Run-length encoding for mostly black images (#26)

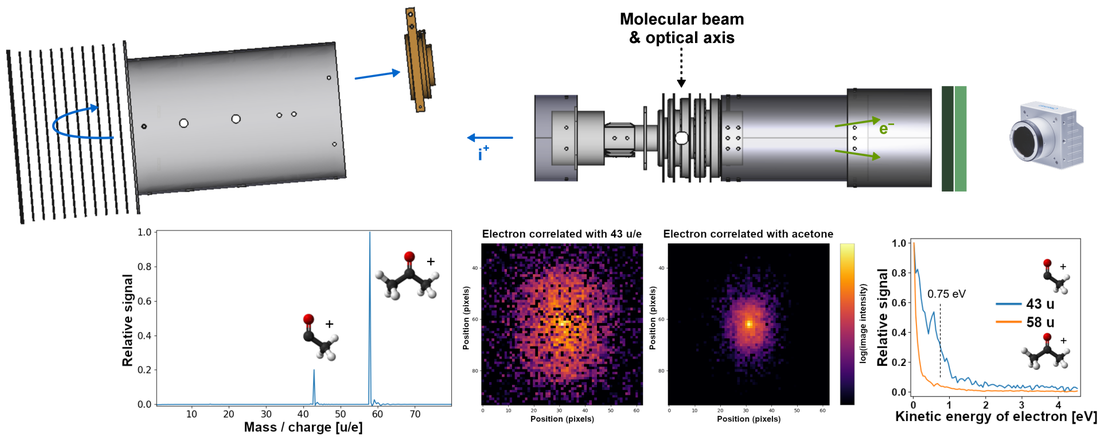
E. P. Månsson[1,*], L. Colaizzi[1,2], S. Ryabchuk[2], K. Saraswathula[1], V. Wanie[1], A. Trabattoni[1] and F. Calegari[1,2]

[1] CFEL, DESY, Hamburg, Germany. [2] Physics Department, Universität Hamburg, Hamburg, Germany. * erik.maansson@desy.de

Principle of covariance experiments

For multi-detector covariance analysis, e.g. between photoelectron energy and mass of ionic fragments of a molecule, it is necessary to save raw data (e.g. images and spectra) separately for each laser shot, rather than only saving the average over many laser shots. This can lead to higher rates of raw data than a typical computer or storage medium can handle at the required repetition rate of the experiment, unless suitable (lossy) compression is applied.

In the CFEL-ATTO group at DESY, we perform pump–probe experiments on molecules and use a spectrometer that has one side with a MCP & phosphor screen to image image the 2D velocity of electrons (i.e. angularly-resolved electron spectra) and another side for mass spectra of ionic fragments. By correlating the information from both sides, we aim to link the electron's information about the initial photoionization to an excited state with the ion's information about how the molecular dynamics played out (dissociation, localization of charge).



Data rates

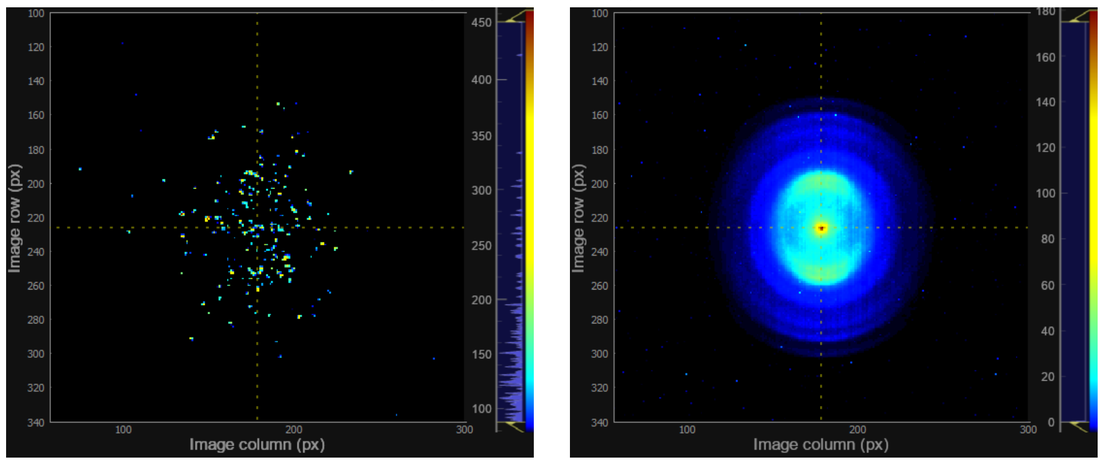
The main challenge for acquiring at the 1 kHz set by the laser system is the image data from a CMOS camera (Optronis CP70-1-M-1000). If it's maximum 1024x1024 pixel region of interest is chosen, with 12-bit sensor data padded to 16 bit integerss, the camera delivers 2.1 GB/s over CoaXPress to a PCIe card. We typically acquire for 10 seconds to 10 minutes before changing conditions, e.g. pump–probe delay in a scan, which would mean 0.13 to 1.3 TB raw data per file or 7 TB/hour. This is too much to save locally or even have a single computer compress using HDF5 (via pytables in Python & numba, on an Intel Xeon E5-1620 v4 3.5 GHz from 2018). The simultaneously acquired mass spectra (ADC waveforms) is two orders of magnitude smaller and therefore not the main topic here.

So far, we reduce the image data by hardware binning of 2x2 pixels and cropping (the entire detector isn't always needed) until the data rate can be handled (see below). Since the images often contain only a few hundred bright spots (detected electrons) it is acceptable to apply a lossy compression scheme where pixels darker than a threshold are set to zero. Currently the resulting image is passed to HDF5 for compression with LZO to about 1/10th to 1/40th of its raw size.

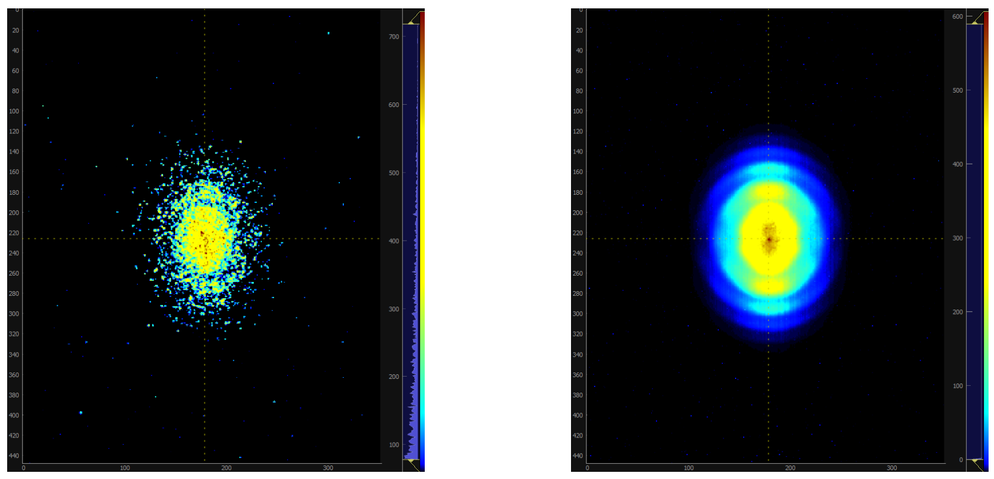
It would be desirable to be able to use the full 1024x1024 pixels for better energy resolution and this may be possible by doing an application-specific compression so that less data will need to be transferred to the HDF5-library. Generic compression algorithms like LZO and GZIP probably wastes CPU-time by trying to be "smart" when the main way that my noisy but mostly-black images can be compressed is to get rid of all the pixels having the value zero.

Example images

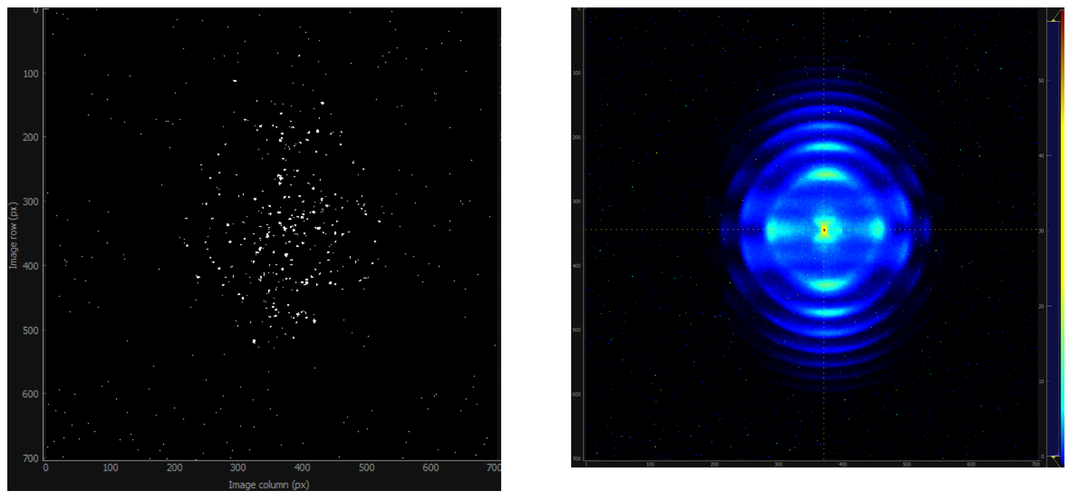
On the left is the image for one laser shot and on the right is the average of many images.
The horizontal and vertical axes correspond to 2D-velocity of electrons, with zero in the middle.



a) With 2x2 pixel binning and cropping by the camera. Intermediate number of electrons per image.



b) With 2x2 pixel binning and cropping by the camera. Many (too many) electrons per image.



c) Unbinned image too large to fully process (background-subtraction, thresholding, summing and HDF5-saving with LZO-compression) was not possible at 1 kHz (a screenshot is used on the left).

Compression approaches

The initially implemented and HDF5-performance-tuned approach has been to:

1. Subtract a dark image background, for things like dark current and CMOS readout noise.
2. Set pixels with differential intensity values below a threshold to zero.
3. (optional, but desired for users) Increment the "sum of all images" with the current image's data, so the GUI can show the average image live.
4. Send the image for HDF5-saving, appending it to an EArray with LZO-compression. The chunk size is at least 1 image (more if they fit in 1 MiB) and the chunk cache is 64 MiB.

(For covariance acquisition, another EArray with mass-spectrum TOF-waveform data is also written within the same file – only making the performance slightly worse. The program aims to work in blocks of about 128 buffered waveforms and 128 buffered images.)

Steps 1 & 2 are implemented within the same loop over the pixel array, compiled to machine code via LLVM by numba.
(*TODO: include step 3 in this loop.*)

The approach under development is to **change step 4** to avoid copying the large but mostly zero-filled array of pixels (2 MiB if the full camera resolution would be used) and also avoid any HDF5-compression filter. The compression instead needs to be handled in the application layer, so that it can be inlined in a single loop over the array (steps 1, 2 and maybe 3).

While "spot finding and centroiding" to only store particle coordinates would allow early data reduction, it is not suitable when the number of electrons is large enough that the spots (partly) overlap.

The proposed "darker-than-threshold run length encoding" (DRLE) processes the image as an 1D-array of signed 16-bit integers (the camera data uses 12 bits).

- For a run of length $L \geq 2$ successive pixels below the user-chosen threshold, the output buffer gets a negative value $-L$ appended.
- A single dark pixel is encoded as 0 to make the decoder's job easier (not needing to change the -1 for $L = 1$ to a single 0).
- Other pixels have positive values and are copied to the output buffer.

This lossy variant of RLE will never make the output longer than the input.

Performance benchmark

Test case		Shuffled LZO HDF5-filter		**DRLE**, no HDF5-filter	
Pixels in image	≥ threshold	Rate (kHz)	File size (MB)	Rate (kHz)	File size (MB)
a) 0.26 × 10^6	0.56 %	**1.12**	838	**6.74**	498
a) 0.5 × 10^6	0.56 %	0.59	839	**3.81**	500
a) 1.0 × 10^6	0.56 %	0.26	837	**1.96**	499
b) 0.5 × 10^6	5.51 %	0.53	3637	**2.50**	3887
b) 1.0 × 10^6	5.51 %	0.24	3630	**1.28**	3880

For typical threshold choices and numbers of electrons per image the old approach with only HDF LZO-compression gives a 71:1 compression ratio while the new DRLE gives 120:1. For images with too much signal (fewer black pixels) the ratios even out to 17:1 and 15:1, respectively.

Higher compression ratio as well as a simpler compression algorithm that avoids large array copies seems to allow full-sized images to be processed and saved at 1 kHz. The statistics reported here are from a script that attempts to be relevant but it is not the exact acquisition program (no mass spectra, background subtraction or GUI communication). 30 000 images were saved (repeating 100 MB of loaded real data) with pytables. Variable-length arrays with the circa 1 MB compressed data from chunks of 50 images perform as well as concatenating into one extendable 1D-array.

Although DRLE will never give a larger output than the input image, **achieving a high compression ratio is essential for making the file-writing work at 1 kHz**, especially when no further HDF5-compressor is used. The **choice of threshold level** is therefore important, done by the user while seeing the live data (possible also when file-writing is disabled).

Since covariance experiments rely on finding correlations in the shot-to-shot variations between the images (and mass spectra), they will typically not be performed with so strong signal that all pixels are bright or even "many electrons per images", so reasonable compression should be possible.

CUI THE HAMBURG CENTRE FOR ULTRAFAST IMAGING · CFEL SCIENCE · Universität Hamburg DER FORSCHUNG | DER LEHRE | DER BILDUNG · HELMHOLTZ · POLITECNICO MILANO · CNR IFN Istituto di Fotonica e Nanotecnologie · STARLIGHT · erc European Research Council Established by the European Commission