PUBLISHED BY IOP PUBLISHING FOR SISSA MEDIALAB

RECEIVED: *May 23, 2014*
REVISED: *September 4, 2014*
ACCEPTED: *October 14, 2014*
PUBLISHED: *November 5, 2014*

TECHNICAL REPORT

Fast GPU-based spot extraction for energy-dispersive X-ray Laue diffraction

F. Alghabi,[a,1] **S. Send,**[b] **U. Schipper,**[a] **A. Abboud,**[b] **N. Pashniak,**[b] **U. Pietsch**[b] **and A. Kolb**[a]

[a] *Institute for Vision and Graphics, University of Siegen, Hölderlinstr. 3, 57076 Siegen, Germany*
[b] *Solid State Physics, University of Siegen, Walter-Flex Str. 3, 57072 Siegen, Germany*

E-mail: farhoosh.alghabi@uni-siegen.de

ABSTRACT: This paper describes a novel method for fast online analysis of X-ray Laue spots taken by means of an energy-dispersive X-ray 2D detector. Current pnCCD detectors typically operate at some 100 Hz (up to a maximum of 400 Hz) and have a resolution of 384×384 pixels, future devices head for even higher pixel counts and frame rates.

The proposed online data analysis is based on a computer utilizing multiple Graphics Processing Units (GPUs), which allow for fast and parallel data processing. Our multi-GPU based algorithm is compliant with the rules of stream-based data processing, for which GPUs are optimized. The paper's main contribution is therefore an alternative algorithm for the determination of spot positions and energies over the full sequence of pnCCD data frames. Furthermore, an improved background suppression algorithm is presented.

The resulting system is able to process data at the maximum acquisition rate of 400 Hz. We present a detailed analysis of the spot positions and energies deduced from a prior (single-core) CPU-based and the novel GPU-based data processing, showing that the parallel computed results using the GPU implementation are at least of the same quality as prior CPU-based results. Furthermore, the GPU-based algorithm is able to speed up the data processing by a factor of 7 (in comparison to single-core CPU-based algorithm) which effectively makes the detector system more suitable for online data processing.

KEYWORDS: Data processing methods; Pattern recognition, cluster finding, calibration and fitting methods; Analysis and statistical methods

[1] Corresponding author.

doi:10.1088/1748-0221/9/11/T11003

Contents

1 Introduction

Modern X-ray synchrotron radiation experiments profit from the availability of advanced detectors. Most of them are 2D pixel detectors providing good position resolution, high dynamic range and fast readout, but insufficient energy resolution. One of the most advanced detectors is the pn-junction charge coupled device (pnCCD), an energy-dispersive area detector with 384×384 pixels and a readout frequency of up to 400 Hz producing 3D data in location (x, y) and energy E. Typically the number of generated pixels is 5.9×10^7 per second corresponding to a data generation rate of 112 MB/s (assuming 2 bytes of data per pixel).

Considering various detector parameters and data processing steps, this amount of data cannot be processed interactively, i.e. in a time which allows for user effective feedback of the experiment by means of standard processing approaches. Thus, the traditional offline data treatment has to be replaced by fast online data processing. One of the experiments profiting from the use of a pnCCD detector in an online fashion is the Laue experiment, where a stationary crystal generates a characteristic Laue pattern after exposure to white X-rays [1, 2]. The clear advantage of the Laue method is that in contrast to monochromatic diffraction techniques, data can be collected without sample rotation. However, the spectroscopic performance of conventional area detectors does not allow to extract enough structural information from a recorded Laue pattern which is based on the simultaneous position and energy resolution of individual Bragg peaks. Hence, in macromolecular crystallography, precise X-ray structure analyses of organic materials are usually carried out by

2014 JINST 9 T11003

recording series of diffraction images of a rotatable crystal exposed to monochromatic X-rays. For that purpose, fully-automated goniometer systems widely serve to collect complete data sets of protein crystals at high resolution. The use of fully depleted pnCCDs [3–5] allows energy-dispersive Laue diffraction as a new tool for structure analysis [6]. In this case, the crystal is illuminated with white synchrotron radiation and analyzed without sample rotation. Previous studies have shown that by means of energy-dispersive Laue diffraction both monocrystalline and polycrystalline materials can be quantitatively characterized within a single X-ray shot [7]. In a recent application, the capability of the pnCCD for simultaneous position and energy resolution was used to investigate streaked Bragg reflections from plastically deformed regions of a Cu cantilever [8].

Currently, the University of Siegen is equipped with a frame store pnCCD module of the eROSITA generation [9]. The sensitive volume consists of a fully sideward depleted Si bulk of 450 μm thickness with 384×384 pixels in the image area (pixel size 75×75 μm^2) and in the frame store area (pixel size 75×51 μm^2) allowing for frame rates of up to 400 Hz. The gain of structural information from the pnCCD data sets requires relatively long processing times depending on the raw data volume (i.e. the number of recorded signal frames), the detected number of photons and the performance of the computer. Usually, processing times in the order of minutes are needed to apply the necessary steps for raw data correction [10] and to find the positions and energies of the Bragg peaks collected within an energy-resolved Laue pattern. In the second step, these coordinates are used to calculate the real lattice of the crystal by converting them into reciprocal-space coordinates and to extract integrated intensities of the indexed reflections for structure refinement [11].

In recent years, Graphics Processing Units (GPUs) became an inevitable device for processing large data sets. This is especially true for streamed data, i.e. given as a sequence of homogeneous data sets ("frames"). For more than 10 years, GPUs are extremely successful in fast data processing in areas like physically-based simulation, signal and image processing and even data mining [12]. Presently, the potential of GPU-accelerated scientific computing is exploited in various fields like medical physics [13, 14], high-energy physics [15–17] and lattice quantum chromodynamics [18, 19].

In this paper we present a data parallel, stream-based algorithm for Laue spot extraction from a sequence of pnCCD images using multiple GPUs. The algorithm is fast enough to provide online data processing for online spot extraction, thus making an interruption-free application of the pnCCD-based Laue experiment possible. The major algorithmic innovation is the parallel spot extraction; further optimizations such as an improved algorithmic suppression of background intensity are presented. Furthermore, the comparison of GPU-based results with those of a CPU-based reference algorithm reveals that the new multi-GPU processing algorithm yields on par quality.

The remainder of the paper is structured as follows. In section 2 we discuss the constraints and challenges we are facing while developing the GPU-based Laue spot extraction algorithm. The GPU-based algorithm is presented in section 3 and evaluated in section 4. Finally, section 5 provides conclusion as well as outlook.

2 Constraints & challenges

Although GPUs provide huge computational power, efficient exploitation of this power necessitates the imposition of some constraints on the programming style. Furthermore, since we want to use multiple GPUs we are exploiting a software framework designed for streaming computation on

2014 JINST 9 T11003

multiple GPUs [20]. Proper use of this framework needs some familiarity with it. Section 2.1 provides this familiarity as well as a discussion of constraints imposed by GPU processing. On the other hand, the developed GPU-based algorithm is intended to grasp the same processing ideas of the sequential CPU-based algorithm as much as possible in order to provide comparable results. This constraint is best understood by explaining the sequential CPU-based approach as done in section 2.2. Finally, there are a number of inherent issues in the sequential CPU-based algorithm related to the quality of results. We were also challenged to resolve these issues while developing the GPU-based algorithm (see section 2.3).

2.1 Implementation constraint

Overview on GPU programming. Graphics Processing Units (GPUs) have gained broad attraction as they have been successfully applied to various computational tasks in different application areas [12–19]. The main factor giving rise to the computational power of GPUs is their large number of processing cores. Current GPUs have more than a thousand cores, i.e. GPUs are considered to be *many-core processors*. However, the computational power of GPUs comes at the cost of a restricted programming model, which is commonly known as *stream processing*. GPUs are extremely efficient, if the same sequence of instructions, i.e. the program, is executed on a large set of data elements with identical structure and each of the data elements can be processed independently. Furthermore, GPUs have a comparably low amount of random access memory (RAM). Thus, GPU programs usually process a series of *frames* of identically structured data elements in a serial manner, one frame at a time; the sequential processing of video frames is one of the classic examples.

The major rule of thumb for designing efficient GPU programs is to avoid the following algorithmic structures:

- random write access, as this introduces data dependency and requires synchronization of potentially independent computations,
- random read access outside the current frame, as only a limited number of frames can be stored on a GPU,
- sequential data processing, e.g. algorithms that prevent parallel (and thus independent) handling of data elements (as an example: most sorting algorithms are sequential), and
- programs that have extremely varying runtimes for individual data elements within a single frame, as the slowest data element will dominate the overall processing time.

Multi-GPU framework. The software framework is developed in CUDA which is a programming interface specifically designed for general-purpose computing on CUDA-enabled NVIDIA GPUs [21]. The framework's underlying assumption is that the data to be processed is in the form of streams, where only part of the full data stream, in most cases only a single frame, can be processed at one point in time. A stream is considered to be a sequence of homogeneous data sets, a concept which can well be adopted to our problem where we have a sequence of pnCCD images to be processed.

2014 JINST 9 T11003

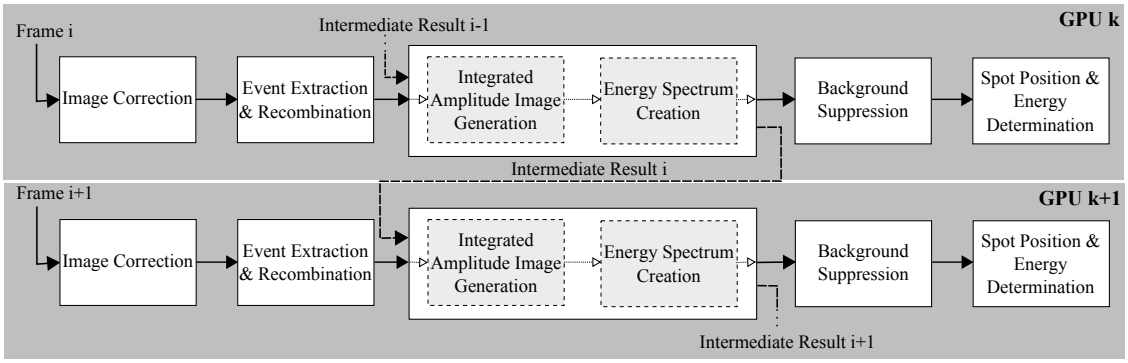

Figure 1. Implementation of the Laue spot extraction on a multi-GPU framework. The *Energy Spectrum Creation* and *Integrated Amplitude Image Generation* modules require synchronization and data transfer between adjacent GPUs.

The framework requires to partition a software solution into smaller subsets, that are processed by so-called *modules*. A module has a specific number of inputs (e.g. a single frame and/or the processed result of another module) and produces a result, that can be used by follow-up modules for further processing. Modules can be executed on GPUs or CPUs, but, as a rule of thumb, the more modules are efficiently implemented on GPUs the more efficient the data processing. Utilizing several GPUs is realized by splitting the raw data stream, i.e. the pnCCD images, and processing the frames alternating on the GPUs with the same module configuration (see figure 1). Thus, each GPU has very similar workload which leads to a good load-balancing between the GPUs. The major drawback of this approach is, that modules which require to process all frames, e.g. to compute some statistics, enforce a synchronization and data transfer between GPUs. In our case, the *Energy Spectrum Creation* module accumulates the per-pixel energies and collects them in an energy spectrum (see figure 1).

As an application example in the area of crystallography, the framework has already been successfully applied for correcting pnCCD images [22]. See [20, 23] for more information on the scope of problems that can be addressed by this kind of (multi-)GPU software framework.

2.2 Result comparability constraint

Sequential CPU-based Laue-spot extraction. The data volume accumulated by a pnCCD detector in an energy-dispersive Laue diffraction experiment comprises two image sequences with pre-defined numbers of recorded frames, respectively. The first image sequence consists of the so-called dark frames which are recorded in the absence of X-rays in order to determine the mean noise of the detector. Typically 300 *dark frames* are sufficient to analyse the noise level of each pixel with good statistics. The mean dark noise of the used pnCCD system is described by an equivalent noise charge (ENC) of about 8 electrons allowing for a relative energy resolution below 1% in the energy range between 6 keV and 20 keV [5]. Further data collection is carried out by recording a second image sequence, the so-called *signal frames*. In this case, the sample is illuminated with white synchrotron radiation and kept at fixed orientation during the exposure time. In this paper, the Laue pattern of a tetragonal hen egg-white lysozyme crystal was detected within 100000 signal frames corresponding to a total exposure time of 1150 s with the pnCCD operated

2014 JINST 9 T11003

at a cycle time of 11.5 ms (frame rate 87 Hz). The experiment was performed at the EDR beamline of the storage ring BESSY II in Berlin providing highly brilliant bending magnet radiation with a white primary beam spectrum extending from 8 keV to 25 keV at the sample position.

In this sense, a single frame is to be understood as an image of charge signals distributing over an area of 384×384 pixels which were measured during the integration time of the pnCCD. These charge signals are also called events if they exceed a certain threshold which is usually set to four times the mean noise. The simultaneous spatial and energy resolution of incident X-ray photons can be achieved in the single-photon counting (SPC) mode of the pnCCD. For that purpose, the average count rates at the positions of the Laue spots should be of the order of 0.1 per pixel and frame.

In the SPC mode, individual photons generate charge clouds of electrons which drift to the pixel structure of the detector and, as a consequence of diffusion and electrostatic repulsion within the detector bulk, extend in space to a typical radius of 10 μm. Depending on the interaction position of the X-ray photon in the detector material the charge cloud is collected in one pixel (single event) or several pixels (split events). For X-ray energies below 30 keV the charge cloud can spread over two (double events), three (triple events) or four adjacent pixels (quadruple events). The best possible energy resolution can be obtained from the single events whereas triple and quadruple events enable sub-pixel resolution with a spatial accuracy of below 3 μm [24]. Four valid event patterns exist for every type of split events according to the various splitting directions in the pixel array of the pnCCD (see also figure 2 in [24]). In the case of triple events, the measured signal is called "valid" if the highest signal amplitude is located in the corner of the event. For quadruple events, the amplitudes need to be distributed over a square with a size of 2×2 pixels in such a way that the highest and lowest signals are recorded along one diagonal of the event.

The Laue diffraction experiment on tetragonal hen egg-white lysozyme was performed with an average count rate of about 400 events per frame so that less than 1% of the pixels contained a signal. However, previous studies have shown that even under this condition, the real data pattern generated within the pnCCD at the Laue spot positions exhibits a more complex structure [5]. The detected signals are dominated by high local count rates at the positions of the collected Bragg peaks. In this case, the condition of a safe position and energy discrimination of single photons can no longer be fulfilled resulting in spatial and temporal overlaps of different photons. These so called pile-up events constitute a significant contribution to the pnCCD data pattern and need to be analyzed by means of statistical methods [11]. In most cases, pile-up signals give rise to event patterns which either spread over more than four adjacent pixels (also denoted as cluster events) or create triple and quadruple events with "forbidden" amplitude distributions in the sense of SPC. In both situations, the recorded event pattern is denoted as "invalid".

Finally, due to the fact that the experiment itself is situated in an air environment, elastic scattering of incident white X-rays on air molecules gives rise to a continuous background signal which distributes over the complete detector area. Under real experimental conditions, the background level effectively depends on the position on the detector and can be influenced to a large extent by means of beam stops.

Prior to structure analyses based on the knowledge of the crystallographic unit cell of the investigated material, the precise position and energy determination of the recorded Bragg peaks is essential. For that purpose, several steps of data processing are required in order to provide the desired information (see figure 2, top row):

2014 JINST 9 T11003

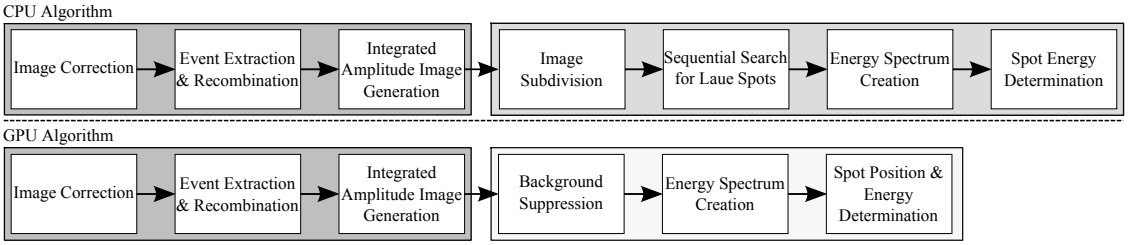

Figure 2. The CPU- and the GPU-based block diagram. The first three processing steps are conceptually identical for CPU- and GPU-based implementations. Mainly the sequential search for Laue spots needs to be adopted.

Image correction and event extraction & recombination. After subtracting noise, offset and common mode from the raw data sets according to the method described in [10], the images containing the diffraction signal of the crystal need to be corrected for gain and charge transfer efficiency (CTE). The gain and CTE correction factors were calculated for every channel of the pnCCD by previously calibrating the detector with the CuK$_\alpha$ fluorescence line of a metallic foil exposed to white X-rays. Event filtering is performed using a threshold of 6σ to identify real X-ray signals where $\sigma = 8e^-$ (ENC) is the mean dark noise. The subsequent recombination to individual photon hits relies on the center-of-mass approach taking into account every type of events (valid and invalid events). The center-of-mass of a split event is then given in floating-point numbers. In many applications, where only single-photon information is of interest, the recombination is restricted to valid event patterns. However, due to the fact that also pile-up events contribute to the integrated Bragg peak intensities required for X-ray structure analysis, invalid events are included in this step. Finally, the calibration data served to convert the signal amplitude of every event measured in arbitrary digital units (adu) into suitable energy units (keV). At this stage, besides pixel and energy coordinates, also a time stamp is ascribed to every photon hit according to the number of the frame in which the event occurred. However, the time coordinate will not be further considered here as it is only of importance for dynamic measurements. The different steps used to extract the simultaneous position, energy and time information about individual events from the raw data can be considered as a standard method which is not restricted to energy-dispersive Laue diffraction. The following techniques of data processing depend on the application itself and need to be modified for other types of experiments using a pnCCD.

Integrated amplitude image generation. The accumulated data volume can be visualized by a single two-dimensional matrix $I_0(x,y)$ with 384×384 pixels. I_0 is to be understood as a distribution of integrated signal amplitudes of recombined events recorded in the pixel structure of the pnCCD. For that purpose, the previously calculated center-of-mass coordinates of individual split events are rounded to integer values (x,y) which correspond to the physical pixel coordinates of the detector. Single events do not need to be rounded since they are already described by integer pixel positions. In this way, the recombined energy E_i of every photon hit located within the pixel (x,y) contributes to the integrated amplitude image I_0 at

2014 JINST 9 T11003

this position. Mathematically, I_0 is given by

$$I_0(x,y) = \sum_{i=1}^{N(x,y)} E_i. \tag{2.1}$$

with $x,y = 1,\ldots,384$ and $N(x,y)$ is the number of recombined events with rounded pixel coordinates (x,y) and energies E_i. In order to properly select the energy range of interest and to avoid contributions from low-energetic background signals, only events with energies above an externally defined threshold (here set to 8.5 keV) are taken into account in this step. In principle, the integrated amplitude image $I_0(x,y)$ is obtained by summing up the recombined signal amplitudes detected in every pixel over the number of recorded frames (100000 in this experiment). In contrast to the reduced intensity distribution I^*, which is calculated in the GPU approach (section 3, eq. (3.1)), both valid and invalid event types including pile-up signals are included in I_0. The choice of 2D matrices for further analyses is advantageous since it allows for fast data processing. Though the full experimental information can be stored within a 3D intensity distribution which elongates the data along the energy direction, such a 3D matrix has a size of several hundreds of MB decreasing the processing speed significantly.

Image subdivision. The purpose of converting the single-photon information into the 2D matrix I_0 is to localize the Bragg peaks of the sample within the recorded data volume. Previous tests of the developed algorithms revealed that in the case of a sequential search for Laue spots (described in the next item), the efficiency of the CPU-based software is improved by subdividing I_0 into smaller images of equal size. In this way, the number of correctly identifiable reflections relative to the number of erroneously localized spots can be enhanced. This effect is caused by an inhomogeneous background continuum which superimposes the diffraction signal of the crystal. Using image subdivision, the signal-to-background ratio is effectively increased for weak Bragg peaks referred to higher background levels in other regions of the detector which are excluded during the sequential search for Laue spots in the corresponding subimage. Typically, the total data volume is divided into four subimages $I_{0,1}$, $I_{0,2}$, $I_{0,3}$ and $I_{0,4}$ with a size of 192×192 pixels each.

Sequential search for Laue spots. From a conceptual point of view, a Bragg peak of the sample detected by the pnCCD can be considered as a local maximum of the integrated amplitude image I_0. The collected Laue spots are searched sequentially by means of an appropriate iterative algorithm applied to the defined subimages. In the first step, the absolute maximum of the subimage $I_{0,1}$ is found and its pixel coordinates (x_1,y_1) are stored as the position of the most intense Bragg reflection in this subimage. Depending on experimental aspects like the primary beam size ($100\times100\,\mu\mathrm{m}^2$ in this experiment) and parallax effects within the sensitive volume of the detector, the Laue spots are not confined to one single pixel but extend over small pixel ranges. Usually, a quadratic area with a size of 5×5 pixels is suitable to define the spot region in which the energy of the Bragg peak can be calculated from the extracted energy spectrum in this area. In the next step, a reduced subimage $I_{1,1}$ is obtained from the original subimage $I_{0,1}$ by setting $I_{1,1} = I_{0,1}$ and $I_{1,1}(x,y) = 0$ if $x_1 - 2 \le x \le x_1 + 2$ and

2014 JINST 9 T11003

$y_1 - 2 \leq y \leq y_1 + 2$. In this way, the region of the first successfully found Laue spot is marked in the subimage $I_{1,1}$. The position (x_2, y_2) of the second Laue spot is then given by the pixel coordinates of the absolute maximum of $I_{1,1}$. As soon as this position is determined, the next subimage $I_{2,1}$ is defined by $I_{2,1} = I_{1,1}$ with $I_{2,1}(x,y) = 0$ if $x_2 - 2 \leq x \leq x_2 + 2$ and $y_2 - 2 \leq y \leq y_2 + 2$ thereby marking the positions of the first and the second identified Laue spots in $I_{2,1}$. This iterative procedure is continued until all the Bragg peaks of the subimage $I_{0,1}$ are localized. Finally, the described search algorithm is applied to the remaining subimages $I_{0,2}$, $I_{0,3}$ and $I_{0,4}$ providing the positions of the Laue spots collected in these subimages based on the corresponding reduced subimages. In order to reduce the overall processing time of this algorithm, the number of steps is limited to the maximum number of expected reflections per subimage.

Energy spectrum creation. Since a crystalline material diffracts X-rays with sharp wavelengths under fixed angles according to Bragg's law, only discrete energies can be ascribed to the Bragg reflections associated with the absolute maxima of the processed subimages. If the crystal is exposed to white X-rays, several diffraction orders can be measured simultaneously at the Laue spots' positions depending on the incident spectrum and the atomic structure of the sample. In general, the diffracted energies of higher-order reflections are multiples of the first-order energy. The energetic profiles of the identified Bragg peaks are resolved at this stage by calculating the energy spectra measured by the pnCCD at the spot positions. For that purpose, the data set needs to be processed again on the level of individual events: in the first step, a filter is used which selects all the recombined events (i.e. valid and invalid events) whose rounded pixel coordinates belong to the 5×5 pixel area of a Laue spot. Then, an energy spectrum is created from these events extending up to the highest detected energy in this pixel range. The suitable bin size for the histogram depends on the energy resolution of the pnCCD. Single events generated by monochromatic X-rays in the energy range between 6 keV and 20 keV are detected with a resolution between 200 eV and 280 eV (FWHM) [5] corresponding to a standard deviation between σ = 85 eV and $\sigma = 120$ eV. In previous applications, it could be verified that bin sizes between $\sigma/2$ and σ give reliable results of spectral analyses with sufficient statistics. Hence, a bin size of 100 eV is defined for energy spectrum creation and spot energy determination. Under this condition, an energy spectrum was extracted from the pnCCD data for every identified Laue spot.

Spot energy determination. As already mentioned above, an energy threshold of 8.5 keV was introduced to separate the diffracted signal of the sample from low-energetic background (see item Integrated Amplitude Image Generation). This threshold is also applied in the last step of data processing defining the lowest possible energy of a Laue spot. The dominating peak with the highest integrated intensity in the spectrum is assumed to be a Bragg peak of the crystal. Fitting this peak by means of a Gaussian distribution with polynomial background, its energy E and standard deviation σ can be obtained. The value of $\sigma(E)$ is crucial to either accept or reject the identified Laue spot for structure analysis. If the standard deviation of the Gaussian is comparable to the expected resolution of the pnCCD ($\sigma(E)/E \approx 1\%$, 80 eV $\leq \sigma(E) \leq$ 300 eV) the reflection is used for unit-cell determination. The upper limit of 300 eV for σ takes into account that in general, structural effects inside the crystal may

2014 JINST 9 T11003

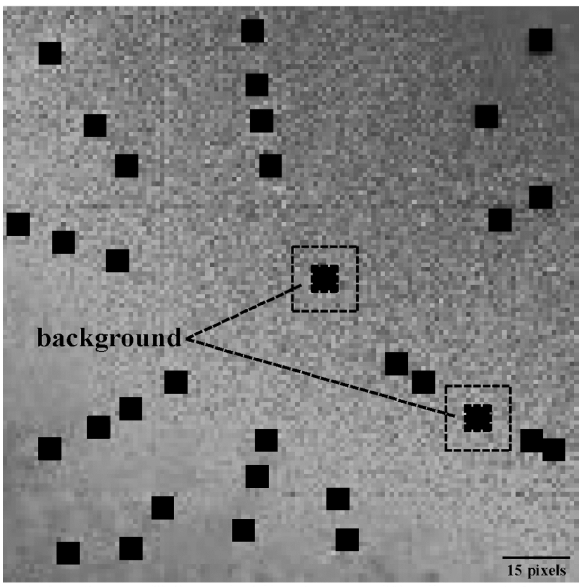

Figure 3. Laue pattern of hen egg-white lysozyme (image size 128×128 pixels): the black squares indicate the selected spot regions with a size of 5×5 pixels. The identified Bragg peaks of the crystal are accompanied by background signals which distribute over the whole detector area. The background level is typically measured in the direct vicinity of individual reflections (dashed frames) [11]. Dark pixels represent areas with lower background levels.

lead to an energetic broadening of the Bragg peaks. Moreover, since also split events are included in the energy spectrum, the width of the fitted peak is necessarily larger compared to the situation where only single events are considered.

Figure 3 shows the result of this procedure in the case of a recorded Laue pattern which was generated by a tetragonal hen egg-white lysozyme crystal after illumination with white synchrotron radiation in the energy range between 8 keV and 25 keV.

In the last step of data treatment, the unit cell of the exposed crystal is calculated from the extracted Bragg peaks' positions and energies. For that purpose, the three-dimensional spot coordinates (x, y, E) including their measurement errors are converted into the corresponding data volume in reciprocal space. Later, every possible combination of three distinct reciprocal lattice vectors undergoes appropriate reduction and refinement procedures and is then tested if it spans a conventional unit cell of the reciprocal lattice within suitable error limits [11]. After conversion into real-space coordinates the best matching unit cell is finally used for a complete indexing of the recorded Laue pattern.

2.3 Challenges of the sequential CPU-based Laue-spot extraction method

Besides the goal of implementing the Laue spot detection on the GPU (see section 3), we additionally address two issues related to the quality resulting from the CPU-based method as described above. Firstly, the CPU-based method assumes a symmetric area (square of size 5×5 pixels) around each local maximum. This square shape then forms the basis for further analysis of the data and

2014 JINST 9 T11003

Figure 4. An integrated amplitude image with non-symmetric-shaped spots (image size 384×384 pixels): this image is generated using the GPU algorithm and pixel values are normalized and then mapped to shades of grey (black colour represents large amplitudes).

subsequently spot information determination (see figure 3), i.e. only events within this area are used to determine spot characteristics. However, due to parallax effects this assumption is not always met, i.e. scattered beams can create non-symmetric shapes like oval or ellipsoidal shapes (see figure 4). Secondly, higher pixel values are more abundant in areas with stronger background scattering. Thus it is difficult to find spots in areas with lower background levels and the number of 'artificial' spots in areas with higher background values increases. To alleviate this problem, the CPU-based method proposes to divide the image into four quarters and then starts searching for a certain number of maximum pixel values in each quarter separately.

3 GPU-based Laue-spot extraction

The major challenges to provide a GPU-based Laue spot extraction is to formulate algorithms using a fixed data layout that allows for independent data processing without enforcing global data access. As a key concept, we represent most information on a per-pixel level, i.e. on the level of individual data frames acquired with the pnCCD detector. Figure 2 (bottom row) shows the major steps of the algorithm developed for running on GPU. Conceptually, only a few steps are different from the CPU version, specifically the two steps of background suppression (see

2014 JINST 9 T11003

section 3.3) and spot position & energy determination (see section 3.5). Most of the steps, however, are identical to their CPU counterparts with minor modifications. The main modifications are described in the following.

3.1 Image correction and event extraction & recombination

This step is similar to the CPU version with the difference that only single events and valid split events (doubles, triples and quadruples splitting over two, three and four adjacent pixels, respectively) are taken into account. The split events are then recombined into individual photon hits by calculating the center-of-mass coordinates for each event and then rounding them to the nearest integer pixel positions (rounded recombined events). For each recombined event the energy is calculated by accumulating the signal amplitudes of constituent pixels. Also in this case, only events with energies above a threshold of 8.5 keV are considered for further spot analyses.

3.2 Integrated amplitude image generation

In this step, we create an integrated amplitude image $I^*(x,y)$. For each pixel (x,y) the integrated amplitude image is calculated by summing up the signal amplitudes of all single events and rounded recombined events over the whole set of recorded frames (100000 frames in this experiment) located at the corresponding pixel position:

$$I^*(x,y) = \sum_{i=1}^{N(x,y)} E_i. \tag{3.1}$$

where $x,y = 1,\ldots,384$ and $N(x,y)$ is the number of all single and recombined valid events with pixel coordinates (x,y) and energies E_i. This is in contrast to the CPU version where all events including clusters are used to generate the integrated amplitude image. The reason for the restriction to valid event patterns is that in this experiment we are only interested in computing the unit cell of the crystal. Furthermore, it is already proven that a consideration of only valid events is sufficient for unit-cell computation (see [6]). Therefore, we decided to consider only valid events which, in turn, speeds up processing by decreasing the amount of computation required.

3.3 Background suppression

As already explained in section 2.2, the CPU algorithm searches for Laue spots in a sequential manner. This method is suitable for running on a single-core CPU since it keeps the processor busy all the time thereby fully exploiting its computational power. This approach, however, leads to underutilization of processors with more cores. This is because only one processor core is working while the other cores sit idle thus leading to poor exploitation of the processor. To address this issue, we have developed an algorithm with comparable results which can more efficiently exploit many-core processors. In this algorithm, we fit a polynomial function of degree n to the integrated amplitude image $I^*(x,y)$ as follows:

$$F(x,y) = \sum_{i+j\leq n} a_{i,j}x^i y^j = a_{n,0}x^n + a_{n-1,1}x^{n-1}y + \ldots + a_{0,n}y^n + a_{n-1,0}x^{n-1} + \ldots + a_{1,0}x + a_{0,1}y + a_{0,0}. \tag{3.2}$$

2014 JINST 9 T11003

The coefficients $a_{i,j}, i,j \in \mathbb{N}, i+j \leq n$ are found by using a least-squares fitting method taking into account all the pixels (x,y) jointly. Afterwards, a measure of the fitting error is computed. The measure of error we have chosen is called Mean Absolute Error and is computed as follows:

$$\mathrm{MAE} = \frac{1}{N} \sum_{x,y} |I^*(x,y) - F(x,y)| . \tag{3.3}$$

where $N = 384 * 384$ is the total number of pixels in the integrated amplitude image. Later, we try to discard the background and keep only the pixels which comprise spots. For this purpose, we define a threshold function $T(x,y)$:

$$T(x,y) = F(x,y) + s \cdot \mathrm{MAE}. \tag{3.4}$$

where s is a scalar constant. Then, we compute the Background-Suppressed Image defined by the following formula:

$$\mathrm{BSI}(x,y) = \begin{cases} I^*(x,y) & \text{if } I^*(x,y) \geq T(x,y) \\ 0 & \text{if } I^*(x,y) < T(x,y) \end{cases} \tag{3.5}$$

For the polynomial fitting (see eq. (3.2)), we tried polynomials of degrees 1, 2 and 3. Then we computed the fitting error using Normalized Absolute Error measure defined as:

$$\mathrm{NAE} = \frac{\sum\limits_{x,y} |F(x,y) - I^*(x,y)|}{\sum\limits_{x,y} I^*(x,y)}. \tag{3.6}$$

The computed error values are 21.04%, 20.53% and 19.23% for linear, quadratic and cubic polynomials, respectively. As the difference in errors are low we decided to use linear polynomial to make the calculations easier:

$$F(x,y) = a_{1,0}x + a_{0,1}y + a_{0,0}. \tag{3.7}$$

In order to choose the constant s in eq. (3.4) properly, we investigated the effect of the threshold function $T(x,y)$ on the quality of the obtained Laue pattern. It turned out that too small values of s cause pixel clusters for separate spots to be connected to each other while too large values of s typically lead to sparser spot sets. Considering these effects the choice $s = 3$ seemed to be convenient for further data processing thus defining the threshold function as:

$$T(x,y) = F(x,y) + 3 \cdot \mathrm{MAE}. \tag{3.8}$$

3.4 Energy spectrum creation

This step is similar to its CPU version counterpart in the sense that all previously-found events (singles and valid split events in our case) are used to form energy spectra. In contrast to the CPU version where energy spectra are extracted spot-wise, we create the energy spectra pixel-wise, i.e. we form an energy spectrum for each pixel separately by considering all single events and rounded recombined events in that pixel position. In other words we can write:

$$N(x,y) = \sum_{i=1}^{c} \mathrm{PES}(x,y,b_i). \tag{3.9}$$

2014 JINST 9 T11003

where $x, y = 1, \ldots, 384$, $N(x,y)$ is the number of all single and recombined valid events with pixel coordinates (x,y), c is the number of energy bins (which is the same for all pixels) and $\mathrm{PES}(x,y,b_i)$ (Pixel-wise Energy Spectrum) is the number of events in pixel position (x,y) and energy bin b_i. Even though conceptually very similar, the implementation on multiple GPUs is not straightforward due to the large amount of storage required for the per-pixel spectra (see "A Note on Implementation" in the following).

Later, a spectrum maximum image, denoted by $\mathrm{SMI}(x,y)$, is created by storing the energy with the highest count for each spectrum in the corresponding pixel position.

A note on implementation. We combine the two conceptual blocks of *Integrated Amplitude Image Generation* and *Energy Spectrum Creation* in a single processing module (see figure 1). This module collects per-pixel energy data for all frames in order to compute the integrated amplitude image and set up a per-pixel spectrum with some 600 bins of size 122 eV each. For a frame size of 384×384 and 4 bytes per bin, this would result in a data block of some 337.5 MB. As described above, collecting this kind of information for all frames requires synchronization and data transfer between GPUs. A naive implementation would transfer the full data block between the GPUs, resulting in extreme bandwidth problems and, thus, slowing down the overall processing significantly.

A much more efficient solution is to transfer only the last three frames, in general the last n-1 frames for n GPUs, between the GPUs in a round-robin scheme, so each GPU has the full information required for processing this module. This approach allows for a fully incremental online-processing of the data stream, i.e. for the current frame the result on the basis of all prior frames is available. For the sake of comparison with the CPU implementation, which does not provide incremental processing (see section 2.2), we do not apply this incremental approach on the GPU as it would reduce performance. We simply run the processing without transfering data for the module and collect the partial results from each GPU after all frames have been processed. This result is then passed to the subsequent modules. This is computationally more efficient, since only one final data transfer is required.

3.5 Spot position & energy determination

Pixels in $\mathrm{BSI}(x,y)$ are either of two groups: zero-valued pixels which denote background and non-zero pixels. Non-zero pixels are distributed across the image in form of connected components (clusters). Connected components are defined based on the concept of connectivity of two pixels. There are two types of connectivity: 4-connectivity and 8-connectivity. If a pixel has coordinates (x,y), then 4 pixels with coordinates $(x \pm 1, y)$ and $(x, y \pm 1)$ are said to be 4-connected. If a pixel has coordinates (x,y), then 4-connected pixels besides 4 pixels with coordinates $(x \pm 1, y \pm 1)$ are said to be 8-connected. Based on this definition, we have a number of connected components each consisting of one or more non-zero pixels. A connected component with at least m pixels and at most M pixels is considered as a spot. The position of each spot is computed as follows:

$$P_{\mathrm{spot}} = (x_{\mathrm{spot}}, y_{\mathrm{spot}}) = \left(\left[\frac{\sum_{i=1}^{p} \mathrm{BSI}(x_i, y_i) \cdot x_i}{\sum_{i=1}^{p} \mathrm{BSI}(x_i, y_i)} \right], \left[\frac{\sum_{i=1}^{p} \mathrm{BSI}(x_i, y_i) \cdot y_i}{\sum_{i=1}^{p} \mathrm{BSI}(x_i, y_i)} \right] \right). \tag{3.10}$$

2014 JINST 9 T11003

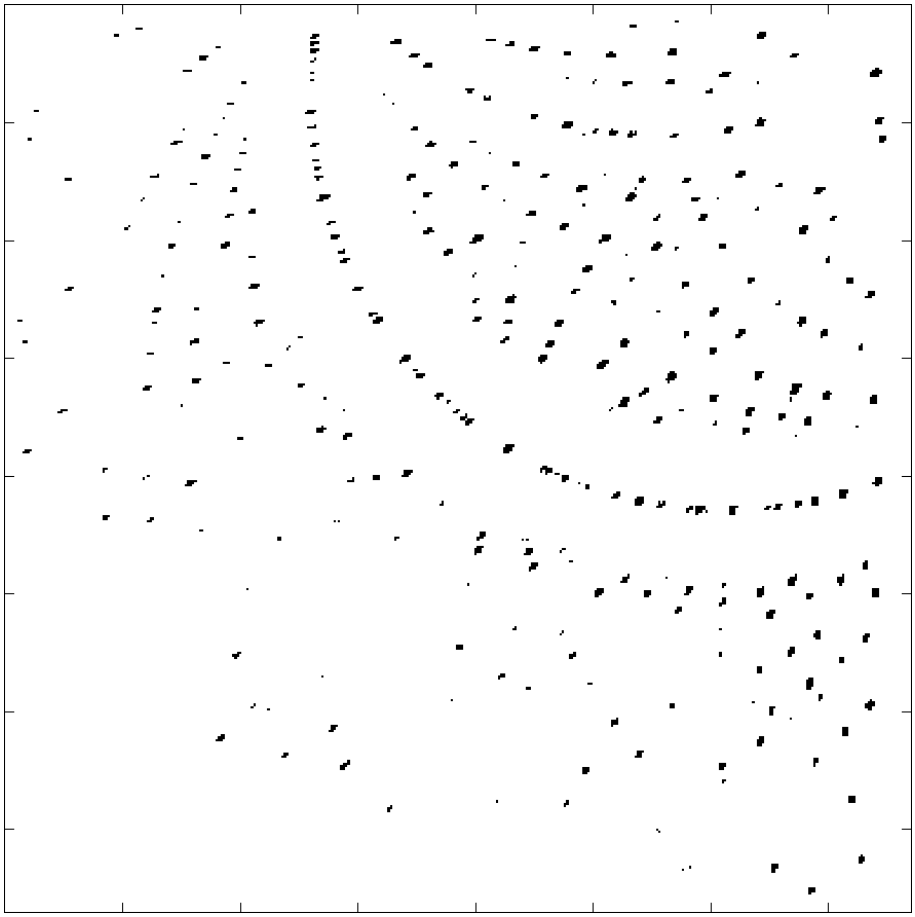

Figure 5. Background-suppressed image as computed by the GPU algorithm (image size 384×384 pixels): Zero-valued pixels are shown in white colour whereas non-zero clusters are shown in black colour.

where p is the number of pixels in the spot, (x_i, y_i) denotes the coordinates of pixel i in the spot $(1 \leq i \leq p)$ and $[x]$ denotes the closest integer to x. The energy of each spot is likewise computed by:

$$E_{\text{spot}} = \frac{\sum_{i=1}^{p} \text{BSI}(x_i, y_i) \cdot \text{SMI}(x_i, y_i)}{\sum_{i=1}^{p} \text{BSI}(x_i, y_i)}. \tag{3.11}$$

Furthermore, the implemented connectivity type is 8-connectivity and connected components of all sizes (i.e. $m = 1$ and $M = \infty$) are considered as a spot.

As stated in section 2.3, our method tries to deal with non-regular spot shapes, e.g. elliptical ones. Actually the shape of each spot is merely determined by how the background is suppressed. Figure 5 shows the connected components as computed by the GPU algorithm after background suppression using the integrated amplitude image in figure 4. The other issue, i.e. the inhomogeneous background values, is resolved by considering an adaptive threshold for the background suppression according to eq. (3.8).

2014 JINST 9 T11003

Table 1. Unit cell parameters obtained for the GPU method and the CPU method in comparison with the expected cell parameters of tetragonal hen egg-white lysozyme.

	a_1[Å]	a_2[Å]	a_3[Å]	α_1[°]	α_2[°]	α_3[°]
expected	79.1	79.1	37.9	90.0	90.0	90.0
GPU method	79.5	80.0	38.3	89.6	89.0	88.7
CPU method	79.6	77.8	38.1	88.2	89.8	91.8

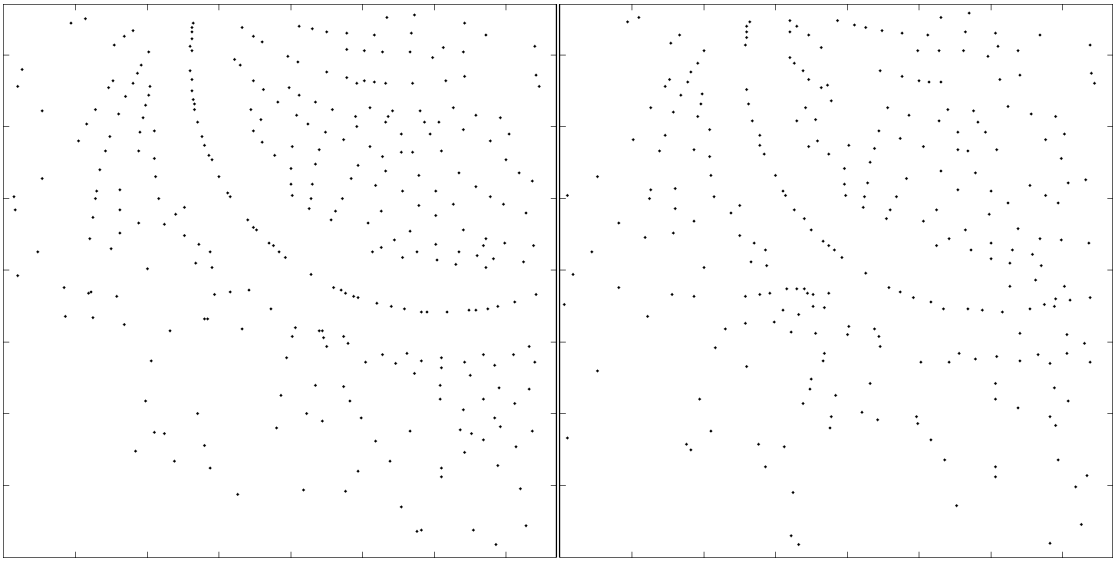

Figure 6. Position of spots as computed by the GPU- (left) and the CPU-based algorithm (right): there are 313 spots in the left image and 281 spots in the right one. Both spot sets are determined from the same sequence of pnCCD images (image size 384×384 pixels).

4 Results

The described methods for Laue spot extraction were applied to the collected test data set of a hen egg-white lysozyme crystal and compared after cell calculation and indexing of the Laue pattern as well as on the level of spot coordinates (x, y, E). Figure 6 (left) shows the calculated spot positions resulting from the background-suppressed image depicted in figure 5. The data volume comprises 313 Laue spots in total. On the other hand, the sequential CPU-based spot finding algorithm provided 281 Laue spots (figure 6, right) from the same set of pnCCD images. The fact that the number of reflections found by the GPU method is about 10% larger than in the case of the CPU method is based on the more efficient background correction. For both methods the sets of spot coordinates were then used to compute the conventional unit cell of the crystal on a standard CPU by means of reduction and refinement procedures [11]. The extracted unit cells are characterized by the lattice constants and cell angles listed in table 1.

Within the instrumental resolution of the pnCCD the relative errors are in the range of 2.5% for the lattice constants and 3.5% for the cell angles. Hence, the calculated unit cells are in satisfying agreement with each other and with the expected lattice structure of tetragonal hen egg-white lysozyme.

2014 JINST 9 T11003

Table 2. Indexing of GPU- and CPU-computed Laue spots for various tolerances.

tolerance	GPU method	CPU method
0.25	297/313 (94.9%)	256/281 (91.1%)
0.20	284/313 (90.7%)	255/281 (90.7%)
0.15	268/313 (85.6%)	253/281 (90.0%)
0.10	206/313 (65.8%)	226/281 (80.4%)
0.05	109/313 (34.8%)	119/281 (42.3%)

In the final step of data analysis these cells served to index the individual Bragg peaks using the Laue equations. At this stage the quality of indexation was quantitatively evaluated by investigating the absolute deviation of the computed Miller indices h, k and l (given in floating-point numbers) from their next neighboring integers. Table 2 gives an overview of the obtained results expressed in terms of the fraction of indexed reflections, where tolerance is the maximum accepted deviation of h, k and l from integers.

Assuming a reliable indexation of individual Bragg peaks for a tolerance of 0.20 more than 90% of the extracted Laue spots could be indexed with a sufficient accuracy for both methods. The comparison between the two approaches at the level of indexed reflections at this tolerance is shown in figure 7. Quantitative analyses revealed that 216 out of 255 spots indexed by the CPU method were correctly reproduced by the GPU method corresponding to an amount of about 85%. Obviously, the agreement at the level of indexed reflections is significantly better than at the level of the spot coordinates (x, y, E). In the latter case a reasonable agreement could be found for 153 out of 255 indexed spots (60%) where a spatial deviation of one pixel in each direction and one standard deviation of the energy between the same Bragg peaks of different spot sets were accepted.

The obtained results demonstrate that the spot sets calculated both by the CPU algorithm and the GPU algorithm lead to the expected conventional unit cell of the crystal within the achievable resolution and provide a consistent indexation of the recorded Laue pattern. The important improvement associated with the GPU method is a significant reduction of the overall processing time. The (single-core) CPU algorithm takes approximately 30 minutes to extract spots from a raw data set of 100000 frames on a computer with 24 GB of RAM. This number decreases to less than 4 minutes for the GPU algorithm running on a computer with 4 Tesla C2050 GPUs (each consisting of 448 cores) and the same amount of RAM. This corresponds to a processing frame rate of 420 Hz which is well in the range of online processing for current pnCCD sensors with frame rates of up to 400 Hz. This achievement effectively makes the system more suitable for interactive use while running experiments.

5 Conclusion and outlook

In this paper, we developed a GPU-based algorithm which is suitable for computing the positions and energies of Laue spots collected by means of a pnCCD detector during an energy-dispersive Laue diffraction experiment. To this end, an existing algorithm running on a single-core CPU was modified in such a way that parallel data processing, allowing for effective utilization of GPUs,

2014 JINST 9 T11003

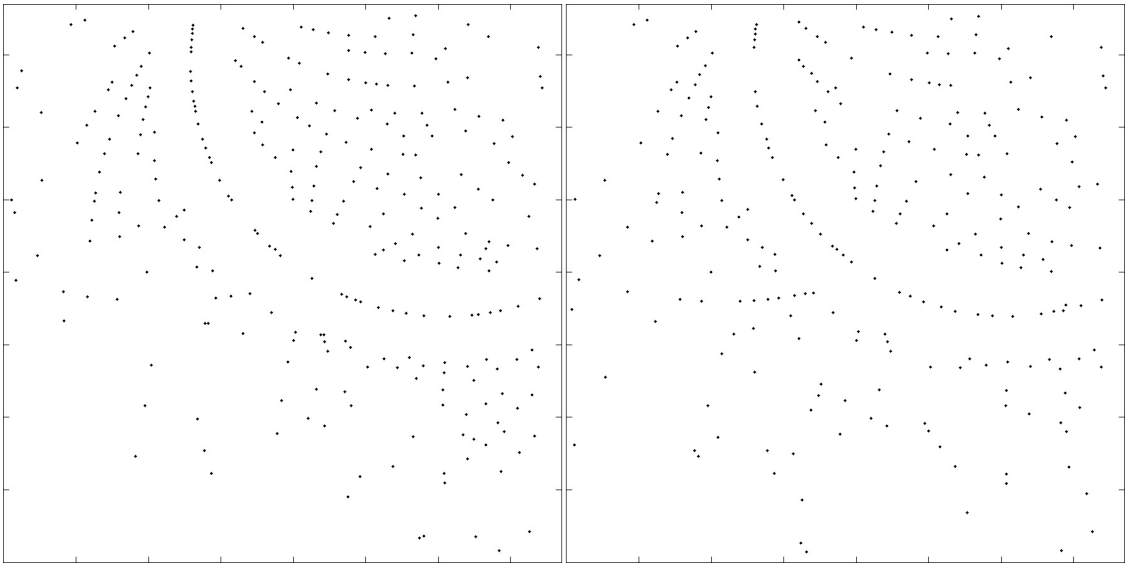

Figure 7. Comparison of indexed reflections, left: GPU result (284 spots), right: CPU result (255 spots) (image size 384×384 pixels).

is ensured. The new approach is based on a software framework for multi-GPU processing. The GPU algorithm enables a significant reduction of the data processing time by a factor of 7 (using four Tesla C2050 GPUs). By directly comparing the energy-resolved Laue pattern of tetragonal hen egg-white lysozyme obtained by the GPU algorithm with the results of the CPU algorithm it could be demonstrated that the two methods yield Laue spot sets of similar quality. In both cases, the obtained accuracy is sufficient to reproduce the expected unit cell of the crystal within the instrumental resolution of the pnCCD and to index the recorded Laue pattern. Moreover, as a consequence of an improved background correction, the number of correctly identified Laue spots could be increased by about 10% using the GPU algorithm. In the next step of GPU-based data processing, pile-up events will be taken into account in order to extract integrated Bragg peak intensities from the spectral information about individual Laue spots. The computed values can then be used for X-ray structure analysis on an atomic level based on the knowledge of experimental structure-factor amplitudes.

From a conceptual point of view, the potential of the developed method can be exploited in a variety of applications. Though presently restricted to energy-dispersive Laue diffraction experiments with a pnCCD, the concept of multi-GPU processing allows for a safe separation of the diffraction signal from the accompanying background. In this sense, the method can be generalized for a larger class of experiments where the spatial profile of the diffraction signal is not spot-like. Moreover, GPU algorithms of the presented form are applicable to any type of X-ray area detector (e.g. CCDs and image plates) producing data in terms of an image sequence.

Acknowledgments

This research was mainly funded by the German Ministry for Research and Education (BMBF) under grant No. 05K10PSB.

2014 JINST 9 T11003

References

[1] K. Moffat, *Laue diffraction*, *Methods Enzymol.* **277** (1997).

[2] K. Moffat et al., *X-ray Laue diffraction from protein crystals*, *Science* **223** (1984) 4643.

[3] L. Strüder et al., *The European photon imaging camera on XMM-Newton: the pn-CCD camera*, *Astron. Astrophys.* **365** (2001) L18.

[4] N. Meidinger et al., *PNCCD for photon detection from near infrared to X-rays*, *Nucl. Instrum. Meth.* **A 565** (2006) 251.

[5] S. Send et al., *Characterization of a pnCCD for applications with synchrotron radiation*, *Nucl. Instrum. Meth.* **A 711** (2013) 132.

[6] S. Send et al., *Energy-dispersive Laue diffraction by means of a frame-store pnCCD*, *J. Appl. Cryst.* **42** (2009) 1139.

[7] S. Send et al., *Analysis of polycrystallinity in hen egg-white lysozyme using a pnCCD*, *J. Appl. Cryst.* **45** (2012) 517.

[8] A. Abboud et al., *Energetic Structure of µLaue Peaks from a Plastically Deformed Cu Cantilever Revealed by a Three-Dimensional X-ray Detector*, *Rev. Sci. Instrum.* (2014), submitted.

[9] S. Granato et al., *Characterization of eROSITA PNCCDs*, *IEEE Trans. Nucl. Sci.* **60** (2013) 3150.

[10] R. Andritschke et al., *Data Analysis for Characterizing pnCCDs*, *IEEE Nucl. Sci. Symp. Conf. Rec.* **2008** (2008) 2166.

[11] S. Send, *Utilization of a frame store pnCCD for energy-dispersive Laue diffraction with white synchrotron radiation*, Ph.D. Thesis, University of Siegen, Siegen Germany (2013).

[12] D. Luebke et al., *GPGPU: general-purpose computation on graphics hardware*, *Proceedings of the 2006 ACM/IEEE conference on Supercomputing*, Tampa U.S.A. (2006), pg. 208.

[13] X. Jia et al., *GPU-based high-performance computing for radiation therapy*, *Phys. Med. Biol.* **59** (2014) R151.

[14] G. Pratx and L. Xing, *GPU computing in medical physics: A review*, *Med. Phys.* **38** (2011) 2685.

[15] J. Kanzaki, *Application of graphics processing unit (GPU) to software in elementary particle / high energy physics field*, *Procedia Comput. Sci.* **4** (2011) 869.

[16] G. Lamanna et al., *GPUs for fast triggering and pattern matching at the CERN experiment NA62*, *Nucl. Instrum. Meth.* **A 628** (2011) 457.

[17] G. Collazuol et al., *Fast online triggering in high-energy physics experiments using GPUs*, *Nucl. Instrum. Meth.* **A 662** (2012) 49.

[18] M.A. Clark et al., *Solving Lattice QCD systems of equations using mixed precision solvers on GPUs*, *Comput. Phys. Commun.* **181** (2010) 1517.

[19] R. Babich et al., *Scaling lattice QCD beyond 100 GPUs*, *Proceedings of International Conference for High Performance Computing, Networking, Storage and Analysis*, Seattle U.S.A. (2011), Article No. 70.

[20] F. Alghabi et al., *A scalable software framework for stateful stream data processing on multi-GPU systems and applications*, in *Symp. GPU Programming and Applications*, Springer, Berlin Germany (2013), accepted for publication.

[21] NVIDIA Corporation, *NVIDIA CUDA C programming Guide*, Ver. 4.0 (2011).

2014 JINST 9 T11003

[22] F. Alghabi et al., *Real-Time processing of pnCCD images using GPUs*, *14th International workshop on radiation imaging detectors*, Figueira da Foz Portugal (2012), pg. 178.

[23] M. Macedonia, *The GPU enters computing's mainstream*, *IEEE Computer* **36** (2003) 106.

[24] A. Abboud et al., *Sub-pixel resolution of a pnCCD for X-ray white beam applications*, 2013 *JINST* **8** P05005.

2014 JINST 9 T11003