Noise Cancellation Test Pipeline Manual

Henning Jansen
henning.jansen@uibk.ac.at

March 2023

1 Disclaimer

During the process of finishing the paper, we found that KiDS uses a modification for the half-light radius of galaxies. This modification multiplies the half-light radius measured with `Galfit` by $\sqrt{q}$, where q is the axis ratio of the ellipse. This has to be done, because `Galsim` interprets the given half-light radius as a circularized half-light radius, while the measurement was done along the semi-major axis of the ellipse. Translating one definition into the other requires this correction factor. Since it does not affect the main results of the paper, we did not run everything again with this modification. But for further work, this is implemented in the code now. So if you get different absolute biases compared to the paper, this is due to the slight modification of galaxy sizes.

2 Introduction

This pipeline is designed to test different noise cancellation methods for shear calibration. It is mainly written in Python and uses the `galsim` module for galaxy simulation. The methods, which can be studied with this pipeline are shape noise cancellation, pixel noise cancellation, and the response method. It is possible to do this both on a grid and on more realistic scenes with randomly positioned galaxies. The output of this pipeline consists of binned improvements (in runtime and equivalent area) and biases for the desired setup of simulations. These scripts are **not** a product of the Euclid Consortium Science Ground Segment. They are created exclusively for the present analysis, and made public for reproducibility of the results presented in this paper.

3 Requirements

To run the pipeline, the required packages can be installed by creating a (mini-)conda environment from the provided .yml file. You can also find a `random_galsim.py` and a `noise_galsim.py`, which need to be exchanged in the respective directory of `galsim` to have the functionality of the Poisson noise generator, which is used in the pipeline.

Just overwrite them in your `/miniconda3/envs/galsim/lib/python3.9/site-packages/galsim-` directory. Now activate the environment and you are good to go. A minimal functional version of the required file structure is also available and can be copied whereever you like.

4 Usage

To run the whole pipeline for either the grid or the random position simulations, one only needs the two bash scripts `run_grid.sh` and `run_random_positions.sh` and the respective configuration files `config_grid.ini` and `config_rp.ini`. There is also an experimental scripts `run_variable_shear.sh`, which is supposed to estimate the impact of blends with different shear. **In order for this script to function correctly, the option *"same_but_shear"* needs to be activated in the configuration file.** This script is still in development and does not belong to the main functionality of the pipeline, so treat this carefully. Extensive descriptions of the parameters in the configuration files can be found in Appendix A. We recommend adjusting the parameters in the configuration files first and then running the bash script, since some parameters will be changed depending on your inputs to the bash script. When the script asks you for a working path, the absolute path to the simulation directory is required. If you want to do only the analysis, since the simulations are already done, you need to give the paths to the finished simulations as well. Here you only need to give the name of the directory and not the absolute path. The other inputs are self-explanatory.

If you want to dive deeper in the functionality of the pipeline and change individual scripts yourself, here is a summary of the scripts. For each of the two possible setups, the main steps of the pipeline with individual modules handling them are:

1. Simulation of images and generation of catalogs
2. Analysis of those catalogs
3. Bias estimation with uncertainties
4. Study of the uncertainty behavior with runtime / simulated area
5. Plotting the final results

A list of all the scripts available with their task is given in Appendix B.

5 Output

If you run the whole pipeline, you will find several outputs. These are listed in the following with their respective process number from before. All those files can be found somewhere in the output directory of the simulations directory.

1. Directories named with a timestamp in either "grid_simulations" or "rp_simulations". These directories contain only an input catalog and a shear catalog after the first step. If you turn on the option to output the images themselves, you can find them in the subdirectory "FITS_org".
2. After the analysis step, these directories also contain additional files. For the fit method this is either an "analysis.dat" file on random positions or the "results.dat" files for the grid. These files essentially contain the information about the input shear against the measured shear, which is used for the fit later on.
3. The corresponding simulation directory will after this step inhabit a "fits.txt" file, which contains the fitted biases for the fit method and the results from the response method. The plots corresponding to these results can be found in the "fits" directory, which is a subdirectory of "plots". The option to save the plots can be turned off in the configuration file, as many plots will be created.
4. After this step, the runtime improvements are fitted. The fitting results are saved in the "error_scaling.txt" file and the plots corresponding to the fits can be found in the plots directory.
5. Hereafter, you will see the final results. A binned bias comparison and a binned runtime improvement can be found for both biases in the plots directory.

A Configuration files

In general the configuration files have several adjustable parameters in common with some unique options for each of the setups (grid and random positions). The common parameters are listed in Table 1 and the unique parameters for grid and random positions in Table 2 and Table 3 respectively.

B Scripts

Parameter	Description	Options
	IMAGE	
pixel_scale	Pixel scale of the instrument in arcsec	any float number
exp_time	Exposure time of the image in seconds	any float number
gain	Gain of the instrument in electrons / ADU	any float number
read_noise	Read noise of the instrument in electrons	any float number
sky	Sky level of the image in mag/arcsec^2	any float number
zp	Magnitude which generates one electron per second per pixel	any float number
stamp_xsize	The length of a stamp in pixel	any integer number
stamp_ysize	The height of a stamp in pixel	any integer number
ssamp_grid	The subsample factor for measurement	any integer number
shift_galaxies	If the galaxy center shall be shifted	True or False
shift_type	Shifting in a circle or a square around the center	SQUARE or CIRCLE
shift_radius	The maximum shift distance in arcsec	any float number
	PSF	
psf	Which kind of PSF shall be used	EUCLID, AIRY or GAUSS
lam_min	smallest wavelength in the bandpass in nm (max is 900)	any float number
step_psf	The stepsize in which monochromatic PSF's are sampled	any integer number
tel_diam	Diameter of the telescope in m	any float number
	SIMULATION	
bootstrap_repetitions	How many bootstrap samples to generate	any integer number
reps_for_improvements	How many different fits to do for improvements uncertainty	any integer value
bins_mag	How many magnitude bins	any integer number
min_mag	Brightest magnitude to consider	any float number
max_max	Faintest magnitude to consider	any float number
num_cores	How many workers to use for the parallelization	any integer number
random_seed	The random seed used to control the noise if needed	any integer number
ellip_rms	Standard deviation for the Rayleigh distribution of ellipticities	any float number
ellip_max	Ellipticity cut to avoid too elliptical galaxies	any float number
g2	Options for the second shear component	ZERO, UNIFORM, GAUSS
sn_cut	The implemented signal-to-noise cut	any float number
save_fits	Do you want to save the individual fits for the biases?	True or False
	TIMINGS	
noise_plus_meas	Relative runtime of noise generation and KSB measurement	any float number

Table 1: Shared parameters of both config files

Parameter	Description	Options
	SIMULATION	
same_galaxies	Use the same galaxies for each constant input shear?	True or False
output	Do you want a fits file of the produced stamps (can become large)	True or False
selection	Only consider full cancellations?	True or False
two_in_one	Do shape noise and pixel noise cancellation in one image	True or False
same_noise_and_shift	Use same noise seed and sub-pixel shift for shape cancel	True or False
bin_type	Bin in measured magnitude (adamom) or input magnitude	GEMS or MEAS
sel_bias	Do you want the output shear to be the true input shear?	True or False

Table 2: Specific parameters for grid simulations

Parameter	Description	Options
	IMAGE	
cut_size	**half** of the side length of the cut-out that is used for KSB measurement	any integer value
	SIMULATION	
summarize_pujol	Summarize always two runs belonging to each other?	True or False
bin_type	Magnitude estimate to use for the binning	MAG_AUTO or GEMS
shear_bins	How many constant input shear bins to use	any integer value
same_but_shear	Ongoing work for variable shear fields	True or False
same_but_shear_diff	The generated difference in shear between two versions	any float value
variable_field_mag	The magnitude of the (random) variable shear field	any float value
puj_analyse_every	Analyse only every n-th run for runtime save	any integer value
skip_first_lf	Skip the first n points of the linear fit for the analysis	integer > 5
plot_every_lf	For better visibility plot only every n points of the linear fit	any integer value
skip_first_rm	Skip the first n points of the response method for the analysis	integer > 5
	MATCHING	
max_dist	Maximum distance in pixel to search for a partner in the input catalog	any float value
max_neighbors	Maximum number of neighbors to consider for a matching in magnitude space	any integer value
output	Do you wish to save the generated images for later usage?	True or False
	TIMINGS	
scene_creation	Relative runtime to create one of the scenes from individual stamps	any float value

Table 3: Specific parameters for random position simulations

Script name	Task number	Description
`grid_simulation.py`	1	Generates a catalog of measured shears for galaxies with different constant input shear. This is used for the fit method later on.
`pujol_grid.py`	1	Generates a catalog of measured shears for n galaxies simulated with different shears but the same noise. This is used for the response method later on.
`rp_simulation.py`	1	Generates scenes with randomly positioned galaxies, extracts sources with `SourceExtractor` and creates a catalog with the measured ellipticities of those sources
`pujol_rp.py`	1	Generates several versions with slightly different shear of the same scene with randomly positioned galaxies, extracts the sources with `SourceExtractor` and creates a catalog of measured ellipticities of those sources.
`grid_analysis.py`	2	Reads in the catalog and generates an output file with the input shear and the measured shears with the respective runtimes and uncertainties
`pujol_grid_analysis.py`	2, 3	Reads in the catalog generated by `pujol_grid.py` and determines the responses and biases from the measured ellipticities.
`rp_analysis.py`	2	Reads in the catalog from `rp_simulation.py` and generates an output file with the input shears and the measured shears for all scenes and all magnitude bins. The uncertainty here is just the standard deviation of the measured ellipticities. The bootstrapping happens in the another script.
`pujol_rp_analysis.py`	2, 3	Reads in the catalog generated by `pujol_rp.py` and determines the responses and biases for every magnitude bin.
`plot_data.py`	3	Takes the analyzed data from `grid_analysis.py` and determines the biases by fitting the data for each magnitude- and each time bin.
`catalog_plot.py`	3	Reads in the analyzed data from `rp_analysis.py` and determines the biases by fitting the data for each magnitude bin and after each run. Therefore, the measured ellipticities of the first n runs are summarized and the uncertainty is determined by bootstrapping the first n runs.
`error_plot_grid.py`	4	Fits the uncertainties of the biases against the needed theoretical runtime to determine the runtime improvement of each method for each runtime bin.
`error_plot.py`	4	Does the same as `error_plot_grid.py` for the random positions. This is only a different script due to the slightly different data structure.
`plot_binned_data.py`	5	Takes the runtime improvements determined before and plots them against the magnitude bin used to display the binned runtime improvement.
`variable_shear_diff.py`	5	Takes the results from the analysis of the response with and without variable shear to compare the impact of it.
`bias_comparison.py`	5	Compares the absolute biases in each magnitude bin.
`functions.py`	Other	Contains the main part of the functionality with all the functions to bootstrap and generate simulations in general. This module is imported for most of the scripts.
`merge_catalogs.py`	Other	Useful script if you want to generate smaller simulations and merge the catalogs later on. Just give the paths to both catalogs and either "lf", "rp" or "grid" depending which kind of catalog you are trying to merge (different data structure).
`modify_config.py`	Other	Used to modify the config file with the inputs from the bash pipeline.

Table 4: All the provided scripts and a brief description of their tasks