TU thumbnail generation library

Outline

- How do we currently create galaxy thumbnails?
- TU thumbnail generation library
- Learning about GalSim

Input catalogue

- Two components, a bulge and a disk
- Two populations, the Bulge- (~15%) and the Disk-dominated galaxies, using (g-r) in restframe and Mr
- disk_length = 0 for bulge-dominated galaxies. For disk-dominated galaxies, the disk_length is the major-axis exponential scale length in arcsec
- The bulge_length is the major-axis half-light radius in arcsec
- http://arxiv.org/pdf/1210.8201v1.pdf

Simulation Release 1.C

- python code using GalSim, run at SDC-ES orchestrated by BT2
- Input: MICECATv2.0 galaxy catalog
  - area: ra [70.5, 73.0] , dec [-51.0, -48.0]
  - 236517 galaxies (sdss_r < 24.5)
- Output (which is part of the TU input):
  - 12 GALCAT fits files with galaxy catalog information (ra, dec, shear, shape, etc..) (1 catalog per square degree)
  - 3967 THUMBNAIL fits files with 200 galaxies per file (total size 588GB)
  - 1 file including the output file list and its metadata (EAS bypass)
- total time: 3 galaxies/min

Python code

```
# Galaxy object
disk = galsim.Sersic(self._SERSIC_DISK, scale_radius=params['disk_length'])
bulge = galsim.Sersic(self._SERSIC_BULGE, half_light_radius=params['bulge_length'])
gal = params['bulge_fraction'] * bulge + (1 - params['bulge_fraction']) * disk

# Galaxy shape
gal_shape=galsim.Shear(q=
params['disk_axis_ratio'],beta=params['disk_angle']*galsim.degrees)
gal = gal.shear(gal_shape)

# Galaxy shear
mu = 1./((1. - params['kappa']**2) - (params['gamma1']**2 + params['gamma2']**2))
reshear1 = params['gamma1'] / (1. - params['kappa'])
reshear2 = params['gamma2'] / (1. - params['kappa'])
gal = gal.lens(reshear1,reshear2,mu)

# Flux normalization
gal = gal.withFlux(self._TOTAL_FLUX)

# Creating thumbnail
base_image =
galsim.ImageF(params['nx_pixel'],params['ny_pixel'],scale=params['pixel_scale'],init_value=0)
gal_image = gal.drawImage(image = base_image, method = params['method'])
image = gal.draw(image = base_image)
```

GalSim parameters

- Galaxy input parameters:
    - shape parameters: disk_length, bulge_length, bulge_fraction, disk_axis_ratio, disk_angle
    - shear parameters: kappa, gamma1, gamma2
- Image parameters:
    - nx_pixel, ny_pixel (we assume nx_pixel = ny_pixel, max. size = 500)
    - pixel_scale (is currently fixed to 0.01 arcsec)
    - render method (‘no_pixel’ or ‘real_space’) depending on the galaxy size to optimize the total flux in the image

TU thumbnail library

- Generated in advance, stored and loaded by the simulators when needed

- Created on-demand by the simulators

In advance (currently)

- Large files
- How to locate the thumbnails?
- Storage requirements
- I/O overhead (network / disk)
- Latency (cannot simulate only one image)
- All simulators use the same thumbnail files (e.g. same pixel resolution)

GalSim in c++?

- GalSim is a free software to simulate pixel images. The core is written in c++ and there is a python API to use it

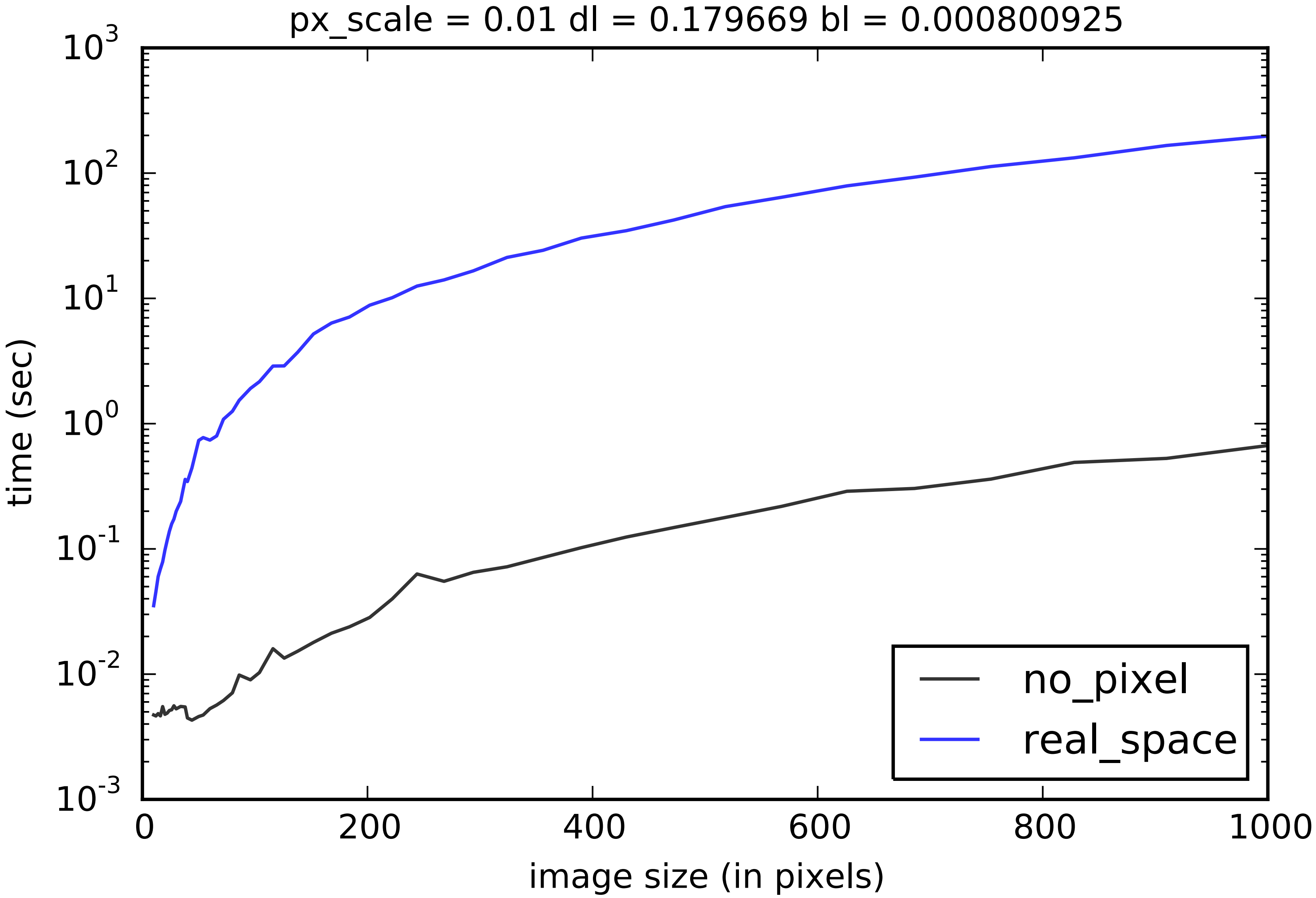
- Authors propose two ways of using it:

  - API in python (what we are using)

  - python script with config files

Time

- Time to create each galaxy thumbnail does not depend on the galaxy type or size but on the image size and the render method



Learning about GalSim

- We are in the process of exploring how the flux in the image varies as a function of the render method (‘no_pixel’ or ‘real_space’), the pixel scale and the image size for different galaxy types and sizes