel: the relative deviation between the two, given in terms of 1 uncertainties.

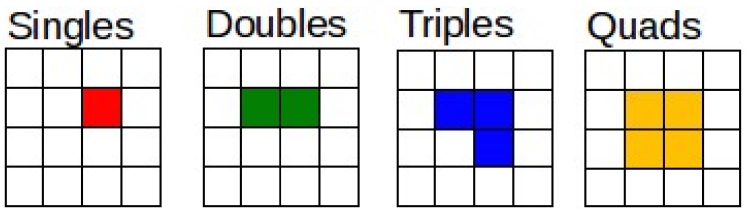


Figure 6.20: Legend of event types, defined by the shape of the spread on the pixel grid in any rotation.

Overall, i.e. across the complete spectrum, triple and quadruple patterns are both over estimated (by 0.40% and 13.60%) while double patterns are underestimated (20.32%). The inclusion of background noise in the statistics of the measured singles and first singles but not the simulated singles makes comparison of these patterns (51.03% and 52.73% in simulation) difficult. When considering only the immediate peak region for singles, i.e. events above 7000 ADU, thus eliminating the low energy background in the measured data, simulation singles and first singles are underestimated (40.06% and 42.34%).

For the individual regions it is found that the shape of the continuum is generally represented well by the simulation, although a consistent underestimation

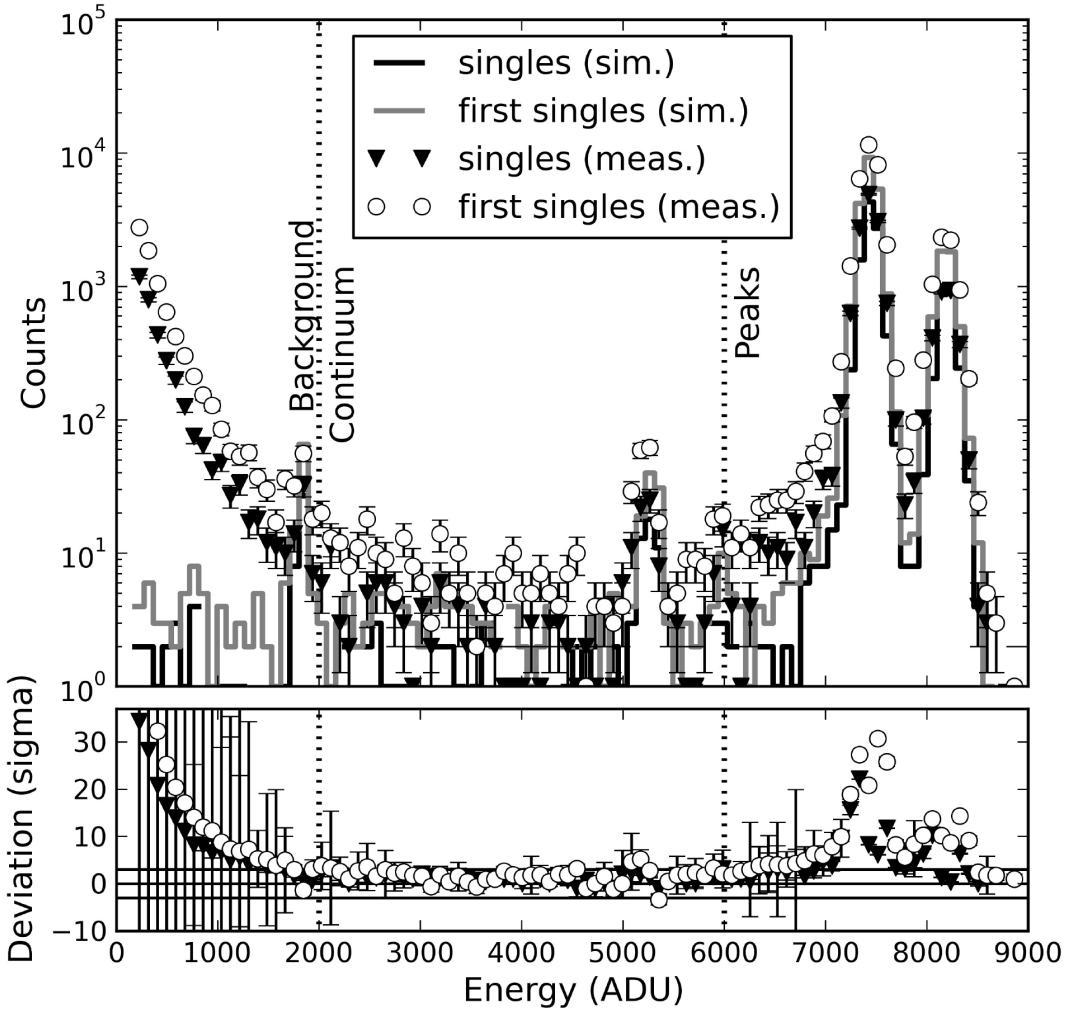


Figure 6.21: Energy histogram of single and first single events.

exists, except for quadruple split patterns. The peak region is consistently overestimated by the simulation, both for singles and higher multiplicity patterns. The large deviations in the background region can be attributed to the fact that a simplified geometry was simulated and as such scattered photons from experimental components other than the detector and the source are missing. This is a systematic effect, as is evident from the residuals shown in the plots and the large variance of deviations in this region.

Fig. 6.24 shows a comparison between fully calibrated measured and simulated data, which has been corrected for offset, common-mode and Charge Transfer Inefficiency (CTI), alongside fits to the 55 Mn K and K lines. The numerical components of these fits and how they match to the expected emission lines are given

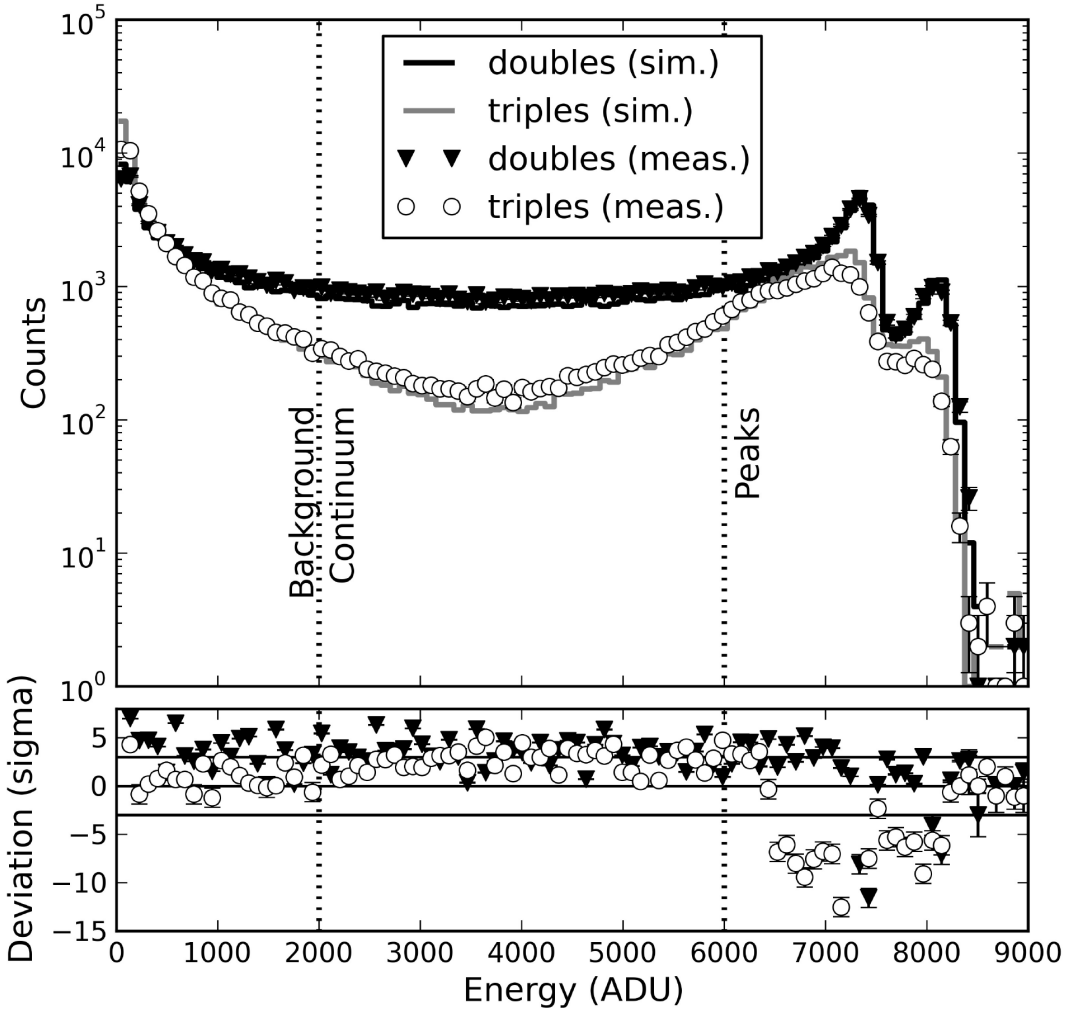


Figure 6.22: Energy histogram of double and tripple events.

in Table 6.5. Additionally, the residuals between the two data sets are shown as $(\text{measured} - \text{simulated})/\sqrt{\text{measured}}$. The charge sharing excess in the simulation causes a deficit of singles, this causes higher uncertainties in the simulation fit. A match within 1 of fitted energy is seen between the simulated and measured data in both peaks as well as the width of the 55 Mn K line. The width of the fit of the 55 Mn K line is larger for the simulated data set by a statistically significant amount, but the source of this deviation has not yet been identified. The deviation between the fully calibrated simulated and measured first singles match the deviations observed for non-calibrated first single events.

These results indicate reasonable agreement between measurements and simulation, within 3 on the uncorrected spectrum. However systematic deviations, es-

Table 6.4: Mean deviations between simulated and measured data in terms of 1 uncertainties of the measured data for different event types and regions. The values in brackets give the variance of the data in a specific region.

Region	Background	Continuum	Peak
Singles	9.3 (80.1)	1.3 (1.0)	4.9 (23.9)
First singles	14.1 (191.4)	1.6 (2.4)	9.1 (61.0)
Doubles	2.3 (33.2)	3.5 (2.0)	0.9 (14.4)
Triples	2.1 (190.5)	2.7 (1.3)	4.2 (28.3)
Quads	14.8 (424.2)	7.7 (8.6)	6.1 (59.6)
Invalids	11.1 (958.8)	0.7 (2.4)	4.2 (45.2)

Table 6.5: Tabulated data of the fits to the Mn peaks in figure 6.24

	Simulated	Measured
Mn Kα	5892.78 (+-2.61) eV	5891.41 (+-2.46) eV
FWHM Mn Kα	151.63 (+-0.58) eV	141.20 (+-0.35) eV
Mn Kβ	6490.19 (+-3.37) eV	6488.13 (+-3.05) eV
FWHM Mn Kβ	158.50 (+-1.81) eV	157.67 (+-1.17) eV

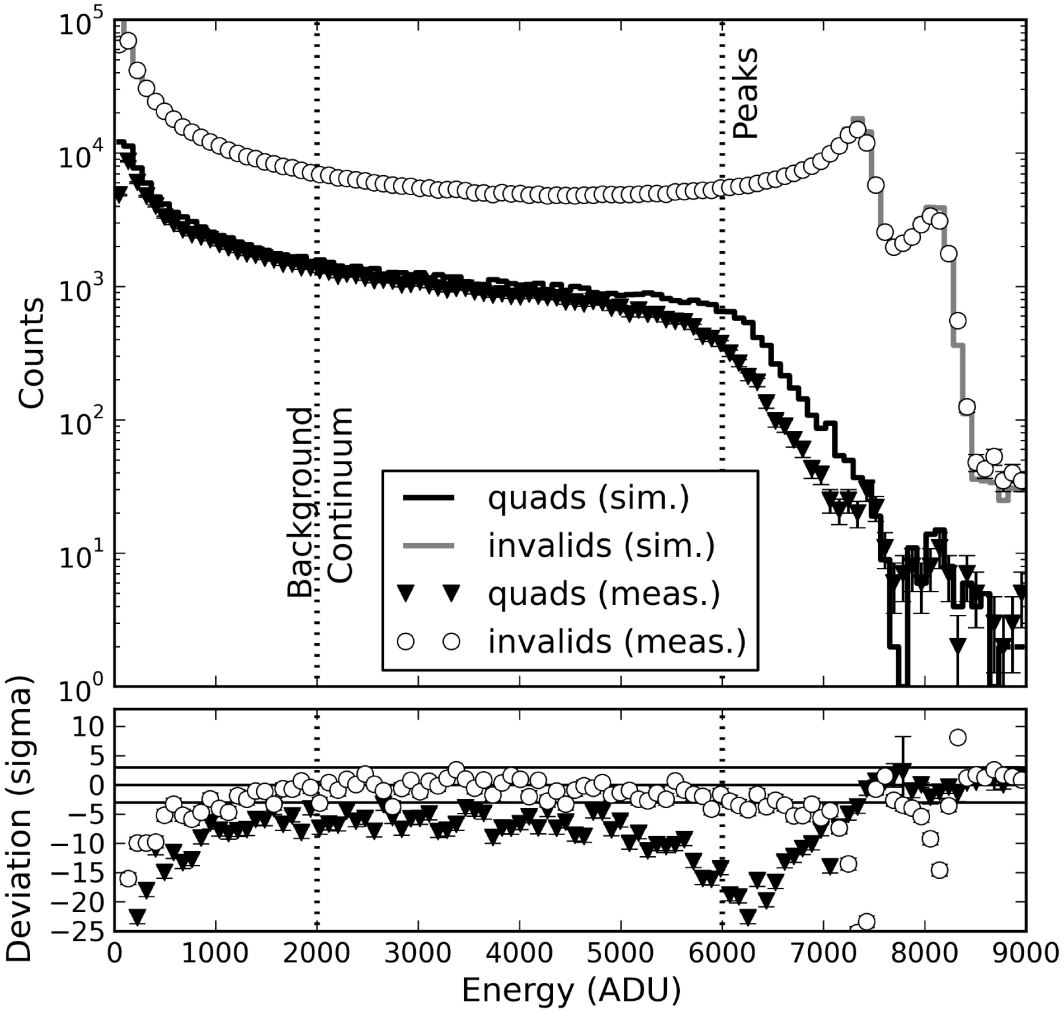


Figure 6.23: Energy histogram of quad and invalid events. Invalid corresponds to any other event shape, or patterns with a diagonally offset event adjacent to it.

pecially concerning charge sharing, remain. In particular single, first single and double events are underestimated while triple and quad events are overestimated, indicating that charge sharing may be overestimated in X-CSIT.

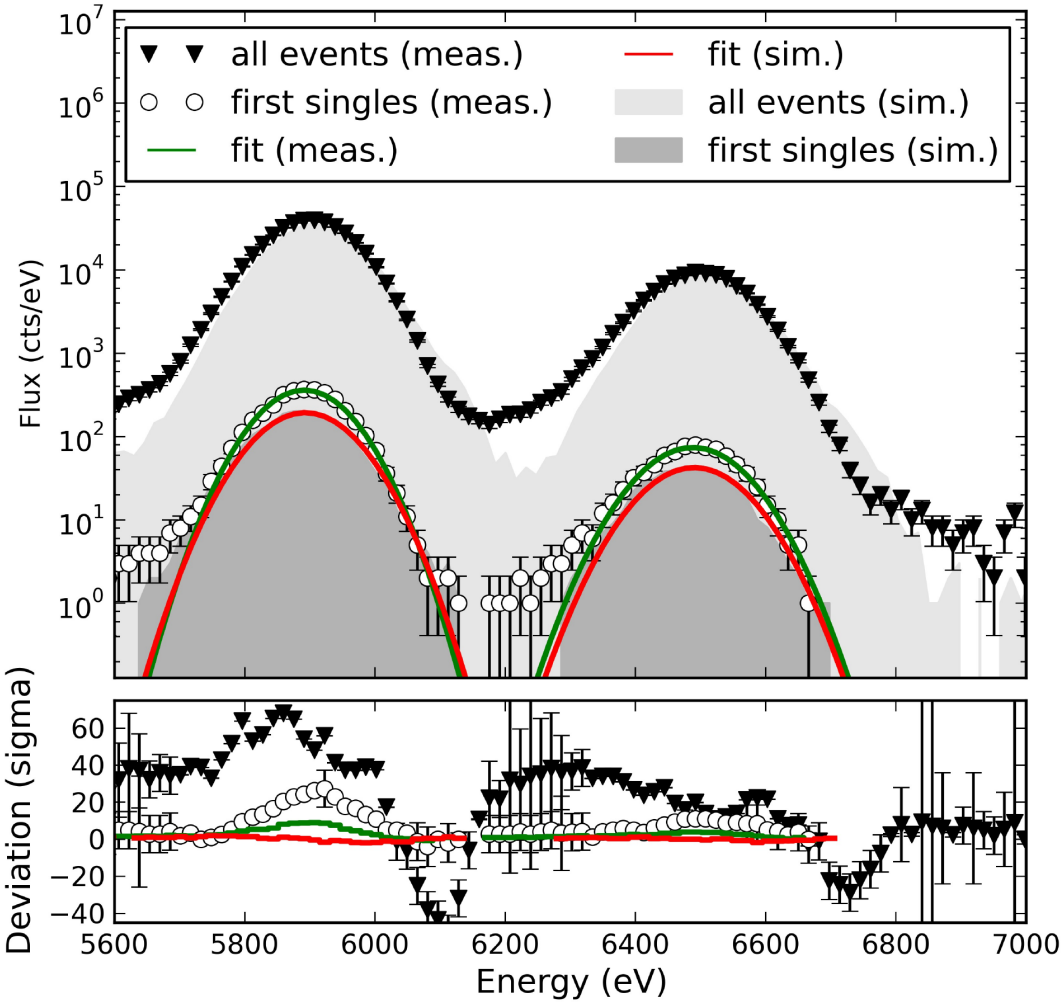


Figure 6.24: Photo peak region showing a comparison of fully corrected data alongside the fit models used to determine the gain.

Chapter 7

Analysis of X-CSIT charge simulation with LPD

7.1 Experimental setup

To test the X-ray Camera SImulation Toolkit (X-CSIT), experiments were performed with the Large Pixel Detector (LPD) at the Advanced Photon Source (APS) facility in Argonne, USA, between commissioning and test experiments performed for LPD itself. APS produces an asymmetric, squashed ovoid beam (Fig. 7.1 and 7.2) of 18 keV photons. This beam was collimated into a rectangular beam, approximately 5 mm horizontally and several tens of μm vertically. The exact vertical size of the beam varied between experiments. This beam was then scanned vertically across LPD as shown in Fig. 7.3. Because the beam was scanned in the vertical direction, it is the vertical axis of the beam that is known as the beam width.

The data from LPD shows the progression of the beam as it passes from one row of pixels to the next. It is this transition between rows of pixels, known as crossing a pixel boundary, where charge sharing can be observed and measured. This pixel boundary does not actually represent a physical feature in the detector. The sensor is a continuous block of semiconductor and the pixels are points on one face of this where charge is collected. The pixel boundary is therefore actually a midway point between adjacent pixels.

In a scenario where the photon beam is smaller than the scale of charge sharing,

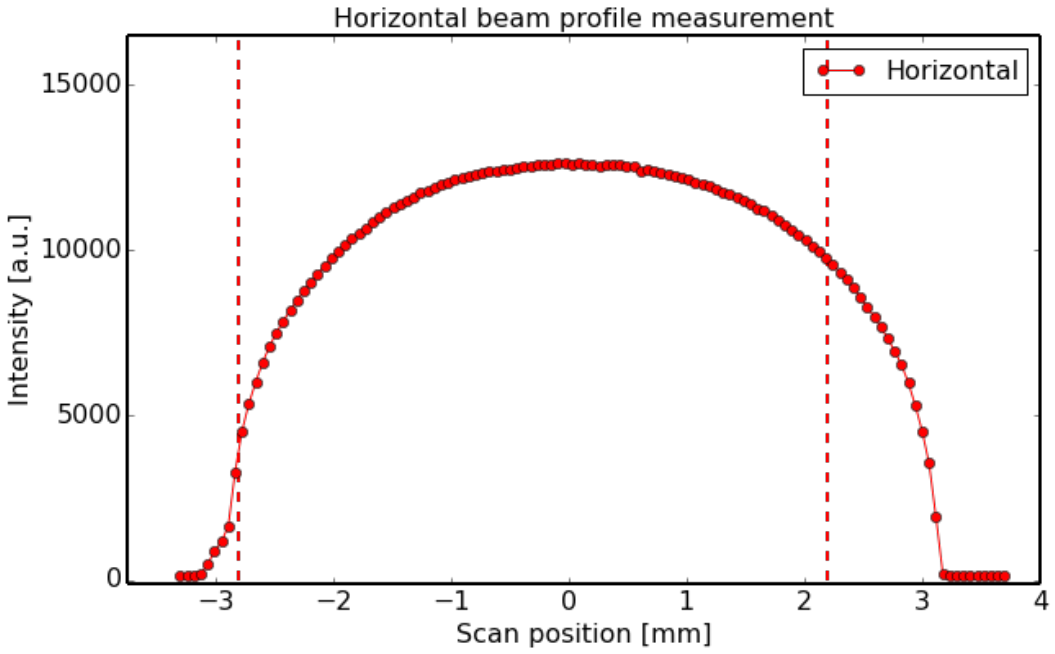


Figure 7.1: Horizontal line scan measurements of the APS beam before collimation. The dashed lines indicate where the beam was collimated to 5 mm *wide.*

it would be possible to measure charge sharing directly from the ratio of charge in each pixel as the beam is scanned from one to the other. In this ideal experiment the proportion of the total beam that is collected in one of the pixels would be the Cumulative Distribution Function (CDF) of the spread function when plotted against scan position. It would be necessary to consider the variable depth of photon absorption, as well as possibly particle scattering, but the result would be a direct observation and measurement of charge sharing.

However, such an experiment would require a beam and step sizes smaller than the scale of charge sharing, preferably very much smaller and on the scale of $\sim$1 µm. Neither of these parameters were available at APS for use with LPD. Both the minimum beam size (width) and step sizes available were in the 10s of µm. In this scenario the split in charge between two pixels is no longer solely attributable to charge sharing, but also to the width of the beam itself. This is known as the charge split, or charge splitting.

Despite this it is still possible to observe and measure charge sharing with a

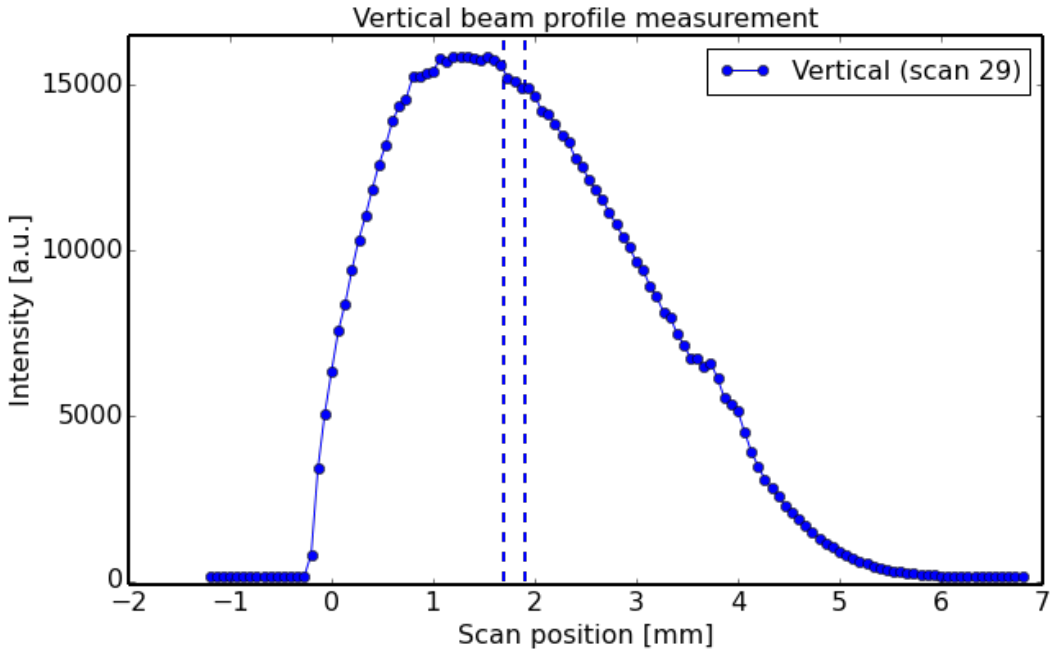


Figure 7.2: Vertical line scan measurements of the APS beam before collimation. The dashed lines indicate the initial region of collimation during testing, the experiments described in this chapter were collimated further within this region.

wide beam, so long as there is an imbalance of charge sharing between adjacent pixels. So long as more charge passes from one pixel to the other than the reverse, the proportion of charge in one of the pixels will be perceptibly different to the proportion of the beam landing in that pixel. With a monolithic beam, this would be achieved simply by moving the beam so that more area of the beam is incident upon one pixel than the other, and maximised by positioning the beam so that the edge of the beam is aligned with the pixel boundary. Simulations demonstrating this can be

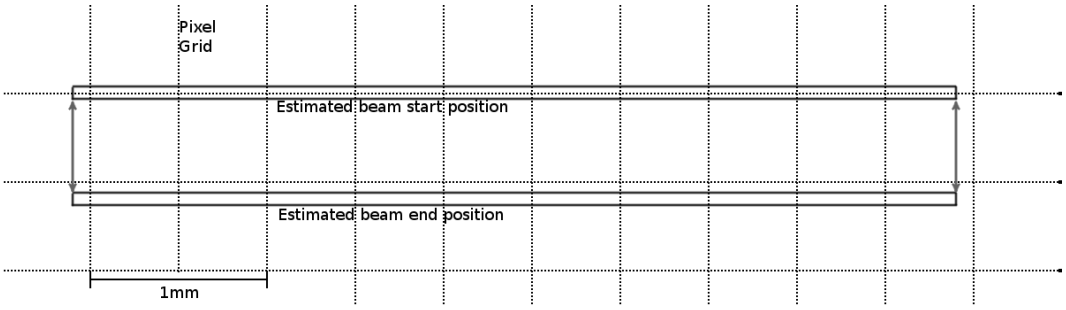


Figure 7.3: Diagram showing the pixel grid and the beam scan of the 50 µm *dataset.*

seen in Fig. 7.4, which show $F(x,\sigma)$ numerically integrated where

$$F(x,\sigma) = \frac{1}{4w}\int_{-\infty}^{x} Erf\left(\frac{y+w}{\sigma\sqrt{2}}\right) - Erf\left(\frac{y-w}{\sigma\sqrt{2}}\right) \mathrm{dy}. \tag{7.1}$$

Here y is a dummy variable, w is half the width of the initial beam, and the term of the integration is the spread function of the beam along the x axis. The beam spread function is found by integrating the normal function along the width of the beam, from $-w$ to w, a function of its mean μ. Fig. 7.5 shows the difference between two X-CSIT simulations, one with and one without charge sharing, that also demonstrates the beam edges to be the most sensitive region to charge sharing.

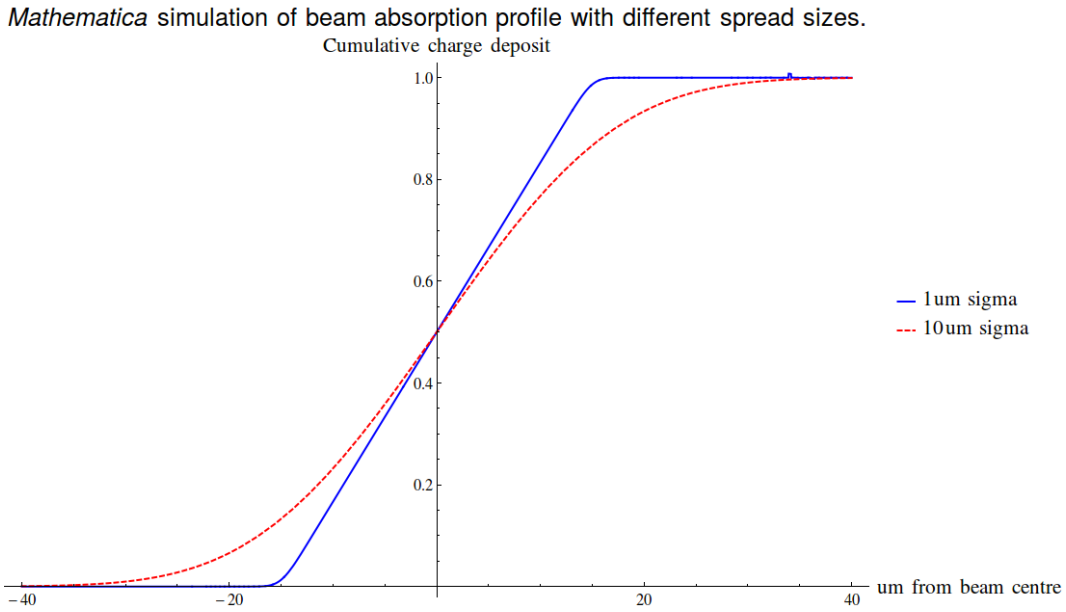


Figure 7.4: Numerical predictions of charge absorption as a beam 30 µm *wide is scanned from one pixel to the next. Two predictions are shown, of charge spreading with* 1 µm *and* 10 µm *sigma.*

To prove that positioning the beam edge along the pixel boundary produces the most charge sharing, and therefore the most observable result, we consider a monolithic, rectangular beam incident upon a pixel detector. We position this beam such that it lands on two adjacent pixels or rows of pixels, and then observe it along the axis of the pixel boundary as in Fig. 7.6, as we are only concerned with charge movement perpendicular to the pixel boundary. The beam has a fixed width of

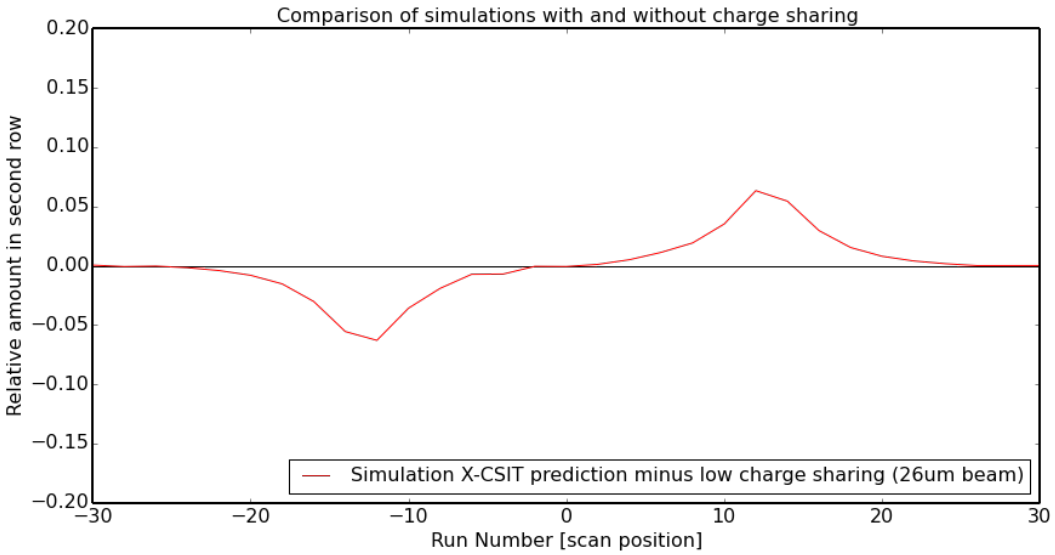


Figure 7.5: Difference diagram showing the result of X-CSIT simulation with charge sharing minus the result of an identical simulation without charge sharing.

$W = B_1 + B_2$, where B_1 is the distance from the boundary crossing to the edge of the beam in pixel 1, and B_2 is the distance from the boundary crossing to the edge of the beam in pixel 2. We define pixels 1 and 2 such that $B_2 \leq B_1$. Here B_1 and B_2 vary with the beam position, but W is constant. Now we consider a small element of the beam ε_1, which is a distance R from the pixel boundary and is in pixel 1, and contains many photons. Each photon from ε_1 that is absorbed by the detector will have its own charge spread that is dependent on absorption position and secondary particle scatter. However we can define a charge spread S_1 which is the average spread of the sum of all spreads caused by photons in ε_1. A proportion of this spread, $P_{S1}(R)$, will cross the pixel boundary and be absorbed by pixel 2 instead of pixel 1.

Another small element of the beam, ε_2, is also a distance R from the pixel boundary, but lands in pixel 2. This element also has an associated spread S_2, and a proportion of this spread, $P_{S2}(R)$, will cross the pixel boundary and be absorbed in pixel 1 not 2. Because S_1 and S_2 are symmetrical, and identical to each other except in their position, $P_{S1}(R) = P_{S2}(R)$. Therefore the charge spreading from ε_1 is exactly opposite to ε_2 and the charge absorbed by each pixel is the same as

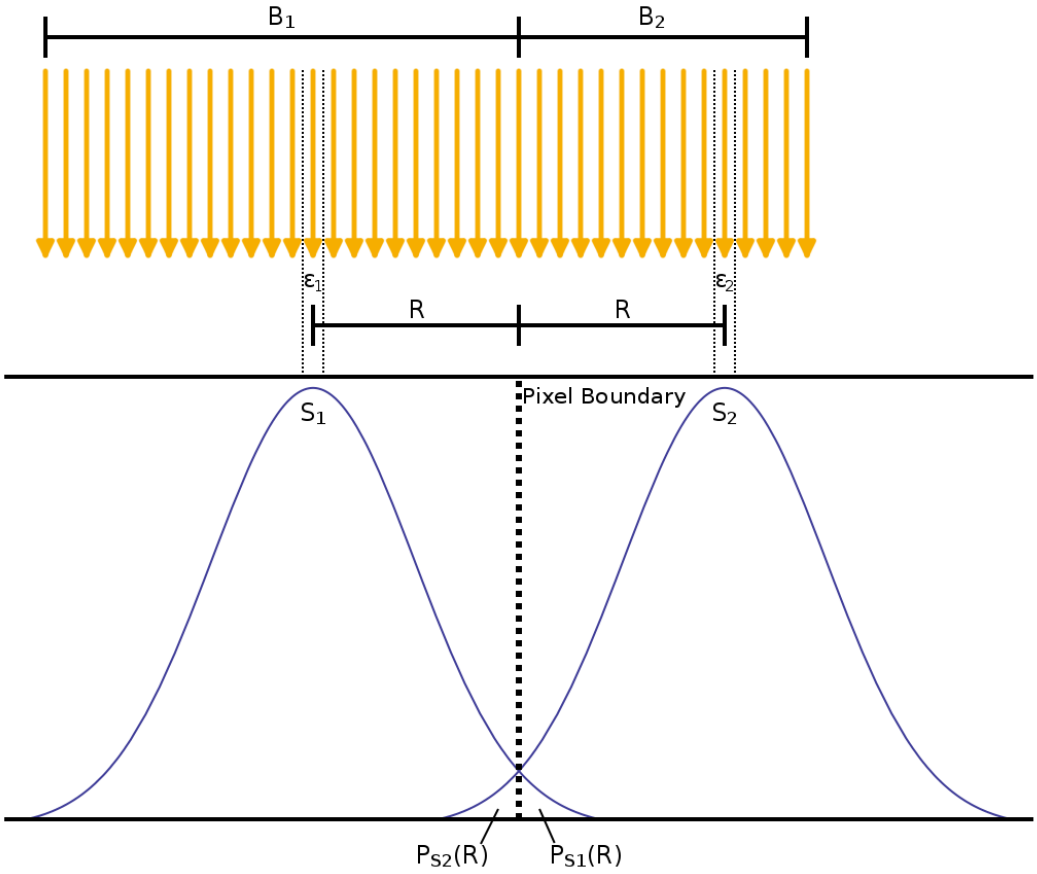


Figure 7.6: A diagram showing how two small beam elements ε_1 and ε_2, equidistant from the pixel boundary, can cancel out each others charge sharing, such that no charge sharing is observed from them.

the charge incident on each pixel. Without an imbalance in charge sharing, charge sharing is not observed from the elements ε_1 and ε_2. Charge sharing will only be observed from ε_1 with $R > B_2$ and so has no equivalent ε_2.

The proportion of charge that is observed to cross a pixel boundary C_{observed}, and is not exactly negated by beam on the other side of the pixel boundary, is

$$C_{\text{observed}} = \int_{B_2}^{B_1} P(R)\,\mathrm{d}R \tag{7.2}$$

For $P(R)$, the proportion of a spread that crosses a pixel boundary distance R away, that decreases with R, it can be seen that C_{observed} is maximised when the beam position is chosen such that $B_2 = 0$ and $B_1 = W$.

It should also be noted, that if B_1 and B_2 are redefined such that B_2 is the

distance from the pixel boundary to the closest beam edge, and B_1 to the furthest beam edge, Equation 7.2 is still valid even when the beam is wholly within one pixel. The method of maximising Equation 7.2 is also still the same, though it is evident without it that moving the beam closer to the pixel boundary will increase charge spreading across that boundary.

With an imbalance of charge sharing across a pixel boundary crossing, the proportion of total beam absorbed by one of the pixels can again be plotted against beam position, and compared to the zero charge sharing hypothesis and simulations with charge sharing. An additional consideration is beam stepping size. It was not possible to measure the exact beam position relative to the location of the pixels to the μm precision that would be required, and without such a measurement it would be impossible to unlink the charge imbalance due to charge sharing from that due to beam incidence with a single data point. At least two measurements showing a charge split are required, given a known beam step size. With at least two measurements, simulations can be matched to the data to find the beam position and measure charge sharing.

7.2 Analysis of LPD data

7.2.1 Available data and file structure

Two datasets were taken with LPD at APS in November 2015. These are referred to as the 50 μm and 20 μm datasets for the stepping sizes the detector was moved, relative to the beam, between each data point. This beam of 18 keV photons was rectangular, approximately 5 mm horizontally and several tens of μm vertically. The vertical size of the beam was 70 μm for the 50 μm dataset, and unknown for the 20 μm dataset. Because the beam was scanned in the vertical direction, it is the vertical axis of the beam that is known as the beam width.

A single super-module of LPD was used, however only six out of sixteen modules possible were attached, indicated in Fig. 7.7. The other regions of the detector still output data because they were still part of the same super-module, even though no sensor or Aplication Specific Integrated Circuit (ASIC) chips were present. The

nature of the experiment is such that APS and the beam it produced were stationary, with the detector moved relative to it. However, this is usually referred to as the beam being scanned, since in the detector output it is the beam that appears to move. Both datasets have differing numbers of runs (step positions), images taken per step and differing file structures. They also had different beam widths and correspondingly, different photon counts per image.

The 50 µm dataset consists of 13 runs, where each run was a beam position along the scan. Between each run the beam position was moved 50 µm in the vertical direction, for a total scan of 600 µm, or 1.2 LPD pixels. For each run and beam position, LPD was switched on ten times, each time producing a separate data file. Each file contained 504 complete images. These images are split into trains of 24 images, each train appearing in the file sequentially. LPD was designed to operate at European XFEL where beam flashes would arrive in trains with empty periods between them, LPD therefore stores multiple images from a train in separate analogue memory cells and outputs between trains. LPD was linked to the APS clock so that it would begin taking a 24 image train at the same point relative to the synchrotron phase. The APS beam normally flashes at a higher frequency than LPD can image, with the expectation that experiments treat it as effectively continuous. However, for LPD, APS redistributed the electron bunches in the ring. Most of the electron bunches were moved closer together, covering approximately $1/5$th of the ring circumference, while on the other side of the synchrotron a single bucket was given the maximum possible fill. In LPD this appears as three sequential images with a low signal from the many small bunches, and a single high signal image from the high intensity flash. Sixteen images[1] were taken by LPD per full cycle of the APS synchrotron, with image 12 containing the single intense flash. Because only 24 images were taken by LPD per train in the 50 µm dataset, only a single image with good data was recorded per train. Images 3 to 5 contained the remaining, lower intensity flashes. These images are each noticeably less intense than image 12 and on recommendation from T. Rüter, who took the data, these images were not used

[1] Images were counted from 0.

because of their lower signal/noise ratio. With 21 trains per file, the 50 µm dataset therefore contains 210 images per beam position.

The 20 µm dataset, which was captured after the 50 µm dataset but analysed before it, used a slightly different file layout. The number of images per train was increased to 32, and the number of trains cut to 16 for 512 total images per file. Because of the larger train size, image 28 also contained usable beam data, so the number of usable images per file increased to 32, and per run to 320. Each run still represents a single step of the beam position vertically, though this time 20 µm, with ten files per run. The 20 µm dataset contains 21 runs compared to the 13 of the 50 µm dataset. This is because the 50 µm dataset was intended to have 21 runs, for a total scan of two LPD pixels, or 1000 µm, to guarantee a boundary crossing. However, the 50 µm experiment crashed after 13 runs and recorded no further data. For the 20 µm dataset the same value of 21 runs was used. It was an oversight on the operators that with the smaller step size, this no longer corresponded to a two pixel scan [117].

7.2.2 Corrections to data

PyDetLib [118] was used to read the data files, as well as apply corrections to the datasets. PyDetLib is a data analysis, correction and calibration toolset written by S. Hauf for use at European X-ray Free Electron Laser (European XFEL). PyDetLib is written in python for use in ipython [100] shells, and can also be integrated into the Karabo software framework [99]. It provides tools for accessing data in the native output formats of LPD and Adaptive Gain Integrating Pixel Detector (AGIPD), including functions to re-arrange AGIPD datasets, since AGIPD does not output in a continuous row and columnwise format [2].

PyDetLib also provides correction tools for data analysis, in particular offset and common mode corrections. Offset corrections are calculated from a dark image

[2]This is because each AGIPD module has two rows of ASIC chips where the bottom row is rotated 180 degrees compared to the top row. The ASIC chips still output in the same pixel order however, so elements of each AGIPD image need to be reordered and inverted to represent their physical position.

dataset, where the detector is activated without input signal. This provides the baseline offset for each pixel that can be applied as a correction directly to the dataset. The other major correction PyDetLib provides is common mode correction. This takes a the median value of a row or column and applies it as a correction to the dataset. The choice of row or column can be made when applying the correction, as well as limiting the rows and columns to subsets of the image, such as by considering each ASIC separately. This correction is intended to correct for row or column wide offsets that occur at the time of data collection, and therefore cannot be corrected for in advance. PyDetLib also integrates into the European XFEL calibration database, which keeps track of calibration constants on a per-module basis, along with time preiods for which they are valid.

PyDetLib also provides analysis tools, such as display of data images. This has been used to create Figs. 7.7, 7.8, 7.9 and 7.10. Other tools provided PyDetLib but that were not used in this analysis include histogramming of data, statistics calculation, event clustering and computation on Graphical Processing Units (GPUs). The Karabo integration of PyDetLib was not required as data analysis was performed on a single machine. Additionally dark images taken after the experiment were used for calibration, as the calibration database was not in operation for the test detector taken to APS.

Following the 20 µm dataset experiment, 100 files of dark images, where the beam was blocked and only background was recorded, were taken with the 20 µm dataset file structure. This provided calculation of the background pixel offset and noise values on a per pixel and memory cell basis. These were then used by PyDetLib to correct the data, and by the X-CSIT simulation of the experiment.

However, problems with the initial dark image set were discovered. The output files produced by LPD have an inverted intensity scale. With the 12-bit Analogue to Digital Converter (ADC) this meant that the maximum output number of 4095 represented a signal of no charge, and 0 represented maximum input signal. This is subsequently flipped in PyDetLib. The dark image dataset taken for the stepping experiments had data values in the range 8192 to 12 287. Because only six mod-

ules were attached to a sixteen module super-module, ten modules in each image represent detector regions where no signal was received by the Front End Module (FEM). Some of these regions always output a signal of zero, some an entirely random signal, and some a maximum signal. Comparing these regions in the datasets and the dark images (e.g. 7.7 and Fig. 7.8) shows that the dark image dataset is still inverted. In addition to this mismatch in intensities, several detector artefacts, including dead pixels, that were noticeable in some or all of the dataset images[3] were not present in the dark image dataset. In light of these inconsistencies it was decided that the dark images did not accurately represent the detector baseline, possibly due to a configuration or output error. Instead alternate dark images were sourced from the following experiment. This consisted of 30 files with the same data format as the 20 µm dataset. These dark image files were taken after the intervening experiment[4]. However, this experiment would not have caused significant radiation damage to the detector, and therefore the dark images remain a good description of the detector baseline for the 50 µm and 20 µm experiments. Comparing Fig. 7.9 to Fig. 7.7 it can be seen that these dark images much better match the data images than the original dark image seen in Fig. 7.8.

The new dark image set was then used with PyDetLib to generate a per-pixel and image number offset and noise map. These maps were then used by PyDetLib to perform offset and common mode corrections (as described above) respectively to all of the data images in both the 20 µm and 50 µm datasets. Additionally all images, including the offset and noise maps, were cropped down to a 64×64 pixel region of columns 192 to 255 and rows 160 to 223. This encompasses the right half of the module the beam landed in, as seen in Fig. 7.7, and the right half of the module above it. This significantly cuts on image processing time and memory usage, while still retaining the data region and unilluminated background pixels both on the same module, and on an adjacent module.

[3] Including images that did not contain beam.

[4] This experiment was part of the LPD testing process and is not relevant to this analysis.

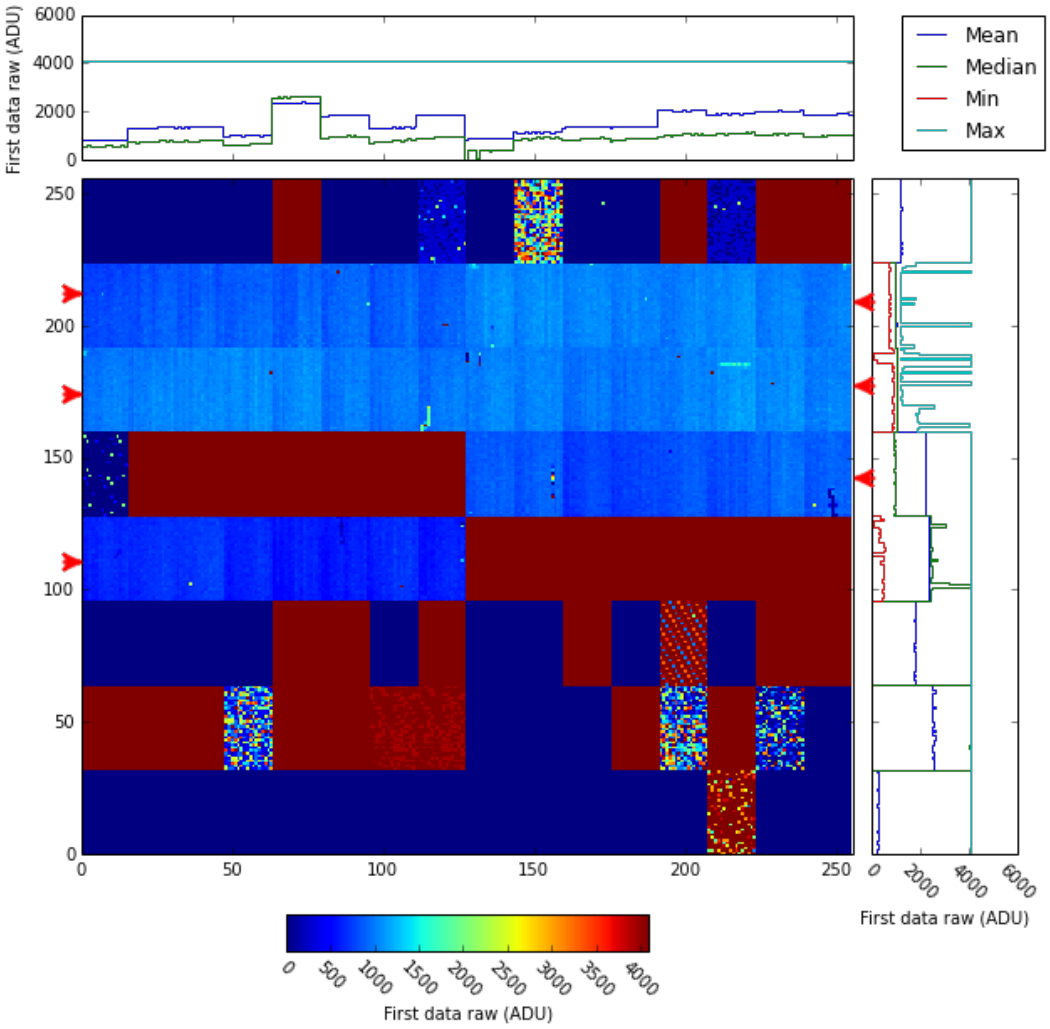


Figure 7.7: Example image, showing the first beam-containing image in run 1344. *Red arrows indicate the six modules that were attached to the detector, and the beam can be seen as a horizontal line at* $x = 210, y = 185$, *in the middle module on the right side of the detector.*

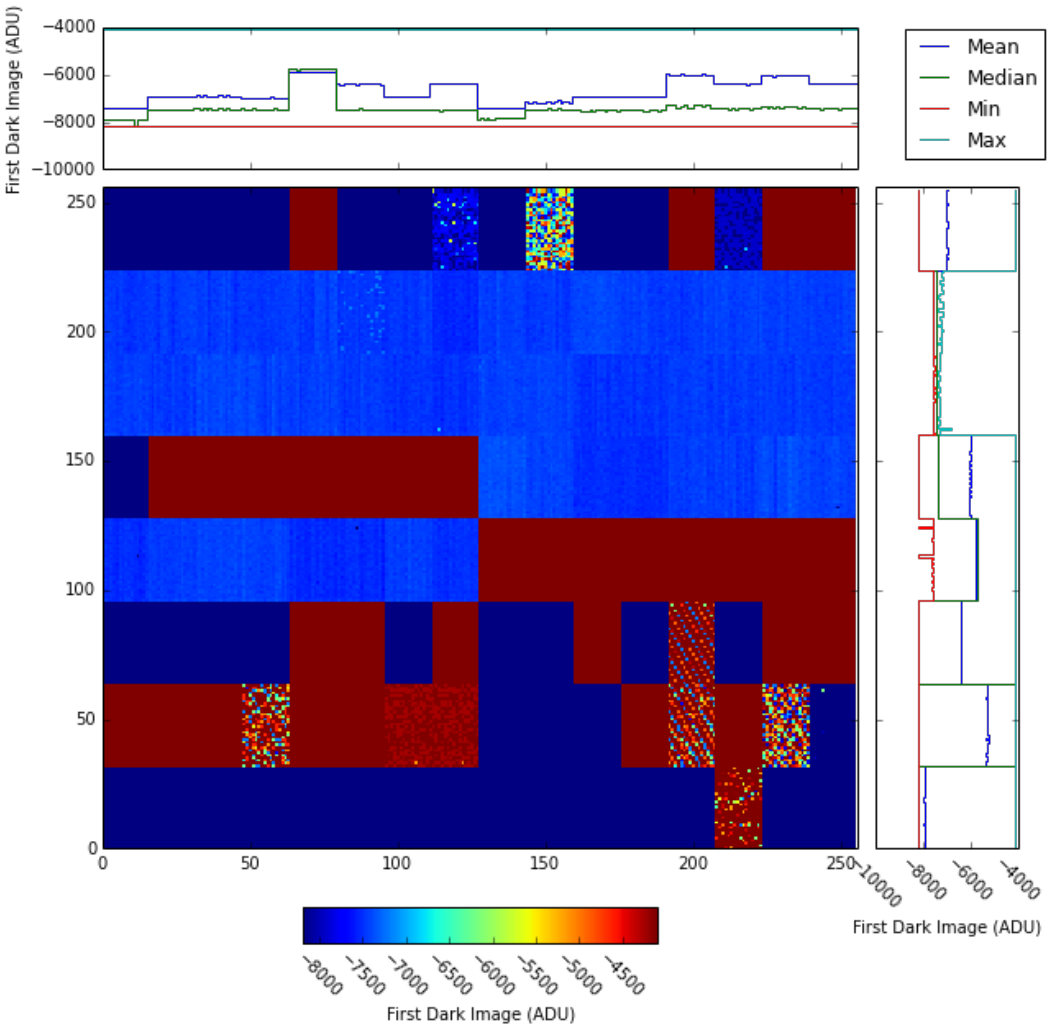


Figure 7.8: An example image from the initial, faulty dark image dataset. The beam signal seen in Fig. 7.7 is not present as expected, but most of the background noise and detector artifacts are also missing. The automatic scale PyDetLib has drawn is also different (-8200 *to* -4100*) to the one in Fig. 7.7 (*0 *to* 4100*).*

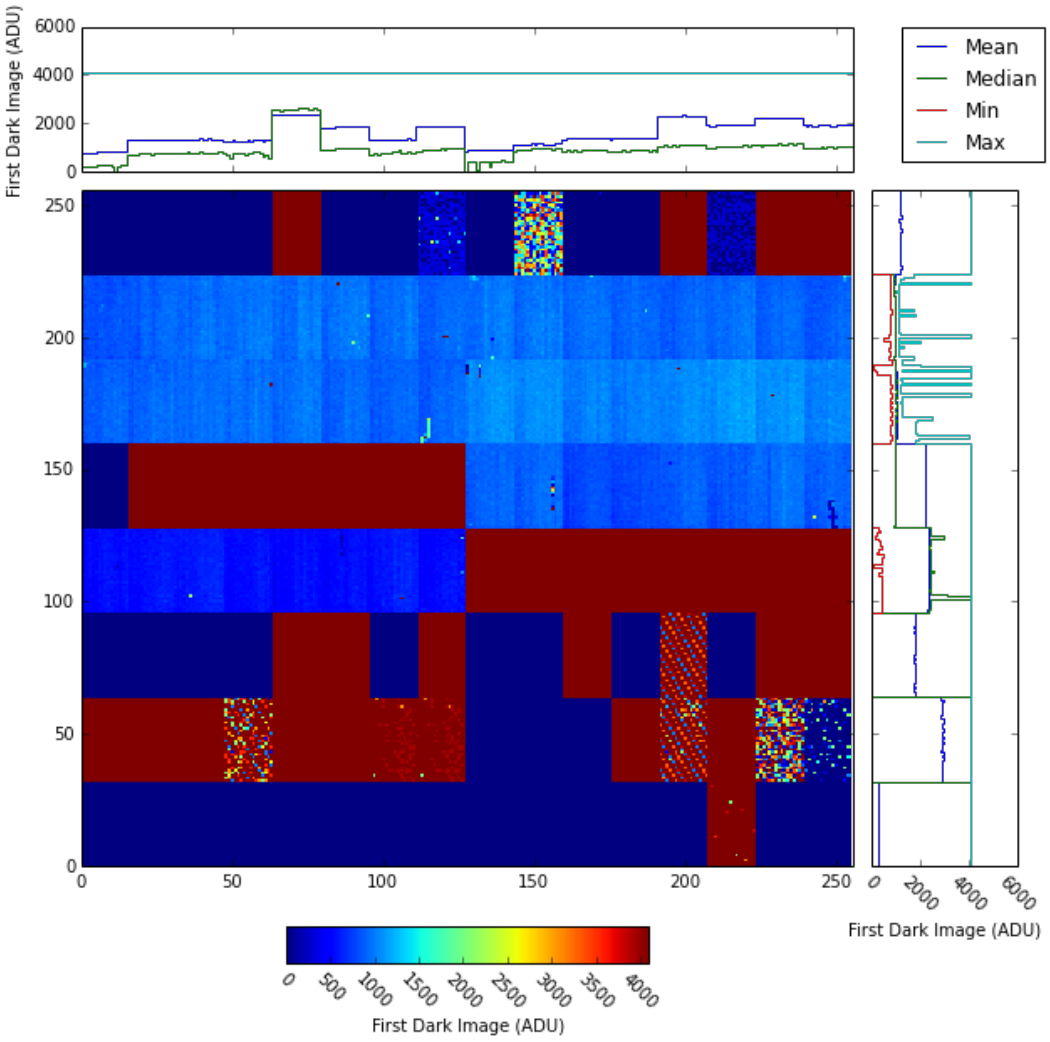


Figure 7.9: An example image from the new dark image dataset. The detector artefacts seen in the data images are replicated here, as is the scale.

7.2.3 Background signal

During analysis of the 20 µm dataset, compilation images were created for each run, summing together all data images in that run. It was noticed that in all of these images the values of background pixels, those pixels that neither contained beam signal or detector artefacts, were all negative. Because these images had already received the full set of offset and common mode corrections from PyDetLib, the average value of these background pixels should have been zero.

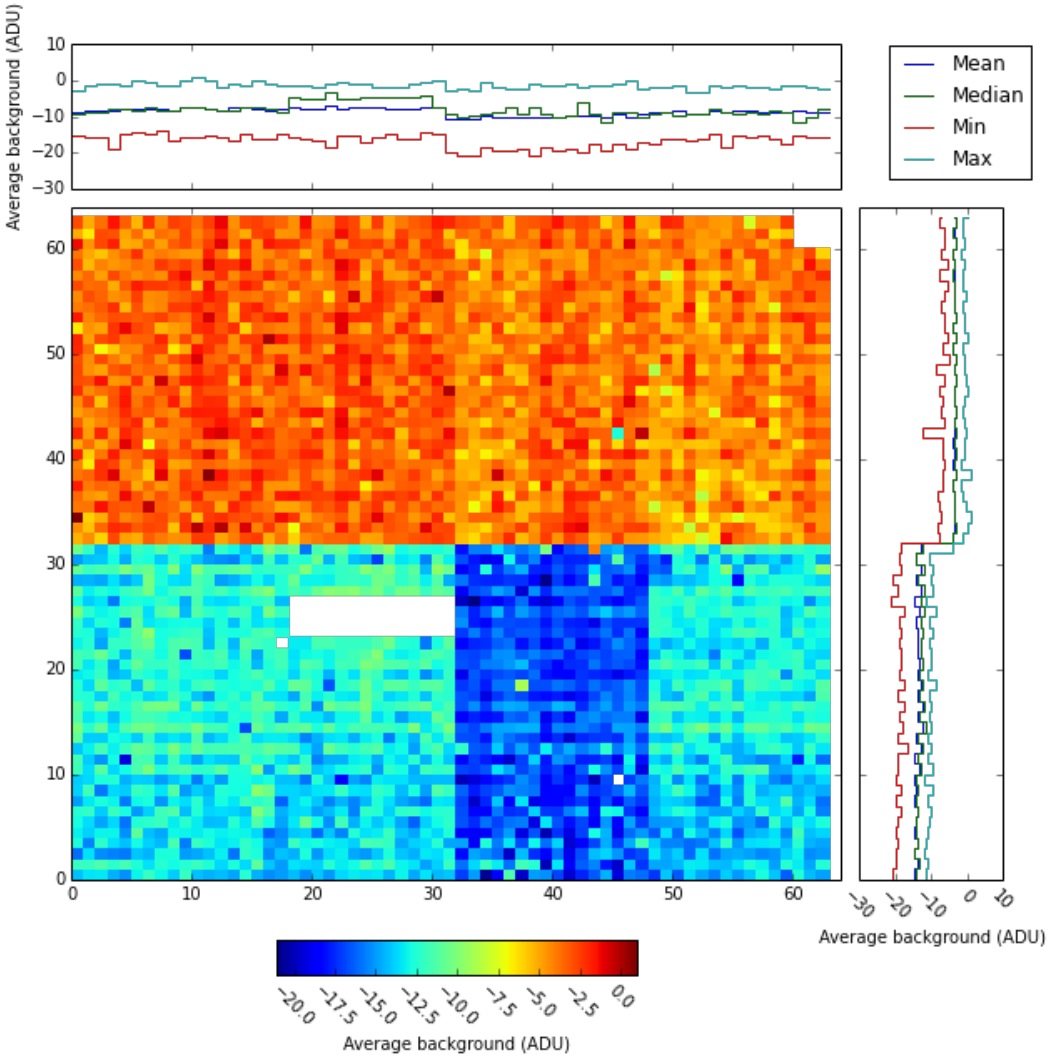


Figure 7.10: The average intensity of all beam images in the complete 20 µm *dataset, with all detector artefacts and the beam region masked. The negative value of all background pixels can be seen.*

It was seen in these compilation images that the average background offset varied from ASIC to ASIC, with the ASICs in the unilluminated sensor that did not receive beam having higher, but still negative baseline values. A sum of all data images, normalised to each constituent image, can be seen in Fig. 7.10. Plotting the mean background value of each ASIC in the illuminated sensor, where pixels in the beam region and pixels with detector artefacts were excluded, in Fig. 7.11 shows similar trends per run of ASICs 1, 2&4 with roughly the same background, while ASIC 3 returns a lower value. However there are large variations between each run, with no discernible pattern. A histogram of the average value of each pixel over all images, Fig. 7.12, also shows the variation between ASICs and sensor, as well as showing the spread of values and confirming the large variation of all ASIC offsets from zero.

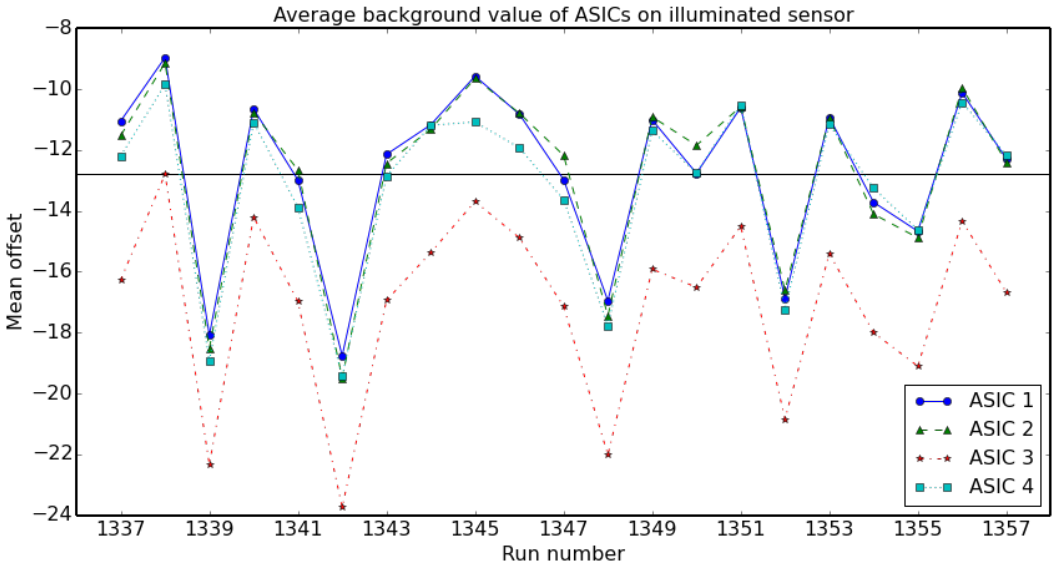


Figure 7.11: The average background pixel value of ASICs on the illuminated sensor, over all runs in the 20 µm *dataset.*

The cause of this background offset was discovered by calculating the average background levels of all individual images, including those that did not contain beam data. For consistency, all images used the same definition of background pixels, excluding the data region and all dead pixels even in images that did not contain them. Plotting these background means over image number showed a repeating pat-

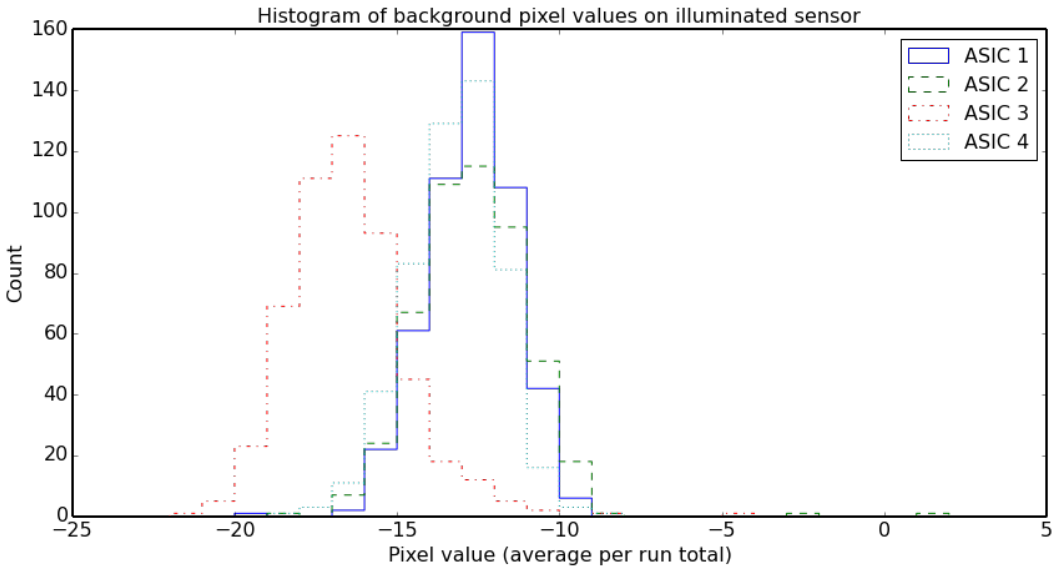


Figure 7.12: A histogram of background pixel value of ASICs on the illuminated sensor in the 20 µm *dataset.*

tern seen in Fig. 7.13. This background signal appears as an asymmetric sinusoidal wave, with the negative half of each cycle greater in intensity than the positive half. This miss match in the intensity of the positive and negative halves of the cycle is why the average contribution to the data was negative, and thus why the signal was noticed. In addition, only every other wavelength appears, with a gap between them showing no background offset at all. This pattern appears in all trains of 32 images, however only maintains phase within each train. Within each train, each image was captured at a constant rate and the image number can be considered a time axis, however between each train an unknown time period elapsed. This consistency of the signal within each train, but not between them, seems to indicate that the signal of LPD is affected at the image capture stage, not at the output and digitization stage between trains.

As shown in Fig. 7.14, the signal is present in both sensors, though the signal orientation was reversed in the unilluminated sensor compared to the illuminated sensor. Both signals show a greater negative half of the cycle than the positive half. Additionally, in the illuminated sensor, a smaller peak appears following some of the signals. Interestingly the signals in the illuminated and unilluminated sensor do

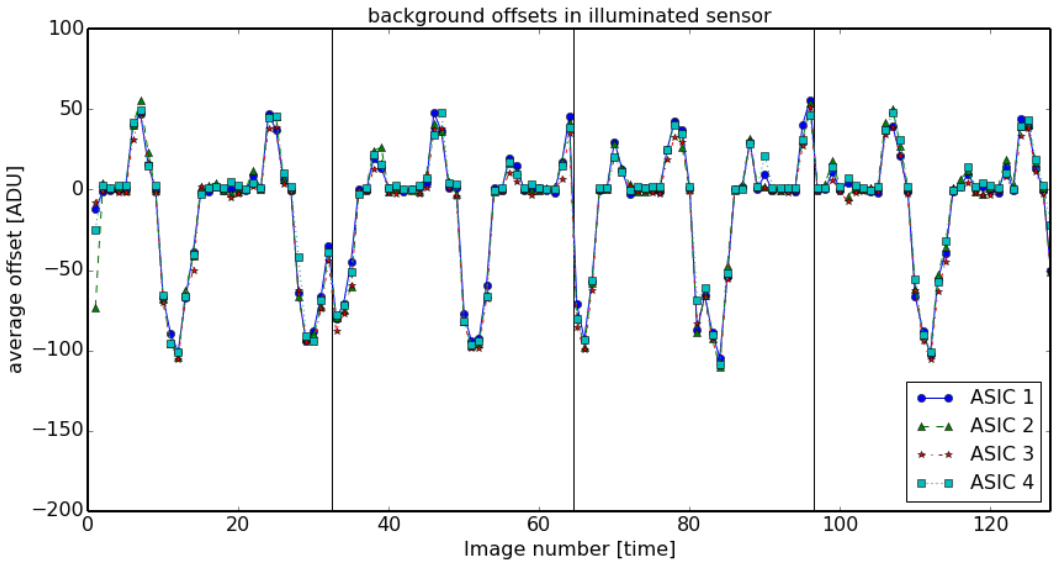


Figure 7.13: Plot of the average background value of each ASIC of the illuminated sensor, over the first 128 *images in the second file of the* 20 µm *dataset. Each black line denotes the break point between 32 image trains.*

not keep phase with each other between trains.

This background signal was found in all data available, including the 50 µm and 20 µm datasets as well as the dark images. As can be seen from the plots however, data images that do not contain background signal have a correct average value of zero, and therefore the offset values calculated from the dark images are correct. So the offset calculator function of PyDetLib is correctly excluding images that show this background signal.

The background signal was initially corrected for by applying the per-image average background of the data-containing ASIC as an offset value on the data region. Assuming the background signal is a static offset value, this would correct the data region to its original value. To test this correction a scatter plot, Fig. 7.15, shows the total sum of beam intensity after the background signal correction, against the average background that was applied as the final correction. Because the beam intensity is after all corrections it should have a consistent zero, and therefore there should be no correlation between the two quantities.

It can be seen from this plot that the data points can be split into four regions.

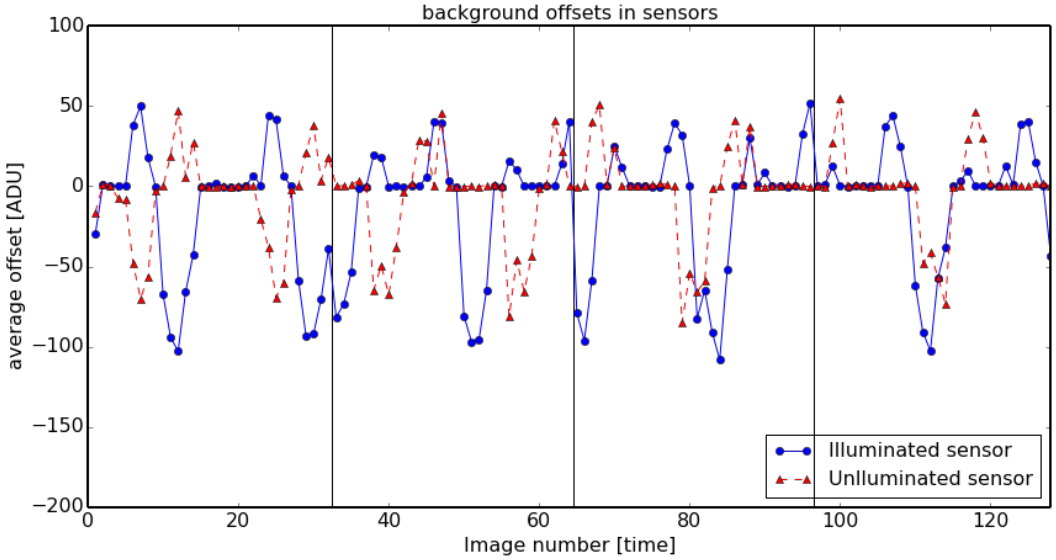


Figure 7.14: Plot of the average background value of each sensor over the first 128 *images in the second file of the* 20 µm *dataset. The sensor see a reversed signal compared to each other, and the signal does not keep relative phase across trains.*

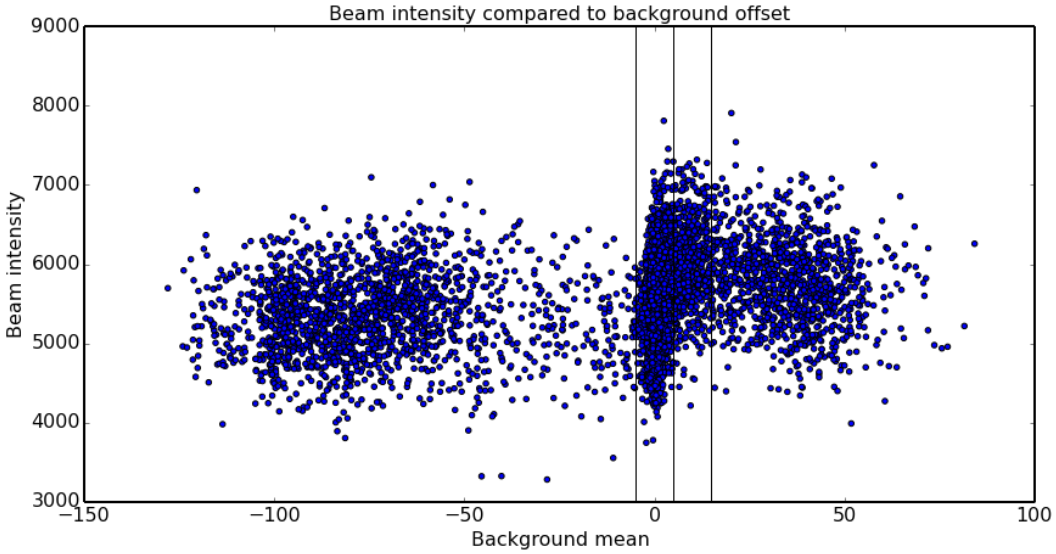


Figure 7.15: Two dimensional histogram of each image in the 20 µm *dataset, showing the beam intensity after background correction, against the per pixel background offset in that image. Vertical lines delineates the four regions of interest considered in the following histogram.*

The first (lower than -5 per pixel) where the background offset is significantly negative, the last (more than $+15$ per pixel) where background offset is significantly positive. The other two regions are the images unaffected by background offset (between -5 to 5 per pixel) and a smaller region between 5 to 15 per pixel, where the beam intensity is noticeably higher than in any other region. Compared to the baseline region without background offset, the images with negative background appear to have a lower average beam intensity, though the difference is smaller than the spread of beam intensities. Likewise images with high background also show a slightly higher average beam intensity.

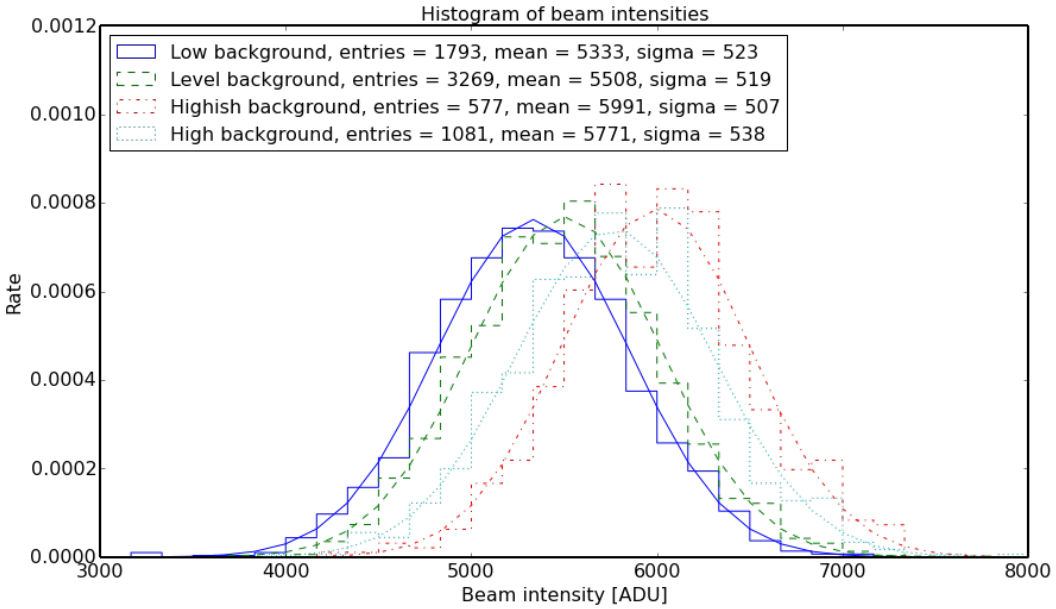


Figure 7.16: A histogram of beam intensities of the four regions of background offset from the previous graph.

This finding is expanded on in a histogram of beam intensity, divided into the regions discussed above, Fig. 7.16. Here the trend of images with negative background offsets to have lower beam intensities, and images with positive background offsets to have higher beam intensities can be seen clearly. The fourth region, where a moderate positive offset is seen, is also shown, which has the highest average beam intensity.

The same trends and conclusions were also seen in identical plots, Fig. 7.17

and 7.18, made with the 50 µm dataset.

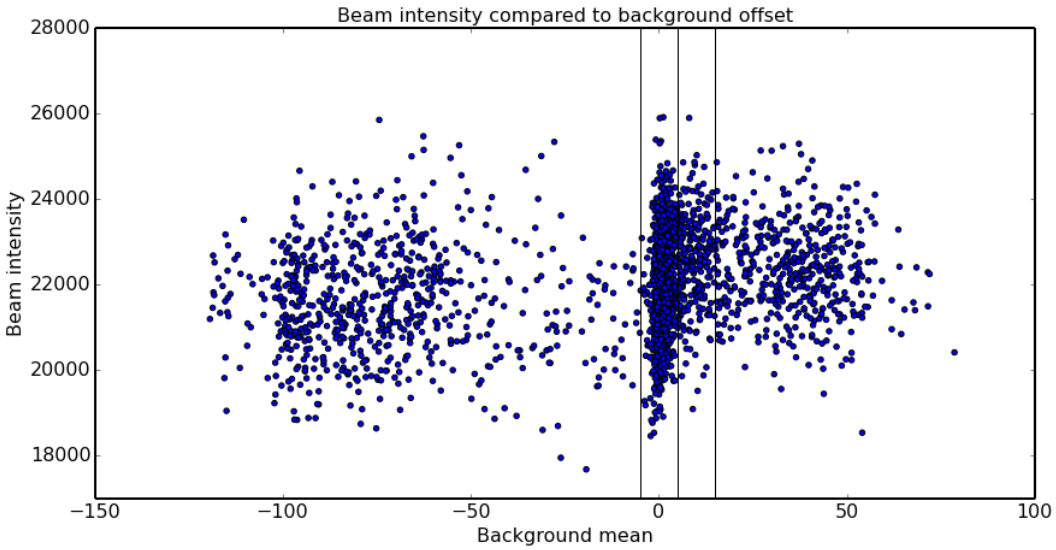


Figure 7.17: Scatter plot of each image in the 50 µm *dataset, showing the beam intensity after background correction, against the per pixel background offset in that image.*

The difference in average beam intensities varies between regions by about, or slightly less than, the standard deviation of the spreads themselves. While the correction goes a long way towards adjusting the images with background offset to be more consistent with those that do not have background offset, it cannot be said that it completely compensates for it. The correction must be deemed to have failed, and in all datasets, images containing background offsets will be excluded entirely. This removes approximately 50% of all data images, although the random position of the signal within trains means that the amount of images removed varies from run to run.

7.3 X-CSIT simulations

X-CSIT was configured to simulate the same 64×64 pixel region used to analyse the original data files. To do this, two full LPD modules of 128×32 pixels were simulated in the physics and charge simulations. The electronics simulation used the offset and noise maps calculated from the dark image datasets as input to the offset

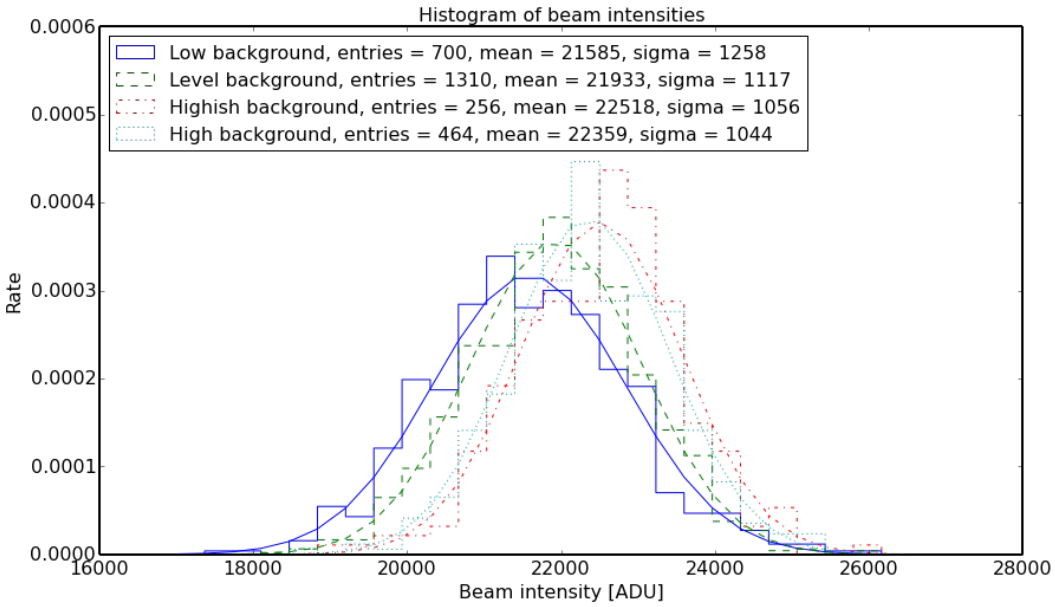


Figure 7.18: A histogram of beam intensities of the four regions of background offset from the 50 µm *dataset.*

and noise functions respectively. Here only the 64×64 pixel region of interest was given full simulation attention.

The simulation was started by firing photons into the detector at 18 keV. Each photon was randomly positioned within the beam area, which was 5000 µm horizontally and of a variable vertical width. Position within the beam region used a flat random distribution profile. The number of photons to be fired in total was also varied, this was randomised using a Gaussian distribution.

Each simulation run using X-CSIT used a number of images corresponding to the number of images taken with the original datasets. Simulations replicating the 50 µm dataset used 210 images per run, with the noise and offset maps from image 12. Simulations replicating the 20 µm dataset used 160 images with the noise and offset maps for each of image 12 and 28, for 320 images per run.

7.4 Comparison of simulations to data

7.4.1 50 µm dataset

The 50 µm dataset consists of 13 runs, each a single beam position in the scan. The first run in the dataset positioned the beam such that charge was split between rows 29 and 30, with approximately 59% of the charge deposited into row 29. Unfortunately the following run does not split the charge between two rows, with all of the charge deposited in row 29. Therefore while the first run does show a pixel crossing, without at least a second associated data point the beam crossing cannot be used for analysis. Later in the dataset, the tenth and eleventh runs split the charge of the beam between rows 28 and 29. This boundary crossing is seen in two data points and can therefore be used for analysis. Usefully run number 1333 appears to be positioned close to the beam edge, where sensitivity to charge sharing is greatest, with 97% of the charge deposited in row 29. Run number 1334 splits charge approximately 36/64% between rows 29 and 28. Comparisons with simulations shown later in this subsection will demonstrate that this data point is still within the charge sharing range of the boundary.

It was known from analysing the experimental data that the total amount of charge absorbed from the beam varied from image to image, as seen in Fig. 7.18. In the case of the 50 µm dataset there was a variation of about 5%. This variation in measured beam intensity is a combination of photon absorption statistics, because many photons fully penetrate the detector and are not absorbed, and variations in the initial input beam. Because neither quantity was specified in the experiment, the simulation was first run without variation on the input beam, and the intensity was adjusted until the average simulation output matched the average measured beam. The variation on the simulated beam was then adjusted to match the data, as shown in Fig. 7.19. A sigma of 5% on the input beam was required to match the 50 µm dataset.

The simulation was configured to recreate the scan of the original experiment, but with step sizes of 5 µm, ten times smaller than the step sizes of the original. The beam was scanned from −50 µm to 50 µm from the pixel boundary. The beam

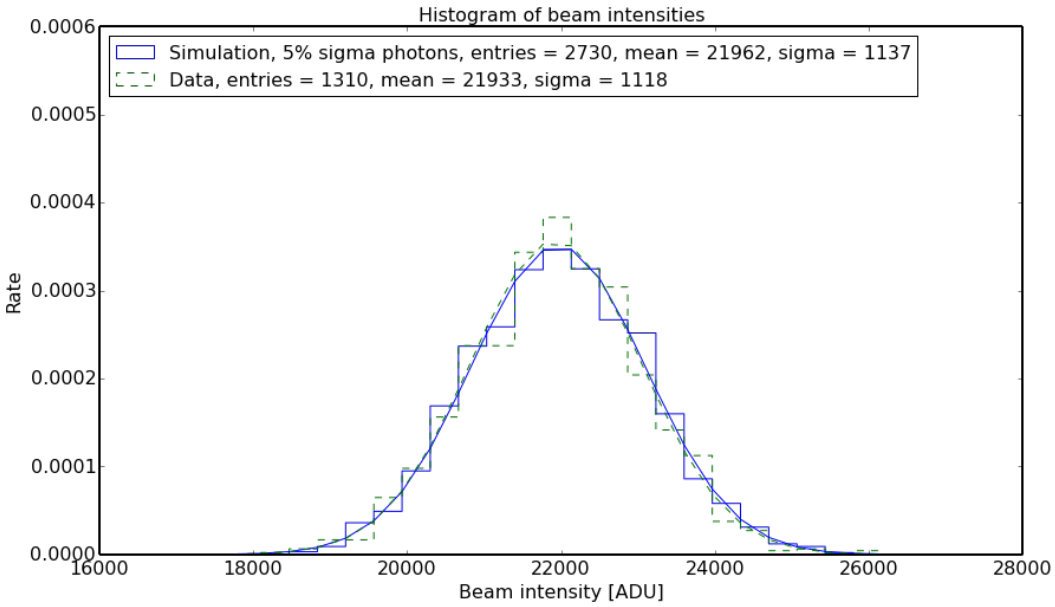


Figure 7.19: 50 µm dataset and simulation beam spread.

width (vertical size) was set to the same 70 µm that the original experiment used. The exact position of the pixel boundary relative to the beam centre is not known in the original data, only the distance between each point. Thus the central point of Fig. 7.20 has been positioned to match the simulations, while the other two points are placed 50 um relative to it. To match the data points, all charge sharing in the charge simulation was linearly scaled compared to the X-CSIT default prediction. The Gaussian shape of the spread (see Section 6.2) was not altered, only the size of it. Fig. 7.20 shows the result of these simulations. Here the y axis shows

$$R = \frac{p_2}{p_1 + p_2} \tag{7.3}$$

where p_1 and p_2 are the number of photons seen in the first and second rows, and R is the relative amount of energy absorbed in the second row. R is therefore a proportion of the beam energy, going from 0 to 1 as the beam is scanned from the first row to the second. It was found that increasing all charge sharing by a factor of 1.4 provided the best fit to the data, indicating that for 18 keV photons, X-CSIT underestimates charge sharing.

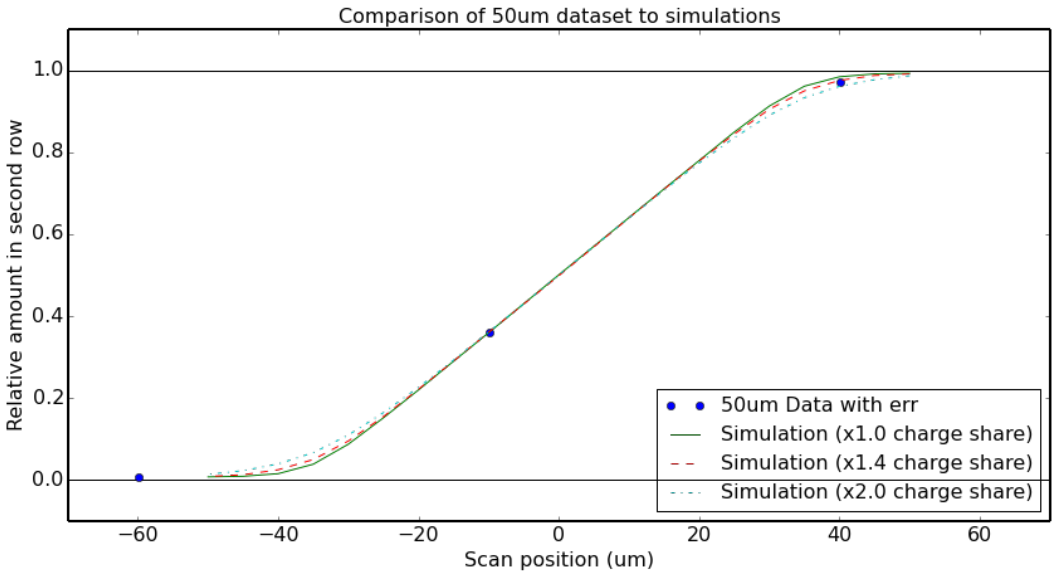


Figure 7.20: Comparison of the 50 µm dataset to X-CSIT simulations with various charge sharing scale factors.

Fig. 7.21 and 7.22 show the same plot as Fig. 7.20, but zoomed in on the two relevant data points, and with vertical error bars calculated as

$$(\Delta R)^2 = \frac{p_1 p_2}{(p_1 + p_2)^3}. \tag{7.4}$$

The horizontal error bars on the data points are taken from the <1 µm repeatability specification[119] of the LIMES 150 high precision linear stage used to increment the detector. The repeatability specification is the deviation on the actual position of the stage if it is repeatedly moved to the same position from the same direction. It is thus taken as the minimum error on the accuracy of the stage. The positional accuracy of the stage is specified as <10 µm/100 mm, which would give an error of 5 nm over the 50 µm step the stage is moved. Thus the higher <1 µm repeatability is used for the error.

7.4.2 20 µm dataset

It was initially envisioned that the 20 µm step dataset would be a repeat of the 50 µm dataset, only changing the step size to a finer value. However, two errors occurred. The first was that the number of steps was not increased to compensate for the

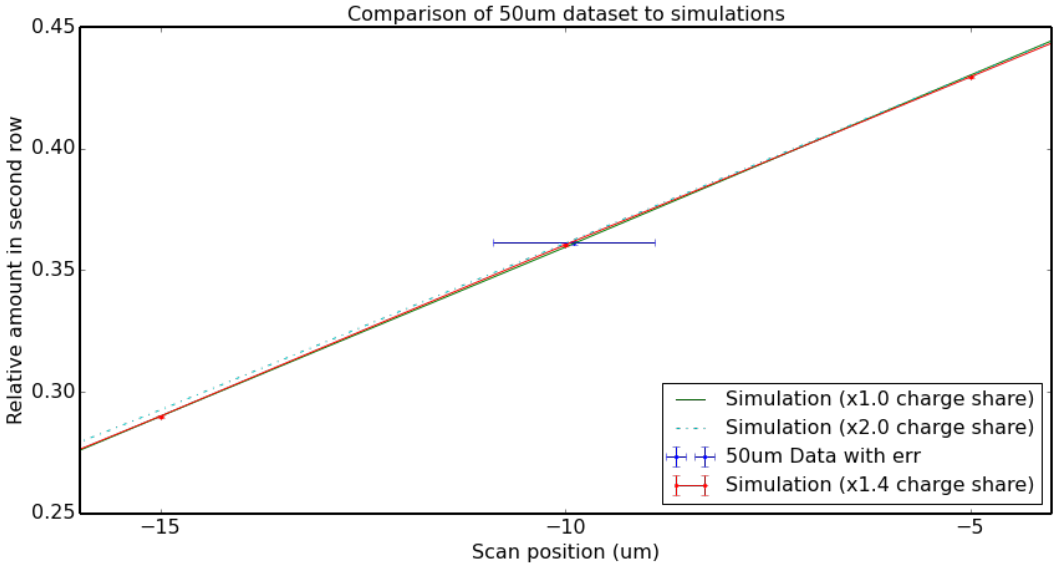


Figure 7.21: Comparison of the 50 µm dataset to various simulations. This is a zoomed view of Fig. 7.20, showing error bars for the data and the $\times 1.4$ charge sharing simulation.

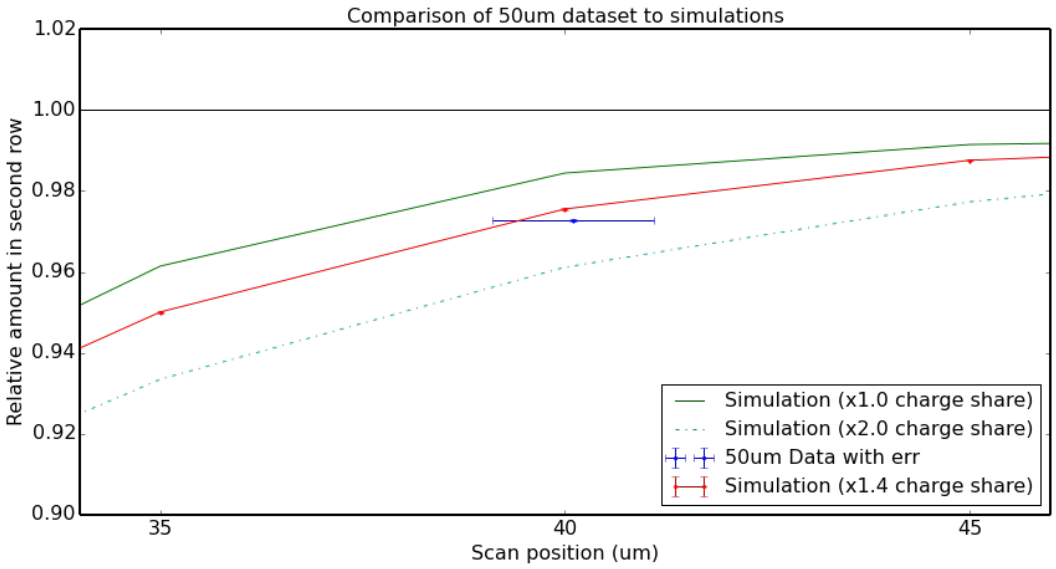


Figure 7.22: Comparison of the 50 µm dataset to various simulations. This is a zoomed view of Fig. 7.20, showing error bars for the data and the $\times 1.4$ charge sharing simulation.

smaller step sizes. As a result only 21 runs were conducted, with a total scan distance of 400 µm, instead of the intended 1000 µm. Luckily, the dataset still contains a pixel boundary crossing, despite the scan distance being smaller than a single pixel of LPD.

The second error was with the beam width. The 20 µm experiment was initially going to use the same beam settings as the 50 µm dataset, including the 70 µm beam width. However, between the experiments efforts were made to lower the beam width further. These efforts were deemed unsuccessful, and the experiment recommenced. As a result of the changes, the beam aperture as recorded in the logbook was set to a width of 200 µm, much larger than the intended 70 µm. Assuming that the scan did still take place, this beam width setting would still have produced good data. However, analysis of the data shows that only two data points, with a spacing of 20 µm, showed a split in charge between two rows. This alone would limit the actual size of the beam to a maximum of 60 µm[5]. Comparison to simulations and the photon intensity of the 50 µm dataset limit the maximum possible beam width to less than 30 µm. Therefore for the 20 µm dataset, the width of the beam is also a free parameter, alongside the beam position and degree of charge sharing. With only two data points, it is not possible to simultaneously determine both the beam width and the degree of charge sharing. Only one can be determined given a fixed assumption of the other.

This subsection will therefore consist of four comparison simulations to the same dataset, under four different assumptions.

- The assumption that X-CSIT correctly predicts charge sharing. This will determine if the X-CSIT prediction is compatible with the 20 µm dataset at all.

- The assumption that charge sharing approaches zero and is very much smaller than the beam width and step sizes. This is not a physically realistic scenario, but provides the absolute upper limit on the beam width.

[5]This assumes three step lengths, no charge sharing at all and the beam perfectly positioned to only cover the pixel crossing in two of the runs.

- The assumption that the beam width is very small compared to charge sharing. This is also not a realistic scenario, but provides an upper limit on how much charge sharing could be occurring, assuming a linear scaling of charge sharing as with the 50 µm dataset simulations.

- Finally the beam width can be estimated by comparing beam intensities in the 20 µm dataset and the 50 µm dataset, the latter of which has a known beam width. Because the initial, uncollimated beam and the width of the collimated beam should not have changed between experiments, the ratio of beam intensities between the two experiments should also provide us with the ratio of beam widths as well.

Before simulation commenced however, as with the 50 µm dataset the variation on the input beam was first calibrated to match that seen in the experimental data. Fig 7.23 shows the spread on measured beam in the images of the 20 µm dataset, compared to the same result from the X-CSIT simulation. As with the 50 µm dataset this was matched by trial and error, and it was found that a 9% sigma on the randomised photon count was required to match the experimental data.

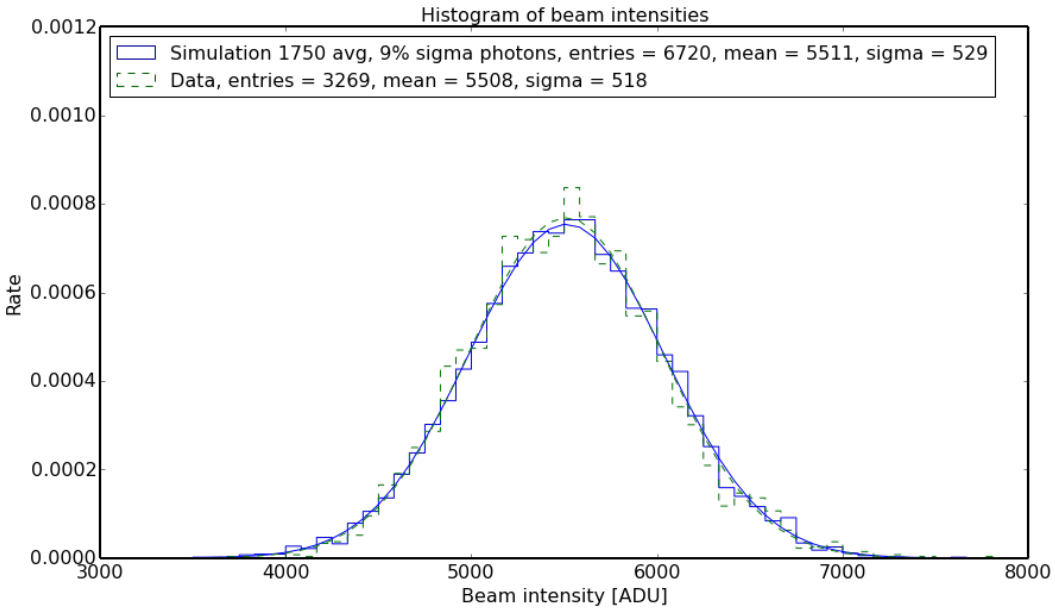


Figure 7.23: 20 µm dataset and simulation beam spread.

X-CSIT prediction

The first series of simulations used an unmodified X-CSIT prediction of charge sharing. The beam position was varied between −30 µm to 30 µm from the pixel boundary in increments of 2 µm, one tenth the step size of the original experiment. The total travel distance of 30 µm either side of the pixel boundary was chosen so that the ends of each simulation would match the next nearest data points to the pixel boundary. It was noted in this simulation, and follow-up simulations discussed later in this section, that the simulations did not predict the long tails of charge splitting seen only in the 20 µm dataset.

As can be seen in Fig. 7.24, simulations with 20 µm and 30 µm were trialled and were found to under and over predict charge splitting respectively. Varying the beam width determined that a beam width of 26 µm gave the best match to data.

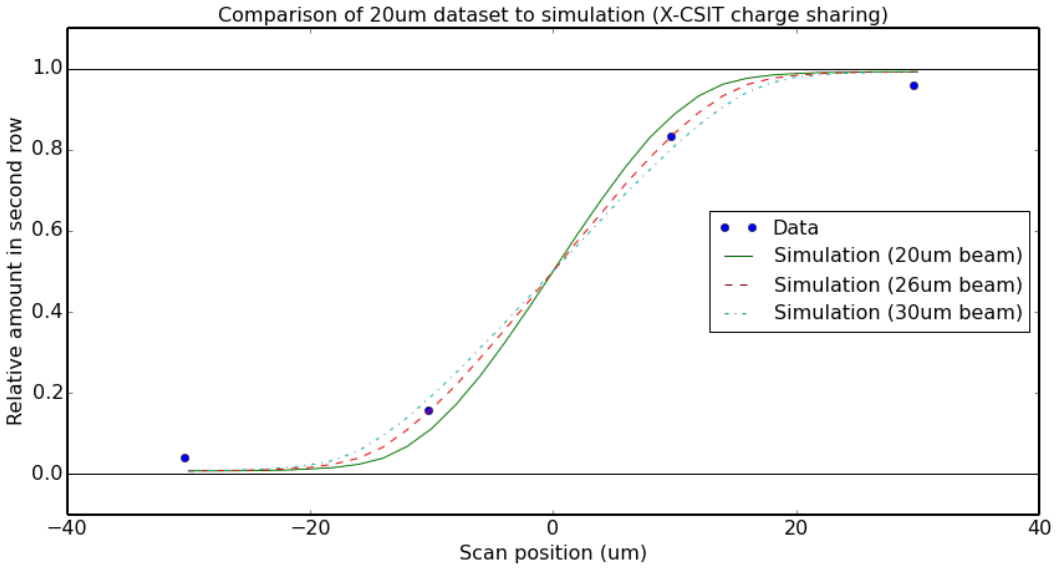


Figure 7.24: Comparison of the 20 µm dataset to unmodified X-CSIT prediction of charge sharing but various beam widths.

Maximum beam width

For this experiment, the charge sharing simulation of X-CSIT was linearly scaled so that it would predict all charge sharing to occur at 1% of the value X-CSIT would normally predict it at. This was done to determine the maximum beam width that would still be consistent with the dataset. While charge sharing was made very

small, it was not turned off for two reasons. The first is that charge sharing is a real observed effect, and removing it entirely would be more non-physical than making it occur over a range two orders of magnitude smaller than predicted. The second is for reliability and consistency reasons. No mechanism was ever written into X-CSIT to turn off the charge sharing function, or to make the range always zero. While it would be possible to write in new functions to not perform charge sharing, this would entail new code, which could lead to bugs or inconsistencies with the other experiments performed using X-CSIT.

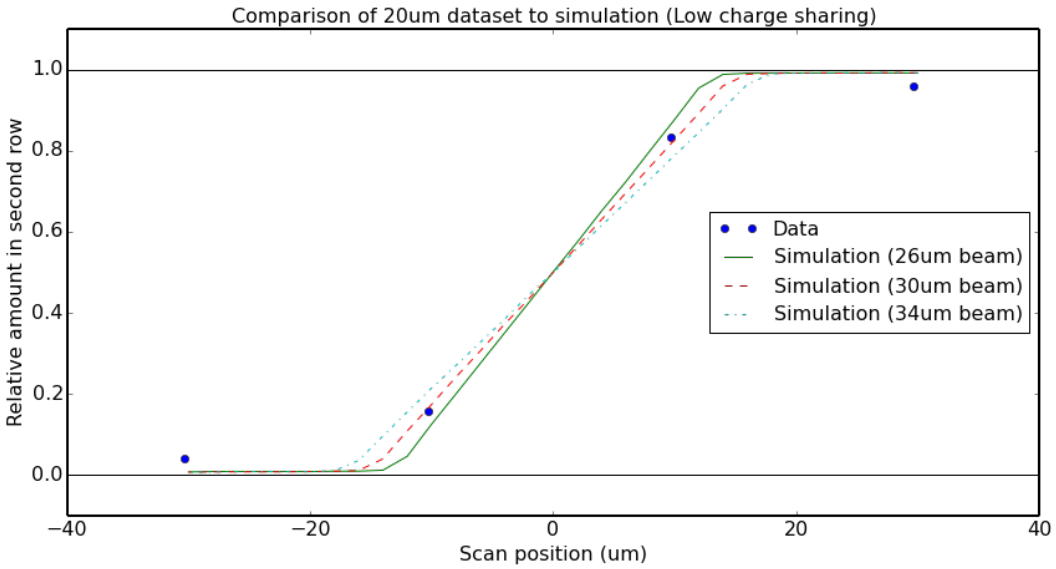


Figure 7.25: Comparison of the 20 µm dataset to X-CSIT simulations with 1% charge sharing and various beam widths.

As can be seen in Fig. 7.25, all simulations show a very linear charge splitting expected from a flat beam with minimal charge sharing. A beam width of 30 µm was found to fit the dataset best, and therefore represents the maximum possible beam width that remains consistent with the dataset. It is interesting to note that the difference in beam widths between this simulation and the one with the X-CSIT prediction of charge sharing is only 4 µm. Therefore a 13.33% variation in the beam width corresponds to a two order of magnitude difference in the charge sharing prediction. The conclusion from this is that not only is this experiment very sensitive to beam size along the charge splitting axis, but also that the data points in

the 20 µm dataset are not ideally placed for greatest sensitivity.

Maximum charge sharing

In these simulations the beam width was set at a fixed value of 1 µm in order to establish an upper limit on the amount of charge sharing that will still be consistent with the dataset. As with before, all charge sharing was extended in range by a linear value, presented in a percentage value compared to X-CSIT default prediction. Fig. 7.26 shows that increasing X-CSIT default charge by a factor of 1.9 sharing fits the dataset best, and is the upper limit on charge sharing still consistent with the 20 µm dataset.

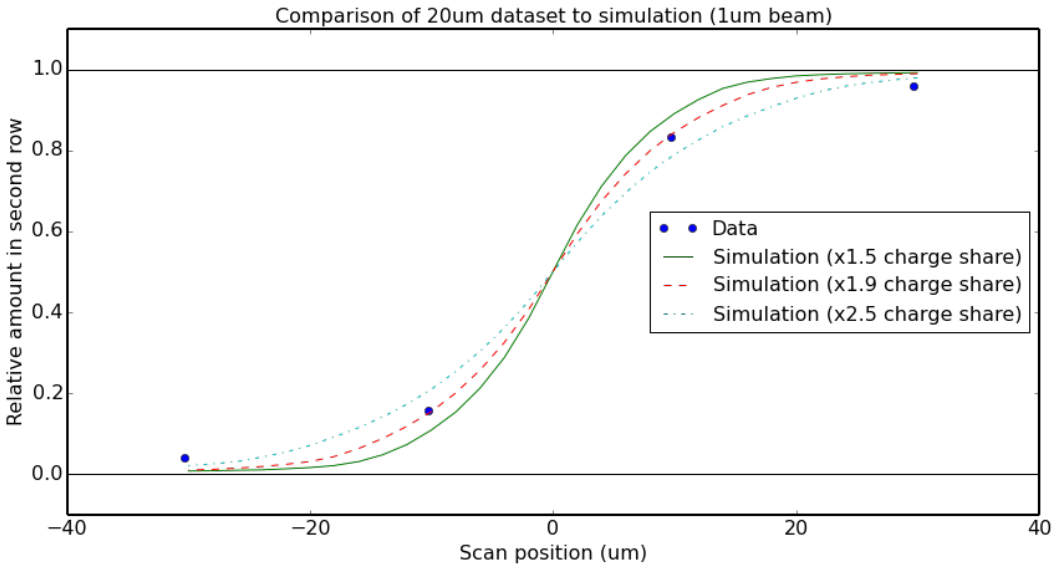


Figure 7.26: Comparison of the 20 µm dataset to X-CSIT simulations with a very small input beam and various charge sharing factors.

Prediction of beam width from 50 µm dataset

Because the beam width of the 50 µm dataset is known to be 70 µm, comparing the beam intensities of the 20 µm and 50 µm datasets can be used to produce an estimate of the beam width in the 20 µm dataset. This comparison assumes that the only change in the beam between the two datasets was a change in beam width. It should be noted that the 20 µm to 30 µm beam widths predicted in this section are smaller than what the hardware used in the experiment was normally capable of

being set to. The reason why the beam width is smaller than in the 50 µm dataset and also not the 200 µm recorded in the logbook is not known. This estimate relies on an assumption that cannot be proven.

Shown in the data box of Fig. 7.19 and 7.23 are the mean beam intensities in ADU of 21933 and 5508 respectively, which with a 70 µm wide beam in the 50 µm leads to an estimation of a 17.58 µm wide beam for the 20 µm dataset.

As can be seen from Fig. 7.27 the 1.4 charge sharing factor from the 50 µm dataset provides a reasonable fit to the 20 µm dataset when using the estimated beam width. Given the similarity of the two experiments it would be expected that the two datasets return a similar or identical result. One possibility for this discrepancy is that the long tails seen in the other data points of the 20 µm dataset are also affecting the positions of run numbers 1344 and 1345.

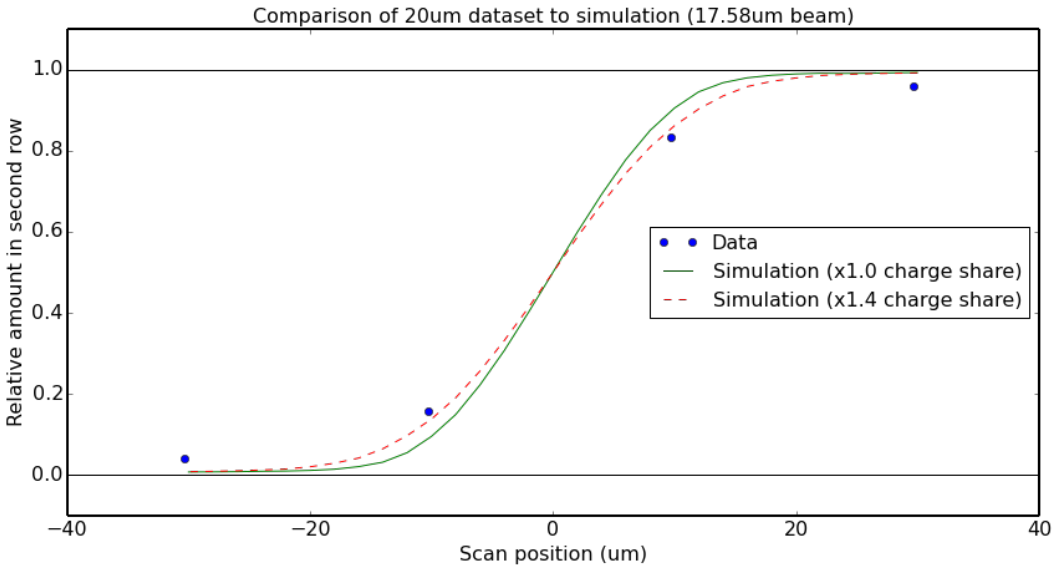


Figure 7.27: Comparison of the 20 µm dataset to X-CSIT simulations with a 17.58 µm wide beam.

To test the effect of the long tails on the simulation, the proportion of charge seen undergoing charge sharing in runs 1343 and 1346, the next nearest data points to the pixel boundary, were removed from the main beam, and instead added directly to the two pixels. This quantity was chosen because the long tails are seen to have a very flat distribution, only going to zero over a range of 200 µm, so the change

in proportion should not change significantly over the range of the beam scan. The beam was also shortened to 16 µm wide as we are now assuming that some 8% of the absorbed photons are not part of the beam. Fig. 7.28 shows that the charge sharing factor of 1.4 found in the 50 µm dataset is a much better fit. While this analysis of the 20 µm dataset does rely on the accurate measurement of the beam aperture in the 50 µm dataset, and is therefore not independent of it, this finding does reinforce the result found previously with the 50 µm dataset.

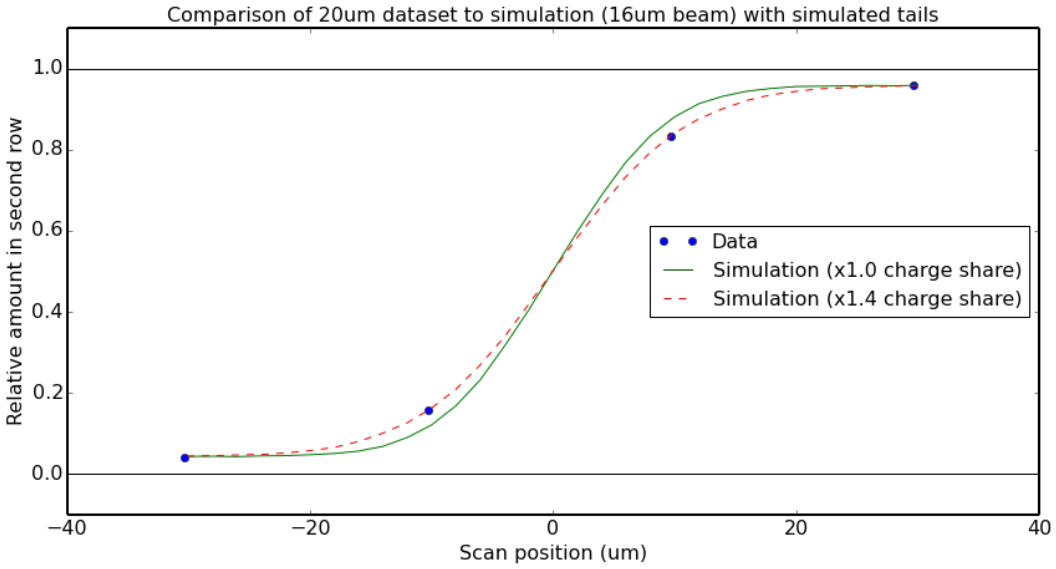


Figure 7.28: Comparison of the 20 µm dataset to X-CSIT simulations with a 16 µm wide beam with simulated tails based upon the next nearest data points.

Analysis of long tails

As has been mentioned previously in the Chapter, in addition to the beam crossing observed in the 20 µm dataset, charge splitting between the two rows is also observed on a much longer scale, the data points seen below run 1344 and above 1345 do not approach 0 and 1 respectively until more than 100 µm from the pixel boundary. Fig. 7.29 shows a simulated scan using X-CSIT predicted charge sharing. The beam is scanned from −30 µm to 30 µm from the pixel boundary in 2 µm step sizes as with previous simulations. From 30 µm to 250 µm the beam is scanned in 5 µm step sizes to lower runtime. The two lines are otherwise identical simulations. As can be seen, X-CSIT does not predict the long tails seen in the 20 µm dataset.

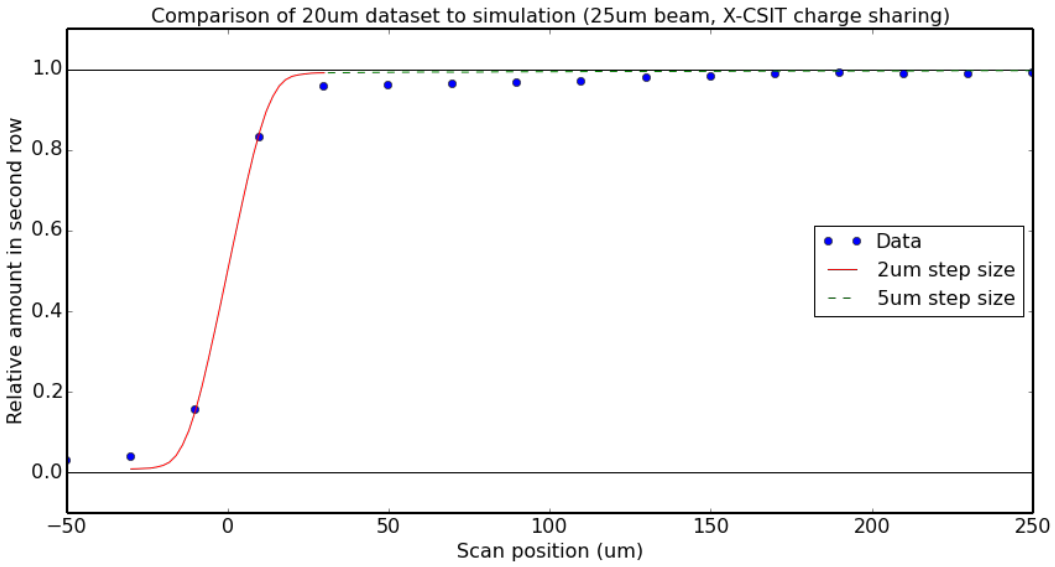


Figure 7.29: Comparison of the 20 µm dataset to X-CSIT simulations with a 25 µm wide beam.

One theory for the presence of the long tails, was that they might be back scatter from inside the detector. The 18 keV photons incident on a 500 µm thick silicon detector only have around a 50% absorption rate[6], with the other half of the photons penetrating through to the detector. To test this, a more advanced model of the LPD module was built, including the Molybdenum memory shields and mounting plate. A comparison between the two models of LPD is shown in Fig. 7.30. No change in absorption rates or charge proportions was observed.

However, in light of their absence in the 50 µm dataset it should be considered that the long tails are probably not a detector artefact that should be simulated, but an external effect to the detector. It is possible that the presence of the long tails is related to the small beam width observed in the 20 µm dataset, such as an obstruction to the beam line imperfectly collimating it and producing a small amount of off-axis scatter, seen as the long tails.

[6]The 50 µm beam test in Fig. 7.19 shows an average beam intensity of 21962 ADU in the simulation, where 4096 ADU is calibrated to 1000 photons at 12 keV in LPD at this gain stage. This corresponds to the energy of 3575 18 keV photons, or 50% of the 7000 input photons.

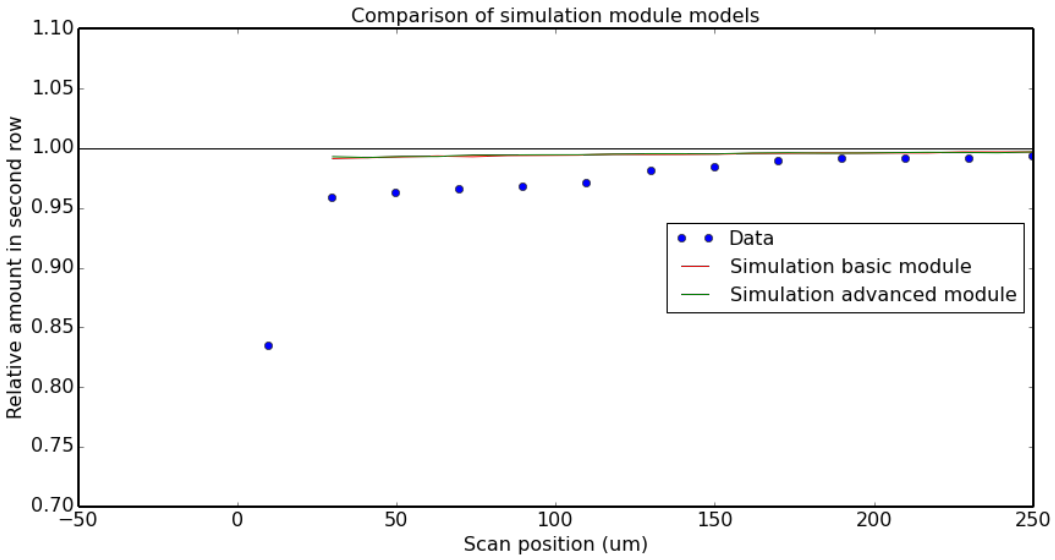


Figure 7.30: Comparisons of the 20 µm dataset and two versions of the detector model in X-CSIT, neither the original, more simple model, nor the more advanced model predict the long tail.

7.5 Conclusion

The analysis of the APS/LPD dataset concludes that X-CSIT underestimates charge sharing at 18 keV, and that to match the dataset, all charge sharing in X-CSIT must be linearly scaled up by a factor of 1.4. As will be discussed in Chapter 8, this charge sharing underestimation may be due to an energy dependent factor to the charge spread. This analysis should therefore be used as part of a larger process of testing and validating X-CSIT with various photon energies.

X-CSIT is principally focused around its modular design the split of physical regimes into different, but linked, simulations. Bespoke simulations were chosen for the charge and electronics simulations

Chapter 8

Status of X-CSIT

8.1 X-CSIT Overview

The X-ray Camera SImulation Toolkit (X-CSIT) is a modular simulation of semiconductor pixel detectors. It is designed to be easily adapted to differing detector designs, including all of the detectors intended to be used at European XFEL. So far, X-CSIT has been used to simulate two of the detectors mentioned in Chapter 3, the Large Pixel Detector (LPD) and pnCCD, and has also been incorporated into Karabo [99], the European XFEL control and data management software framework.

X-CSIT simulates three stages of physics in pixel detectors. It simulates particle interactions, deposition of energy in the semiconductor layer of the detector and scattering using Geant4. The choise of Geant4 for the particle simulation was made because it was felt that a bespoke particle simulation of sufficient accuracy, and which modelled all the desired interactions, would not be an effective use of time given that Geant4 provided all the functionality envisioned. X-CSIT simulates charge spreading and charge sharing between pixels using a statistical spread of charge according to the detector design and the depth of the initial charge deposit. X-CSIT also simulates the detector electronics in a highly modular fashion, wherein functions representing electronic components are applied to a matrix of pixels. A number of such functions are provided with X-CSIT, as is the ability to modify, expand or add to them to create a model of the detector electronics. Bespoke sim-

ulations were chosen for the charge and electronics simulations instead of existing simulations such as Technology Computer Aided Design (TCAD) [19, 20] or Simulation Program with Integrated Circuit Emphasis (SPICE) [120] to maintain control over their features and capabilities, particularly with a view to including plasma simulations in the charge simulation and the on-ASIC storage used by European X-ray Free Electron Laser (European XFEL) detectors. This simplified the process of achieving a fully functional version of X-CSIT and, therefore, proving the core concept of simulation separation. The separation of simulations would then allow the modification of sub-simulations or complete replacement with alternatives should that be desired.

A simulation of charge plasma effects has also been designed for X-CSIT. However, as discussed in Section 6.4, additional test data is required to complete this physics model, and it is therefore not functional at the time of writing.

8.2 Validation of X-CSIT

Two sets of experimental data have been compared to simulations created with X-CSIT.

The first was an experiment conducted with a pnCCD [104, 113]. The sensor was exposed to an Iron-55 source, measuring the K-alpha X-ray emissions, in photon counting mode. The photon flux rates were sufficiently low that for the most part, individual photons were observed by the detector. The analysis centred around counting the rates at which charge was observed being deposited in neighbouring pixels, these mostly representing events where single photons had shared their charge between two, three and four pixels. This experiment was replicated in X-CSIT, the output from which was analysed using the same method.

Comparison of the charge sharing rates, and the energy spectrum of each event type, gave indirect insight into the range over which charge sharing occurred, given an even distribution of photons. The conclusion was that at the 5.9 keV photon energy observed, X-CSIT gave a reasonable match to simulation.

It should also be noted, that at $\sim$5.9 keV, the incident photons have a much shorter mean free path in the semiconductor than those expected to be used at European XFEL. Most of the photons will have been absorbed close to the illuminated surface of the detector. Because pnCCD is a reverse side illuminated detector, this will have maximised charge sharing, but also minimised the spread in interaction depth and the effect of the particle simulation on the output.

The second comparison of X-CSIT to real data is the analysis detailed in Chapter 7. This consisted of a thin beam scanned across a pixel boundary. Charge sharing was then observed from the rate at which the charge moved from one pixel to the next as the beam was scanned. The split in charge between the two pixels was due to both the width of the beam and charge sharing. However, comparisons could still be made to simulations because of the known beam width, and charge sharing could be measured from the relationship between scan position and charge splitting.

This analysis concluded that with photons of 18 keV, X-CSIT under-estimated charge sharing. To match the data, all charge sharing ranges had to be increased by

a factor of 1.4.

Taken together, these two analyses would seem to indicate the existence of a charge sharing factor dependent on the energy of the charge deposit. As discussed in Section 6.2, this should logically exist, but was not quantified and was expected to be small enough to ignore for X-CSIT.

The next steps for X-CSIT should therefore be to search for this charge deposit energy factor to the charge sharing simulation. The two analyses presented would on their own not be sufficient to determine the nature of this effect, three datasets with different energies should be used, ideally all using the same method. Section 7.1 provides a description of both the experiment used with the 18 keV dataset, and also of how the experiment could be refined, and this would present the best option for investigating charge sharing further, particularly as European XFEL begins its commissioning process. Despite its shortcomings, the 18 keV dataset presented in Chapter 7 is still good data, and could be used to help create and test any future addition of energy dependent charge sharing.

Chapter 9

Conclusion

X-ray Camera SImulation Toolkit (X-CSIT) is a simulation toolkit that enables users to create simulations of a wide variety of X-ray semiconductor pixel detectors. The initial design goal was to produce a common software framework that could be used to simulate all of the detectors to be used at the European XFEL, focusing initially on the three bespoke detectors designed for it (see Chapter 3).

Meeting this goal required designing a modular and highly adaptable system, that is open to modification of core components with minimal or no effect on the rest of the simulation. As part of this design, X-CSIT is split into three sub-simulations. The particle simulation, using Geant4 [3, 4], simulates the particles incident upon the detector. This includes travel through the detector materials, interactions, energy depositing and generation of secondary particles. The charge simulation is a bespoke design that generates mobile charge in the semiconductor layer from the energy deposits in the particle simulation. It then calculates the spread of that charge and how it will be collected by the pixel grid, including charge sharing. Finally the electronics simulation sequentially applies the effect of the electronics of the detector to the charge collected in the charge simulation. Since the electronics of detectors are bespoke and can differer greatly, this is the most open component of X-CSIT. Many functions are provided to simulate common components or effects. A user is then able to apply them as needed to a matrix representing the pixel grid, or introduce their own functions. The open design of X-CSIT also allows modification to improve the base simulation, so that the accuracy of the simulation can be

improved without re-writing each detectors simulation individually. This includes up to whole sub-simulations of X-CSIT, which could be replaced entirely without affecting the other simulations due to the clearly defined communication between them.

The adaptable design of X-CSIT, intended for the many detectors expected at European XFEL, also opens the software to applications beyond its initial scope. The simulation of pnCCD demonstrates that X-CSIT can simulate the multitude of CCDs in use, and it is just as capable of simulating pixel detectors at other laser light sources as at European XFEL.

Additional to the physics currently simulated by X-CSIT, the high peak photon fluxes expected to be seen at the European XFEL are expected to create charge plasmas. X-CSIT has been designed with the expectation that at some concentration of charge, it will stop considering charge clouds individually and switch to a model of them as a group experiencing charge plasma effects. A preliminary design for how to identify regions where charge plasmas are occurring has been written and implemented in X-CSIT. This work is available to form the foundation of an expansion to X-CSIT that simulates charge plasma effects. However, no data was available to tune the simulation to where charge plasma effects should take over, or what effect they should have on the spread of charge in the semiconductor. Creating charge plasmas with photons requires large concentrations of X-rays onto a µm sized target over a few ns. Varying the beam intensity and scanning the beam between two pixels would allow measurement of where charge sharing begins to occur, and the effect it has on charge sharing. These two quantities are likely to be linked and energy dependent. The European XFEL itself will be capable of producing these events, and it is hoped by members of the group that time will be found during commissioning to conduct such an experiment.

The rest of the simulation, at the time of writing, remains partially validated. The analysis of pnCCD data by T. Rüter [104, 113] showed an approximate match of charge sharing in X-CSIT to real data with $\sim$6 keV photons. The Large Pixel Detector (LPD) analysis, see Chapter 7, showed an underestimation with 18 keV photons.

Further testing and tuning of the charge simulation is therefore desired. Chapter 7 describes the setup used to test LPD at Advanced Photon Source (APS), as well as how such an experiment could be improved. Conducting such experiments at differing energy bands, particularly at the principle European XFEL energy of 12 keV, should then be used to identify any energy dependence to charge sharing.

Bibliography

[1] M. Benoit, et al., "Simulation of charge collection processes in semiconductor CdZnTe γ-ray detectors", *Nucl. Instr. and Meth. A*, 606, 508, (2009).

[2] M.E. Myronakis, et al., "Computational modelling of pixelated CdZnTe detectors for x- and γ- ray imaging applications", *JINST*, 7, P03004, (2012).

[3] S. Agostinelli, et al., "Geant4-a simulation toolkit", *Nucl. Instr. and Meth. A*, 506, 250, (2003).

[4] J. Allison, et al., "Geant4 developments and applications", *IEEE Trans. Nucl. Sci.*, 53, 270, No 1 (2006).

[5] F. Marinho, et al., "A GEANT4 based simulation for pixelated X-ray hybrid detectors", *Nucl. Instr. and Meth. A*, 777, 50, (2015).

[6] D. Wu, et al., "A hybrid Monte Carlo model for the energy response functions of X-ray photon counting detectors", *Nucl. Instr. and Meth. A*, 830, 397, (2016).

[7] D. Krapohl, et al., "Simulation of the spectral response of a pixellated X-ray imaging detector operating in single photon processing mode", *IEEE NSS/MIC proceedings*, 3843, (2010).

[8] A. Schubel, et al., "A Geant4 based framework for pixel detector simulation", *JINST*, 9, C12018, (2014).

[9] D. Krapohl, et al., "Validation of Geant4 Pixel Detector Simulation Framework by Measurements With the Medipix Family Detectors", *IEEE Trans. Nucl. Sci.*, 63, 1874, No 3 (2006).

[10] X. Llopart, et al., "Medipix2: A 64-k pixel readout chip with 55-μm square elements working in single photon counting mode", *IEEE Trans. Nucl. Sci.*, 49, 2279, No 5 (2002).

[11] X. Llopart, et al., "Timepix, a 65k programmable pixel readout chip for arrival time, energy and/or photon counting measurements", *Nucl. Instr. and Meth. A*, 581, 485, (2007).

[12] R. Ballabriga, et al., "Medipix3: A 64 k pixel detector readout chip working in single photon counting mode with improved spectrometric performance", *Nucl. Instr. and Meth. A*, 633, S15, (2011).

[13] R. Ballabriga, et al., "The Medipix3RX: a high resolution, zero dead-time pixel detector readout chip allowing spectroscopic imaging", *JINST*, 8, C02016, (2013).

[14] K. Mathieson, et al., "The simulation of charge sharing in semiconductor X-ray pixel detectors", *Nucl. Instr. and Meth. A*, 477, 191, (2002).

[15] J. Kalliopuska, et al., "3D simulations of 3D silicon radiation detector structures", *Nucl. Instr. and Meth. A*, 568, 27, (2006).

[16] J. Schwandt, et al., "Design of the AGIPD sensor for the European XFEL", *JINST*, 8, C01015, (2013).

[17] A. Ellakany, et al., "Modeling and simulation of a hybrid 3D silicon detector system using SILVACO and Simulink/MATLAB framework", *International Conference on Microelectronics (ICM)*, (2016).

[18] M.M. Vignetti, et al., "Simulation study of a novel 3D SPAD pixel in an advanced FD-SOI technology", *Solid-State Electronics*, 128, 163, (2017).

[19] Silvaco TCAD website, *https://www.silvaco.com/products/tcad.html*.

[20] Synopsys TCAD website, *https://www.synopsys.com/silicon/tcad.html*.

[21] C. Bostedt, et al., "Ultra-fast and ultra-intense x-ray sciences: first results from the Linac Coherent Light Source free-electron laser", *J. Phys. B*, 46, 164003, (2013).

[22] C. Bostedt, et al., "Linac Coherent Light Source: The first five years", *Rev. Mod. Phys.*, 88, 015007, (2016).

[23] H. Tanaka, et al., "A compact X-ray free-electron laser emitting in the sub-angstrom region", *Nature Photon.*, 6, 540, (2017).

[24] M. Yabashi, et al., "Status of the SACLA Facility", *Appl. Sci.*, 7, 604, (2017).

[25] European XFEL website, *http://www.xfel.eu*.

[26] A. Madsen, "Scientific Instrument MID Conceptual Design Report", *TECHNICAL NOTE XFEL.EU*, TR-2011-008, (2012).

[27] A. Madsen, et al., "Scientific Instrument Materials Imaging and Dynamics (MID) Technical Design Report", *TECHNICAL NOTE XFEL.EU*, TR-2013-005, (2013).

[28] M. Nakatsutsumi, et al., "Scientific Instrument High Energy Density Physics (HED) Conceptual Design Report", *TECHNICAL NOTE XFEL.EU*, TR-2013-003, (2013).

[29] M. Nakatsutsumi, et al., "Scientific Instrument High Energy Density Physics (HED) Technical Design Report", *TECHNICAL NOTE XFEL.EU*, TR-2014-001, (2014).

[30] M.P. Mancuso, "Scientific Instrument Single Particles, Clusters, and Biomolecules (SPB) Conceptual Design Report", *TECHNICAL NOTE XFEL.EU*, TR-2011-007, (2012).

[31] A.P. Mancuso, et al., "Scientific Instrument Single Particles, Clusters, and Biomolecules (SPB) Technical Design Report", *TECHNICAL NOTE XFEL.EU*, TR-2013-004, (2013).

[32] Ch. Bressler, "Scientific Instrument FXE Conceptual Design Report", *TECHNICAL NOTE XFEL.EU*, TR-2011-005, (2011).

[33] Ch. Bressler, et al., "Scientific Instrument FXE Technical Design Report", *TECHNICAL NOTE XFEL.EU*, TR-2012-008, (2012).

[34] M. Meyer, "Scientific Instrument SQS Conceptual Design Report", *TECHNICAL NOTE XFEL.EU*, TR-2011-003, (2011).

[35] T. Mazza, et al., "Scientific Instrument SQS Technical Design Report", *TECHNICAL NOTE XFEL.EU*, TR-2012-007, (2012).

[36] A. Scherz, et al., "Scientific Instrument Spectroscopy and Coherent Scattering (SCS) Conceptual Design Report", *TECHNICAL NOTE XFEL.EU*, TR-2013-006, (2013).

[37] M. Altarelli, et al., "The European X-Ray Free-Electron Laser Technical Design Report", *DESY XFEL Project Group*, (2007).

[38] W. Decking, et al., "Layout of the beam switchyard at the European XFEL", *Proceedings of EPAC08*, WEPC073 , (2007).

[39] M. Altarelli, "Structural biology at the European X-ray free-electron laser facility", *Phil. Trans. of the Royal Soc. Lon. B*, 369, 20130311, (2014).

[40] A.P. Mancuso, "Coherent imaging of biological samples with femtosecond pulses at the free-electron laser FLASH", *New J. Phys.*, 12, 035003, (2010).

[41] W. Ackermann, "Operation of a free-electron laser from the extreme ultraviolet to the water window", *Nat. Photon. A*, 1, 336, (2007).

[42] H.N. Chapman, "Femtosecond diffractive imaging with a soft-X-ray free-electron laser", *Nat. Phys.*, 2, 839, (2006).

[43] H.N. Chapman, "Femtosecond time-delay X-ray holography", *Nature*, 448, 676, (2007).

[44] European XFEL website, *Filming Chemical Reactions*, *http://www.xfel.eu/research/examples/chemical_reactions/*.

[45] U. Zastrau, "How X-ray Free Electron Lasers Enable Ultrafast Dynamics Studies in High-Energy-Density Science", *Synch. Rad. News*, 29, 24, (2016).

[46] M. Nakatsutsumi, "Femtosecond laser-generated high-energy-density states studied by x-ray FELs", *Plasma Phys. Control. Fusion*, 59, 014028, (2017).

[47] S.H. Glenzer, "Matter under extreme conditions experiments at the Linac Coherent Light Source", *J. Phys. B*, 49, 092001, (2016).

[48] M.J. Berger, et al., *XCOM: Photon Cross Sections Database*, *www.nist.gov/pml/data/xcom/*.

[49] R.D. Evans, "The Atomic Nucleus", *pp.695*, *McGraw-Hill book company*, (1955).

[50] R.D. Evans, "The Atomic Nucleus", *pp.673*, *McGraw-Hill book company*, (1955).

[51] R.D. Evans, "The Atomic Nucleus", *pp.683*, *McGraw-Hill book company*, (1955).

[52] E.H.M. Heijne, "Muon Flux Measurements with Silicon Detectors in the CERN Neutrino Beams", PhD Thesis, CERN, (1983).

[53] C.A. Klein, "Bandgap Dependence and Related Features of Radiation Ionization Energies in Semiconductors", *J. Appl. Phys.* 39, 2029, (1968).

[54] U. Fano, "Ionization Yield of Radiations. II. The Fluctuations of the Number of Ions", *Phys. Rev.*, 72, 26, (1947).

[55] L. Rossi, et al., "Pixel Detectors From Fundamentals to Applications", *pp.35*, *Springer*, (2006).

[56] H. Spieler, "Semiconductor Detector Systems", *pp.19*, *Oxford University Press*, (2005).

[57] W.R. Leo, "Techniques for Nuclear and Particle Physics Experiments", *pp.221*, *Springer-Verlag*, (1987).

[58] M.N. Mazziotta, "Electronhole pair creation energy and Fano factor temperature dependence in silicon", *Nucl. Instr. and Meth. A*, 584, 436, (2008).

[59] H. Spieler, "Semiconductor Detector Systems", *pp.56*, *Oxford University Press*, (2005).

[60] L. Rossi, et al., "Pixel Detectors From Fundamentals to Applications", *pp.40*, *Springer*, (2006).

[61] H. Spieler, "Semiconductor Detector Systems", *pp.59*, *Oxford University Press*, (2005).

[62] W.R. Leo, "Techniques for Nuclear and Particle Physics Experiments", *pp.215*, *Springer-Verlag*, (1987).

[63] H. Spieler, "Semiconductor Detector Systems", *pp.14*, *Oxford University Press*, (2005).

[64] L. Rossi, et al., "Pixel Detectors From Fundamentals to Applications", *pp.42*, *Springer*, (2006).

[65] H. Spieler, "Semiconductor Detector Systems", *pp.27*, *Oxford University Press*, (2005).

[66] R.F. Fowler, et al., "Electronhole pair creation energy and Fano factor temperature dependence in silicon" *Nucl. Instr. and Meth. A*, 450, 75, (2000).

[67] G.L. Miller, et al., "Silicon p-n Junction Radiation Detectors", *IRE Trans. Nuc. Sci.*, 7, 185, (1960).

[68] R.N. Williams, et al., "The plasma effect in silicon semiconductor radiation detectors", *Nucl. Instr. and Meth.* 120, 261, (1974).

[69] W.R. Leo, "Techniques for Nuclear and Particle Physics Experiments", *pp.238*, *Springer-Verlag*, (1987).

[70] G. Livadiotis and D.J. McComas, "Electrostatic shielding in plasmas and the physical meaning of the Debye length", *J. Plasma Physics*, 80, 341, (2014).

[71] J. Becker, et al., "Impact of plasma effects on the performance of silicon sensors at an X-ray FEL", *Nucl. Instr. and Meth. A*, 615, 230, (2010).

[72] H. Spieler, "Semiconductor Detector Systems", *pp.93*, *Oxford University Press*, (2005).

[73] W.R. Leo, "Techniques for Nuclear and Particle Physics Experiments", *pp.269*, *Springer-Verlag*, (1987).

[74] H. Spieler, "Semiconductor Detector Systems", *pp.203*, *Oxford University Press*, (2005).

[75] W.R. Leo, "Techniques for Nuclear and Particle Physics Experiments", *pp.282*, *Springer-Verlag*, (1987).

[76] H. Spieler, "Semiconductor Detector Systems", *pp.204*, *Oxford University Press*, (2005).

[77] H. Spieler, "Semiconductor Detector Systems", *pp.205*, *Oxford University Press*, (2005).

[78] W.R. Leo, "Techniques for Nuclear and Particle Physics Experiments", *pp.283*, *Springer-Verlag*, (1987).

[79] H. Spieler, "Semiconductor Detector Systems", *pp.207*, *Oxford University Press*, (2005).

[80] European XFEL, internal communication.

[81] A. Koch, et al., "Performance of an LPD prototype detector at MHz frame rates under Synchrotron and FEL radiation", *JINST*, 8, C11001, (2013).

[82] M. Hart, et al., "Development of the LPD, a high dynamic range pixel detector for the European XFEL", *IEEE NSS/MIC proceedings*, 534, (2012).

[83] Th. Tschentscher, et al., "Layout of the X-Ray Systems at the European XFEL", *TECHNICAL NOTE XFEL.EU*, TN-2011-001, (2011).

[84] G. Potdevin, et al., "Performance simulation of a detector for 4th generation photon sources: The AGIPD", *Nucl. Instr. and Meth. A* 607, 51, (2009).

[85] D. Mezza, et al., "Characterization of AGIPD1.0: The full scale chip", *Nucl. Instr. and Meth. A*, 838, 39, (2016).

[86] M. Porro, et al.,"Development of the DEPFET Sensor With Signal Compression: A Large Format X-Ray Imager With Mega-Frame Readout Capability for the European XFEL", *IEEE Trans. Nucl. Sci.*, 59, 3339, No 6 (2012).

[87] M. Porro, et al., "Expected performance of the DEPFET sensor with signal compression: A large format X-ray imager with mega-frame readout capability for the European XFEL", *Nucl. Instr. and Meth. A*, 624, 509, (2010).

[88] G. Lutz, et al., "DEPFET sensor with intrinsic signal compression developed for use at the XFEL free electron laser radiation source", *Nucl. Instr. and Meth. A*, 624, 528, (2010).

[89] I. Ordavo, et al., "A new pnCCD-based color X-ray camera for fast spatial and energy-resolved measurements", *Nucl. Instr. and Meth. A*, 654, 250, (2011).

[90] L. Rossi, et al., "Pixel Detectors From Fundamentals to Applications", *pp.213*, *Springer*, (2006).

[91] J.R. Janesick, *Scientific Charge-Coupled Devices*, *pp.22*, *SPIE Press*, (2001).

[92] S.E. Holland, et al., "Physics of fully depleted CCDs", *JINST*, 9, C03057, (2014).

[93] E. Motuk, et al., "Design and development of electronics for the EuXFEL clock and control system", *JINST*, 7, C01062, (2012).

[94] J. Coughlan, et al., "The Train Builder data acquisition system for the European-XFEL", *JINST*, 6, C11029, (2011).

[95] G. Potdevin, et al., "HORUS, an HPAD X-ray detector simulation program", *JINST*, 4, P09010, (2009).

[96] J. Becker, et al., "The detector simulation toolkit HORUS", *JINST*, 7, C10009, (2012).

[97] A. Joy, et al., "Physics and electronics simulations of the Large Pixel Detector at EuXFEL", *IEEE NSS/MIC record*, 468, (2012).

[98] S. Thomas, et al., "New ASIC 2 specification for the LPD detector", *EuXFEL internal document*, (2012).

[99] B.C. Heisen, et al., "Karabo: an integrated software framework combining control, data management, and scientific computing tasks", *ICALEPS Proceedings*, FRCOAAB02, (2013).

[100] F. Prez, et al., "IPython: A System for Interactive Scientific Computing", *Computing in Science and Engineering*, 9, 21, (2007).

[101] QT for developers website, *https://www.qt.io/developers/*.

[102] PyQT website, *https://www.riverbankcomputing.com/software/pyqt/intro*.

[103] C. Fortmann-Grote, et al., "SIMEX: Simulation of Experiments at Advanced Light Sources", *presented at the 11th NOBUGS conference*, arXiv:1610.05980, (2016).

[104] T. Rüter, "A Modular Detector Simulation Toolkit for the European XFELs Pixel Detectors", Masters Thesis, Free University of Berlin, (2015).

[105] CERN, "Physics reference manual, version: geant4 10.3", *geant4.web.cern.ch /geant4/UserDocumentation/UsersGuides/PhysicsReferenceManual/fo/PhysicsReferenceManual.pdf*, Dec, 2016.

[106] Geant4 Collaboration, "Geant4 User's Guide for Application Developers, version: geant4 10.3", *geant4.web.cern.ch/geant4/UserDocumentation/UsersGuides/ForApplicationDeveloper/fo/BookForAppliDev.pdf*, Dec, 2016.

[107] R. Ayers, *Developing the Simulation of Semiconductor Pixel Detectors for the World's Most Brilliant X-ray Source*, MSci Report, UCL, (2015).

[108] J. Becker, "Signal development in silicon sensors used for radiation detection", Thesis, University of Hamburg, (2010).

[109] W. Seibt, et al., "Charge collection in silicon detectors for strongly ionizing particles", *Nucl. Instr. and Meth.*, 113, 317, (1973).

[110] M. Campbell, et al., "Study of the charge sharing in a silicon pixel detector by means of -particles interacting with a Medipix2 device", *Nucl. Instr. and Meth. A*, 591, 38, (2008).

[111] L. Rossi, et al., "Pixel Detectors From Fundamentals to Applications", *pp.59*, *Springer*, (2006).

[112] S. Aiello, et al., "Plasma effects for heavy ions in implanted silicon detectors", *Nucl. Instr. and Meth. A*, 427, 510, (1999).

[113] A. Joy et al., "X-CSIT: a toolkit for simulating 2D pixel detectors", *JINST*, 10, C04022, (2015).

[114] N. Meidinger, et al., "pnCCD for photon detection from near-infrared to X-rays", *Nucl. Instr. and Meth. A*, 565, 251, (2006).

[115] L. Strüder, et al., "Large-format, high-speed, X-ray pnCCDs combined with electron and ion imaging spectrometers in a multipurpose chamber for experiments at 4th generation light sources", *Nucl. Instr. and Meth. A*, 614, 483, (2010).

[116] M. Kuster, et al., *The x-ray telescope of CAST*, *New J. Phys.*, 9, 169, (2007).

[117] T. Rüter, private communication.

[118] S. Hauf, et al., "Calibration processing at the European XFEL Implementation and concepts", *IEEE NSS/MIC record*, (2015).

[119] Product information sheet for LIMES 150 High-Precision Linear Stage, *www.owis.eu/fileadmin/user_upload/owis.eu/products/pdf/pi_limes_150.pdf*, 11.04.2017.

[120] L.W. Nagel, et al., "SPICE (Simulation Program with Integrated Circuit Emphasis)", Thesis, University of California, Berkeley, (1973).

Appendices

Appendix A

Works published during this PhD

- A. Joy, M. French, M.Hart, P. Seller and M. Wing, "Physics and electronics simulations of the Large Pixel Detector at EuXFEL", *IEEE NSS/MIC record*, 468, 2012.

- M. Hart, C. Angelsen, S. Burge, J. Coughlan, R. Halsall, A. Koch, M. Kuster, T. Nicholls, M. Prydderch, P. Seller, S. Thomas, A. Blue, A. Joy, V. O'shea and M. Wing, "Development of the LPD, a high dynamic range pixel detector for the European XFEL", *IEEE NSS/MIC proceedings*, 534, (2012).

- A. Joy, M. Bohlen, S. Hauf, B.C. Heisen, A. Koch, M. Kuster, J. SztukDambietz, M. Turcato, M. Wing and C. Youngman, "Simulation Tools for Detector Performance and Calibration at European XFEL", *IEEE NSS/MIC record*, 2013.

- A. Joy, M. Wing, S. Hauf, M. Kuster, T. Rüter, "X-CSIT: a toolkit for simulating 2D pixel detectors", *JINST*, 10, C04022, (2015).

- T. Rüter, S. Hauf, M. Kuster, A. Joy, R. Ayers, M. Wing, C.H. Yoon and A.P. Mancuso, "X-ray Detector Simulation Pipelines for the European XFEL", *IEEE NSS/MIC record*, 2015.

- C. Fortmann-Grote, A.A. Andreev, R. Briggs, M. Bussmann, M. Garten, A. Grund, A. Hübl, S. Hauf, A. Joy, Z. Jurek, N.D. Loh, T. Rüter, L. Samoylova, R. Santra, E.A. Schneidmiller, A. Sharma, M. Wing, S. Yakubov, C.H. Yoon, M.V. Yurkov, B. Ziaja and A.P. Mancuso, "SIMEX: Simulation of Experiments at Advanced Light Sources", presented at the 11th NOBUGS conference, arXiv:1610.05980, (2016).