Recent advances in the azimuthal integration library PyFAI

Giannis ASHIOTIS 1 , Jérôme KIEFFER 1

1

Data analysis / ESRF, Grenoble, France

E­mail: [jerome.kieffer@esrf.fr](mailto:jerome.kieffer@esrf.fr)

Azimuthal integration allows the use of area detectors for recording powder diffraction patterns

Data reduction step is very time-consuming → bottleneck

2D raster scans with pencil beam

diffraction tomography

Recent developments in PyFAI address such this issue by achieving the required execution speeds, without sacrificing accurancy

Here, we present recent developments in PyFAI which focuses both on the accuracy of the results and execution speed

. This data reduction step is usually a big bottleneck in the data processing pipeline and often limits the productivity of modern synchrotron beamlines, where diffraction is used to probe samples with a or experiments by using very fast detectors.

.

It is written in Python to facilitate the incorporation into the processing pipeline.

PyFAI is an open-source software package designed to perform azimuthal integration and 2D­regrouping on area detector frames for SAXS, WAXS and XRPD experiments. This is done by histogramming pixel positions weighted by the pixel intensity. Pixels spanning over more than one histogram bin contribute to the neighbouring bins accordingly. As a result This the noise in the diffraction pattern is greatly reduced. To achieve the necessary execution speeds required by online data analysis, the concept of a look-up table was introduced when working on sets of images sharing the same geometrical configuration. Furthermore the code is parallelized using OpenCL, which achieves very good performances even on very cost-effective graphics cards.

[1] J. Kieffer and D. Karkoulis (2013) Journal of Physics: Conference Series, 425, 202012

[2] J. Kieffer and J.P. Wright (2013) Powder Diffraction, 28 , S339­S350

PyFAI is an open-source software package for performing azimuthal integration and 2D-regrouping on area detector frames for SAXS, WAXS and XRPD experiments. This is done by histogramming pixel positions weighted by the pixel intensity. Pixels spanning over more than one histogram bin contribute to the neighbouring bins accordingly. This results in greatly reduced noise in the diffraction pattern. To achieve the necessary execution speeds required by such experiments, the concept of a look-up table was introduced. Furthermore the code is parallelized using OpenCL, which achieves very good performances even on very cost-effective graphics cards.