PlotInteraction

March 13, 2017

```
In [ ]: from silx.gui import qt
        %gui qt
```

1 simple plot of a 2D image

- using Plot2D

1.1 load data from data/lena.hdf5

```
In [1]: # input using .hdf5
        import h5py
        import numpy
        dataPath='data/ascent.h5'
        f=h5py.File(dataPath)
        from silx.io.utils import h5ls
        h5ls(dataPath)
```

```
Out[1]: '<HDF5 dataset "data": shape (512, 512), type "<i8">\n'
```

```
In [2]: image=numpy.array(f['data'], dtype='float32')
```

1.2 plot the image

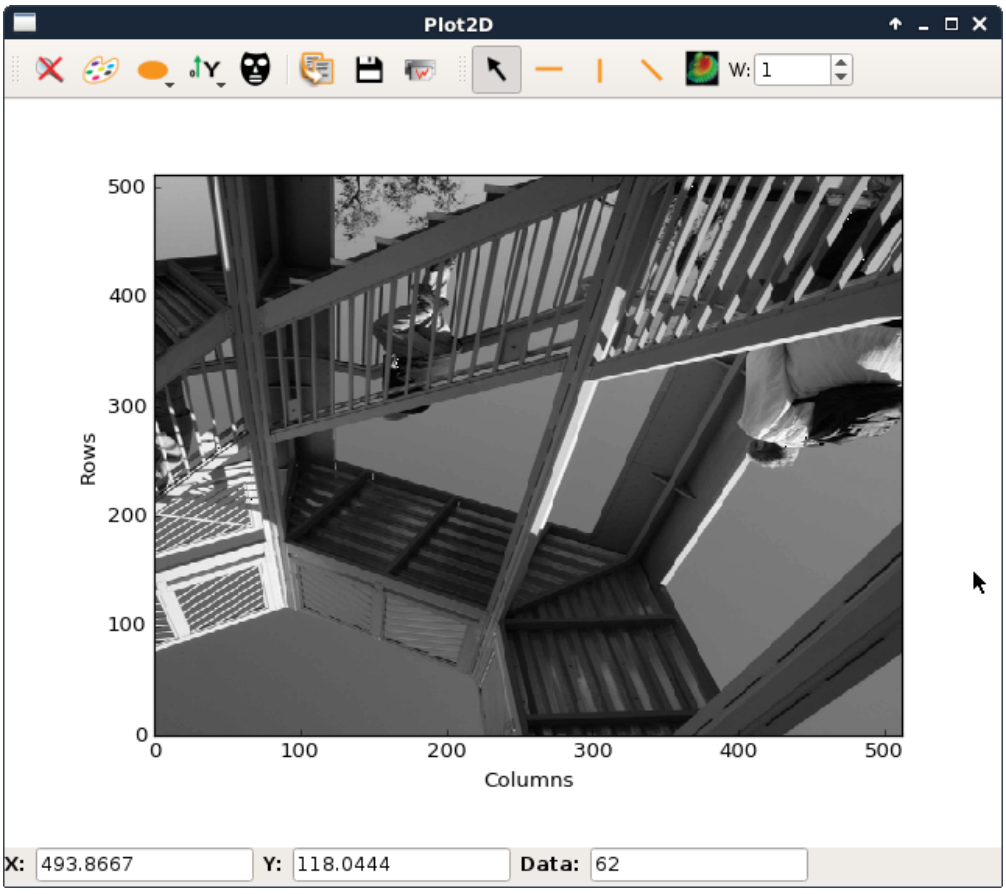
```
In [ ]: from silx.gui.plot import Plot2D
        plotImage=Plot2D()
        plotImage.addImage(image, origin=(0, 0), legend='sino')
        plotImage.show()
```

2 display the pixel intensity distribution

2.1 create the histogramnd

- using silx.math.histogram.Histogramnd
- http://www.silx.org/doc/silx/dev/modules/math/histogram.html

```
In [ ]: from silx.math.histogram import Histogramnd
        histo, w_histo, edges = Histogramnd(image.flatten(),
                                            n_bins=256,
                                            histo_range=[0,256])
```



ascent image

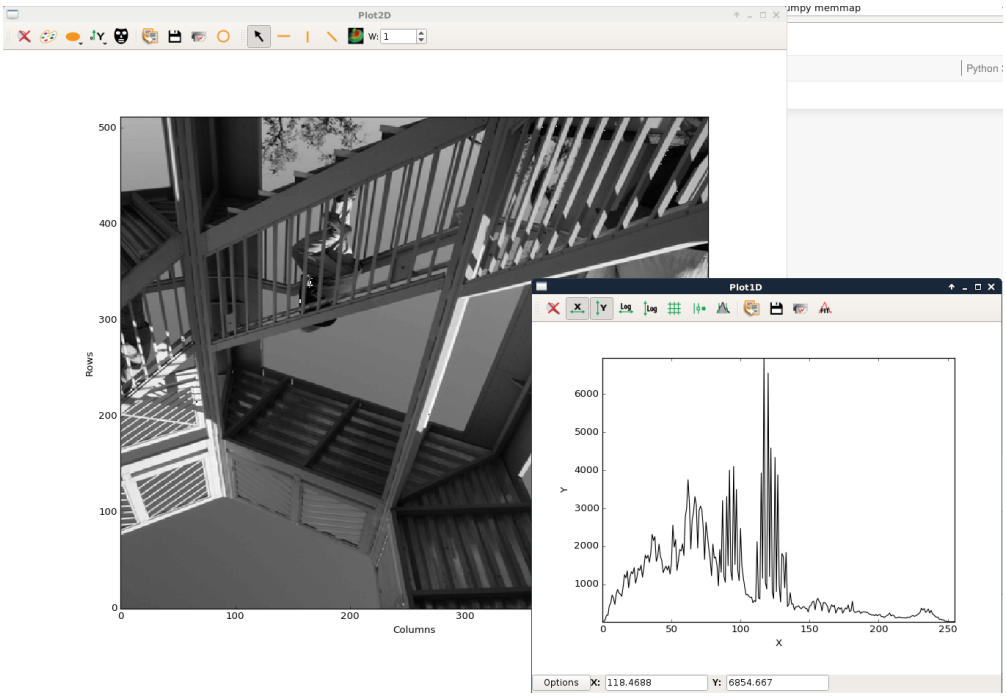
2.2 plot the histogram

- using silx.gui.plot.Plot1d

```
In [ ]: from silx.gui.plot import Plot1D
        plotHisto = Plot1D()
        plotHisto.addCurve(range(256), histo, legend='intensity')
        plotHisto.show()
```

3 create a PlotAction which plot the histogram for the current image

- using silx.gui.plot.PlotActions.PlotAction
- doc@ http://www.silx.org/doc/silx/dev/modules/gui/plot/plotactions_examples.html



simple-image

```
In [ ]: from silx.gui.plot.PlotActions import PlotAction
        from silx.math.histogram import Histogramnd
        from silx.gui.plot import Plot1D

        class ComputeHistogramAction(PlotAction):
            """Computes the intensity distribution on the current image

            :param plot: :class:`.PlotWidget` instance on which to operate
            :param parent: See :class:`QAction`
            """
```

```
        def __init__(self, plot, parent=None):
            PlotAction.__init__(self,
                                plot,
                                icon='shape-circle',
                                text='pixels intensity',
                                tooltip='Compute image intensity distribution',
                                triggered=self.computeIntensityDistribution,
                                parent=parent)
            self.plotHistogram=Plot1D()

        def computeIntensityDistribution(self):
            """Get the active image and compute the image
            intensity distribution"""
            # By inheriting from PlotAction, we get access to attribute
            # self.plot
            # which is a reference to the PlotWindow
            activeImage = self.plot.getActiveImage()

            if activeImage is not None:

                histo, w_histo, edges = Histogramnd(activeImage[0].flatten(),
                                                    n_bins=256,
                                                    histo_range=[0,256])

                self.plotHistogram.addCurve(range(256),
                                            histo,
                                            legend='pixel intensity')
                self.plotHistogram.show()
```

3.1 Add this action into the toolBar of the window

```
In [ ]: plotHisto.clear()
        myaction=ComputeHistogramAction(plotImage)
        toolBar=plotImage.toolBar()
        toolBar.addAction(myaction)
        plotImage.show()
```

4 show automatically the histogram when the image change

- using plotImage.sigActiveImageChanged.connect(plotHisto)

```
In [ ]: from silx.math.histogram import Histogramnd

        def computeIntensityDistribution():
            """Get the active image and compute the image
            intensity distribution"""
            # By inheriting from PlotAction, we get access to attribute
            # self.plot
```

```
            # which is a reference to the PlotWindow
            activeImage = plotImage.getActiveImage()

            if activeImage is not None:
                histo, w_histo, edges = Histogramnd(activeImage[0].flatten(),
                                                    n_bins=256,
                                                    histo_range=[0,256])

                from silx.gui.plot import Plot1D
                plotHistogram = Plot1D()
                plotHistogram.addCurve(range(256),
                                       histo,
                                       legend='pixel intensity')
                plotHistogram.show()
```

In [ ]:
```
plotImage=Plot2D()
plotImage.sigActiveImageChanged.connect(computeIntensityDistribution)
plotImage.addImage(image, origin=(0, 0), legend='lena')
```

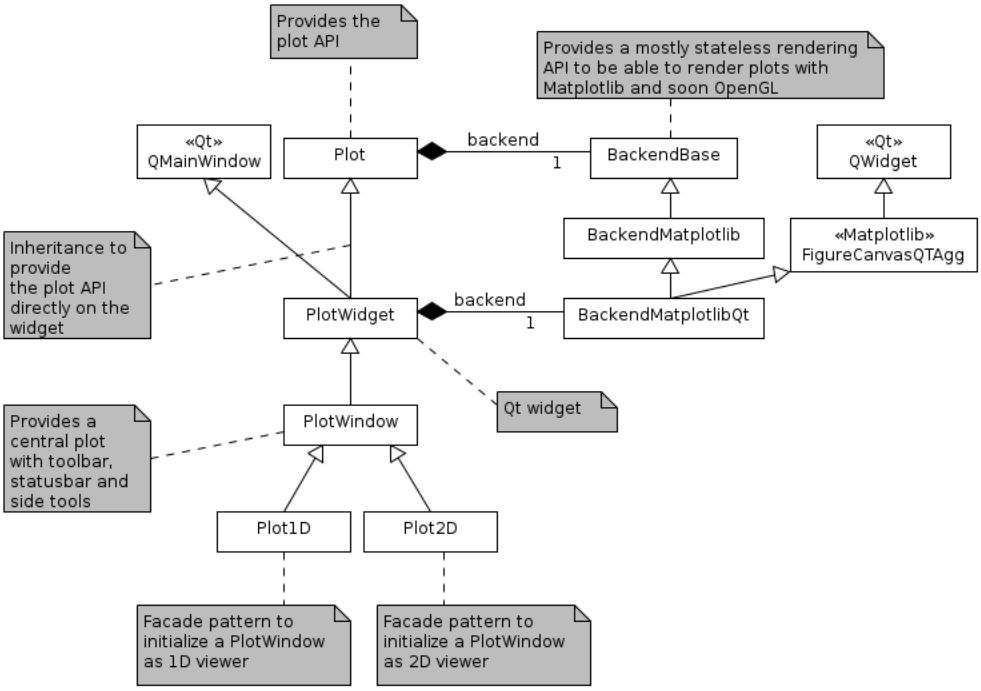
In [ ]:
```
from silx.gui.plot.PlotActions import PlotAction
from silx.math.histogram import Histogramnd
from silx.gui.plot import Plot1D

class ComputeHistogramAction(PlotAction):
    """Computes the intensity distribution on the current image

    :param plot: :class:`.PlotWidget` instance on which to operate
    :param parent: See :class:`QAction`
    """
    def __init__(self, plot, parent=None):
        PlotAction.__init__(self,
                            plot,
                            icon='shape-circle',
                            text='pixels intensity',
                            tooltip='Compute image intensity distribution',
                            triggered=self.computeIntensityDistribution,
                            checkable=True,
                            parent=parent)
        self.plotHistogram=Plot1D()
        self.plot.sigActiveImageChanged.connect(self.update)

    def update(self):
        # By inheriting from PlotAction, we get access to attribute
        # self.plot
        # which is a reference to the PlotWindow
        activeImage = self.plot.getActiveImage()

        if activeImage is not None:
```

```
                    histo, w_histo, edges = Histogramnd(activeImage[0].flatten(),
                                                        n_bins=256,
                                                        histo_range=[0,25

                    self.plotHistogram.addCurve(range(256),
                                                histo,
                                                legend='pixel intensity')

            def computeIntensityDistribution(self):
                """Get the active image and compute the image intensity distributio
                if self.isChecked():
                    self.update()
                    self.plotHistogram.show()
                else:
                    self.plotHistogram.hide()
```

```
In [ ]: plotImage=Plot2D()
        myaction=ComputeHistogramAction(plotImage)
        toolBar=plotImage.toolBar()
        toolBar.addAction(myaction)
        plotImage.addImage(image, origin=(0, 0), legend='ascent')
        plotImage.show()
```

```
In [ ]: plotImage.addImage(image/2.0, origin=(0, 0), legend='acent')
```

5 For information : the class diagram of the Plot module



plot class diagram