08-edge-detection

January 13, 2020

1 Edge Detection

1.1 Line Profile

In [ ]:
```
import skimage.io
from matplotlib import pyplot as plt

image = skimage.io.imread("../data/07-bw.jpg")
plt.plot(image[200, :])
plt.plot(image[400, :])
plt.plot(image[600, :])
```

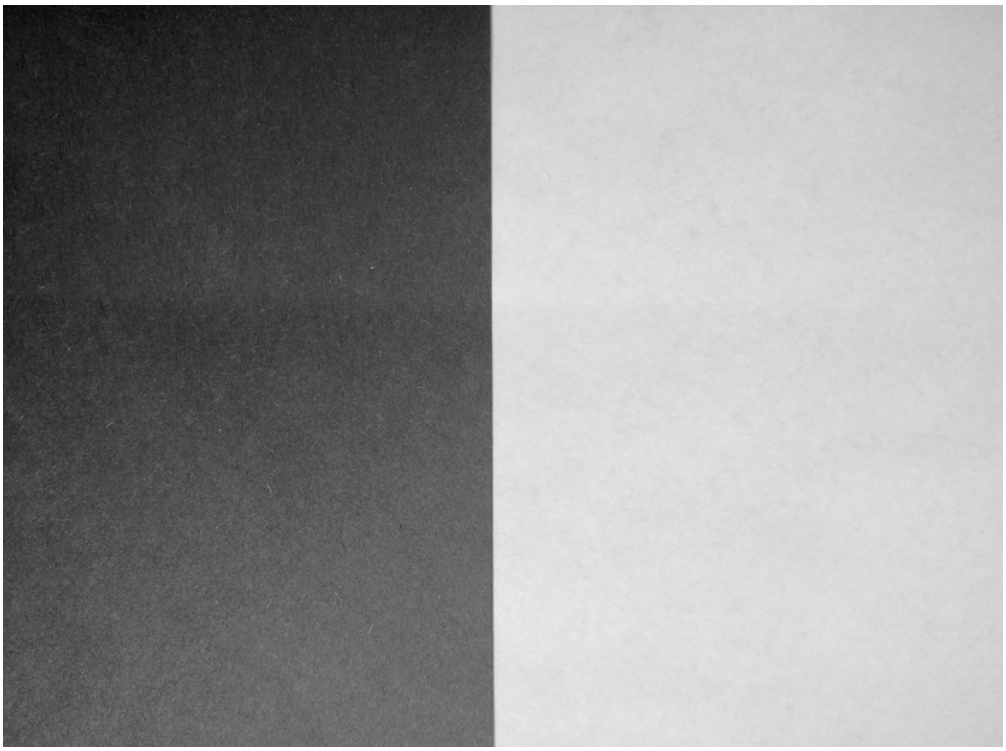
In [ ]:
```
# zoom in on the image
skimage.io.imshow(image[200:250, 380:430])
```

In [ ]:
```
# zoomed in, where would you place the edge?!
plt.plot(range(400, 410), image[200, 400:410])
```

1.2 Canny Edge Detection

In [ ]:
```
# created by John Canny in 1986
# fancy algorithm composed of multiple steps
# algorithm steps:
    # gaussian blur, remove noise; !sigma
    # sobel edge detection: derivative of curve fitted to pixels
    # non-maximum suppression
    # double threshold
    # hysteresis
# important part: parameters -> sigma, low, high threshold
# read image as grayscale:
import skimage.feature
import skimage

image = skimage.io.imread("../data/07-shapes.tif", as_gray=True)
sigma = 2
low_threshold = 0.1
high_threshold = 0.3
edges = skimage.feature.canny(
```



An Edge

```
    image=image,
    sigma=sigma,
    low_threshold=low_threshold,
    high_threshold=high_threshold,
)
plt.imshow(edges)
# to get rid of artifacts
# plt.imshow(edges, interpolation='bicubic')
```

1.2.1 Keypoints

- The `skimage.viewer.ImageViewer` is extended using a `skimage.viewer.plugins.Plugin`.
- We supply a filter function callback when creating a Plugin.
- Parameters of the callback function are manipulated interactively by creating sliders with the `skimage.viewer.widgets.slider()` function and adding them to the plugin.