LARGE SYNOPTIC SURVEY TELESCOPE

Large Synoptic Survey Telescope (LSST)

Lossy Compression WG Report

A. Author, B. Author, and C. Author

DMTN-068

Latest Revision: 2018-02-19

DRAFT

Abstract

We report on the investigation into the use of lossy compression algorithms on LSST images that otherwise could not be stored for general retrieval and use by scientists.

Change Record

Version	Date	Description	Owner name
1	2017-04-17	Initial release. Based on LDM example	Tim Jenness
2	yyyy-mm-dd	Future changes	Future person

Contents

Lossy Compression WG Report

1 Introduction

The Lossy Compression WG was formed in response to RFC-325 with its charter being **?**. The purpose was to investigate whether some pipeline products (images) might be saved after applying a lossy compression algorithm without significantly degrading their suitability for many scientific investigations. The main reason for such a change would be to provide community access to some products without requiring on-the-fly reprocessing from the raw data.

In RFC-325 it was recognized that user experience will be unacceptably impacted by the long latency required to access the LSST data from tape media. Unfortunately, preliminary analysis indicated that retaining all processed images on disk would be too costly and therefore not feasible, unless lossy compression is applied. The same analysis indicated that storing all raw data on disk (w/o lossy compression) is feasible.

The LSST has traditionally avoided lossy compression for any of its image data products (including the large co-added images as well as templates retained for each data release). Anecdotal experience from the Dark Energy Survey (DES) and other surveys (e.g. HSC) indicates that lossy compression can be applied, without loss of scientific fidelity. If this is the case, the reduced disk space needs may enable LSST to retain on low-latency media more data that we otherwise would (rather than regenerate or pull from tape.

2 Methodology

This investigation is not meant to address the specific file format(s) that might be used to store LSST data (e.g.; FITS vs. HDF5). The tests that have been made were performed using images stored using FITS, mainly because the changes necessary could be used within the current LSST pipeline testing infrastructure. The specific images used were a set of HSC data that formed a modest depth tract on which pipeline regression testing was already being routinely performed in the development of the LSST pipelines. For this test set there were 33 images/CCDs, from 11 visit/exposures, at two bands (HSC-R, HSC-I). Included among these images are a 4 images near the edge of the HSC focal-plane, where vignetting causes a portion of the detector unusable for science. These regions are masked and present very different

noise characteristics but are useful because they show some caveats that must be considered when applying compression.

A change has been injected into the pipeline that allows for a quantization to be applied to the science (and weight) images that are traditionally stored as floats. Formally, the quantization factor, q, determines the number of samples/subdivisions of some set number, in this case the standard-deviation of the image pixels that do not contain a detected source. For a FITS image this is expressed as a scale factor (BSCALE) and the image pixel values are converted to the nearest integer multiple of this factor. We then use existing loss-less compression algorithms to compress the integer representation of the image to achieve a compressed image. Our tests varied the factor q from 4 to 128 (stepping by factors of 2).

Metrics are then obtained to understand the impact and efficacy of compression. Broadly, these fall into three categories:

1. **Image Compression benchmarks:** to measure the changes at the pixel level. These include: percent increase in noise/RMS, median difference, and number of pixels that change by more than the quantization level (to catch cases where the integer representation is not able to capture the full dynamic range of the original images).
2. **Catalog/Measurement benchmarks:** to measure the change of aggregate quantities of interest for scientists using the images for scientific measurements. The current benchmarks being measured are source position, flux, and shape along with their associated uncertainties.
3. **Compression algorithm benchmarks:** to measure the compression factor achieved, along with algorithm execution times for compression and decompression.

In addition a second round of image and catalog benchmarks can also be obtained to assess the changes that might be expected when compressed products are combined to form stacked images from which astronomical source measurements are also obtained.

3 Results

3.1 Image Compression Benchmarks

At the image level, independent measurements of the noise in the original science and weight images (I_0, W_0) and the quantized versions (I_q, W_q) are made. The algorithms used are independent of those that performed the estimates used to set the quantization. In most cases we consider only pixels with FLAG=0 or FLAG=32 (which indicates the presence of a source) as heavily masked regions often have values (particularly in the weight image) that can exceed the range accessible in the quantized images. Figure 1 shows the distributions of pixels values for the science and weight planes from two images typical of those in the test set.

A base level check is made to that examines the difference between the quantized and unquantized version of an image ($I_{diff} = I_q - I_0$). First the mean ($\bar{I_{diff}}$) and RMS ($\sigma_{I_{diff}}$) are computed to show that no systematic offset occurs and that the noise in the difference is indeed less than the scale factor. We then also search for pixels where the difference exceeds the quantization level. For most images this latter value is identically zero but in a small number of cases the pixels in a bright object will exceed the range available in the quantized image (i.e. the integer representation has insufficient cardinality to track the dynamic range in the image). If flagged pixels are included, then are typically more pixels that exceed this range and in the worst cases (e.g. images from CCDs that are vignetted) a large fraction of the weight pixels cannot be tracked. **More needed to quantitatively describe these?**

We then measure the standard deviation (RMS) in each science and weight image (σ_{I_q} and σ_{W_q} respectively) to understand the fractional increase in the image noise from the quantization ($\sigma_{grow} = \sqrt{(\sigma^2_{I_q} - \sigma^2_{I_0})}$. Figures 2 and 3 show histograms of these metrics based on the images in this test set. The left panels show the residual noise as measured from the difference between the unquantized and quantized images. The right panels show the fractional additive noise resulting from the quantization. Note that the number of samples in the histograms for q=64 and 128 this is because the measurement of the standard deviation is approaching the machine accuracy (i.e. σ_{I_q} differs from σ_{I_0} by less than a part in 10^6).

3.2 Catalog/Measurement benchmarks

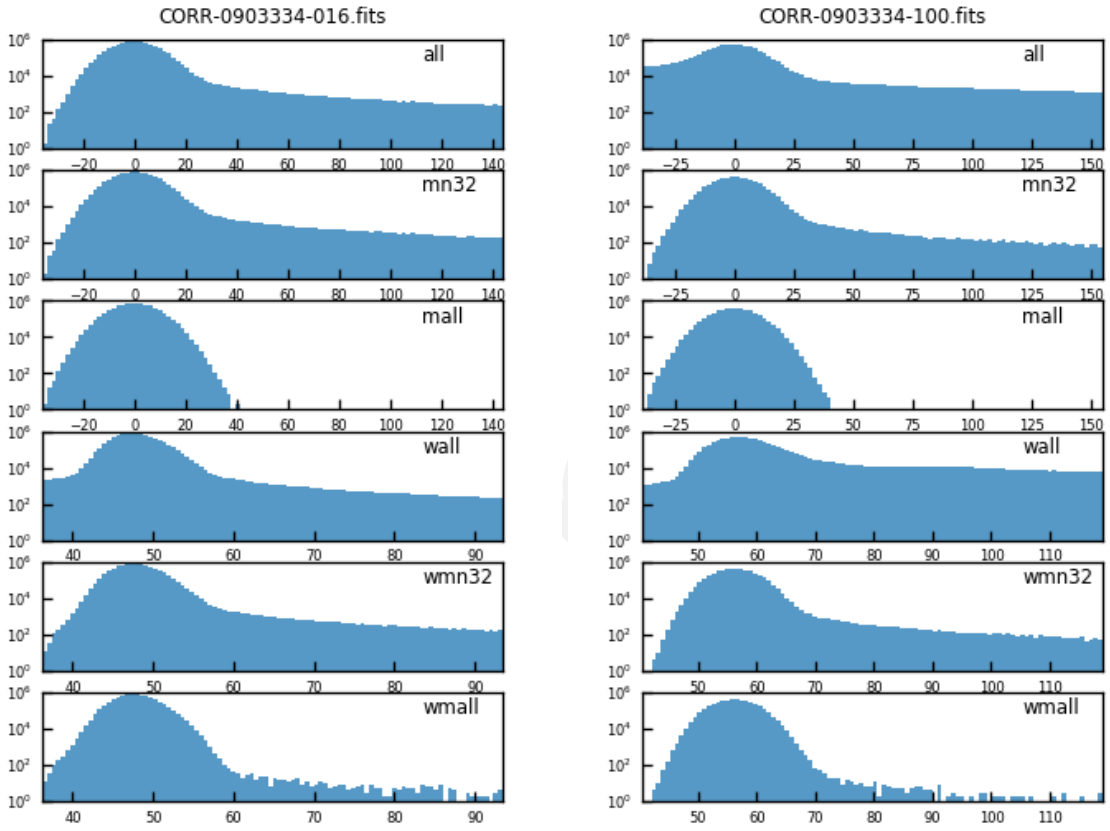

FIGURE 1: Histograms showing distribution of pixel values for two images in the set. (left) panels show distributions for a normal image, while (right) panels show the distributions for an image near the edge of the focal plane with heavy masking. (top) to (bottom) the panels show: all science plane pixels (all), all science pixels with MASK=0 or 32 (mn23), and all unmasked pixels (mall), followed by similar distributions for the weight plane (wall, wmn32, and wmall). Note the "mall" and "wmall" are roughly the distribution that was used to estimate the quantization level.

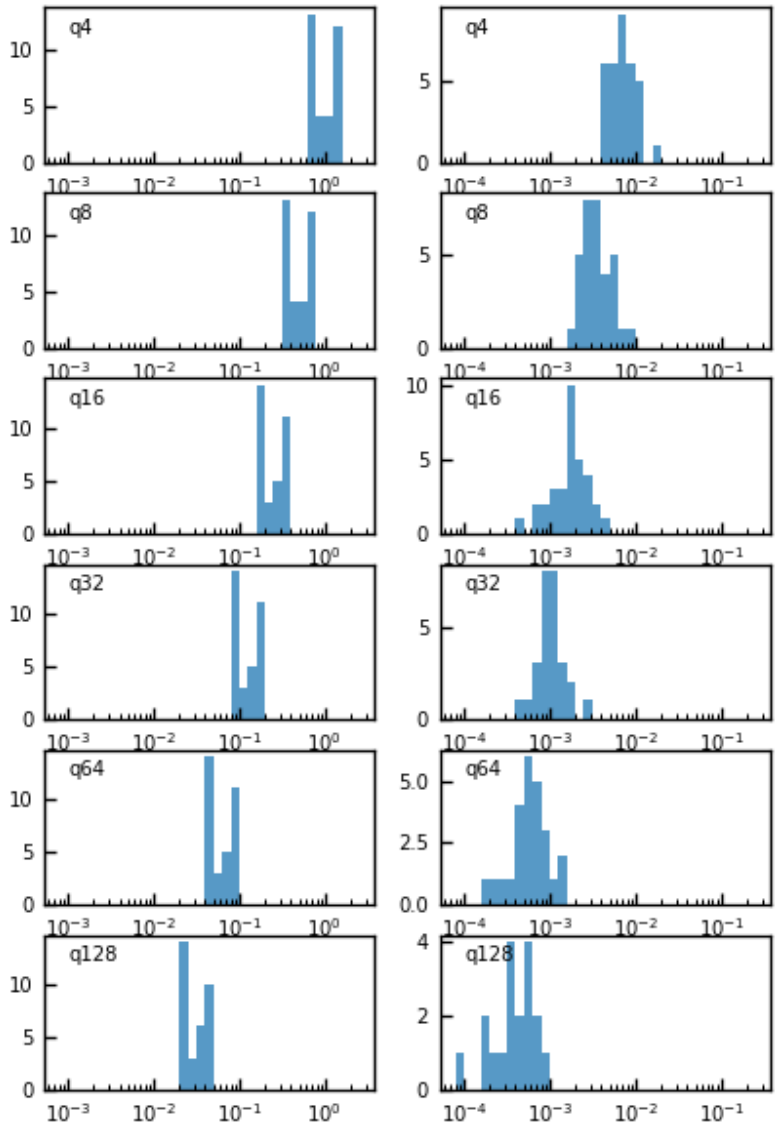

FIGURE 2: Histograms showing image level statistics with respect to the original compressed image. (left panels) are histograms showing the RMS of the difference between the compressed and original image. (right panels) are histograms of the fractional increase in the noise with respect to the original image.

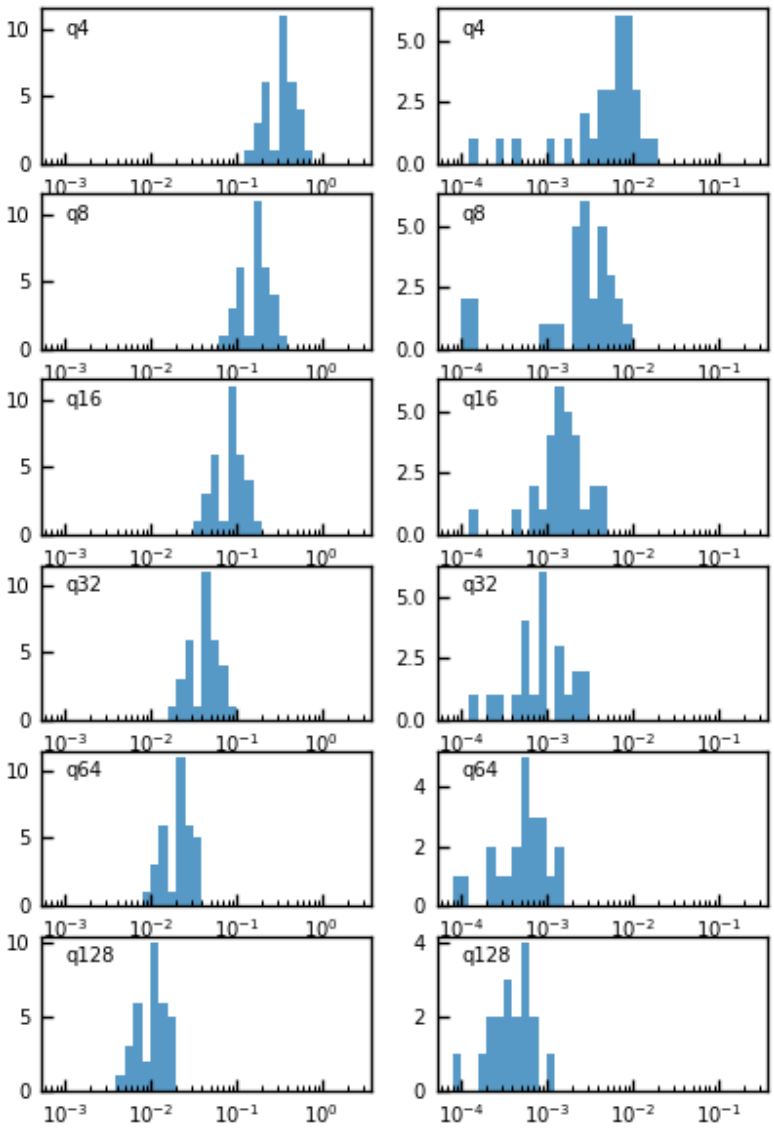

FIGURE 3: Similar to Figure 2 but for the weight image.

3.3 Compression Algorithm benchmarks

We have applied a variety of existing compression algorithms to the quantized images from this study to obtain benchmarks of their efficacy. The values reported reflect those algorithms' performance when running under OS X 10.13.2 (macOS High Sierra) on a MacBook Pro with quad 2.9 GHz processors. A ramdisk was used for storage to minimize the impact of I/O operations within the test.

A range of existing algorithms have been benchmarked, including a number which use threading to achieve greater speed. The algorithms considered were:

1. **gzip:** the standard GNU implementation of Lempel-Ziv (LZ77).
2. **pigz:** a threaded version of gzip.
3. **bzip2:** an implementation of Burrows-Wheeler block sorting (offers the possibility of recovery of undamage block).
4. **pbzip2:** a threaded/parallel implementation of bzip2.
5. **lbzip2:** another threaded/parallel implementation of bzip2.
6. **lz4:** a "typically faster" implementation of LZ77 (favoring speed over compression ratio). Pushing to higher compression ratios significantly degrades performance.
7. **lzop:** a separate implementation that trades a small hit in compression time for an improvement in decompression. The performance trade is not apparent at the file size being used in these tests.
8. **zstd:** Also based on the LZ77 family and includes a parallel implementation. In addition this algorithm has implementations/bindings over a wide variety of languages (including Python). The usage of a pre-computed dictionary may offer improvement in speed/compression factor but rigorous testing was not possible for this small set (performance was identical to the untrained algorithm if the full set was used both to train and then obtain benchmarks).
9. **xz:**

The results from benchmark tests are summarized in Tables 1-3, showing compression factor, time to compress per file, and time to decompress per file, respectively. These times do

Table 1: Compression Factor Achieved

q	gzip	pigz	bzip2	pbzip2	lbzip2	lz4	lzop	zstd	zstdb
q4	6.73	6.73	9.96	9.95	9.96	3.69	3.11	6.29	6.29
q8	5.54	5.53	8.20	8.20	8.21	3.34	2.96	5.42	5.42
q16	4.69	4.69	5.41	7.01	7.03	3.11	2.82	4.82	4.82
q32	4.04	4.03	6.14	6.14	6.14	2.93	2.66	4.35	4.35
q64	3.62	3.62	5.47	5.47	5.48	2.82	2.47	3.94	3.94
q128	3.38	3.37	4.88	4.88	4.88	2.66	2.32	3.56	3.57
vanilla	1.71	1.71	1.80	1.80	1.80	1.50	1.49	1.72	1.72

Table 2: Time to Compress per File

q	gzip	pigz	bzip2	pbzip2	lbzip2	lz4	lzop	zstd	zstdb
q4	4.45	1.18	5.00	1.42	0.85	0.21	0.24	0.36	0.12
q8	6.06	1.64	4.91	1.39	0.82	0.21	0.24	0.42	0.15
q16	8.27	2.24	4.33	1.39	0.82	0.27	0.27	0.55	0.18
q32	10.30	2.76	5.27	1.42	0.79	0.24	0.27	0.58	0.21
q64	11.79	3.00	5.39	1.52	0.88	0.24	0.30	0.61	0.24
q128	12.76	3.21	5.91	1.61	0.94	0.27	0.30	0.67	0.21
vanilla	3.36	0.97	8.94	2.79	1.58	0.15	0.12	0.30	0.15

Table 3: Time so Decompress per File

q	gzip	pigz	bzip2	pbzip2	lbzip2	lz4	lzop	zstd	zstdb
q4	0.21	0.24	2.30	1.21	1.27	0.15	0.18	0.27	0.24
q8	0.24	0.27	2.33	1.12	1.24	0.18	0.18	0.27	0.27
q16	0.27	0.27	2.02	1.12	1.21	0.18	0.18	0.27	0.24
q32	0.30	0.30	2.42	1.24	1.24	0.15	0.18	0.27	0.24
q64	0.30	0.30	2.42	1.27	1.09	0.18	0.21	0.27	0.27
q128	0.30	0.33	2.82	1.30	1.24	0.24	0.21	0.30	0.27
vanilla	0.39	0.36	4.36	1.48	1.27	0.15	0.12	0.24	0.24

not include the time to necessary to obtain and apply scale factor used in the quantization. Furthermore, the set of files being compressed are nearly identical (98 Mb) and therefore do not provide any information about algorithmic performance with respect to file size. When a parallel implementation was available the threading was set to use 4 cores.

4 Mapping Measurements to SRD?

5 Recommendations

The Working Group will provide a recommendation on the suitability of the implementation of a lossy compression algorithm of LSST products with a figure of merit that can be applied with the LSST Sizing Model ([LDM-144]).

6 WG Membership

Membership of roughly four people is optimal and should include persons familiar with weak-lensing and difference imaging concerns. The proposed membership is:

- Robert Gruendl (NCSA; **Chair**),
- Paul Price (Princeton),
- Bob Armstrong (Princeton),
- Krzysztof Findeisen (UW; replacing John Parejko),
- Sophie Reed (Princeton),
- Eric Morganson (DES/NCSA; observer)
- Ben Emmons (EPO Tucson; observer)

References

[LDM-144], Freemon, M., Pietrowicz, S., Alt, J., 2016, *Site Specific Infrastructure Estimation Model*, LDM-144, URL https://ls.st/LDM-144