scikit-image

image processing in python

`skimage.viewer`:

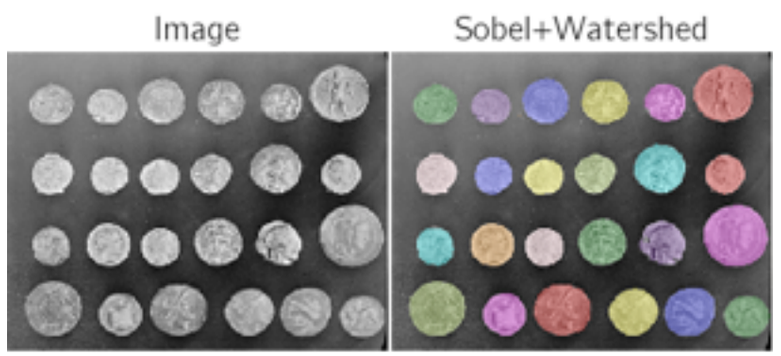
Interactive image viewer

Tony S. Yu

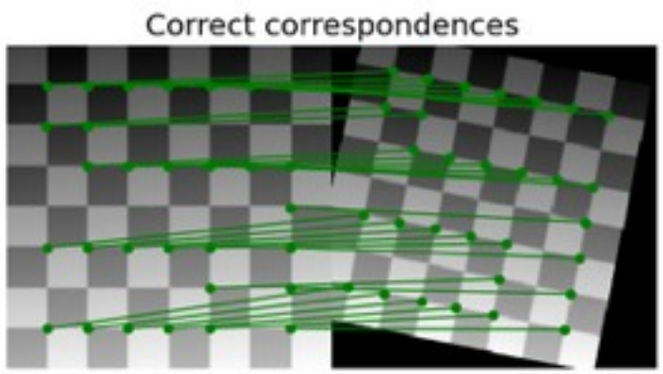
scientific software developer

ENTHOUGHT

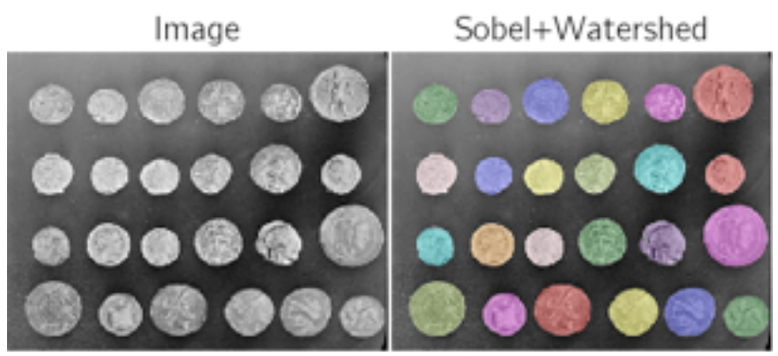
SciPy 2013



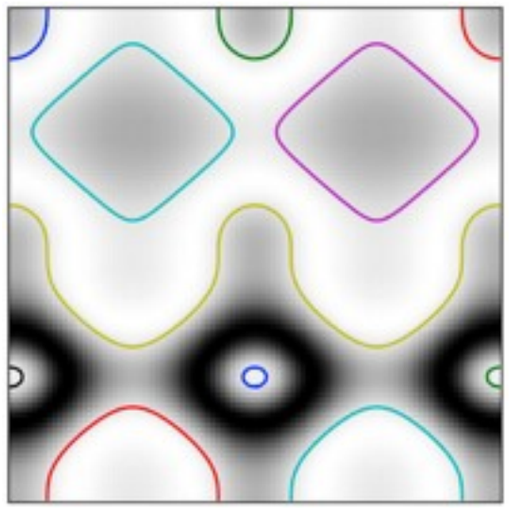
`skimage.segmentation`



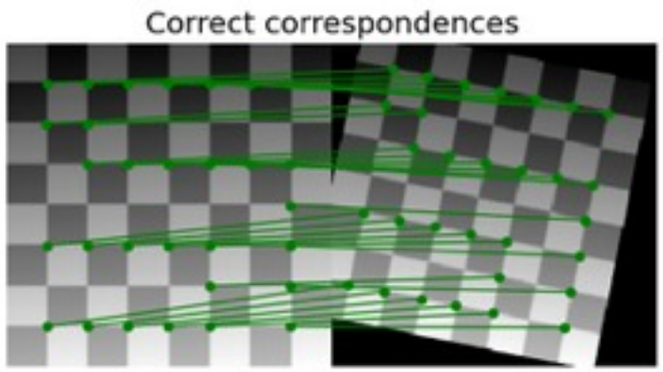
skimage.feature



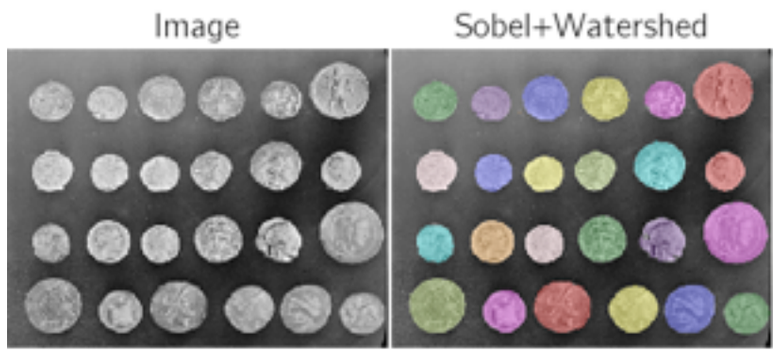
skimage.segmentation



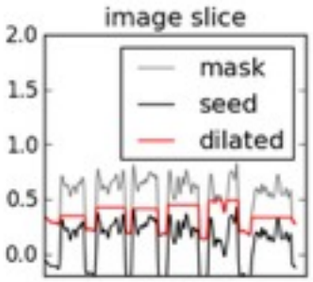
skimage.measure

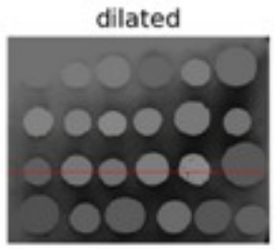


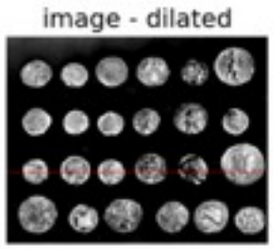
skimage.feature



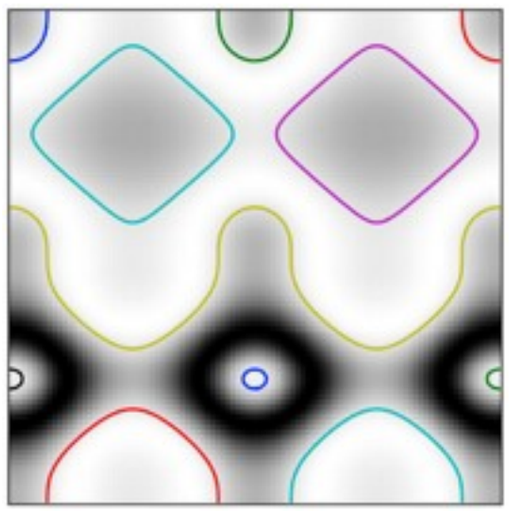
skimage.segmentation



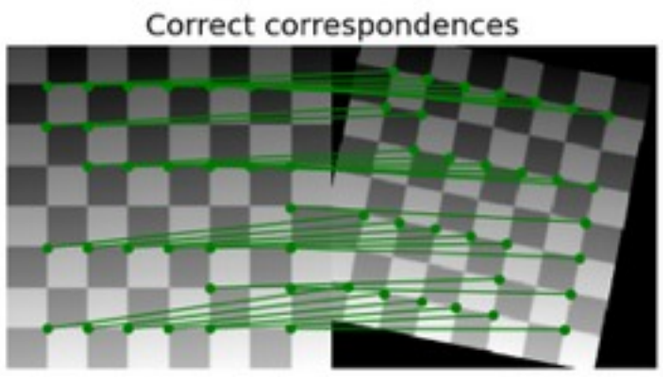




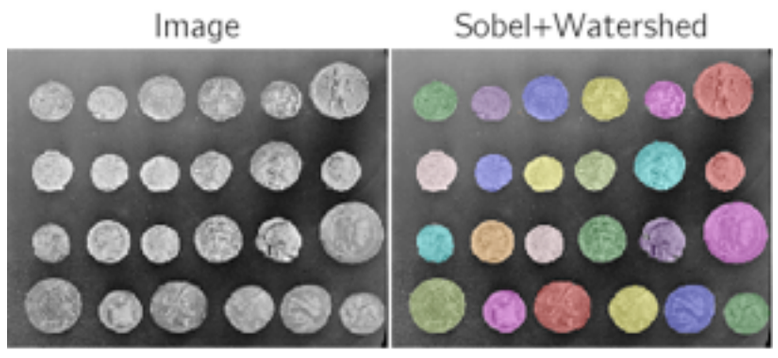
skimage.morphology



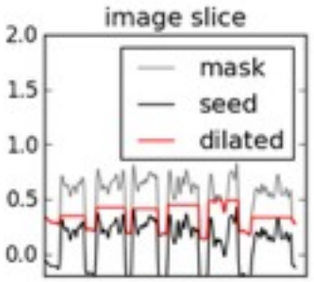
skimage.measure

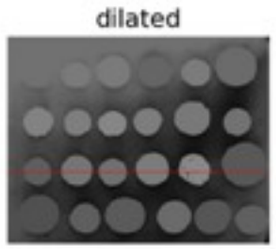


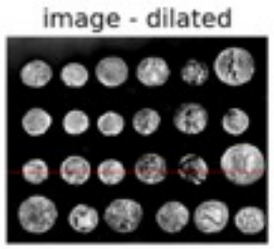
skimage.feature



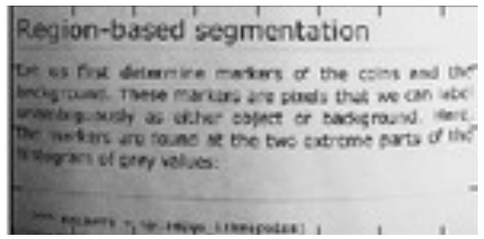
skimage.segmentation

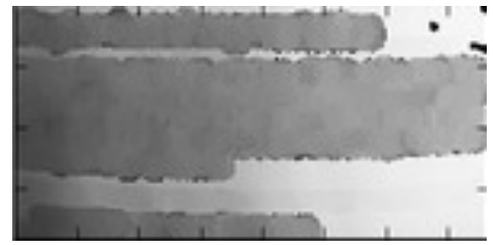




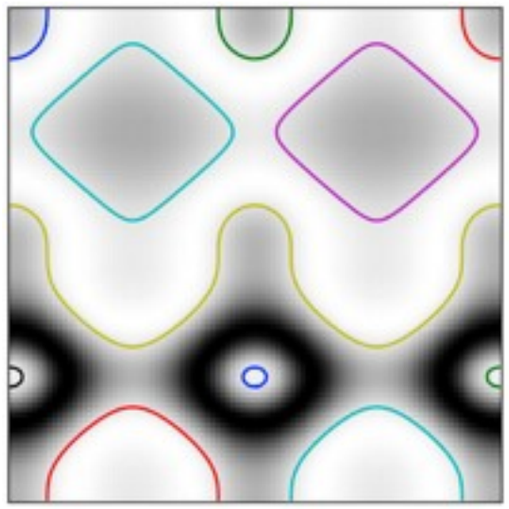


skimage.morphology

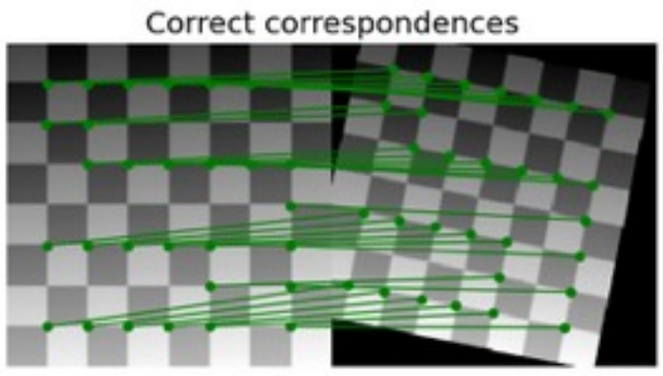




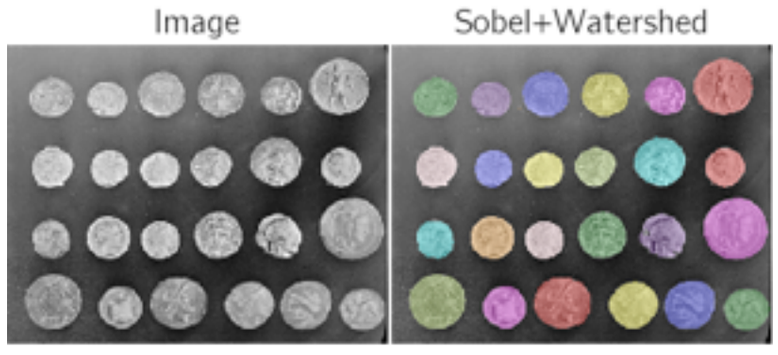
skimage.filter



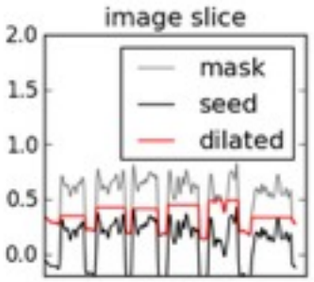
skimage.measure

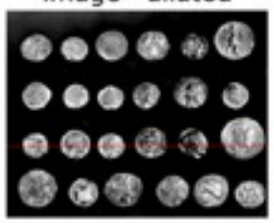


skimage.feature

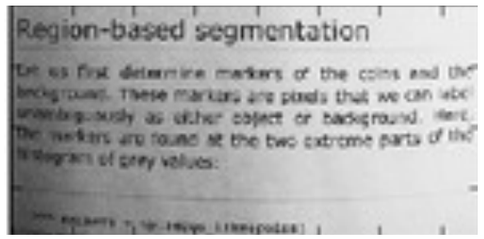


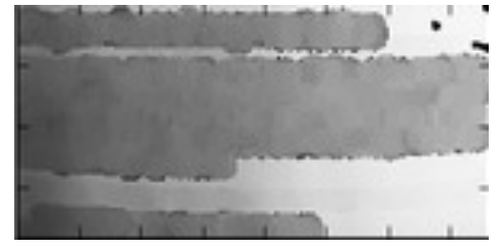
skimage.segmentation



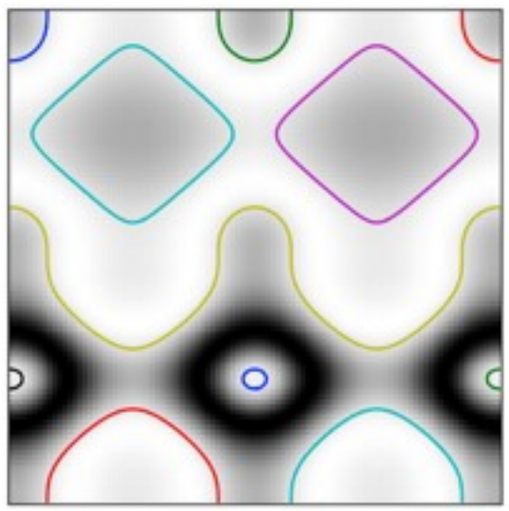


skimage.morphology

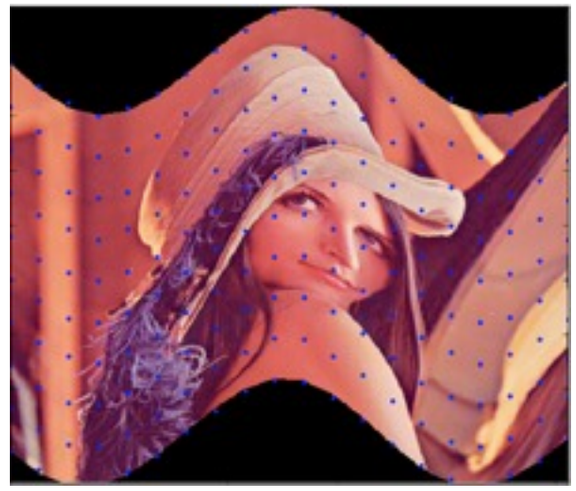




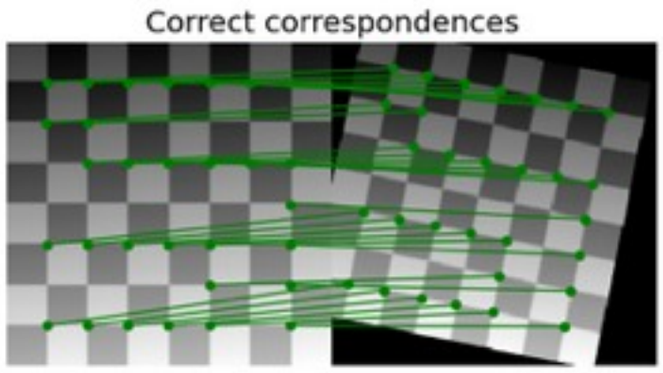
skimage.filter



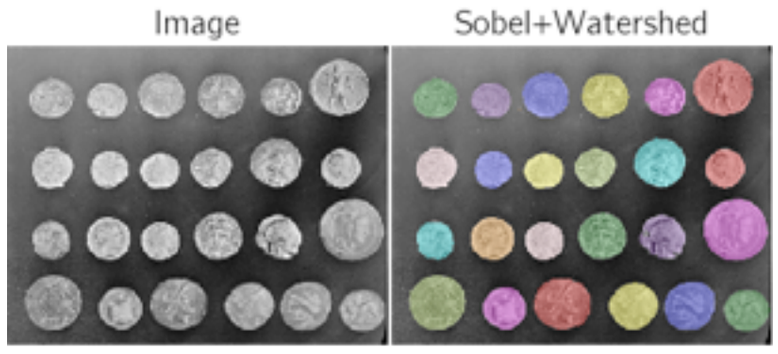
skimage.measure



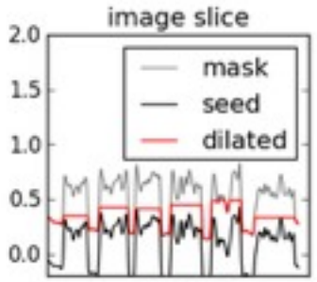
skimage.transform

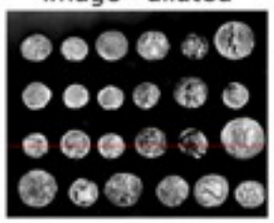


skimage.feature

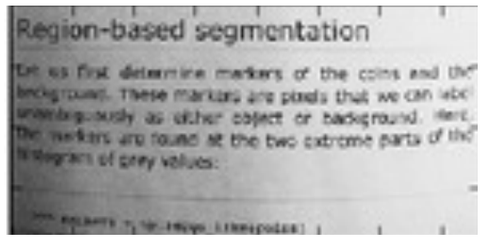


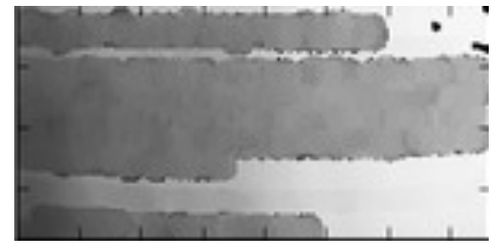
skimage.segmentation



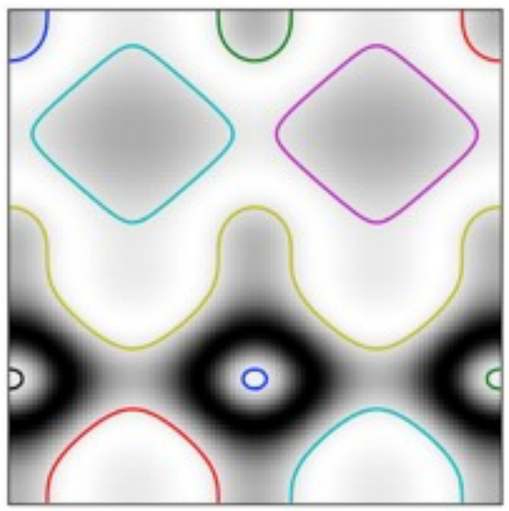


skimage.morphology

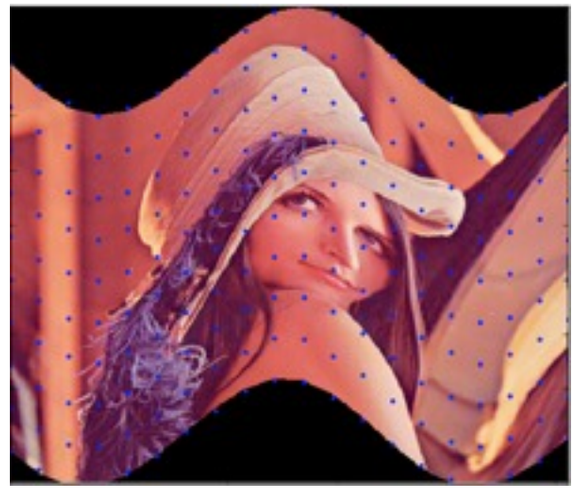




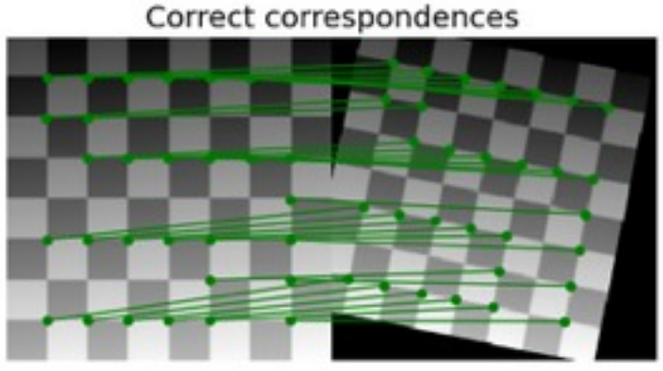
skimage.filter



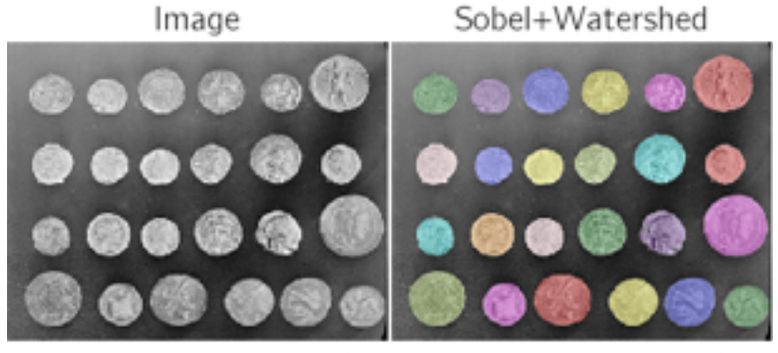
skimage.measure



skimage.transform



skimage.feature



skimage.segmentation

- skimage.io
- skimage.color
- skimage.exposure
- skimage.viewer

scikit-image viewer

- Started by Guilluame Gay as a PR to add a line profile tool

scikit-image viewer

- Started by Guilluame Gay as a PR to add a line profile tool
- Refactored into scikit-loupe (a.k.a. `skloupe`; pure matplotlib): `https://github.com/tonysyu/skloupe`

scikit-image viewer

- Started by Guilluame Gay as a PR to add a line profile tool
- Refactored into scikit-loupe (a.k.a. `skloupe`; pure matplotlib): `https://github.com/tonysyu/skloupe`
- Rewritten as Qt-application with Matplotlib canvas at SciPy 2012

scikit-image viewer

- Started by Guilluame Gay as a PR to add a line profile tool
- Refactored into scikit-loupe (a.k.a. `skloupe`; pure matplotlib): `https://github.com/tonysyu/skloupe`
- Rewritten as Qt-application with Matplotlib canvas at SciPy 2012
  - Qt windows and widgets

scikit-image viewer

- Started by Guilluame Gay as a PR to add a line profile tool
- Refactored into scikit-loupe (a.k.a. `skloupe`; pure matplotlib): `https://github.com/tonysyu/skloupe`
- Rewritten as Qt-application with Matplotlib canvas at SciPy 2012
  - Qt windows and widgets
  - Matplotlib for drawing images, interaction, ease of plotting (histograms, lines, etc.)

The basic viewer

```
from skimage import data
from skimage.viewer import ImageViewer

image = data.camera()
viewer = ImageViewer(image)
viewer.show()
```

The basic viewer

```
from skimage import data
from skimage.viewer import ImageViewer
```

```
image = data.camera()
viewer = ImageViewer(image)
viewer.show()
```

Plugins: interact with scikit-image functions

Plugins: interact with scikit-image functions

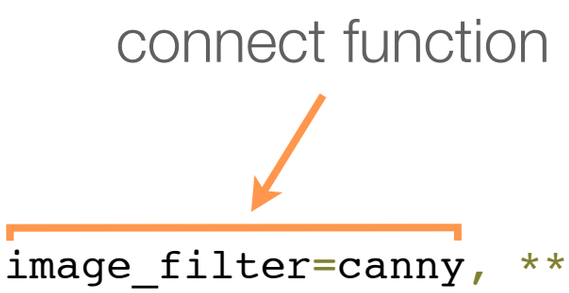
```
from skimage import data
from skimage.filter import canny
from skimage.viewer import ImageViewer
from skimage.viewer.plugins.overlayplugin import OverlayPlugin
from skimage.viewer.widgets import Slider


class CannyPlugin(OverlayPlugin):

    def __init__(self, *args, **kwargs):
        super(CannyPlugin, self).__init__(image_filter=canny, **kwargs)

    def attach(self, image_viewer):
        self.add_widget(Slider('sigma', 0, 5))
        self.add_widget(Slider('low threshold', 0, 255, value_type='int'))
        self.add_widget(Slider('high threshold', 0, 255, value_type='int'))

        super(CannyPlugin, self).attach(image_viewer)


image = data.camera()
viewer = ImageViewer(image)
viewer += CannyPlugin()
viewer.show()
```

Plugins: interact with scikit-image functions

```
from skimage import data
from skimage.filter import canny
from skimage.viewer import ImageViewer
from skimage.viewer.plugins.overlayplugin import OverlayPlugin
from skimage.viewer.widgets import Slider


class CannyPlugin(OverlayPlugin):

    def __init__(self, *args, **kwargs):
        super(CannyPlugin, self).__init__(image_filter=canny, **kwargs)

    def attach(self, image_viewer):
        self.add_widget(Slider('sigma', 0, 5))
        self.add_widget(Slider('low threshold', 0, 255, value_type='int'))
        self.add_widget(Slider('high threshold', 0, 255, value_type='int'))

        super(CannyPlugin, self).attach(image_viewer)


image = data.camera()
viewer = ImageViewer(image)
viewer += CannyPlugin()
viewer.show()
```

Plugins: interact with scikit-image functions

```
from skimage import data
from skimage.filter import canny
from skimage.viewer import ImageViewer
from skimage.viewer.plugins.overlayplugin import OverlayPlugin
from skimage.viewer.widgets import Slider


class CannyPlugin(OverlayPlugin):

    def __init__(self, *args, **kwargs):
        super(CannyPlugin, self).__init__(image_filter=canny, **kwargs)

    def attach(self, image_viewer):
        self.add_widget(Slider('sigma', 0, 5))
        self.add_widget(Slider('low threshold', 0, 255, value_type='int'))
        self.add_widget(Slider('high threshold', 0, 255, value_type='int'))

        super(CannyPlugin, self).attach(image_viewer)


image = data.camera()
viewer = ImageViewer(image)
viewer += CannyPlugin()
viewer.show()
```



Plugins: interact with scikit-image functions

```
from skimage import data
from skimage.filter import canny
from skimage.viewer import ImageViewer
from skimage.viewer.plugins.overlayplugin import OverlayPlugin
from skimage.viewer.widgets import Slider


class CannyPlugin(OverlayPlugin):

    def __init__(self, *args, **kwargs):
        super(CannyPlugin, self).__init__(image_filter=canny, **kwargs)

    def attach(self, image_viewer):
        self.add_widget(Slider('sigma', 0, 5))
        self.add_widget(Slider('low threshold', 0, 255, value_type='int'))
        self.add_widget(Slider('high threshold', 0, 255, value_type='int'))

        super(CannyPlugin, self).attach(image_viewer)


image = data.camera()
viewer = ImageViewer(image)
viewer += CannyPlugin()
viewer.show()
```

connect function

Plugins: interact with scikit-image functions

```
from skimage import data
from skimage.filter import canny
from skimage.viewer import ImageViewer
from skimage.viewer.plugins.overlayplugin import OverlayPlugin
from skimage.viewer.widgets import Slider


class CannyPlugin(OverlayPlugin):

    def __init__(self, *args, **kwargs):
        super(CannyPlugin, self).__init__(image_filter=canny, **kwargs)

    def attach(self, image_viewer):
        self.add_widget(Slider('sigma', 0, 5))
        self.add_widget(Slider('low threshold', 0, 255, value_type='int'))
        self.add_widget(Slider('high threshold', 0, 255, value_type='int'))

        super(CannyPlugin, self).attach(image_viewer)


image = data.camera()
viewer = ImageViewer(image)
viewer += CannyPlugin()
viewer.show()
```

connect function

connect plugin

Quick plugin syntax

```
from skimage import data
from skimage.filter import canny

from skimage.viewer import ImageViewer
from skimage.viewer.widgets import Slider
from skimage.viewer.plugins.overlayplugin import OverlayPlugin
```

```
plugin = OverlayPlugin(image_filter=canny)
plugin += Slider('sigma', 0, 5)
plugin += Slider('low threshold', 0, 255, value_type='int')
plugin += Slider('high threshold', 0, 255, value_type='int')

viewer = ImageViewer(data.camera())
viewer += plugin
viewer.show()
```

Live Demos

Thanks to ...

- Stéfan van der Walt (scikit-image BDFL)
- Guillaume Gay for the line-profile tool and original inspiration
- Steven Silvester for testing / PR-review and suggesting some user-interface improvements
- All scikit-image developers and contributors

scikit-image

- Join the scikit-image Sprint this Friday
- website: `skimage.org`
- repo: `github.com/scikit-image/scikit-image.git`

  `github.com/scikit-image/scikit-image/tree/master/viewer_examples`
- mailing-list: `groups.google.com/forum/#!forum/scikit-image`
- me: `tyu@enthought.com`