
jicbioimage.segment Documentation

Release 0.3.0

Tjelvar Olsson and Matthew Hartley

May 05, 2016

1	The <code>jicbioimage.segment</code> Python package	3
1.1	Features	3
1.2	Related packages	3
2	API documentation	5
2.1	<code>jicbioimage.segment</code>	5
	Python Module Index	9

jicimagelib documentation

The `jicbioimage.segment` Python package

The `jicbioimage.segment` Python package provides a set of standard segmentation methods to the `jicbioimage` namespace package.

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

1.1 Features

- Functionality for segmenting images
- 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.io>
- GitHub: <https://github.com/JIC-CSB/jicbioimage>

1.2.2 `jicbioimage.core`

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

1.2.3 `jicbioimage.transform`

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

1.2.4 jicbioimage.illustrate

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

API documentation

2.1 jicbioimage.segment

Module containing image segmentation functions.

Example usage:

```
>>> import numpy as np
>>> from jicbioimage.core.image import Image
>>> ar = np.array([[1, 1, 0, 0, 0],
...               [1, 1, 0, 0, 0],
...               [0, 0, 0, 0, 0],
...               [0, 0, 2, 2, 2],
...               [0, 0, 2, 2, 2]], dtype=np.uint8)
>>> im = Image.from_array(ar)
>>> connected_components(im)
SegmentedImage([[3, 3, 1, 1, 1],
               [3, 3, 1, 1, 1],
               [1, 1, 1, 1, 1],
               [1, 1, 2, 2, 2],
               [1, 1, 2, 2, 2]])
>>> connected_components(im, background=0)
SegmentedImage([[2, 2, 0, 0, 0],
               [2, 2, 0, 0, 0],
               [0, 0, 0, 0, 0],
               [0, 0, 1, 1, 1],
               [0, 0, 1, 1, 1]])
>>> segmentation = connected_components(im, background=0)
>>> segmentation.history
['Created image from array', 'Applied connected_components transform']
```

class jicbioimage.segment.**Region**

Class representing a region of interest in an image.

The `jicbioimage.core.region.Region` class is a subclass of `numpy.ndarray`.

However, note that it will compress any data given to it to boolean.

```
>>> import numpy as np
>>> ar = np.array([-1, 0, 1, 2])
>>> Region(ar)
Region([ True, False,  True,  True], dtype=bool)
```

To select an particular element use the `jicbioimage.core.region.Region.select_from_array()` class method.

```
>>> Region.select_from_array(ar, identifier=2)
Region([False, False, False,  True], dtype=bool)
```

area

Number of non-zero elements.

Returns int

border

Region formed by taking border elements.

Returns `jicbioimage.core.region.Region`

centroid

Return centroid as (y, x) tuple.

convex_hull

Region representing the convex hull.

Returns `jicbioimage.core.region.Region`

dilate (*iterations=1*)

Return a dilated region.

Parameters **iterations** – number of iterations to use in dilation

Returns `jicbioimage.core.region.Region`

index_arrays

All nonzero elements as a pair of arrays.

inner

Region formed by taking non-border elements.

Returns `jicbioimage.core.region.Region`

perimeter

Return the perimeter.

Returns int

points

Region as a list of points.

classmethod select_from_array (*array, identifier*)

Return a region from a numpy array.

Parameters

- **array** – `numpy.ndarray`
- **identifier** – value representing the region to select in the array

Returns `jicbioimage.core.region.Region`

```
class jicbioimage.segment.SegmentedImage(shape, dtype=<type 'numpy.uint8'>, buffer=None,
                                         offset=0, strides=None, order=None, name=None,
                                         log_in_history=True)
```

Class representing the results of applying a segmentation to an image.

Each unique pixel value represents a different region of the segmentation. 0 represents background and positive integers represent the different regions.

background

Return the segmented image background.

In other words the region with pixel values 0.

Returns *jicbioimage.core.region.Region*

identifiers

Return a set of unique identifiers in the segmented image.

merge_regions (*id1*, *id2*)

Merge two regions into one.

The merged region will take on the *id1* identifier.

Parameters

- **id1** – region 1 identifier
- **id2** – region 2 identifier

number_of_segments

Return the number of segments present in the segmented image.

png (*width=None*)

Return png string of image.

Parameters **width** – integer specifying the desired width

Returns png as a string

pretty_color_image

Return segmentation as a pretty color image.

Returns *jicbioimage.core.image.Image*

region_by_identifier (*identifier*)

Return region of interest corresponding to the supplied identifier.

Parameters **identifier** – integer corresponding to the segment of interest

Returns *jicbioimage.core.region.Region*

remove_region (*identifier*)

Remove region from the segmentation.

Parameters **identifier** – region identifier

unique_color_image

Return segmentation as a unique color image.

Returns *jicbioimage.core.image.Image*

`jicbioimage.segment.connected_components(*args, **kwargs)`

Return *jicbioimage.core.image.SegmentedImage*.

Parameters

- **image** – input *jicbioimage.core.image.Image*
- **connectivity** – maximum number of orthogonal hops to consider a pixel/voxel as a neighbor
- **background** – consider all pixels with this value (int) as background

Returns *jicbioimage.core.image.SegmentedImage*

`jicbioimage.segment.watershed_with_seeds(*args, **kwargs)`

Return `jicbioimage.core.image.SegmentedImage`.

Parameters

- **image** – input `jicbioimage.core.image.Image`
- **seeds** – `numpy.ndarray` of same shape as image, each seed needs to be a unique integer
- **mask** – bool `numpy.ndarray` of same shape as image, only regions that are marked as True will be labelled

Returns `jicbioimage.core.image.SegmentedImage`

j

`jicbioimage.segment`, 5

A

area (jicbioimage.segment.Region attribute), 6

B

background (jicbioimage.segment.SegmentedImage attribute), 6

border (jicbioimage.segment.Region attribute), 6

C

centroid (jicbioimage.segment.Region attribute), 6

connected_components() (in module jicbioimage.segment), 7

convex_hull (jicbioimage.segment.Region attribute), 6

D

dilate() (jicbioimage.segment.Region method), 6

I

identifiers (jicbioimage.segment.SegmentedImage attribute), 7

index_arrays (jicbioimage.segment.Region attribute), 6

inner (jicbioimage.segment.Region attribute), 6

J

jicbioimage.segment (module), 5

M

merge_regions() (jicbioimage.segment.SegmentedImage method), 7

N

number_of_segments (jicbioimage.segment.SegmentedImage attribute), 7

P

perimeter (jicbioimage.segment.Region attribute), 6

png() (jicbioimage.segment.SegmentedImage method), 7

points (jicbioimage.segment.Region attribute), 6

pretty_color_image (jicbioimage.segment.SegmentedImage attribute), 7

R

Region (class in jicbioimage.segment), 5

region_by_identifier() (jicbioimage.segment.SegmentedImage method), 7

remove_region() (jicbioimage.segment.SegmentedImage method), 7

S

SegmentedImage (class in jicbioimage.segment), 6

select_from_array() (jicbioimage.segment.Region class method), 6

U

unique_color_image (jicbioimage.segment.SegmentedImage attribute), 7

W

watershed_with_seeds() (in module jicbioimage.segment), 7