# Lossless compression of LCLS data

2017 SSRL/LCLS Users' Meeting

Mikhail Dubrovin

U.S. DEPARTMENT OF ENERGY
Office of Science

SLAC NATIONAL ACCELERATOR LABORATORY

# Problem

- LCLS-II vs. LCLS data flow and volume increase $\times 10^2 \div 10^3$.
- Need to think about disk space saving.
- What could we get with lossless compression of data?
  - how much disk space can be saved?
  - how much computing time it needs to compress, decompress data? (very arbitrary)

# Definitions

- *Compression:*


- *Compression factor (c.f.)* = <Data size>/<Compressed data size>
- *De-compression:*


- *Lossless compression:* De-compressed data = Data

# Data entropy

- Wikipedia: Entropy (information theory)
- Named after Boltzmann's H-theorem, Shannon defined the entropy H of a discrete random variable X with possible values {x1, ..., xn} and probability mass function P(X) and I(X)-information content, which can be explicitly written as:

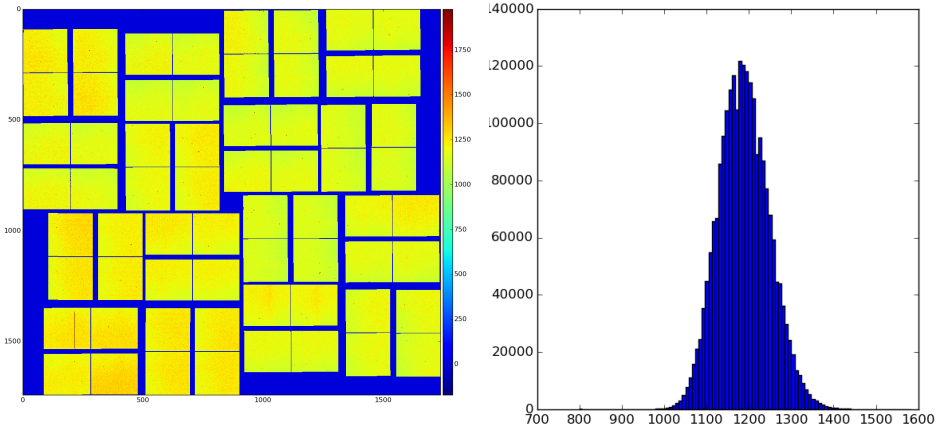
$$\mathrm{H}(X) = \sum_{i=1}^{n} \mathrm{P}(x_i)\,\mathrm{I}(x_i) = -\sum_{i=1}^{n} \mathrm{P}(x_i)\log_b \mathrm{P}(x_i),$$

- H for b=2 is an average number of bits needed to code each data value (word).
- Implemented in pyimgalgos/src/Entropy.py

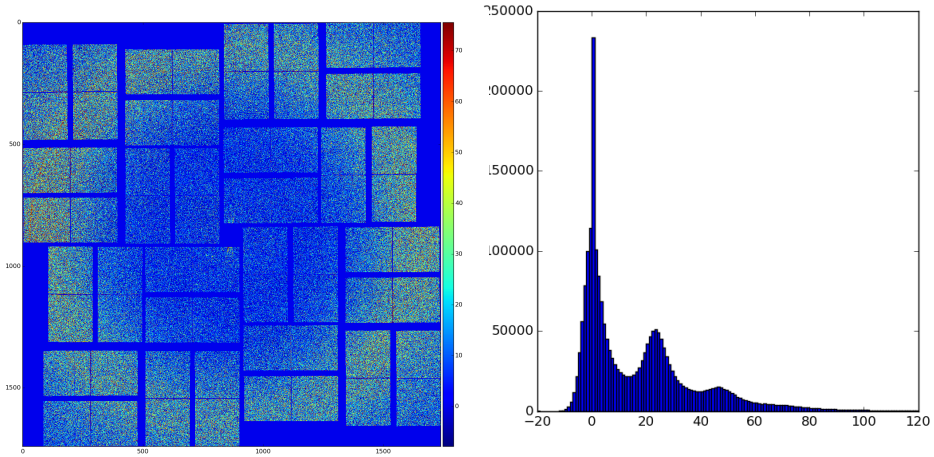
# CSPAD Data

- shape=(32,185,388)
- size=2296960 pix
- dtype=(u)int16

- Raw data entropy H=7.95,
  c.f.= 2.01




- Calibrated data
  (subtracted pedestals)
  H=5.84,
  c.f. = 2.74



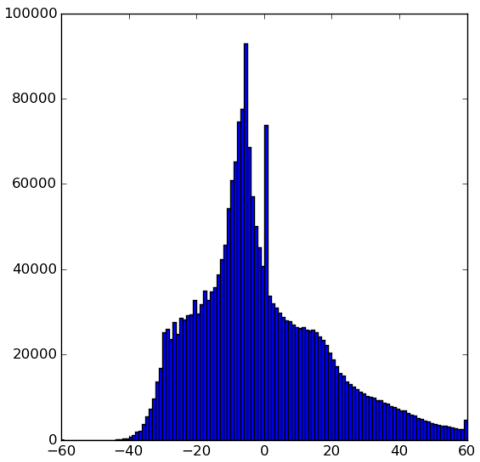
# Other calibration option

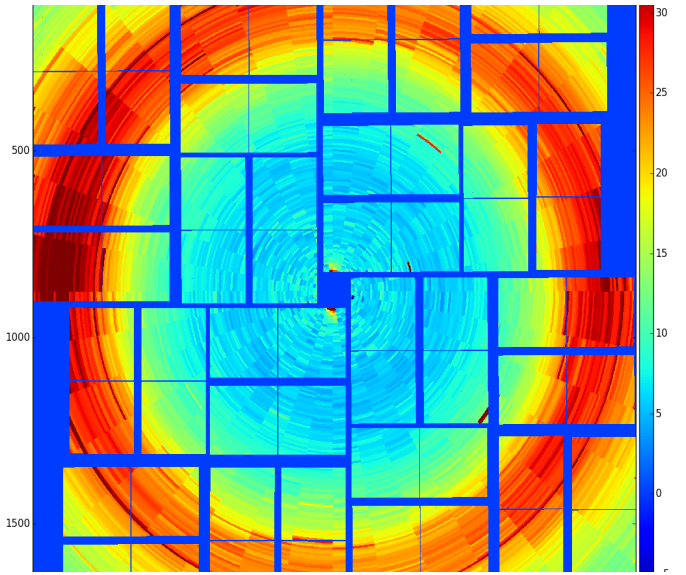
Subtraction of pedestals helps with compression. Does it help if we apply more corrections?

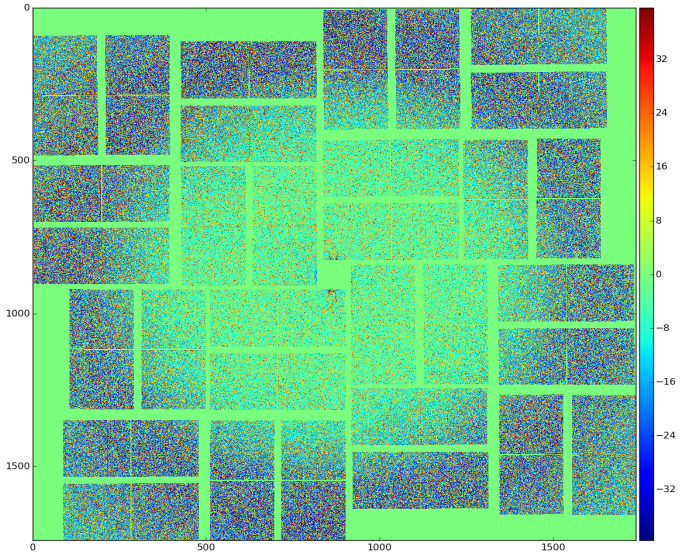
Subtraction of water-ring background

- “shifts” and “smears” spectrum
- Image (16-bit) array entropy H=6.28
- Estimated compression factor 2.55

Data spectrum is worse for compression



# Test of GZIP

- gzip-CLI – most popular and productive file-to-file compressor
  - gzip -c test.xtc > test.xtc.gz
  - compression factor = 1.89, time ≈ 3sec/event…
    - not optimal – xtc contains image and non-image data and metadata
    - slow

# Test of GZIP levels

- zlib-(gzip API) compression for single CSPAD image, test level:

| zlib level | Comp.factor | t(msec) comp. | t(msec) decomp. |
|---|---|---|---|
| 0 | 1.000 | 25 | 6 |
| 1 | 1.572 | 109 | 27 |
| 2 | 1.580 | 125 | 29 |
| 3 | 1.592 | 171 | 28 |
| 4 | 1.591 | 159 | 29 |
| 5 | 1.592 | 261 | 31 |
| 6 | 1.621 | 597 | 27 |
| 7 | 1.624 | 610 | 27 |
| 8 | 1.624 | 636 | 27 |
| 9 | 1.624 | 612 | 27 |

- compression time rises with level, but factor almost flat

# GZIP and LZF in HDF5

HDF5 has a few embedded compressors

- gzip with default level=4
- input size=4594000 byte
- time includes saving in file

| Data | Compressor | Shuffle | Comp. factor | t(msec) comp. |
|---|---|---|---|---|
| raw | gzip | False | 1.58 | 216 |
| | gzip | True | 1.95 | 147 |
| | lzs | False | 1.05 | 108 |
| | lzs | True | 1.70 | 75 |
| calib | gzip | False | 2.07 | 168 |
| | gzip | True | 2.19 | 183 |
| | lzs | False | 1.35 | 101 |
| | lzs | True | 1.47 | 87 |

- Compression factor of calib data is larger then raw data
- lzf is faster than gzip
- gzip compression factor is larger than lzf
- parameter *Shuffle* (word-byte separation) gives significant effect

Filters szip, lzo, blosc, bzip2 are unavailable in our installation of HDF5

# Home-made compressors for LCLS data

## Compressor for LCLS detector int16 data, Igor Gaponenko :

- estimates dataset spread, defines optimal offset,
- use 16- and 8-bit words to save data with positions coded in metadata
- Features
  - Optimized to work with 16-bit detector data only (not with xtc or hdf5 files containing metadata).
  - By design compression factor ≤2.
  - Single array of data is split and processed in multi-threads (inside compression algorithm).
  - Up to ≈two order of magnitude faster than gzip (Igor's statement).
  - Further specialization of data (separation of signal and background regions between threads) may improve compression factor.

## Compressors Hist16 & HistN, Matt Weaver

- Available in package pdsdata/compress
- Hist16 - the same as Igor's compressor, does not use multi-threading - slower than Igor's
- HistN – uses 16-bit and 8,7,6...-bit words, compression factor HistN upto ≈2.

# Summary

- A few standard and home-made lossless compression algorithms were considered against LCLS typical 16-bit detector data. The best compression factor achieved ≈2.2
- Data entropy is estimated as H=5.8 (at 16-bit word) for pedestal subtracted data, c.f.≈2.7
- Per-pixel-in-time entropy may be better, but not too much. Makes problematic per event access.
- Compression factor 2.2÷2.7 is helpful but does not completely solve disk space saving problem.