



Vera C. Rubin Observatory  
Data Management

# Visualization of Calibration Verification

Chris Waters

DMTN-192

Latest Revision: 2021-06-15



## Abstract

This technote will discuss the expected visualization needs to support daily verification of existing calibrations against incoming data.

## Change Record

Version	Date	Description	Owner name
1	YYYY-MM-DD	Unreleased.	Chris Waters

*Document source location:* <https://github.com/lstt-dm/dmtn-192>

## Contents

<b>1 Introduction</b>	<b>1</b>
<b>2 Verification for Certification</b>	<b>1</b>
<b>3 Full Focal-Plane Views</b>	<b>2</b>
<b>4 Online Verification</b>	<b>2</b>
<b>5 Tool Definitions</b>	<b>2</b>
<b>6 Conclusions</b>	<b>3</b>
<b>A References</b>	<b>3</b>
<b>B Acronyms</b>	<b>3</b>

# Visualization of Calibration Verification

## 1 Introduction

Instrument signature removal (ISR) requires calibrations that accurately represent the state of the camera. The `cp_verify` package is designed to apply calibrations to an appropriate set of input exposures, and then attempt to measure a set of statistical parameters that can be used to determine if that calibration is removing the instrumental signature. However, given that each type of calibration can have multiple statistics, and the large number of detectors and amplifiers in the camera, having a set of visualization tools is essential to use these measurements.

The goal of this tech note is to provide examples of the visualization tools and plots that will likely be necessary for calibration verification to be a simple and fast process. There are currently two expected modes that `cp_verify` will be run in: first, to validate a newly generated calibration is satisfactory prior to certification; and secondly to compare newly taken daily calibration frames against existing calibrations to confirm that those existing calibrations still correct all of the instrument specific signals they were designed to. Both of these cases are discussed in the following sections.

## 2 Verification for Certification

To be accepted for certification, calibrations need only pass the tests and quality metrics defined in DMTN-101. The yaml output from `cp_verify` will provide a simple binary answer to that question. Plotting the per-amplifier values for each test as a full focal plane heat map provides a way to quickly check that there are no raft based or gradient like structures in the metrics. As many of the per-amplifier metrics are defined in terms of some residual signal remaining, histograms of those values also gives the user a way to check that the distributions are understood.

Although the focal plane heat map and metric histograms are likely the main visualization methods for certification, scatter plots of the measured values should definitely be supported. This allows two parameters that are coupled to be monitored as well. Plotting the value of a metric (possibly using candlestick points to illustrate the amplifier to amplifier scatter) as a

function of the observation date may help determining the validity range of the calibration.

There is no way currently to directly compare two calibrations in `cp_verify`, except by running independent `cp_verify` runs against the same set of raw images. CZW: per-amp differences; delta function at zero for completely matching values.

Lists of outliers?

### 3 Full Focal-Plane Views

We will probably want to use `visualizeVisit.py` to generate full focal plane images. This either needs to be done automatically as part of construction, or a new pipeline should be defined to do that.

### 4 Online Verification

Combination of the candlestick plots and per-amp difference ideas. Record time series somewhere so each day can add candlestick data. Monitoring the daily changes will identify drifts in the calibration quality. Sudden changes will prompt generation of new calibrations.

### 5 Tool Definitions

A `cp_verify` visualization script will need to be added to that package. It should be capable of generating all of the plots for a particular calibration and writing them out for viewing. Although this will not be a `pipetask`, it will need to access the butler repository to read the `cp_verify` outputs. An alternate solution is to add a jupyter notebook to the `cp_verify` examples directory, which would provide a consistent set of visualization and plotting tools for all calibrations identified in a given repository and collection.

CZW: Andres brought up the good idea that standarized notebooks could probably solve the issue of looking at everything and generating plots, etc. Maybe we do that instead of new tools?

## **6 Conclusions**

## **A References**

## **B Acronyms**