

PDD-Sim MANUAL

Version 1.0

Developed by the Research Group AEGORA

Date: May, 2021

THIS PAGE IS INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES:.....	5
LIST OF TABLES:	5
RELEVANT DOCUMENTS.....	7
LIST OF ACRONYMES.....	8
Preliminary:	9
1. General Description	10
1.1 DATA Input	12
1.2 ACCUMULATION mode	12
1.3. TIME-TAG mode	13
2. Getting started with PDD-Sim	13
2.1 System requirements and compatible systems.....	13
2.2 Installation.....	13
2.3 Running the application	14
2.4 Main screen	15
3. Data input and Mode selection in PDD-Sim	16
4. Settings the parameters of the detector	18
4.1 Setting the parameters of the MCP	19
4.2 Setting the parameters of the Phosphor converter.....	20
4.3 Setting the parameters of the Fiber tapping	20
4.4 Setting the parameters of the CMOS detector	21
4.5 Setting the random seed	23
5. Setting the algorithm for the calculation of the centroids.....	23
6. Data output from PDD-Sim.....	24
7. Working with PDD-Sim.....	26

THIS PAGE IS INTENTIONALLY LEFT BLANK

LIST OF FIGURES:

Figura 1: Baseline configuration of the PDD simulated by PDD-Sim. 10
Figura 2: PDD-Sim operational layout in the two modes: Accum and Photon Counting. 11

LIST OF TABLES:

Tabla 1: Required python modules..... 14

THIS PAGE IS INTENTIONALLY LEFT BLANK

RELEVANT DOCUMENTS

WSO-FCU-TN-0001: Centroid Calculation Baseline for the Microchannel Plate (MCP) detector in the Far Ultraviolet (FUV) channel of the Field Camera Unit (FCU).

Ana I Gómez de Castro, Ivan Prada, Javier Yáñez
May 2020.

The World Space Observatory: ultraviolet mission: science program and status report

M. Sachkov, A. I. Gómez de Castro, B. Shustov,
Proc. SPIE 11444, Space Telescopes and Instrumentation 2020: Ultraviolet to Gamma Ray,
1144473 (13 December 2020); doi: 10.1117/12.2562929

Field camera unit of the WSO-UV mission

M. Sachkov, S. Sichevsky, B. Shustov, A. Shugarov, V. Shmagin, S. Iosipenko, R. Arkhangelsky, E. Kanev, A. I. Gomez de Castro, J. C. Vallejo
Proc. SPIE 11444, Space Telescopes and Instrumentation 2020: Ultraviolet to Gamma Ray,
1144474 (13 December 2020); doi: 10.1117/12.2562962

The detector for the far ultraviolet channel of the imaging instrument (FCU) on board the Spectrum-UV (WSO-UV) space telescope

Ana I. Gómez de Castro, Laura Díez, Javier Yáñez, Iván Prada, Alejandra Araujo, Alejandra Salinas, Xavi Llamas, Mikhail Sachkov, Andrey Shugarov,
Proc. SPIE 11444, Space Telescopes and Instrumentation 2020: Ultraviolet to Gamma Ray,
1144461 (13 December 2020); doi: 10.1117/12.2560330

MCPSim-Py: an open source python-based simulator of the performance of MCP photon-counting detectors

Iván Prada, Javier Yáñez, Ana I. Gómez de Castro
Proc. SPIE 11444, Space Telescopes and Instrumentation 2020: Ultraviolet to Gamma Ray,
114449B (13 December 2020); doi: 10.1117/12.2576256

LIST OF ACRONYMES

CMOS	Complementary metal–oxide–semiconductor
FCU	Field Camera Unit
FITS	Flexible Image Transport System
FPGA	Field-programmable gate array
FUV	Far ultraviolet
FWHM	Full width at half maximum
GUI	Graphical user interface
MCP	Micro-channel plate
PDD	Photon Detection Device
PSF	Point spread function
QE	Quantum efficiency
ROI	Region of interest
UV	Ultraviolet
WSO-UV	World Space Observatory Ultraviolet

Preliminary:

This manual describes the open-source, python coded simulator **PDD-Sim** to evaluate the performance of the Photon Detection Device to be implemented in the World Space Observatory- Ultraviolet far ultraviolet camera. The application is also designed to enable testing of user designed algorithms for event detection and centroiding for all types of sources and possible photon counting regimes. Unlike other tools, the MCP transfer function can be determined empirically from a set of images obtained from the real detector. Also, it is feasible to characterize the MCP transfer function with a suit of parameters.

1. General Description

The Photon Detection Device (PDD) simulated by the application PDD-Sim has a rather standard architecture and it is often used for the detection of high energy photons from astronomical sources. These photons are scarce and the inclusion of an amplification phase (a multichannel plate or MCP) is required. The detection principle is based on the photoelectric effect: photon absorption by the atoms of the substrate generate electrons that are amplified to produce a significant electron current. The electron current is transformed into an optical signal and then, detected digitally by a CMOS detector as displayed in Figure 1.

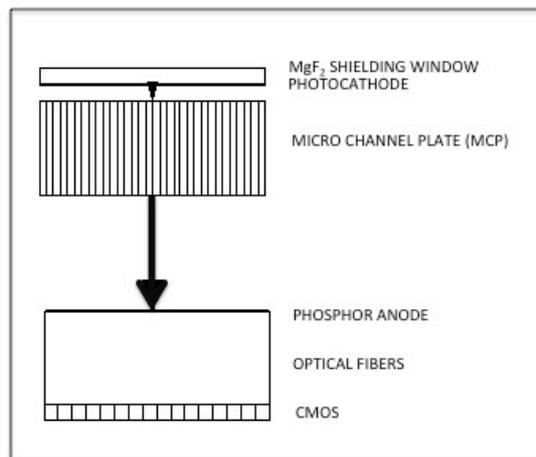


Figure 1: Baseline configuration of the PDD simulated by the application PDD-Sim. The Radiation from the source is transformed in an electron flow at the photocathode, amplified by the MCP and actually detected by a conventional optical detector (CMOS is the election of choice for WSO/ISSIS) after conversion of the electronic signal into optical signal in a Phosphor anode. The Phosphor anode and the CMOS detector are coupled by optical fibers.

The PDD-Sim is a tool designed with tunable parameters that enable the simulation of a broad variety of MCP detectors including:

- Pore section and pitch size in the MCP.
- MCP gain.
- Eccentricity of the electron shower leaving the pores.
- Electron-to-photon conversion factor at the Phosphor anode.
- Optical fiber section at the phosphor anode.
- Demagnification factor of the fibers bundle from the phosphor anode to the CMOS.
- Energy loss (photon loss) within the fiber.
- Energy loss (photon loss) caused by the geometrical coverage of the phosphor anode by the fiber bundle.

Default values are set to the design values of the MCP detector developed for the far ultraviolet channel of the instrument Field Camera Unit (FCU) on board the World Space Observatory-Ultraviolet (WSO-UV).

The simulator has two operation modes (as the operational MCP): *Accum* or *Time-Tag*.

In *Accum* mode, all the events are accumulated over the given time to produce an image that can be downloaded in various formats (fits, png, jpg). Centroiding of the events can be run “*a posteriori*”.

In *Time-Tag* mode, the events are read at the pace set in the input parameters for the CMOS read-out time. The centroid of the events is computed and a list of events is produced in various formats (csv, ascii file...). Several centroiding algorithms are available: 3-cross, 3-square, 5-cross, 5-square, 7-cross and 7-square. An output digital image built from the list of events is also produced.

The overall operational layout is shown in Figure 2.

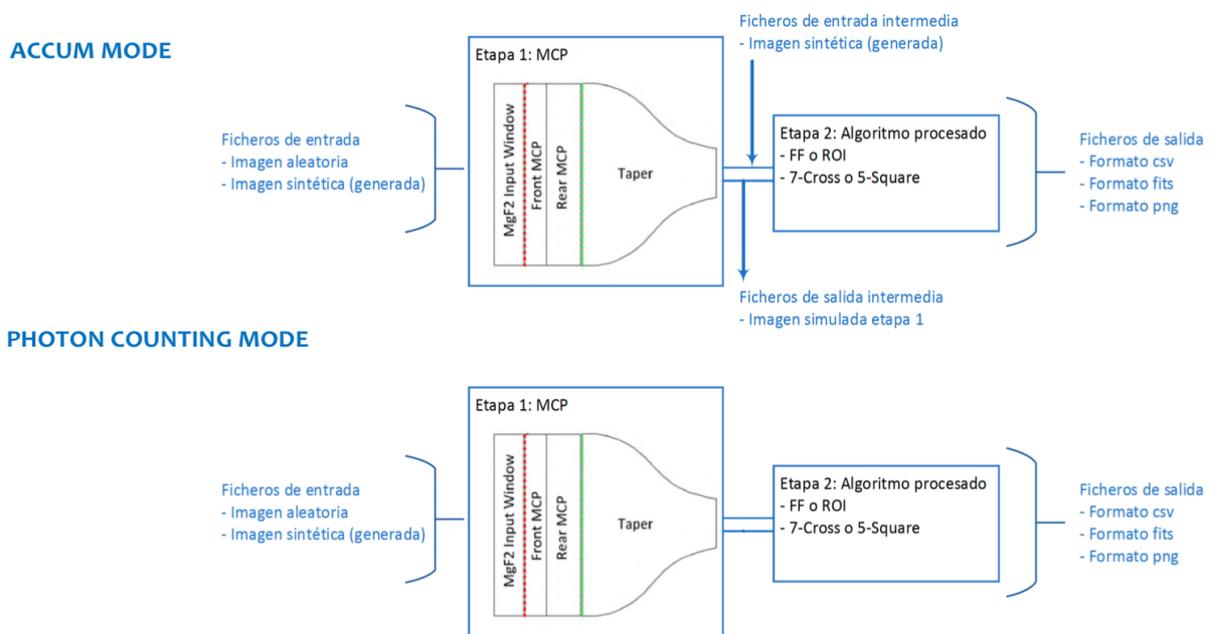


Figura 2: PDD-Sim operational layout in the two modes: Accum and Photon Counting.

1.1 DATA Input

The simulator admits three types of inputs: uniform sky, double source and image.

Uniform sky generates photons over a given area following a uniform distribution. The number of photons to be simulated is provided by the user. Large numbers of photons (more than 1,000) require a long computational time.

Double source generates two sources at predefined distances with predefined relative intensities. The sources are assumed to have a Gaussian Point Spread Function (PSF) and the Full Width Half Maximum (FWHM) is defined by the user.

The image can be provided in JPG or FITS format. The size of the image is scaled according to the plate scale provided by the user (in arcsec per micron) and the MCP design parameters specified in the simulator: number of microchannels and the pitch of the micro-channels in microns.

Example:

An astronomical image of 2048×2048 pixel² with pixel size 0.04 arcsec² and plate scale 0.05 arcsec/micron covers a field of view of 81.92×81.92 arcsec² in the sky and it is projected onto the detector on $1,638 \times 1,638$ micron² or 1.64×1.64 mm². This value is significantly smaller than the size of the MCP (for WSO/FCU/FUV MCP detector the size is a circle of 40 mm diameter).

In the current implementation, the image is re-scaled to cover the MCP at full, i.e., the pixel is assumed to have a pixel of size 0.975 arcsec² ($= 0.04$ arcsec² \times 40 mm / 1.64 mm) unless, otherwise declared in the field: "*Field of View*" of the Input Data Window.

If the fields: "*Field of View*" and "*Center of Coordinates*" are provided in the Input Data Window, PDD-Sim does not make any re-scaling and just select the section of the image specified by the user using the "*Pixel size*" and the "*Plate Scale*" provided by the user.

Also, the number of cosmic ray events can be introduced. They are assumed to have linear trajectories with random orientation on the detector.

1.2 ACCUMULATION mode

In accumulation mode, all the simulated photons reach the detector simultaneously. An intermediate image with the imprint of all the events on the CMOS can be obtained as well as a list of all the events (X, Y coordinates on the detector of the centroids). Both processes are decoupled and a set of 6 algorithms (3-cross, 3-square, 5-cross, 5-square, 7-cross, 7-square) is provided for the determination of the centroids. The statistical treatment of the events made by the simulator can be found in the JCUVA internal document: WSO-FCU-SOC-TN-0001.

1.3. TIME-TAG mode

In time-tag mode, the photons are generated following a Poisson distribution in time. The read-out velocity of the detector can be accommodated as well as the size of the input (and read out) window. Several outputs are produced namely:

- The list of events (time and X, Y coordinates).
- The final image obtained after co-adding all the events.
- The frames resulting after each read-out either as a set of images or as a movie.

2. Getting started with PDD-Sim

2.1 System requirements and compatible systems

PDD-Sim has been developed using Python language. This means that the base compatibility of the system is strictly linked to the compatibility of Python. This application will run on the platforms listed below (in x64 architecture):

- Windows 7 and later systems.
- Mac users: macOS 10.9 (Mavericks) and later systems.
- Ubuntu 18.04 or similar (and later systems).

Minimum required hardware:

- Minimum of 8 GB of RAM.
- Minimum of 1 GB of disk space.
- Minimum an Intel Core i3 or similar processor.

Recommended hardware:

- 16 GB of RAM.
- 10 GB of disk space.
- Intel Core i5 or similar processor.

2.2 Installation

This software is based on python. To be able to execute the program, you must have installed the Python 3.8.7 version and the next specific python packages. Use the following steps to ensure that installation can be achieved correctly.

- 1) Download and install python 3.8.7 in your system.

In windows and mac, you need download this software in the next page:

<https://www.python.org/downloads/release/python-387/>

For Ubuntu or other GNU/Linux distros, you must execute the command:

sudo apt install python3.8.7

- 2) You must have installed the following Python packages (available in pip or conda package installers). It leaves the choice of newer versions of these packages to the user, but the safe functioning of the program cannot be guaranteed. For this example, we will use pip package installer. To install pip, you should run the following command:

python get-pip.py

Once installed, we were to continue installing the packages specified in Table 1 with this command:

pip install package name==version

Example:

pip install numpy==1.20.0

Tabla 1: Required python modules

Package	Version	URL
numpy	1.20.0	https://numpy.org/
matplotlib	3.3.4	https://matplotlib.org/stable/index.html
scikit-image	0.18.1	https://scikit-image.org/
astropy	4.2	https://www.astropy.org/
sklearn	0.24.1	https://scikit-learn.org/stable/
seaborn	0.11.1	https://seaborn.pydata.org/
numba	0.52.0	https://numba.pydata.org/
opencv-python	4.5.1.48	https://opencv.org/
joblib	1.0.0	https://joblib.readthedocs.io/en/latest/
scipy	1.6.0	https://www.scipy.org/

2.3 Running the application

To use the application for the first time, you need access to the previous download folder, (that must be unzipped previously) and execute the following command:

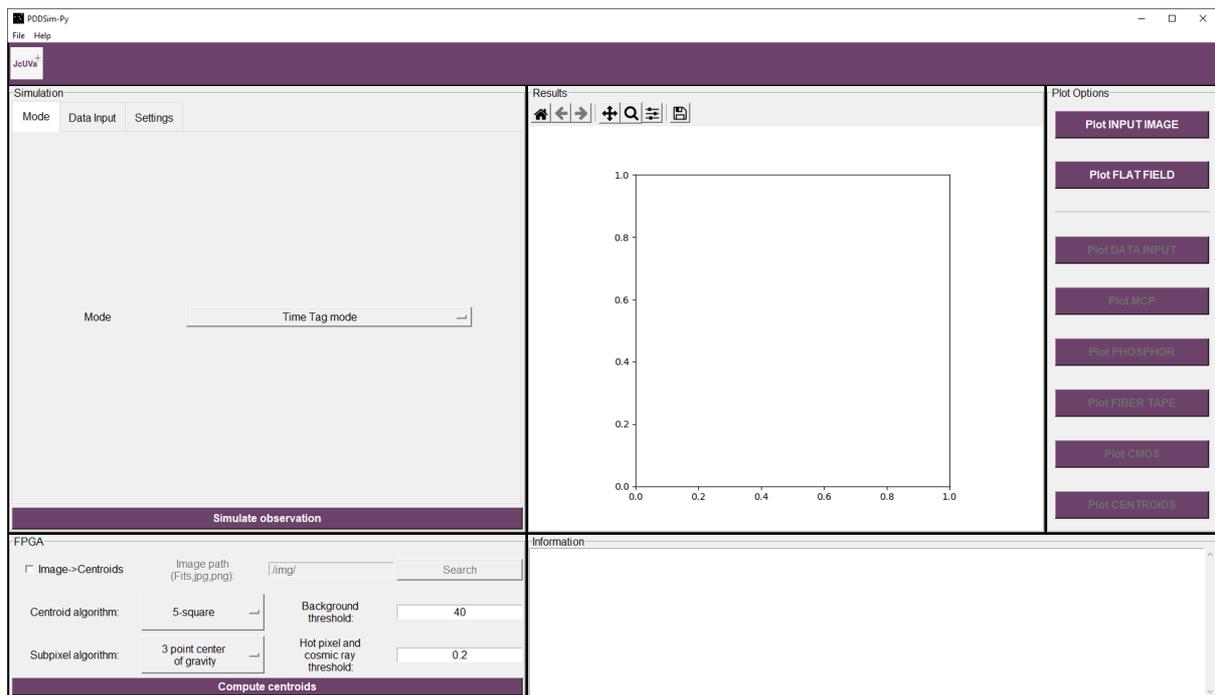
python3 main.py

This command runs the MCPSim-Py program in the foreground. Press **File->Exit** menu bar option or **X top right button** to close the application.

To find out which version of MCPSim-Py you are using, click **Help->About** menu bar option.

2.4 Main screen

When you run the program, a main GUI appear in the screen.



The screen is divided into four sections. The left ones are for input data and the right ones to display the output of the simulations. Input data, settings and MCP operation mode are provided through the interface in the upper left window. The algorithm to be used for the centroiding and its properties are set through the interface at the lower left window. On the right-hand side, the output data are visualized in the upper right window. The bottom right window just provides some statistics on the computational load for the simulator (number of photons simulated, number of secondary electrons simulated...).

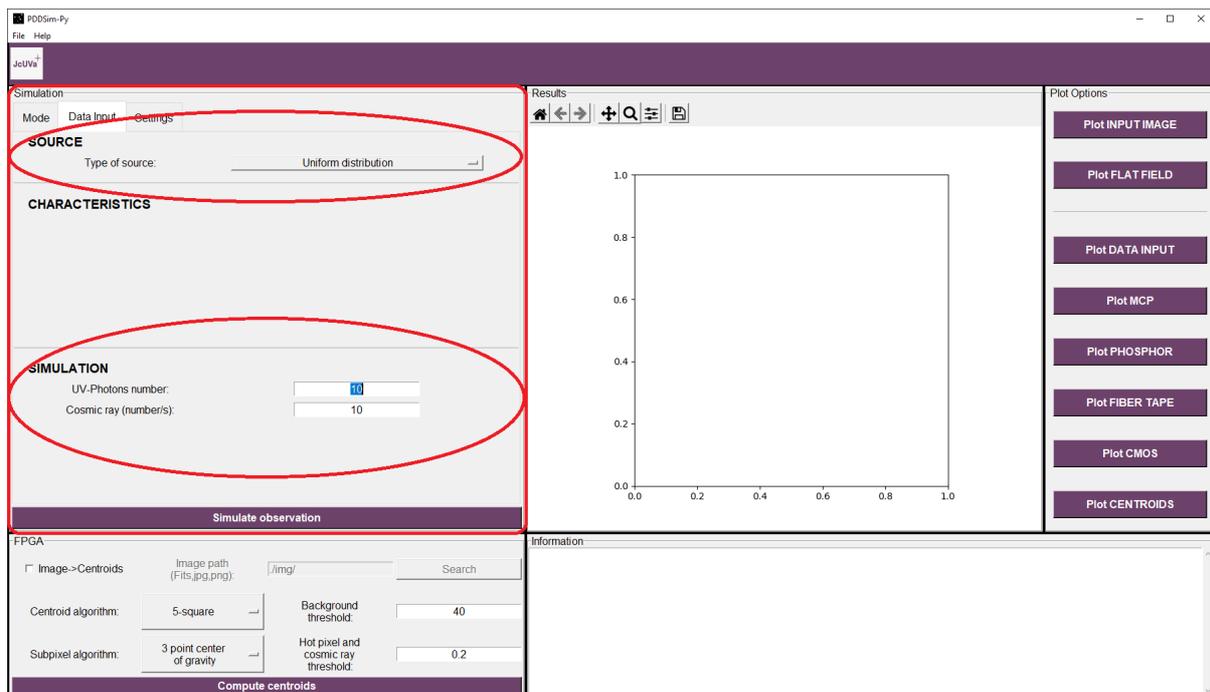
There are two action buttons to simulate an observation and compute the centroids with different algorithms.

3. Data input and Mode selection in PDD-Sim

The application simulates the two operating modes of the MCP detector: Accumulation (Accum) and Time tag. Data input is the same for both modes though in Time-Tag more additional parameters need to be provided on the MCP detector concerning read-out velocity. These parameters are described below.

Three different data inputs are enabled in the simulator:

Uniform data input: the number of photons simulated (provided by the user) are distributed over the detector entry window following a uniform distribution. A number of cosmic ray hits can be provided by the user.



Double source: the number of photons simulated (provided by the user) are distributed in two point like sources, each following a Gaussian point spread function. The user provides the FWHM of the Gaussian, the distance between the two sources and the number of photons coming from each one of them. A number of cosmic ray hits can be provided by the user.

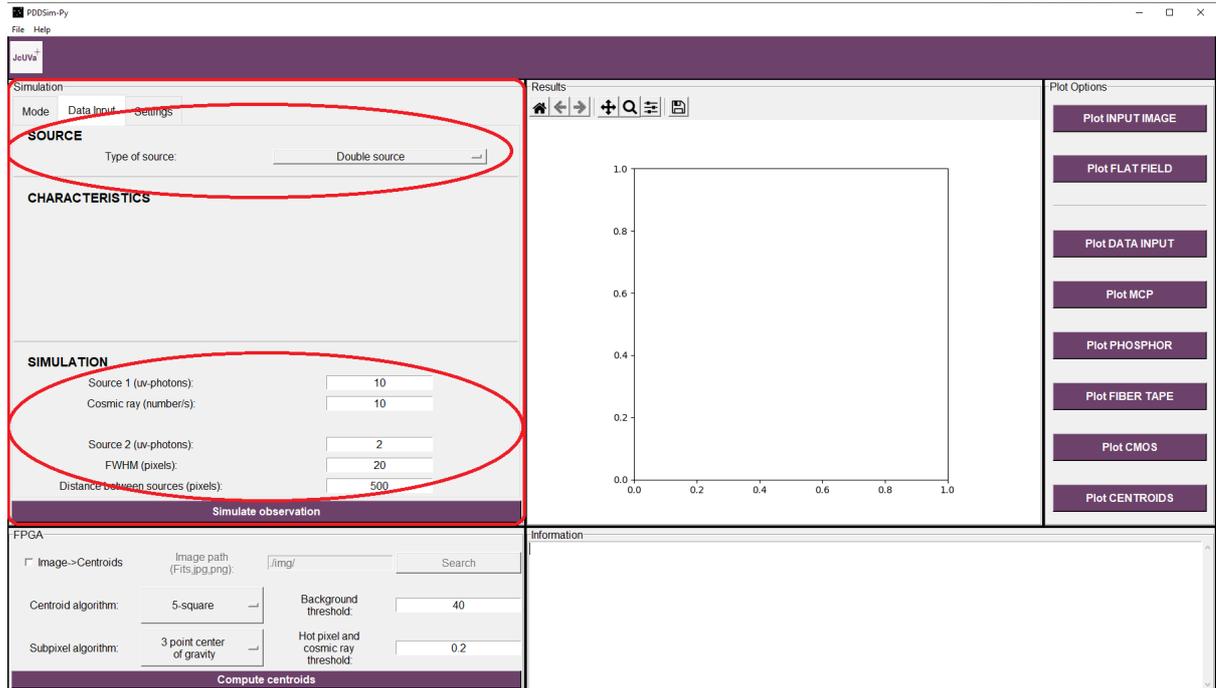


Image: the number of photons simulated (provided by the user) are distributed following the probability distribution of the gray scale image provided by the user. If a color image is provided, the simulator transforms it in a B&W image using the conversion *ITU-R 601-2 luma transform*¹. Basic scaling parameters need to be provided to fit the image into the physical size of the detector (28.3 mm x 28.3 mm)². As, it is not expected that any astronomical image adapts perfectly well to the characteristics of the MCP (in terms of size and number of pixels) PDDSim provides some basic tools to:

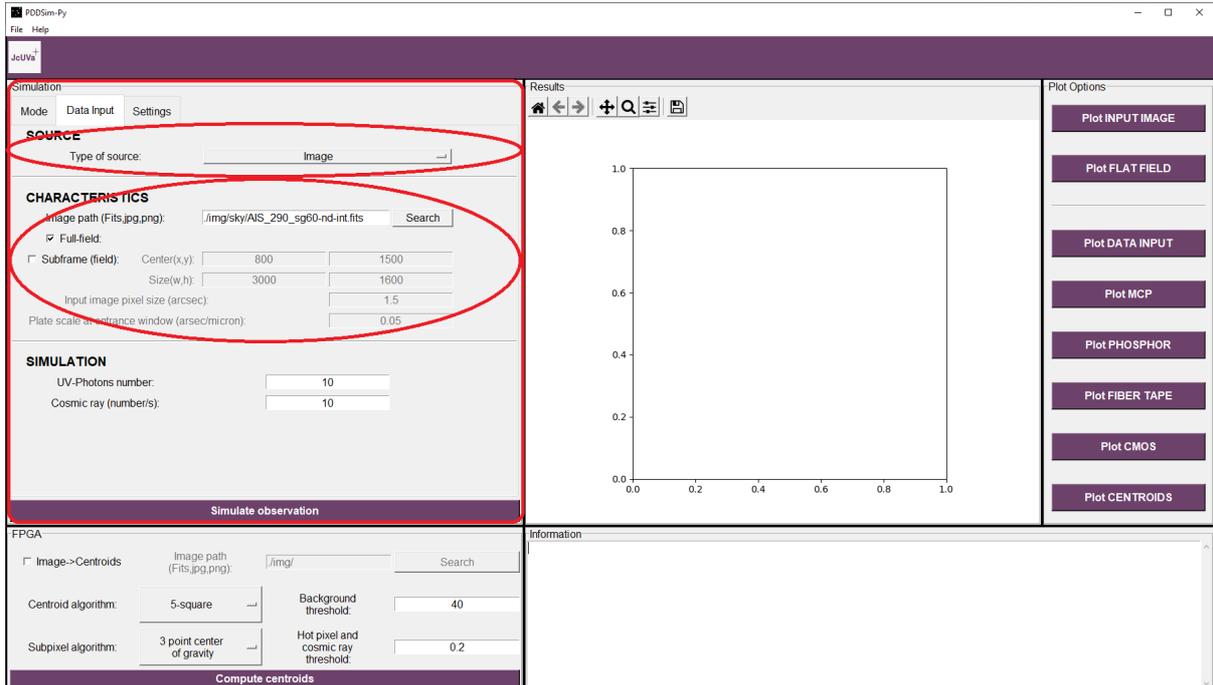
- [1] Scale the image to the size of the detector by selecting the full frame option; if the image is smaller than the field, it is enlarged and if bigger, it is reduced. The scale algorithm uses cubic interpolation³.
- [2] To select region of the image. In this case, the central pixel coordinates (X,Y) and the dimensions of the region need to be provided.

A number of cosmic ray hits can be provided by the user.

¹ <https://pillow.readthedocs.io/en/4.2.x/reference/Image.html#PIL.Image.Image.convert>

² The end detector is a 2048x2048 pix² square CMOS that fits within the circular field (diameter 40 mm) of the MCP detector.

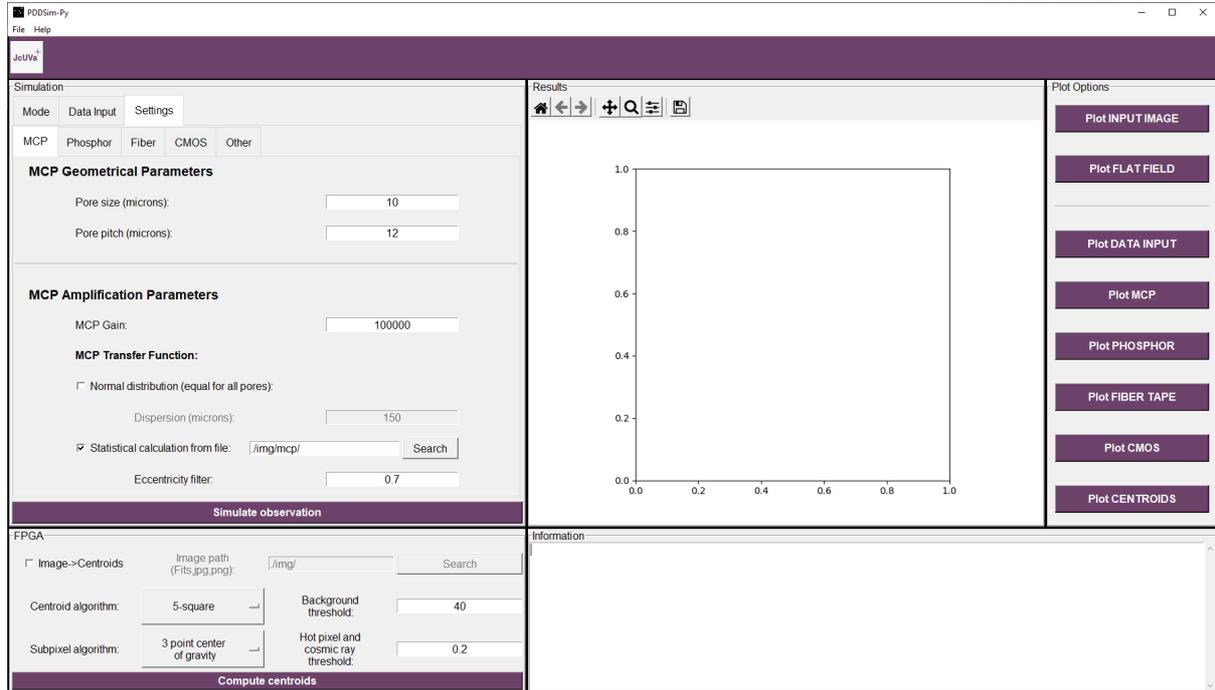
³ <https://pillow.readthedocs.io/en/stable/handbook/concepts.html#PIL.Image.BICUBIC>



4. Settings the parameters of the detector

The main parameters of the detector can be modified by the user though default values are provided for the PDD on the FCU/FUV channel; be careful because some of them maybe modified/updated during the tests and calibration campaigns.

4.1 Setting the parameters of the MCP



The parameters to be set are:

- *Pore size* (in microns).

Pore size (microns):

- *Pore pitch*: sets the effective projected diameter of the pore (in microns).

Pore pitch (microns):

- *MCP Gain*: defines the amplification factor of the photoelectrons in the microtubes.

MCP Gain:

The transfer function defines the shape and properties of the output electron showers from the MCP and can be introduced in two differenced ways:

- *Normal distribution*: the electrons are randomly generated following a bidimensional Gaussian distribution (uncorrelated normal distribution) with fix dispersion and intensity for all microtubes and events.

- *Dispersion*: set the normal deviate of the showers.

Normal distribution (equal for all pores):

Dispersion (microns):

- *Statistical calculation from file*: this mode allows the user to introduce an image of any MCP transfer function for simulating this effect. This image is processed by the simulator to obtain the statistical data that will be used to define the MCP transfer function.
 - *Statistical calculation from file*: images of any MCP transfer function (*.fits format are available).

Statistical calculation from file:

- *Eccentricity filter*: the electron showers are filtered by the eccentricity filter value.

Eccentricity filter:

4.2 Setting the parameters of the Phosphor converter

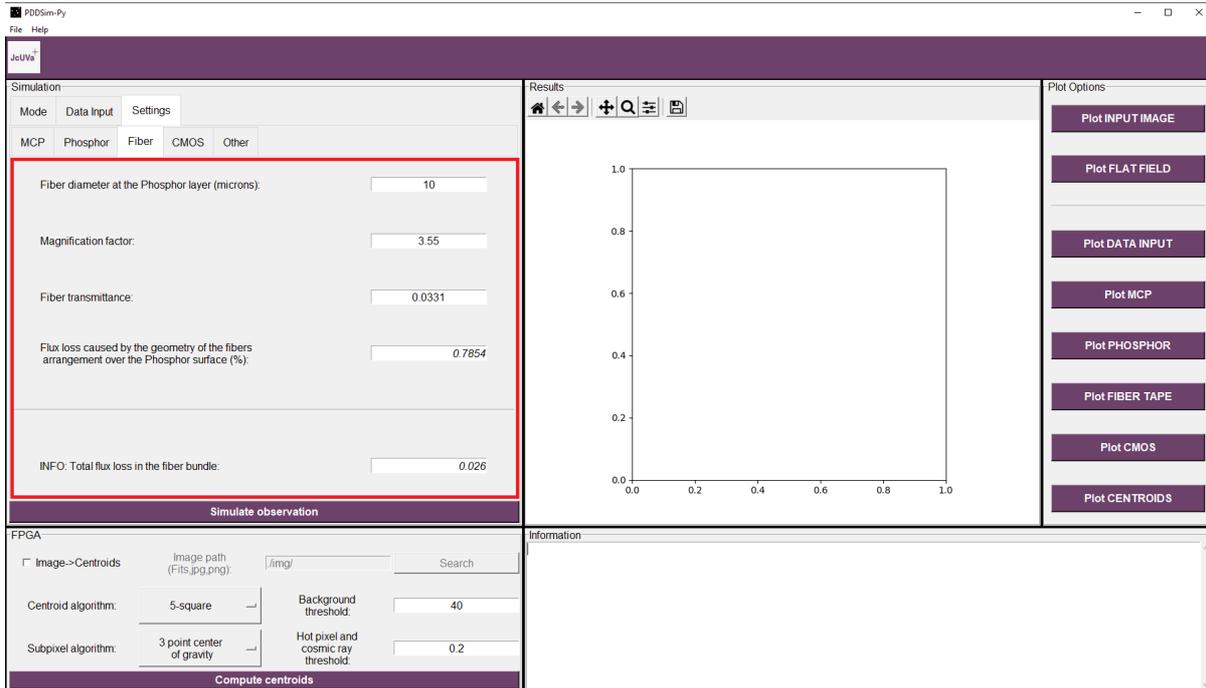
Anyone of these electron pulses is transformed into a photon pulse at the phosphor interface.

- *Phosphor Gain*: The gain of this step is an input parameter (Phosphor Gain).

Phosphor Gain:

4.3 Setting the parameters of the Fiber tapping

The output shower of optical photons produced by the phosphor layer is captured by the optical fibers of the fiber taper. This tab is set to configure the fiber tape design parameters.

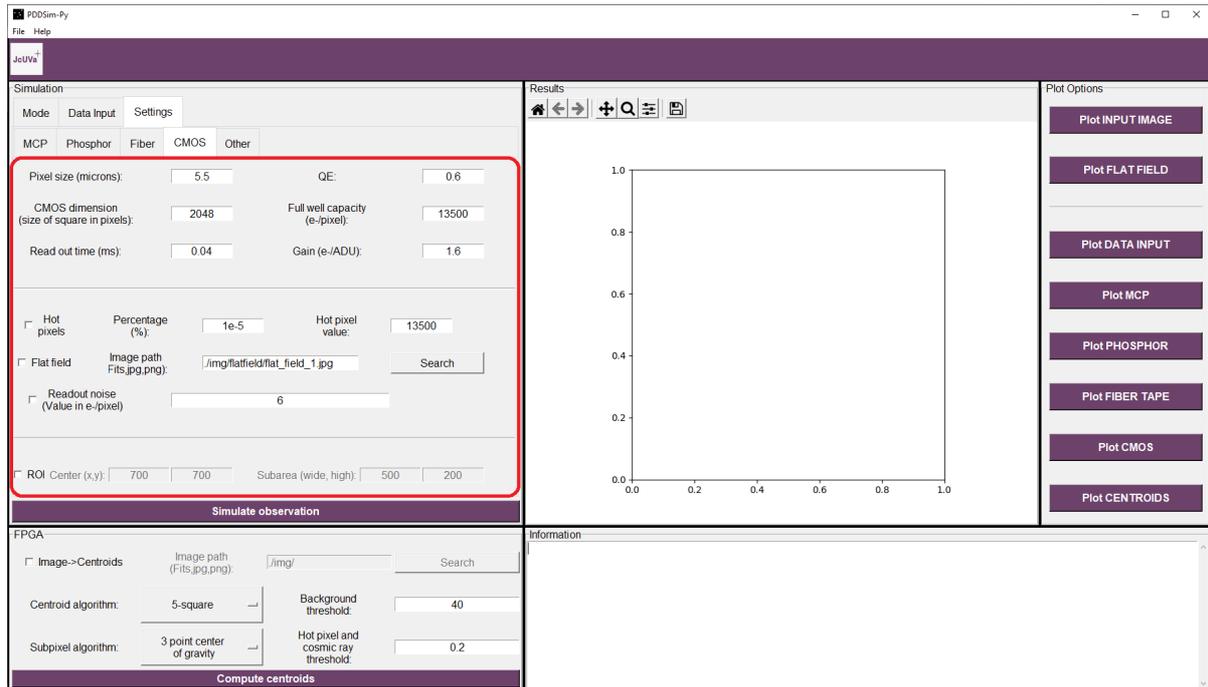


The following parameters need to be set:

- *Fiber diameter at the Phosphor layer*: this parameter sets the size of the fiber picking up the optical photons from the P46 substrate and carrying them into the detector.
- *Magnification Factor*: This is a geometric factor that measures the ratio between the projected area of the bundle in the P46 layer and the projected area on the CMOS surface. The default ratio is 3.55 based on the FCU/FUV MCP detector architecture but can be modified for any other design.
- *Fiber transmittance*: that accounts for the photon losses in the fiber.
- *Flux loss caused by the geometry*: the bundle of optical fibers does not cover perfectly well the P46 surface; even in the optimal case, the fibers have circular sections and a plane cannot be completely covered with circles. The optimal value for an hexagonal arrangement of the fibers is 0.7854, which is the default value.

4.4 Setting the parameters of the CMOS detector

The CMOS is a square grid, where the number of pixels depends on the window size and the pixel size. This tab is set to configure the fiber tape design parameters.



The following CMOS parameters can be set:

- *Pixel size*: the size of the pixel (in microns).
- *CMOS dimension*: the size of the CMOS in pixels.
- *Read-out time* (in ms).
- *Quantum efficiency*
- *Full well capacity (e-/pixel)*
- *Gain of CMOS (e-/ADU)*

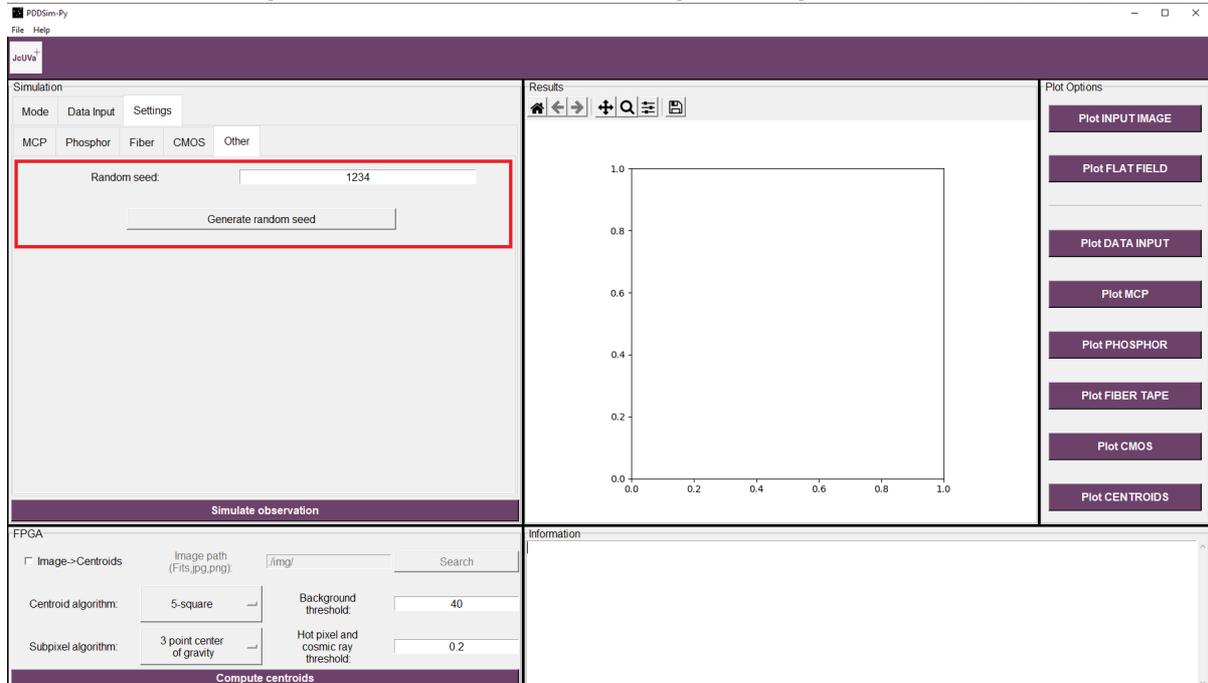
Some additional data on the CMOS performance can be added:

- *Hot pixels*: percentage of the pixels which look much brighter than the rest of the CMOS pixels. You can establish the percent and the value of these pixels.
- *Flat field*: An image can be upload with the response (flat field) of the detector. The image can be uploaded in various format (*.fits, *.png and *.jpg formats are available).
- *Readout noise*: readout noise is the number of electrons that are introduced in the signal due to readout electronics per pixel.

CMOS detectors are configurable and can be read totally or partially. It is also possible to configure a Region of Interest (ROI) to read the data only in this area. Note that this selection only affects the final displays but not the computational time. To select a small region of an image is computationally-wise to do it directly in the data input.

4.5 Setting the random seed

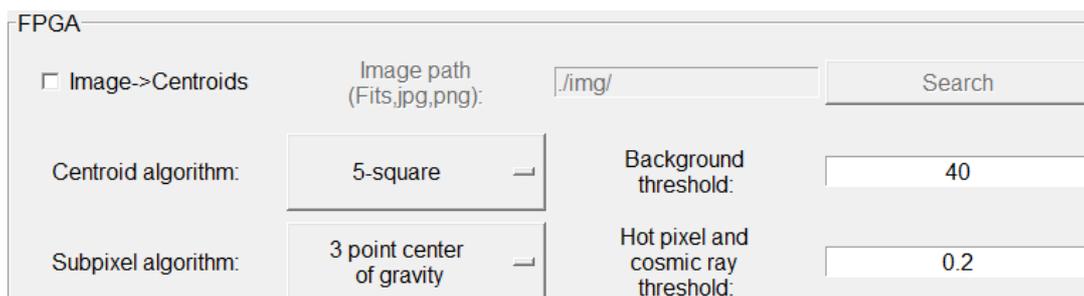
Underlying the simulator, there are many statistical calculations that require a random seed. It is feasible to change the random under the Settings, clicking in Others.



5. Setting the algorithm for the calculation of the centroids

A broad range of centroid algorithms have been implemented (*Centroid algorithm*), as well as subpixel algorithms (*Subpixel algorithm*). The centroid algorithms are based on a threshold value (*Background threshold*). Hot pixel and cosmic ray detection have been implemented. This subsystem is based on sudden changes in the number of counts between a pixel and their boundary (*Hot pixel and cosmic ray threshold*, in [0,1] range).

The *Image->Centroids* check allows to launch the centroid algorithms on an already stored image. Its only necessary to activate the check option, select the local image and press the *compute centroids* button.



6. Data output from PDD-Sim

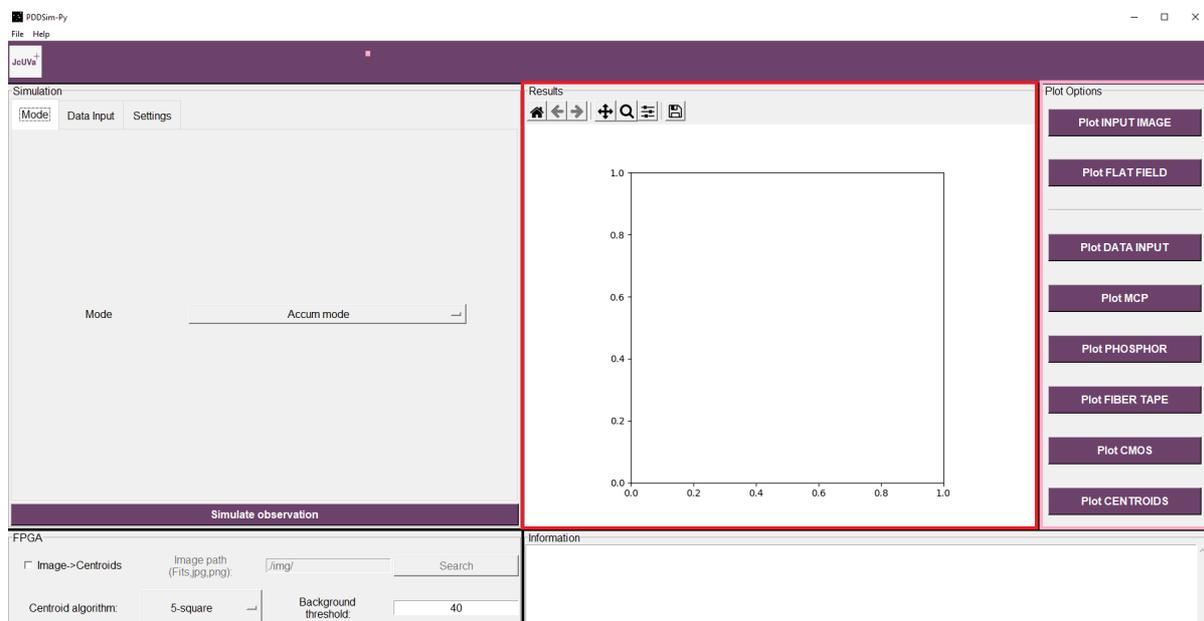
To display the results from the simulation (intermediate results and final) in accumulation mode, six buttons are included in the GUI:

- **Plot DATA INPUT:** shows the input window.
- **Plot MCP:** shows the clouds of photoelectrons generated in the MCP.
- **Plot PHOSPHOR:** shows the distribution of secondary electrons at the P46 layer.
- **Plot FIBER TAPER:** shows the distribution of electrons as captured by the Fiber taper.
- **Plot CMOS:** shows the final distribution of the photons on the square grid of the CMOS.
- **Plot CENTROIDS:** the calculated centroids are displayed for analysis purposes.

The previous buttons are available in the right-side toolbar of the GUI.

In addition, and for all modes, the simulator allows to display the next input information images:

- **Plot INPUT IMAGE:** shows the distribution input image in the case that it is enabled.
- **Plot FLAT FIELD:** shows the flat field image in the case that it is enabled.



For time tag mode, four output files will be generated in the `./img/output` folder:

- An animation, with which it observes the centroid detection of the arrival uv-photons (`./img/output/Video.avi`).

- An image with the number of centroids detected in each pixel, which is generated in *.fits format (*./img/output/Image.fits*).
- A *.csv file with the coordinates (X,Y) of computed centroid for each event centroids (*./img/output/centroid.csv*).
- An image in the *.png format which allows you to easily view the previous image (*./img/output/Image.png*).

For accum mode, two output files will be generated in the *./img/output* folder:

- An image with the number of centroids detected in each pixel, which is generated in *.fits format (*./img/output/Image.fits*).
- A *.csv file with the coordinates (X,Y) of computed centroid for each event centroids (*./img/output/centroid.csv*).

Every time that the simulation is started, the *./img/output* folder will be emptied.

Consequently, we recommend that you save the elements contained in this folder if you find it useful.

7. Working with PDD-Sim

This section provides an execution example behavior for each mode to allow the user to start using the program.

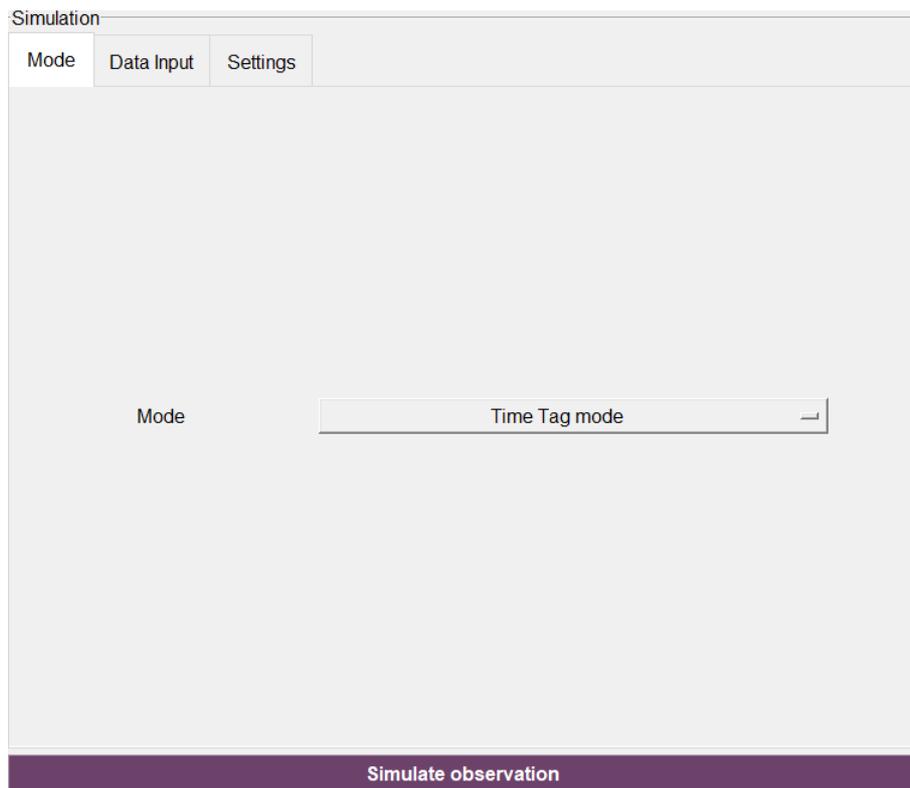
Example 1 (Time tag simulation)

We are going to simulate the behavior of the MCP detector over a specific period of time (*exposition time*). In addition, we will add *cosmic rays*. In order to do this action, we will continue the following checklist:

1. Open MCPSim-Py.

```
\centroidespy>python3 main.py
```

2. Select *Time tag* mode.



3. In data input tab, select *Image* as *Entry distribution sky* and select *AIS_290_sg60-nd-int.fits* image (set as default). Establishing *cosmic ray* to 10.

Simulation

Mode | Data Input | Settings

SOURCE

Type of source:

CHARACTERISTICS

Image path (Fits,jpg,png):

Input image pixel size (arcsec):

Plate scale at entrance window (arsec/micron):

SIMULATION

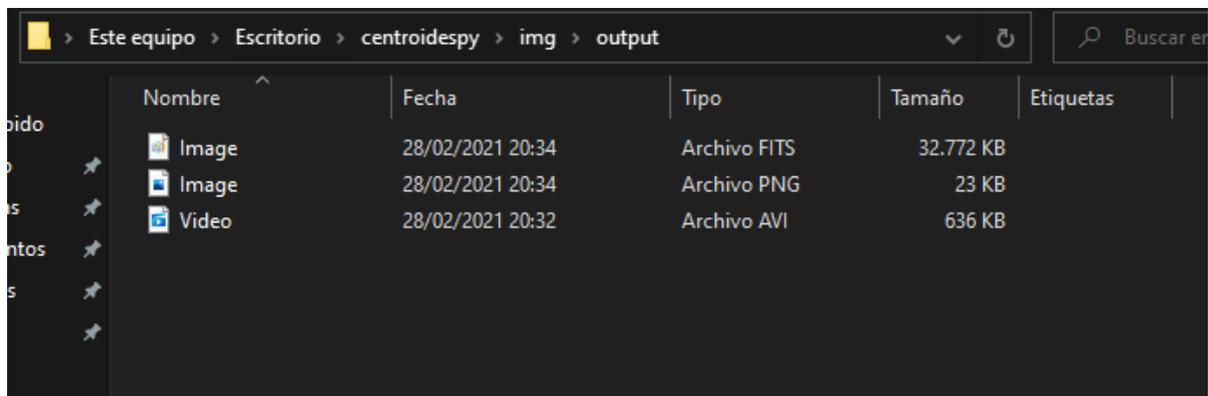
Cosmic ray (number/s):

Exposure time (s):

4. Leave the rest of the parameters as their default value and choose the *5-square centroid algorithm*.
5. Click simulation button. Press Q key to stop the time tag simulation.

Simulate observation

6. Once the test is completed, go to the `./img/output` folder to see the results.



File Explorer path: Este equipo > Escritorio > centroidespy > img > output

Nombre	Fecha	Tipo	Tamaño	Etiquetas
Image	28/02/2021 20:34	Archivo FITS	32.772 KB	
Image	28/02/2021 20:34	Archivo PNG	23 KB	
Video	28/02/2021 20:32	Archivo AVI	636 KB	

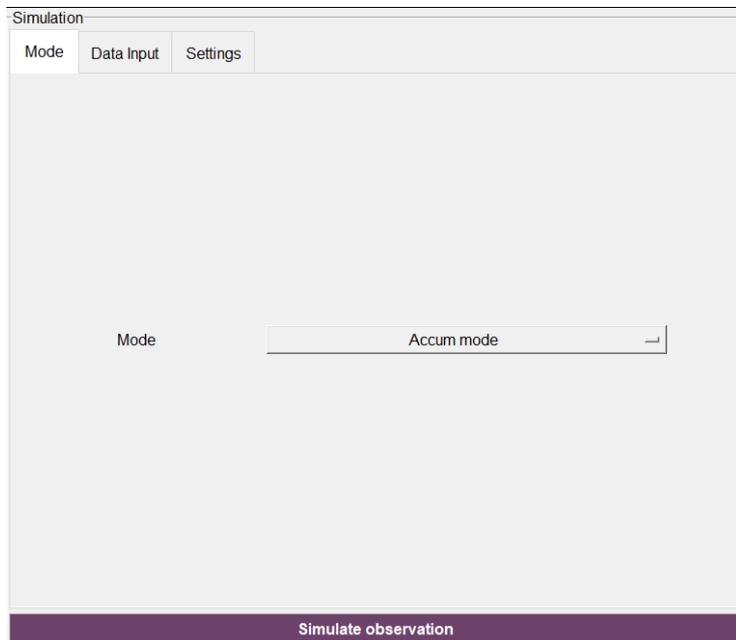
Example 2 (Accumulation mode simulation)

We are going to simulate the behavior of the MCP detector in accumulation mode. In addition, we will add cosmic rays. In order to do this action, we will continue the following checklist:

1. Open MCPSim-Py.

```
\centroidespy>python3 main.py
```

2. Select *Accum mode*.



3. In data input tab, select *Uniform* as *Entry distribution sky* and set to 100 the *UV-photon number*. Establishing *cosmic ray* to 100.

Simulation

Mode Data Input Settings

SOURCE

Type of source:

CHARACTERISTICS

SIMULATION

UV-Photons number:

Cosmic ray (number/s):

Simulate observation

4. Leave the rest of the parameters as their default value.
5. Click simulation button.

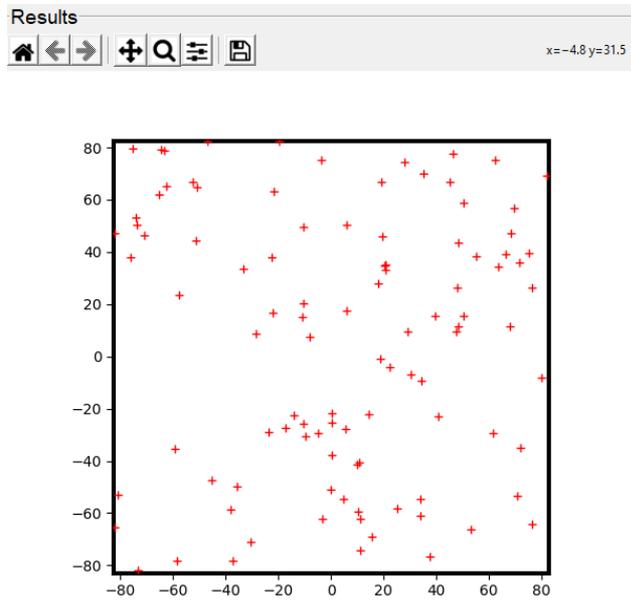
Simulate observation

6. Select 5-square algorithm and press *Compute centroids* button.

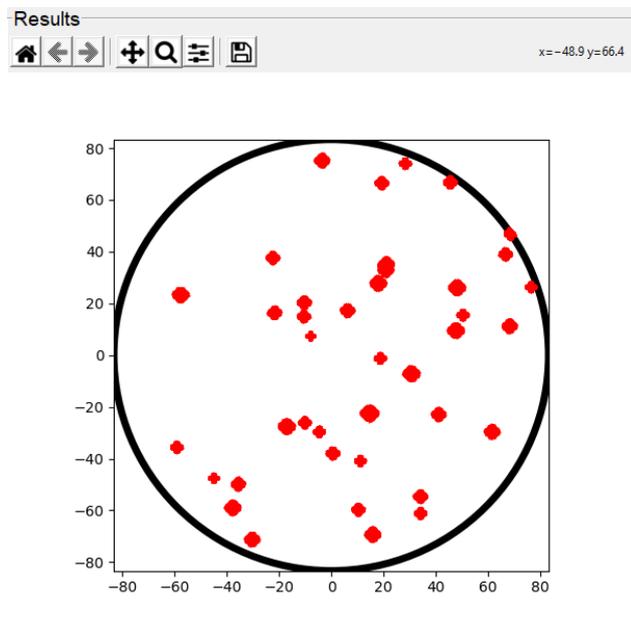
Compute centroids

7. See results clicking Plot CMOS, Plot centroids...

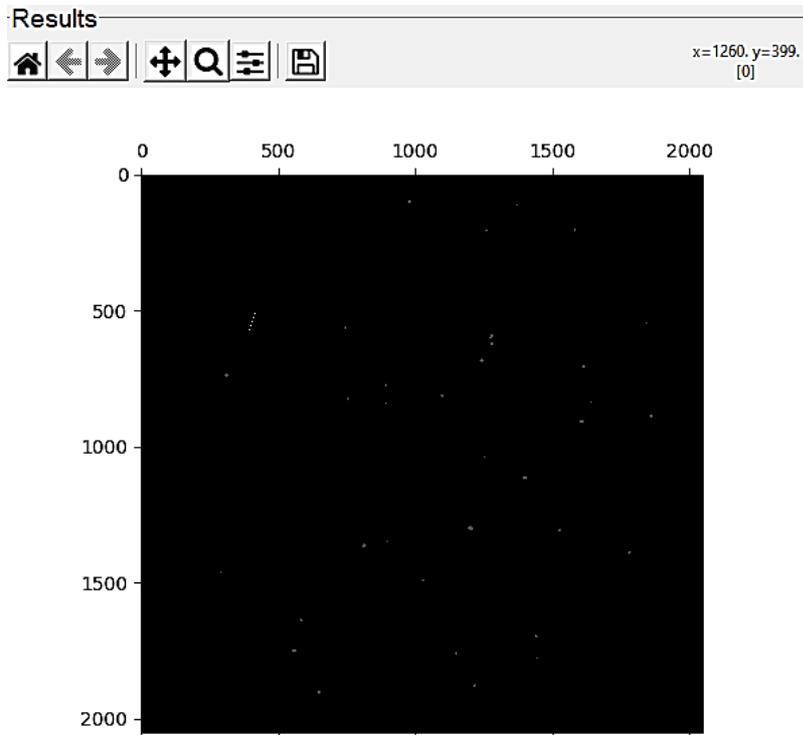
Plot Data Input result:



Plot MCP result:



Plot CMOS result:



Plot Centroids result:

