Neutral Atoms Image Generator

Generated by Doxygen 1.9.5

Chapter 1

README

Chapter 2

t

<col style="width: 100%" > </colgroup> <tbody>

Neutral Atoms Image Generator Model for generating simulated images for neutral atoms in a grid, imaged by different cameras </tbody>

Loading...

Searching...

No Matches

Class Index

C | E | I | T

C

Camera (neutral_atom_imaging_simulation.Camera)

CMOSCamera (neutral_atom_imaging_simulation.Camera)

E

EMCCDCamera (neutral_atom_imaging_simulation.Camera)

Experiment (neutral_atom_imaging_simulation.Experiment)

I

ImageGenerator (neutral_atom_imaging_simulation.ImageGenerator)

T

TweezerArray (neutral_atom_imaging_simulation.Experiment)

Chapter 3

Namespace Index

3.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

Chapter 4

Hierarchical Index

4.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 5

Class Index

5.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 6

Namespace Documentation

6.1 Camera Namespace Reference

Module that implements different cameras.

6.1.1 Detailed Description

Module that implements different cameras.

6.2 Experiment Namespace Reference

Module that implements different experiment setups.

6.2.1 Detailed Description

Module that implements different experiment setups.

6.3 ImageGenerator Namespace Reference

Module for generating images of neutral atoms in a grid.

6.3.1 Detailed Description

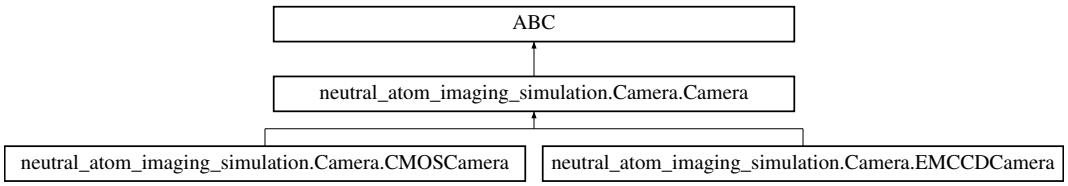
Module for generating images of neutral atoms in a grid.

Chapter 7

Class Documentation

7.1 neutral_atom_imaging_simulation.Camera.Camera Class Reference

Abstract camera class.
Inheritance diagram for neutral_atom_imaging_simulation.Camera.Camera:



Public Member Functions

- def get_image_creation_method (self)
- def set_zernike_coefficients (self, typing.Union[np.ndarray, typing.Tuple[int, int, int, int, int, int, int, int, int, int, int, int, int, int, int]] zernike_coefficients)

  Function for setting the zernike coefficients.
- def set_library (self, ctypes.CDLL library)

  Function for setting the image generation library.
- def library (self)

  Function for getting the image generation library.

Public Attributes

- **zernike_coefficients**

7.1.1 Detailed Description

Abstract camera class.

7.1.2 Member Function Documentation

7.1.2.1 get_image_creation_method()

```
def neutral_atom_imaging_simulation.Camera.Camera.get_image_creation_method (
            self )
```

Reimplemented in neutral_atom_imaging_simulation.Camera.EMCCDCamera, and neutral_atom_imaging_simulation.Camera.CMOS

7.1.2.2 library()

```
def neutral_atom_imaging_simulation.Camera.Camera.library (
              self )
```

Function for getting the image generation library.

Returns

The image generation C library

7.1.2.3 set_library()

```
def neutral_atom_imaging_simulation.Camera.Camera.set_library (
              self,
             ctypes.CDLL library )
```

Function for setting the image generation library.

Parameters

library	The image generation C library

Returns

None

7.1.2.4 set_zernike_coefficients()

```
def neutral_atom_imaging_simulation.Camera.Camera.set_zernike_coefficients (
              self,
             typing.Union[np.ndarray, typing.Tuple[int,int,int,int,int,int,int,int,int,int,int,int,int,int,int
zernike_coefficients )
```

Function for setting the zernike coefficients.

Parameters

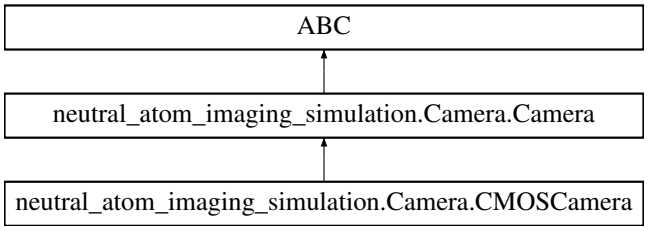
zernike_coefficients	An array of exactly 15 values that represent the coefficients for 1. piston 2. y-tilt 3. x-tilt 4. defocus 5. oblique astigmatism 6. vertical astigmatism 7. vertical coma 8. horizontal coma 9. vertical trefoil 10. oblique trefoil 11. primary spherical 12. vertical secondary astigmatism 13. oblique secondary astigmatism 14. vertical quadrafoil 15. oblique quadrafoil

The documentation for this class was generated from the following file:

- neutral_atom_imaging_simulation/Camera.py

7.2 neutral_atom_imaging_simulation.Camera.CMOSCamera Class Reference

Use this camera if the generated images should look like they are taken by a CMOS camera.
Inheritance diagram for neutral_atom_imaging_simulation.Camera.CMOSCamera:



Public Member Functions

- def __init__ (self, typing.Tuple[int, int] resolution, float dark_current_sampling_alpha=None, float dark_current_sampling_beta=None, float quantum_efficiency=None, float numerical_aperture=None, float physical_pixel_size=None, float magnification=None, float bias_clamp=None, float bias_stdev=None, float row_noise_stdev=None, float column_noise_scale=None, float flicker_noise_scale=None, float preamp-gain=None, float readout_stdev=None, float exposure_time=None, int binning=1)
- def get_image_creation_method (self)

Function for acquiring the function handle of the library that is used to generate images using this camera.

- def apply_settings (self)

  Function for relaying any settings changes to the library.

Public Attributes

- **dark_current_sampling_alpha**
- **dark_current_sampling_beta**
- **quantum_efficiency**
- **numerical_aperture**
- **physical_pixel_size**
- **magnification**
- **bias_clamp**
- **bias_stdev**
- **row_noise_stdev**
- **column_noise_scale**
- **flicker_noise_scale**
- **preampgain**
- **readout_stdev**
- **exposure_time**
- **binning**
- **resolution**
- **zernike_coefficients**

7.2.1 Detailed Description

Use this camera if the generated images should look like they are taken by a CMOS camera.

7.2.2 Constructor & Destructor Documentation

7.2.2.1 __init__()

```
def neutral_atom_imaging_simulation.Camera.CMOSCamera.__init__ (
              self,
             typing.Tuple[int,int] resolution,
             float  dark_current_sampling_alpha = None,
             float  dark_current_sampling_beta = None,
             float  quantum_efficiency = None,
             float  numerical_aperture = None,
             float  physical_pixel_size = None,
             float  magnification = None,
             float  bias_clamp = None,
             float  bias_stdev = None,
             float  row_noise_stdev = None,
             float  column_noise_scale = None,
             float  flicker_noise_scale = None,
             float  preampgain = None,
             float  readout_stdev = None,
             float  exposure_time = None,
             int  binning = 1 )
```

7.2.2.2 Constructor

Initializes all camera specific parameters and relays them to the library

Parameters

resolution	Number of pixels per dimension
dark_current_sampling_alpha	Alpha value for sampling Gamma distribution for dark current
dark_current_sampling_beta	Beta value for sampling Gamma distribution for dark current
quantum_efficiency	Quantum efficiency of the camera for the corresponding wavelength [0.0,1.0]
numerical_aperture	Numerical apperture of the camera setup
physical_pixel_size	Physical pixel size of a single pixel (&#956m)
magnification	Magnification of the optical setup
bias_clamp	Bias clamp of the camera
bias_stdev	Standard deviation of the bias clamp
row_noise_stdev	Standard deviation of the row noise
column_noise_scale	Scale of the zero-mean Gumbel distribution for column noise
flicker_noise_scale	Scale of the zero-mean Gumbel distribution for flicker noise
preampgain	Preampgain of the camera
readout_stdev	Standard deviation of the final readout
exposure_time	Exposure time (s)
binning	Binning factor for the final image

7.2.3 Member Function Documentation

7.2.3.1 apply_settings()

```
def neutral_atom_imaging_simulation.Camera.CMOSCamera.apply_settings (
            self )
```

Function for relaying any settings changes to the library.

Returns

None

7.2.3.2 get_image_creation_method()

```
def neutral_atom_imaging_simulation.Camera.CMOSCamera.get_image_creation_method (
            self )
```

Function for acquiring the function handle of the library that is used to generate images using this camera.

Returns

The library function for generating images using this camera

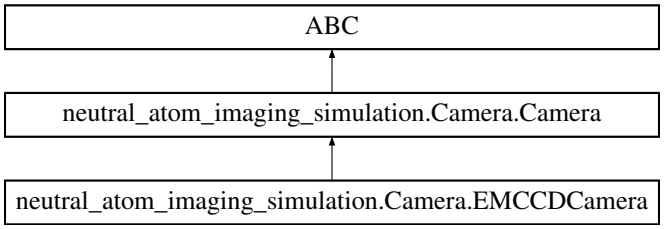
Reimplemented from neutral_atom_imaging_simulation.Camera.Camera.
The documentation for this class was generated from the following file:

- neutral_atom_imaging_simulation/Camera.py

7.3 neutral_atom_imaging_simulation.Camera.EMCCDCamera Class Reference

Use this camera if the generated images should look like they are taken by an electron multiplying charge-coupled device (EMCCD) camera.

Inheritance diagram for neutral_atom_imaging_simulation.Camera.EMCCDCamera:



Public Member Functions

- def __init__ (self, typing.Tuple[int, int] resolution, float dark_current_rate=None, float cic_chance=None, float quantum_efficiency=None, float numerical_aperture=None, float physical_pixel_size=None, float magnification=None, float bias_clamp=None, float preampgain=None, float scic_chance=None, float readout_↩ stdev=None, int number_gain_reg=None, float p0=None, float exposure_time=None, int binning=1)
- def get_image_creation_method (self)

  Function for acquiring the function handle of the library that is used to generate images using this camera.
- def apply_settings (self)

  Function for relaying any settings changes to the library.

Public Attributes

- **dark_current_rate**
- **cic_chance**
- **quantum_efficiency**
- **numerical_aperture**
- **physical_pixel_size**
- **magnification**
- **bias_clamp**
- **preampgain**
- **scic_chance**
- **readout_stdev**
- **number_gain_reg**
- **p0**
- **exposure_time**
- **binning**
- **resolution**
- **zernike_coefficients**

7.3.1 Detailed Description

Use this camera if the generated images should look like they are taken by an electron multiplying charge-coupled device (EMCCD) camera.

7.3.2 Constructor & Destructor Documentation

7.3.2.1 __init__()

```
def neutral_atom_imaging_simulation.Camera.EMCCDCamera.__init__ (
             self,
            typing.Tuple[int,int] resolution,
            float  dark_current_rate = None,
            float  cic_chance = None,
            float  quantum_efficiency = None,
            float  numerical_aperture = None,
            float  physical_pixel_size = None,
            float  magnification = None,
            float  bias_clamp = None,
            float  preampgain = None,
            float  scic_chance = None,
            float  readout_stdev = None,
            int  number_gain_reg = None,
            float  p0 = None,
            float  exposure_time = None,
            int  binning = 1 )
```

7.3.2.2 Constructor

Initializes all camera specific parameters and relays them to the library

Parameters

resolution	Number of pixels per dimension
dark_current_rate	Rate of dark current (photons/s)
cic_chance	Chance for any given pixel to generate a clock-induced charge (photons)
quantum_efficiency	Quantum efficiency of the camera for the corresponding wavelength [0.0,1.0]
numerical_aperture	Numerical apperture of the camera setup
physical_pixel_size	Physical pixel size of a single pixel (&#956m)
magnification	Magnification of the optical setup
bias_clamp	Bias clamp of the camera
preampgain	Preampgain of the camera
scic_chance	Chance for a serial clock-induced charge to occure in any given gain register [0.0,1.0]
readout_stdev	Standard deviation of the final readout
number_gain_reg	Number of gain registers
p0	Chance for generating a secondary electron in any given gain register (1 + p0)^number_gain_reg = gain [0.0,1.0]
exposure_time	Exposure time (s)
binning	Binning factor for the final image

7.3.3 Member Function Documentation

7.3.3.1 apply_settings()

```
def neutral_atom_imaging_simulation.Camera.EMCCDCamera.apply_settings (
             self )
```

Function for relaying any settings changes to the library.

Returns

None

7.3.3.2 get_image_creation_method()

```
def neutral_atom_imaging_simulation.Camera.EMCCDCamera.get_image_creation_method (
            self )
```

Function for acquiring the function handle of the library that is used to generate images using this camera.

Returns

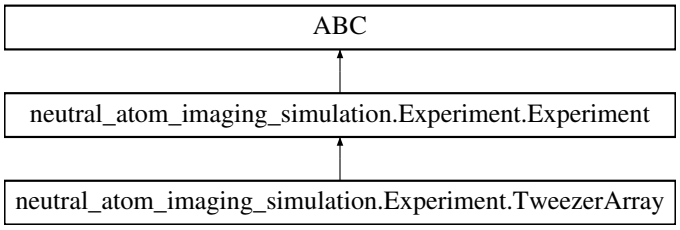
The library function for generating images using this camera

Reimplemented from neutral_atom_imaging_simulation.Camera.Camera.
The documentation for this class was generated from the following file:

- neutral_atom_imaging_simulation/Camera.py

7.4 neutral_atom_imaging_simulation.Experiment.Experiment Class Reference

Abstract experiment class.
Inheritance diagram for neutral_atom_imaging_simulation.Experiment.Experiment:



Public Member Functions

- def get_atom_sites (self)
- def **uses_camera_coords** (self)
- def set_library (self, ctypes.CDLL library)

  Function for setting the image generation library.
- def library (self)

  Function for getting the image generation library.

7.4.1 Detailed Description

Abstract experiment class.

7.4.2 Member Function Documentation

7.4.2.1 get_atom_sites()

```
def neutral_atom_imaging_simulation.Experiment.Experiment.get_atom_sites (
            self )
```

Reimplemented in neutral_atom_imaging_simulation.Experiment.TweezerArray.

7.4.2.2 library()

```
def neutral_atom_imaging_simulation.Experiment.Experiment.library (
            self )
```

Function for getting the image generation library.

Returns

The image generation C library

7.4.2.3 set_library()

```
def neutral_atom_imaging_simulation.Experiment.Experiment.set_library (
            self,
          ctypes.CDLL library )
```

Function for setting the image generation library.

Parameters

library	The image generation C library

Returns

None

The documentation for this class was generated from the following file:

- neutral_atom_imaging_simulation/Experiment.py

7.5 neutral_atom_imaging_simulation.ImageGenerator.ImageGenerator Class Reference

Main class for generating images.

Public Member Functions

- def **__init__** (self)

  Constructor Loads C library for later use.
- def get_library (self)

  Returns the loaded C library.
- def set_camera (self, Camera camera)

  Function for setting the camera.
- def set_experiment (self, Experiment experiment)

  Function for setting the experiment to be imaged.
- def create_image (self)

  Function to be called for generating an image.
- def **read_config_file** (self, str path)

7.5.1 Detailed Description

Main class for generating images.

7.5.2 Member Function Documentation

7.5.2.1 create_image()

```
def neutral_atom_imaging_simulation.ImageGenerator.ImageGenerator.create_image (
            self )
```

Function to be called for generating an image.

Returns

Numpy array of generated image

Numpy array of ground truths per atom site

7.5.2.2 get_library()

```
def neutral_atom_imaging_simulation.ImageGenerator.ImageGenerator.get_library (
            self )
```

Returns the loaded C library.

Returns

The image generation C library

7.5.2.3 set_camera()

```
def neutral_atom_imaging_simulation.ImageGenerator.ImageGenerator.set_camera (
            self,
           Camera camera )
```

Function for setting the camera.

Parameters

camera	The camera to be used for imaging

Returns

None

7.5.2.4 set_experiment()

```
def neutral_atom_imaging_simulation.ImageGenerator.ImageGenerator.set_experiment (
            self,
          Experiment experiment )
```

Function for setting the experiment to be imaged.

Parameters

experiment	The experiment to be imaged

Returns

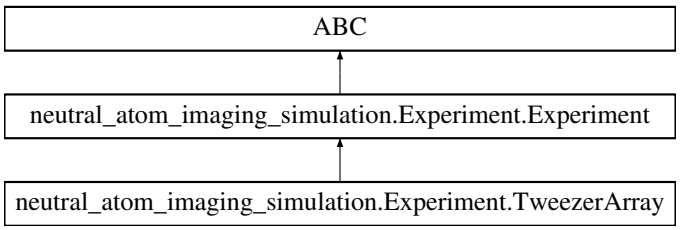
None

The documentation for this class was generated from the following file:

- neutral_atom_imaging_simulation/ImageGenerator.py

7.6 neutral_atom_imaging_simulation.Experiment.TweezerArray Class Reference

Use this if a tweezer array is to be simulated.
Inheritance diagram for neutral_atom_imaging_simulation.Experiment.TweezerArray:



Public Member Functions

- def __init__ (self, stray_light_rate=None, imaging_wavelength=None, scattering_rate=None, survival_↩ probability=None, fill_rate=None)

  Constructor.
- def apply_settings (self)

  Function for relaying any settings changes to the library.
- def set_atom_sites_camera_space (self, atom_sites)

  Function for setting the list of atom sites in normalized camera coordinates.
- def configure_atom_sites_camera_space (self, tuple spacing, tuple count, tuple offset, float angle)

  Function for configuring the layout of atom sites in normalized camera coordinates.
- def set_atom_sites_physical_space (self, atom_sites)

  Function for setting the list of atom sites in physical object space.
- def configure_atom_sites_physical_space (self, tuple spacing, tuple count, tuple offset, float angle)

  Function for configuring the layout of atom sites in physical object space.
- def get_atom_sites (self)

  Function for acquiring the list of atom sites.
- def uses_camera_coords (self)

Public Attributes

- **stray_light_rate**
- **imaging_wavelength**
- **scattering_rate**
- **survival_probability**
- **fill_rate**
- **atom_sites**

7.6.1 Detailed Description

Use this if a tweezer array is to be simulated.

7.6.2 Constructor & Destructor Documentation

7.6.2.1 __init__()

```
def neutral_atom_imaging_simulation.Experiment.TweezerArray.__init__ (
              self,
              stray_light_rate = None,
              imaging_wavelength = None,
              scattering_rate = None,
              survival_probability = None,
              fill_rate = None )
```

Constructor.

Parameters

stray_light_rate	Rate of stray light (photons/s)
imaging_wavelength	The imaging wavelength (um)
scattering_rate	The average of photons emitted by an atom per second (photons/s)
survival_probability	The chance for an atom to survive being imaged [0.0,1.0]
fill_rate	The chance for an atom site to be filled

7.6.3 Member Function Documentation

7.6.3.1 apply_settings()

```
def neutral_atom_imaging_simulation.Experiment.TweezerArray.apply_settings (
              self )
```

Function for relaying any settings changes to the library.

Returns

None

7.6.3.2 configure_atom_sites_camera_space()

```
def neutral_atom_imaging_simulation.Experiment.TweezerArray.configure_atom_sites_camera_space
(
              self,
             tuple spacing,
             tuple count,
             tuple offset,
             float angle )
```

Function for configuring the layout of atom sites in normalized camera coordinates.
On average, there will be column_count $*$ row_count $*$ fill_rate atoms.

Parameters

spacing	The distance between two columns of atoms in normalized camera coordinates (x,y)
count	The number of columns (x,y)
offset	Distance from the image edge to the first atom site in normalized camera coordinates (x,y)
angle	The whole grid is rotated around the first atom site by this angle. Specify in radians

Returns

None

7.6.3.3 configure_atom_sites_physical_space()

```
def neutral_atom_imaging_simulation.Experiment.TweezerArray.configure_atom_sites_physical_↩
space (
              self,
             tuple spacing,
             tuple count,
             tuple offset,
             float angle )
```

Function for configuring the layout of atom sites in physical object space.
On average, there will be column_count ∗ row_count ∗ fill_rate atoms.

Parameters

spacing	The distance between two columns of atoms in &#956m (x,y)
count	The number of columns (x,y)
offset	Distance from the image edge to the first atom site in &#956m (x,y)
angle	The whole grid is rotated around the first atom site by this angle. Specify in radians

Returns

None

7.6.3.4 get_atom_sites()

```
def neutral_atom_imaging_simulation.Experiment.TweezerArray.get_atom_sites (
              self )
```

Function for acquiring the list of atom sites.

Returns

The list of sites

Reimplemented from neutral_atom_imaging_simulation.Experiment.Experiment.

7.6.3.5 set_atom_sites_camera_space()

```
def neutral_atom_imaging_simulation.Experiment.TweezerArray.set_atom_sites_camera_space (
              self,
              atom_sites )
```

Function for setting the list of atom sites in normalized camera coordinates.

Parameters

atom_sites	List of coordinate tuples where atom may reside, each coordinate value should be within [0.0,1.0]

Returns

None

7.6.3.6 set_atom_sites_physical_space()

```
def neutral_atom_imaging_simulation.Experiment.TweezerArray.set_atom_sites_physical_space (
              self,
              atom_sites )
```

Function for setting the list of atom sites in physical object space.

Parameters

atom_sites	List of coordinate tuples where atom may reside, each coordinate value should be within [0.0,1.0]

Returns

None

7.6.3.7 uses_camera_coords()

```
def neutral_atom_imaging_simulation.Experiment.TweezerArray.uses_camera_coords (
              self )
```

Reimplemented from neutral_atom_imaging_simulation.Experiment.Experiment.

The documentation for this class was generated from the following file:

- neutral_atom_imaging_simulation/Experiment.py