



Vera C. Rubin Observatory
Systems Engineering

CBP Pointing and Ghosting Simulation Notebook

Kane Sjoberg

SITCOMTN-134

Latest Revision: 2024-08-06



Abstract

This note gives an overview of a jupyter notebook that performs raytracing through the CBP and LSST and computes ghosting and transmission profiles. The notebook is available at:

https://github.com/lsst-ts/cbp_analysis/tree/b7cd1ce2b2fbd5c0c0ffdd08eabdfc9405884951/Pointing%20%26%20ghosting%20modeling%20notebook The notebook has three main uses:

1. Returns CBP and LSST alt/az pointings for a given positioning of the CBP within the observatory dome and desired CBP pointing on the M1 pupil.
2. Raytracing for any CBP mask, returning final beam positions on the focal plane of LSSTcam and Comcam.
3. Ghosting analysis: raytracing returns realistic estimations of ghosting along with transmission values for the CBP across the wavelength spectrum.

Change Record

Version	Date	Description	Owner name
1	2024-08-06	Version 1.0	Kane Sjoberg

Document source location: <https://github.com/lstt-sitcom/sitcomtn-134>

Contents

1 Introduction	1
2 Dependencies	1
3 Configuring Simulation Parameters	1
4 Usage Notes	2
5 Analysis Sections	2
5.1 Pointing Calculations	3
5.2 Raytracing	3
5.3 Ghosting Analysis	3
6 Summary	3
7 Acknowledgements	3

CBP Pointing and Ghosting Simulation Notebook

CBP Pointing and Ghosting Simulation Notebook

Kane Sjoberg

2024-08-06

1 Introduction

The simulation notebook is designed to support the CBP's integration and simulate the results of different pinhole masks for projection into LSST. Described below is the usage overview.

2 Dependencies

The following Python packages are required to run the notebook: `numpy`, `matplotlib`, `batoid`, `ipyvolume`, `os`, `scipy`, `pandas`, `lsst`

3 Configuring Simulation Parameters

The notebook allows the following parameters to be set:

- **Camera Selection:** Choose between `comcam` and `lsstcam`.
- **Mask Configurations:** Select a CBP pinhole masks for projection; new masks can be defined using a list of tuples where the tuples are (x,y) positions of holes in millimeters. Pre-defined masks include: `one_per_ccd`: places one beam on each ccd of `lsstcam`. `one_per_amp`: places on beam on each amplifier of `comcam`. `center_hole`: a single beam, ideal for efficient interactive 3D renderings of the CBP and LSST for pointing analysis and confirmation.
- **Wavelength and Filter:** Input the wavelength of light and select the appropriate filter. Wavelength range is $320\text{e-}9$ to $1125\text{e-}9$ m. Filter options: `'u'`, `'g'`, `'r'`, `'i'`, `'z'`, `'y4'`, or `None` (no quotations)

- **Raytracing Bundle Distribution and Density:** Two primary methods for initiating bundles of rays at the CBP focal plane. The option 'cross' defines a single bundle of n_x rays at each pinhole, forming a symmetric cross shape of beams. This method is best suited for efficient geometrical pointing modeling and 3D visualization. $n_x \geq 4$ produces true cross-shaped ray bundles from raytracing; $n_x=3$ produces single rays for each pinhole in the mask. For realistic ghosting simulations, the option 'symmetric light cone' allows users to specify the angular density of rays to be traced within each bundle with (n_{θ} , n_{ϕ}). Furthermore, each pinhole in the mask initializes several bundles distributed across the pinhole area for accurate ghosting simulations, with the distribution defined by ($pinhole_nradius$, $pinhole_nphis$). The defaults for each are (6,6); ghosting effects change based upon the density of rays chosen, and render time is very sensitive to the density of rays chosen, especially with large pinhole masks like `one_per_ccd`.
- **Ghosting Cutoff:** Specify the minimum transmission value for non-sequential ghosting tracing. Rays attaining a fractional flux below the specified cutoff will not be included in the raytrace results. $1e-8$ is the default, as this captures third-order reflection ghosts on the detector.
- **Geometrical In-Dome Position:** In cylindrical coordinates, set the radial distance, height, and azimuth position of the CBP relative to the telescope's center of rotation.
- **CBP M1 Pointing:** Define the CBP's pointing on the M1 pupil plane using polar coordinates. The default value is $\theta_{deg} = 0$, $radius_{mm} = 3369$. The available unobscured M1 aperture is 2558 to 4180 millimeters, with 3369mm being the midpoint.

4 Usage Notes

Notebook runtime and load is sensitive to the minimum flux trace cutoff, along with the ray and bundle density parameters; significantly increasing the number of rays, lowering the flux cutoff, or increasing pinhole masks beyond the current parameters is likely to cause the notebook to crash on USDF servers. For best results, load this notebook on a 16gb / 4-core virtual machine. When modifying configuration parameters to compute a fresh raytrace, re-run the dependency import cell.

5 Analysis Sections

5.1 Pointing Calculations

The notebook calculates the altitude and azimuth pointings of the CBP and LSST based on the user-defined geometrical position of the CBP relative to the main telescope. This is done via the `lsst.CoordinateConverter` class to compute the necessary transformations, ensuring that the CBP's pointing aligns accurately with the M1 pupil plane.

5.2 Raytracing

Raytracing is performed using the `batoid` library, which models the optical path through the CBP and LSST systems based on the user geometry specified, and the output of the `lsst.CoordinateConverter`.

5.3 Ghosting Analysis

Ghosting analysis traces rays non-sequentially through multiple back-reflections within the optical system. Accurate transmission values for each optic in LSST are used to calculate the total transmission efficiency of the telescope and trace ghosts.

6 Summary

The CBP Pointing, Transmission, and Ghosting Simulation Notebook is a comprehensive experimental tool for evaluating the optical performance and throughput of the CBP and LSST systems. It is intended to be flexible and modular, accommodating experimentation.

7 Acknowledgements

This notebook builds upon code from Jingyi Zhang (`batoid` initialization), Josh Meyers (`batoid` package), Thierry Souverin (`ccd` graphics), Nathan Amouroux (ghosting initialization), and Elana Urbach (throughput values from the Exposure Time Calculator).