
jicbioimage.transform Documentation

Release 0.5.1

Tjelvar Olsson and Matthew Hartley

November 06, 2015

1	The <code>jicbioimage.transform</code> Python package	3
1.1	Features	3
1.2	Related packages	3
2	API documentation	5
2.1	<code>jicbioimage.transform</code>	5
	Python Module Index	7

jicimagelib documentation

The `jicbioimage.transform` Python package

The `jicbioimage.transform` Python package provides a set of standard image transformations to the `jicbioimage` namespace package.

- Documentation: <http://jicbioimagetransform.readthedocs.org>
- GitHub: <https://github.com/JIC-CSB/jicbioimage.transform>
- PyPI: <https://pypi.python.org/pypi/jicbioimage.transform>
- Free software: MIT License

1.1 Features

- Built in functionality for generating audit trails of the image transforms applied
- Cross-platform: Linux, Mac and Windows are all supported
- Works with Python 2.7, 3.3 and 3.4

1.2 Related packages

1.2.1 `jicbioimage`

- Documentation: <http://jicbioimage.readthedocs.org>
- GitHub: <https://github.com/JIC-CSB/jicbioimage>

1.2.2 `jicbioimage.core`

- Documentation: <http://jicbioimagecore.readthedocs.org>
- GitHub: <https://github.com/JIC-CSB/jicbioimage.core>

1.2.3 `jicbioimage.segment`

- Documentation: <http://jicbioimagesegment.readthedocs.org>
- GitHub: <https://github.com/JIC-CSB/jicbioimage.segment>

1.2.4 jicbioimage.illustrate

- Documentation: <http://jicbioimageillustrate.readthedocs.org>
- GitHub: <https://github.com/JIC-CSB/jicbioimage.illustrate>

API documentation

2.1 `jicbioimage.transform`

Module containing image transformation functions.

The `jicbioimage.transform` module contains a number of built-in general purpose transformations that have had the `jicbioimage.core.transformation()` function decorator applied to them.

`jicbioimage.transform.dilate_binary(*args, **kwargs)`
Return dilated image.

Parameters

- **image** – `jicbioimage.core.image.Image`
- **selem** – neighborhood expressed as 1's and 0's, default is a cross

Returns dilated image

`jicbioimage.transform.equalize_adaptive_clahe(*args, **kwargs)`
Return contrast limited adaptive histogram equalized image.

The return value is normalised to the range 0 to 1.

Parameters

- **image** – numpy array or `jicbioimage.core.image.Image` of dtype float
- **ntiles** – number of tile regions
- **clip_limit** – clipping limit in range 0 to 1, higher values give more contrast

`jicbioimage.transform.erode_binary(*args, **kwargs)`
Return eroded image.

Parameters

- **image** – `jicbioimage.core.image.Image`
- **selem** – neighborhood expressed as 1's and 0's, default is a cross

Returns eroded image

`jicbioimage.transform.find_edges_sobel(*args, **kwargs)`
Return edges detected using the Sobel method.

Parameters

- **image** – `jicbioimage.core.image.Image`

- **mask** – Optional mask indicating regions to ignore

Returns eroded image

`jicbioimage.transform.invert(*args, **kwargs)`

Return an inverted image of the same dtype.

Assumes the full range of the input dtype is in use and that no negative values are present in the input image.

Parameters **image** – `jicbioimage.core.image.Image`

Returns inverted image of the same dtype as the input

`jicbioimage.transform.max_intensity_projection(*args, **kwargs)`

Return maximum intensity projection of a stack.

Parameters **stack** – 3D array from which to project third dimension

Returns `jicbioimage.core.image.Image`

`jicbioimage.transform.mean_intensity_projection(*args, **kwargs)`

Return mean intensity projection of a stack.

Parameters **stack** – 3D array from which to project third dimension

Returns `jicbioimage.core.image.Image`

`jicbioimage.transform.median_intensity_projection(*args, **kwargs)`

Return mean intensity projection of a stack.

Parameters **stack** – 3D array from which to project third dimension

Returns `jicbioimage.core.image.Image`

`jicbioimage.transform.min_intensity_projection(*args, **kwargs)`

Return minimum intensity projection of a stack.

Parameters **stack** – 3D array from which to project third dimension

Returns `jicbioimage.core.image.Image`

`jicbioimage.transform.remove_small_objects(*args, **kwargs)`

Remove small objects from an boolean image.

Parameters **image** – boolean numpy array or `jicbioimage.core.image.Image`

Returns boolean `jicbioimage.core.image.Image`

`jicbioimage.transform.smooth_gaussian(*args, **kwargs)`

Returns Gaussian smoothed image.

Parameters

- **image** – numpy array or `jicbioimage.core.image.Image`
- **sigma** – standard deviation

Returns `jicbioimage.core.image.Image`

`jicbioimage.transform.threshold_otsu(*args, **kwargs)`

Return image thresholded using Otsu's method.

j

`jicbioimage.transform`, 5

D

`dilate_binary()` (in module `jicbioimage.transform`), 5

E

`equalize_adaptive_clahe()` (in module `jicbioimage.transform`), 5

`erode_binary()` (in module `jicbioimage.transform`), 5

F

`find_edges_sobel()` (in module `jicbioimage.transform`), 5

I

`invert()` (in module `jicbioimage.transform`), 6

J

`jicbioimage.transform` (module), 5

M

`max_intensity_projection()` (in module `jicbioimage.transform`), 6

`mean_intensity_projection()` (in module `jicbioimage.transform`), 6

`median_intensity_projection()` (in module `jicbioimage.transform`), 6

`min_intensity_projection()` (in module `jicbioimage.transform`), 6

R

`remove_small_objects()` (in module `jicbioimage.transform`), 6

S

`smooth_gaussian()` (in module `jicbioimage.transform`), 6

T

`threshold_otsu()` (in module `jicbioimage.transform`), 6