

Interactive Software for Efficient Processing of XPCS and SAXS Data from Two-dimensional Detectors

M.A. Borthwick,¹ P. Falus,^{1,2} S.G.J. Mochrie²

¹Massachusetts Institute of Technology (MIT), Cambridge, MA, U.S.A.

²Yale University, New Haven, CT, U.S.A.

Introduction

The use of charge-coupled device (CCD) arrays for correlation spectroscopy began roughly a decade ago [1], and the field of CCD-based x-ray photon correlation spectroscopy (XPCS) is nearly as old [2-4]. Since then, a number of important data collection and processing techniques specific to multispeckle analysis have been developed [5, 6]. Several advances beyond these are underway, and more are undoubtedly ahead.

The field is beginning to attract a broad user base, and, as such, it merits a distillation of the aforementioned collective expertise into a small number of standard software packages. Ideally, an XPCS data-reduction program would be easily accessible to novice users while simultaneously giving developers sufficient flexibility to implement and test new algorithms within a unified framework. It would generate “on-the-fly” results during beam time as well as allow for careful analysis off site.

A first attempt at such an all-in-one software package, called `coherent`, has been developed and tested heavily at beamline 8-ID at the APS. It is now in regular use there. Since correlation function normalization entails the use of the static scattering cross section, `coherent` also doubles as a small-angle x-ray scattering (SAXS) analysis tool. This public-domain software is written in the Yorick interpreted language [7]. Yorick is not only freely available for several operating systems but is also optimized for performing intensive mathematical operations on large, multidimensional arrays of the sort generated by CCD detectors.

Methods and Materials

A typical XPCS analysis in `coherent` proceeds through the following steps:

Interactive setup

- Data set selection
- Acquisition detail confirmation
- Inactive/unwanted pixel removal
- Image partitioning
 - Pixel-to- \mathbf{q} conversion
 - Selection of \mathbf{q} -partitions for pixel averaging

Analysis

- Division of images into pixel segments
- Image segment processing
 - Load segment out of all raw images

- Pixel response correction [8]
- Reorganization of “kinetics mode” slices [6]
- Multiple-tau correlation calculation
- Time-averaging of correlation function
- Correlation normalization
- Scattering cross-section calculation
- Time-averaging of cross section
- Correlation error estimate
- Average correlation within \mathbf{q} -partitions
- Cross-section error estimate
- Averaging of cross section within \mathbf{q} -partitions
- Scattering cross-section normalization

Results display

- Simple fitting of dynamics results
- Plot correlation function results
- Plot dynamics fit results versus \mathbf{q}
- Plot static cross-section versus \mathbf{q}

Saving of results to disk

Each step above nominally represents a module that works independently of all others to the maximum degree possible. This means that any unnecessary modules may be skipped during execution. For example, if the user selects SAXS analysis alone, then no correlation module is run. Ideally, it also allows, transparently to all others, the modification or addition of a module.

Results

Screen shots from `coherent` are shown in Figs. 1-4.

Discussion

The particular organizational structure above was chosen for `coherent` largely because it provides several speed advantages during analysis. First, Yorick’s optimized array syntax can be fully utilized because all readings of a given pixel are arrayed in memory at once, as opposed to all pixels from a single frame being in memory. Second, each image segment is read into memory only once, minimizing delays due to disk access. Third, time- and pixel-averaged result arrays are orders of magnitude smaller than their raw data counterparts, so computations (e.g., normalization) are applied to the processed data whenever possible.

One subtlety arising from this scheme is that image segmentation is necessary to array all readings of a given pixel in memory at once (the first condition in the

First-harmonic energy (keV)
 energy = 7.70744
 Undulator harmonic
 harmonic = []
 First x-coordinate of CCD Region Of Interest
 col-beg = 0
 Last x-coordinate of CCD Region Of Interest
 col-end = 1241
 First y-coordinate of CCD Region Of Interest
 row-beg = 0
 Last y-coordinate of CCD Region Of Interest
 row-end = 1151
 Kinetics mode (1 if used, [] if not)
 kinetics = 0
 Kinetics window size
 kinwinsize = []
 Top row number of visible slice in full-frame mode
 slicetop = []
 Pixel size (mm)
 dpix = 0.0225

FIG. 1. Acquisition detail confirmation. `coherent` is configured to read ASCII data produced by the acquisition software (spec, in this example) rather than to ask the user to input each of the dozens of experimental parameters.

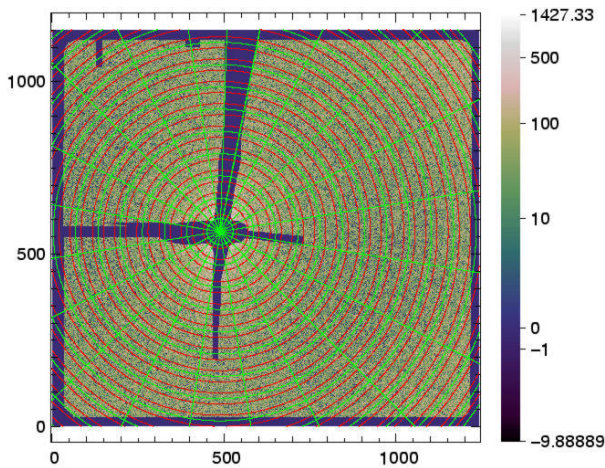


FIG. 2. q -partitioning of a data set. XPCS partitions (green) may be defined separately from SAXS partitions (red).

previous paragraph). This is because most raw data sets are several gigabytes in size, so it is unrealistic to assume that all computers possess sufficient memory to contain entire data sets. However, the loading of data from the disk is especially slow when anything other than full rows of pixels, which are stored contiguously on the disk, are loaded together. On the other hand, `coherent` is also

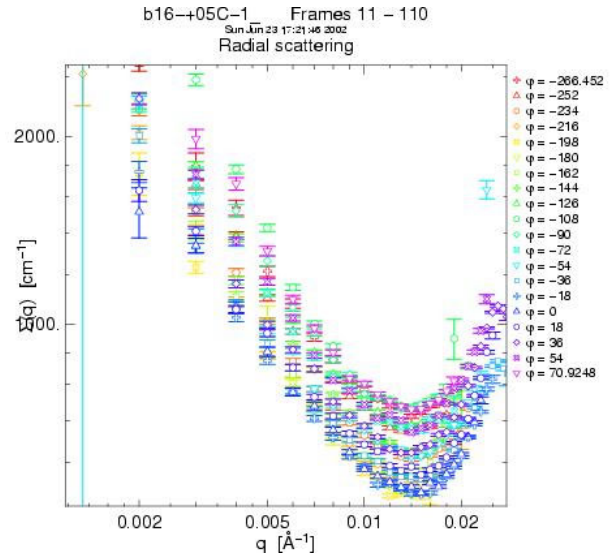


FIG. 3. Static scattering cross section versus $|q|$ at every azimuthal angle in an XPCS data set.

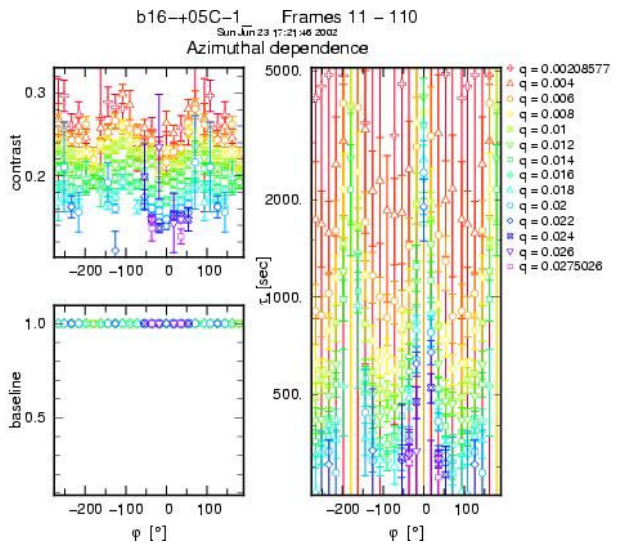


FIG. 4. Azimuthal angle-dependence of single-exponential decay fit parameters for correlation functions at every $|q|$ in an XPCS data set.

designed to allow users to select arbitrary q -partitions in two dimensions of literally any shape (provided a partitioning module able to handle such partitions exists). This means that partitions and segments will not be identical, and careful bookkeeping must be used to properly average within q -partitions that span several image segments. Fortunately, it also means that `coherent` is well-suited to potential future adaptation for parallel processing machines.

Despite the increase in speed by a factor of about 10 that is provided by requiring all readings of a pixel to be loaded into memory for correlation, there is a drawback as well. Namely, this condition makes `coherent` ill-suited to performing real-time analysis, wherein correlations are calculated during acquisition and updated upon the recording of each new frame, much as with point correlators. Even after the technical challenges currently facing such algorithms are overcome and real-time analysis becomes the norm, `coherent` may still have some utility, thanks to its image partitioning capabilities, results display features, and overall flexibility.

Acknowledgments

Work at beamline 8-ID at the APS was developed with support from the National Science Foundation (NSF) Instrumentation for Materials Research Program under DMR 9312543, from the U.S. Department of Energy (DOE) Facilities Initiative Program under DE-FG02-96ER45593, and from the Natural Sciences and Engineering Research Council of Canada (NSERC). Work at MIT was also supported by the NSF under DMR 0071755. Use of the APS was supported by the DOE Office of Science, Office of Basic Energy Sciences, under Contract No. W-31-109-ENG-38. `coherent` was initially undertaken as a means to merge disparate analysis routines written by L.B. Lurio and D. Lumma. Although

the finished product bears far less resemblance to those older packages than first intended, Lurio's and especially Lumma's contributions have nonetheless been substantial. M. Sutton, A. Rühm, and J.-F. Pelletier have also made important additions to several modules and to the overall philosophy of the package.

References

- [1] A.P.Y. Wong and P. Wiltzius, *Rev. Sci. Instrum.* **64**, 2547 (1993).
- [2] S.B. Dierker, R. Pindak, R.M. Fleming, I.K. Robinson, and L.E. Berman, *Phys. Rev. Lett.* **75**, 449 (1995).
- [3] S.G.J. Mochrie, A.M. Mayes, A.R. Sandy, M. Sutton, S. Brauer, G.B. Stephenson, D.L. Abernathy, and G. Grübel, *Phys. Rev. Lett.* **78**, 1725 (1997).
- [4] O.K.C. Tsui and S.G.J. Mochrie, *Phys. Rev. E* **57**, 2030 (1998).
- [5] L. Cipelletti and D.A. Weitz, *Rev. Sci. Instrum.* **70**, 3214 (1999).
- [6] D. Lumma, L.B. Lurio, S.G. J. Mochrie, and M. Sutton, *Rev. Sci. Instrum.* **71**, 3274 (2000).
- [7] Yorick software home page, <ftp://ftp-icf.llnl.gov/pub/Yorick/doc/index.html> (accessed Oct. 2003).
- [8] S.M. Gruner, M.W. Tate, and E.F. Eikenberry, *Rev. Sci. Instrum.* **73**, 2815 (2000).