image_transformation

November 21, 2022

[2]:
```
import numpy as np
import matplotlib.pyplot as plt
from skimage.util import invert
from skimage.color import rgb2gray
from skimage.filters import sobel, gaussian
from skimage.exposure import rescale_intensity
from skimage import io, data, img_as_ubyte, img_as_float
```

1 Image Representation

[ ]:
```
im = data.camera()
print(im[250:260,250:260])
```

```
[[  8   4   8  34  62  74  73  50  25  13]
 [  8   3   8  46  82  77  58  34  15  10]
 [  8   2   9  60 104  82  44  23  12   9]
 [  9   2  11  74 127  89  34  17  14  11]
 [  9   1  11  87 148  97  27  12  14  11]
 [ 10   2  13  95 160 101  23   9  14  12]
 [ 10   4  14  88 145  90  19   7  13  11]
 [ 10   8  19  78 123  76  17   8  13  10]
 [ 10  13  27  74 108  67  16   8  12   9]
 [ 10  21  39  72  93  58  16  10  14  10]]
```

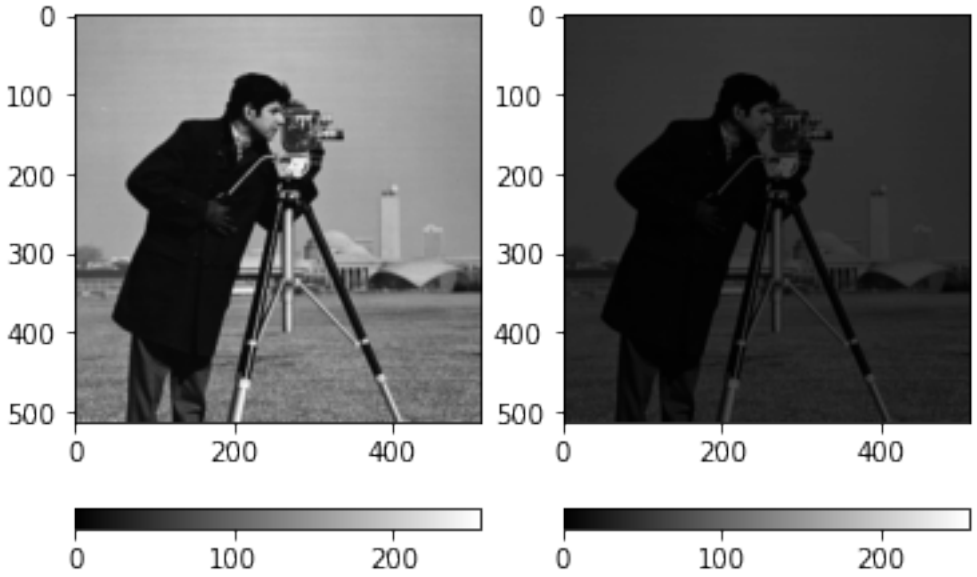
Multiply

[ ]:
```
im = data.camera()
imf = 0.3 * im
imf = imf.astype(np.uint8)

fig = plt.figure()
ax = fig.add_subplot(1, 2, 1)
p = plt.imshow(im, cmap='gray')
c = plt.colorbar(orientation='horizontal')
plt.clim(0, 255)

ax = fig.add_subplot(1, 2, 2)
```

```
p = plt.imshow(imf, cmap='gray')
c = plt.colorbar(orientation='horizontal')
plt.clim(0, 255)
```



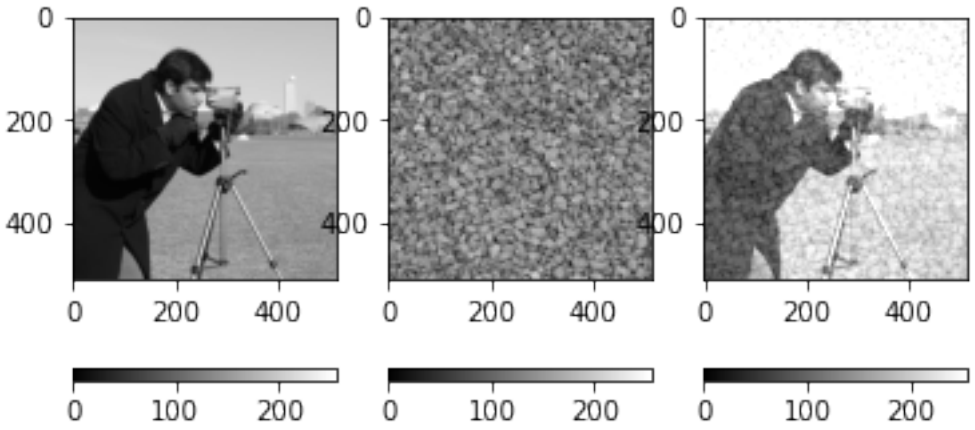
2 Mix

[4]:
```
ima = data.camera()
imb = data.gravel()
im = ima.astype(np.float) + 0.5 * imb.astype(np.float)

fig = plt.figure()
ax = fig.add_subplot(1, 3, 1)
p = plt.imshow(ima, cmap='gray')
c = plt.colorbar(orientation='horizontal')
plt.clim(0, 255)

ax = fig.add_subplot(1, 3, 2)
p = plt.imshow(imb, cmap='gray')
c = plt.colorbar(orientation='horizontal')
plt.clim(0, 255)

ax = fig.add_subplot(1, 3, 3)
p = plt.imshow(im, cmap='gray')
c = plt.colorbar(orientation='horizontal')
plt.clim(0, 255)
```

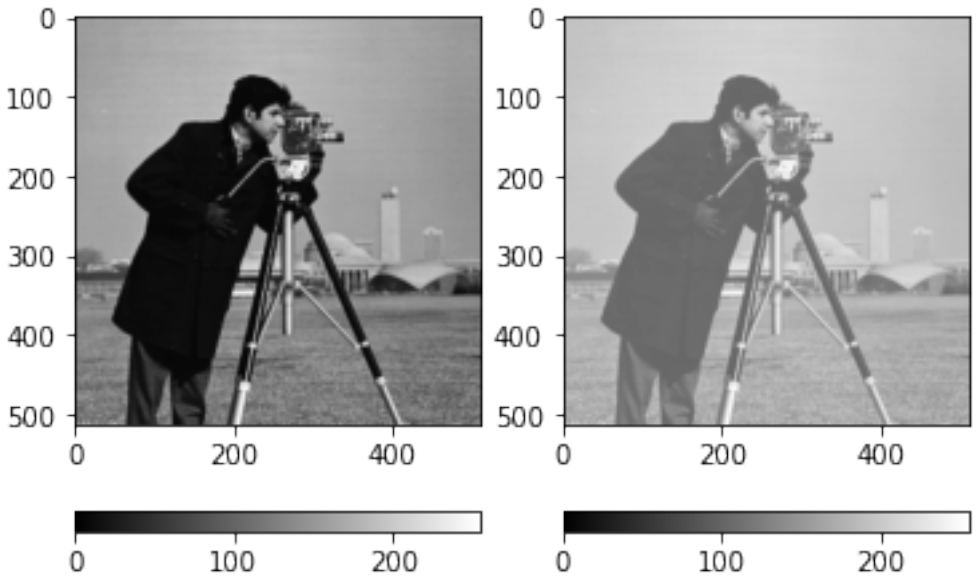


3 Addition

[ ]:
```
im = data.camera()
c = 100
imf = (rescale_intensity(im, out_range=(c, 255))).astype(np.uint8)

fig = plt.figure()
ax = fig.add_subplot(1, 2, 1)
p = plt.imshow(im, cmap='gray')
c = plt.colorbar(orientation='horizontal')
plt.clim(0, 255)

ax = fig.add_subplot(1, 2, 2)
p = plt.imshow(imf, cmap='gray')
c = plt.colorbar(orientation='horizontal')
plt.clim(0, 255)
```



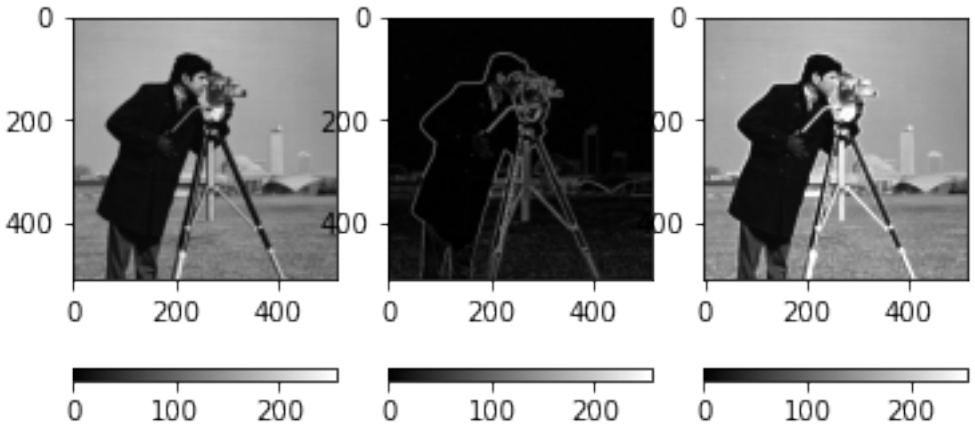
4 Blending

[ ]:
```
im = data.camera()
img = sobel(im)
img =  rescale_intensity(img, out_range=(0, 255))
imb = im + img

fig = plt.figure()
ax = fig.add_subplot(1, 3, 1)
p = plt.imshow(ima, cmap='gray')
c = plt.colorbar(orientation='horizontal')
plt.clim(0, 255)

ax = fig.add_subplot(1, 3, 2)
p = plt.imshow(img, cmap='gray')
c = plt.colorbar(orientation='horizontal')
plt.clim(0, 255)

ax = fig.add_subplot(1, 3, 3)
p = plt.imshow(imb, cmap='gray')
c = plt.colorbar(orientation='horizontal')
plt.clim(0, 255)
```



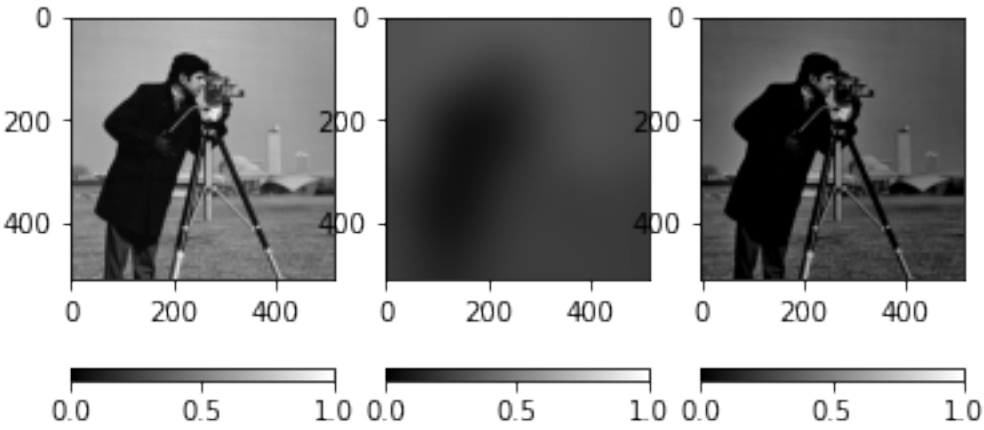
Subtraction

[ ]:
```
im0 = img_as_float(data.camera())
im1 = gaussian(im0, sigma=50)/2
ims = im0 - im1
ims[ims<0] = 0

fig = plt.figure()
ax = fig.add_subplot(1, 3, 1)
p = plt.imshow(im0, cmap='gray')
c = plt.colorbar(orientation='horizontal')
plt.clim(0, 1)

ax = fig.add_subplot(1, 3, 2)
p = plt.imshow(im1, cmap='gray')
c = plt.colorbar(orientation='horizontal')
plt.clim(0, 1)

ax = fig.add_subplot(1, 3, 3)
p = plt.imshow(ims, cmap='gray')
c = plt.colorbar(orientation='horizontal')
plt.clim(0, 1)
```



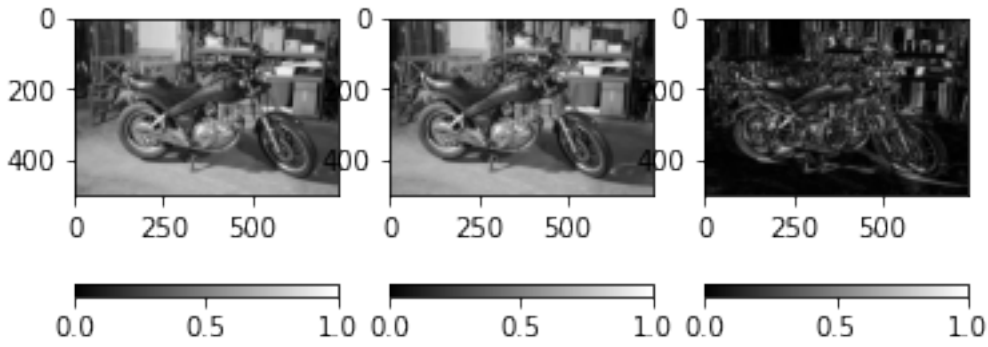
5 Subtraction - move

[ ]:
```
im0, im1, _ = data.stereo_motorcycle()
im0 = rgb2gray(im0)
im1 = rgb2gray(im1)
ims = np.abs(im0 - im1)

fig = plt.figure()
ax = fig.add_subplot(1, 3, 1)
p = plt.imshow(im0, cmap='gray')
c = plt.colorbar(orientation='horizontal')
plt.clim(0, 1)

ax = fig.add_subplot(1, 3, 2)
p = plt.imshow(im1, cmap='gray')
c = plt.colorbar(orientation='horizontal')
plt.clim(0, 1)

ax = fig.add_subplot(1, 3, 3)
p = plt.imshow(ims, cmap='gray')
c = plt.colorbar(orientation='horizontal')
plt.clim(0, 1)
```



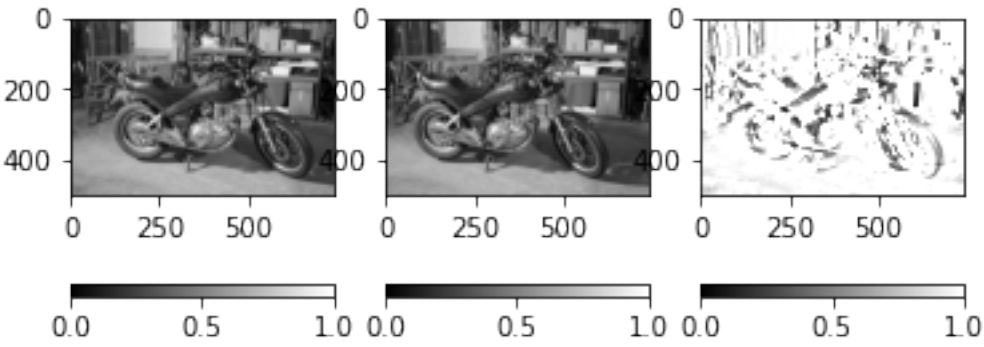
6 Division

[ ]:
```
im0, im1, _ = data.stereo_motorcycle()
im0 = rgb2gray(im0)
im1 = rgb2gray(im1)
im0 = img_as_float(im0)
im1 = img_as_float(im1)
imd = np.zeros_like(im0)
imd[im1!=0] = im0[im1!=0] / im1[im1!=0]

fig = plt.figure()
ax = fig.add_subplot(1, 3, 1)
p = plt.imshow(im0, cmap='gray')
c = plt.colorbar(orientation='horizontal')
plt.clim(0, 1)

ax = fig.add_subplot(1, 3, 2)
p = plt.imshow(im1, cmap='gray')
c = plt.colorbar(orientation='horizontal')
plt.clim(0, 1)

ax = fig.add_subplot(1, 3, 3)
p = plt.imshow(imd, cmap='gray')
c = plt.colorbar(orientation='horizontal')
plt.clim(0, 1)
```

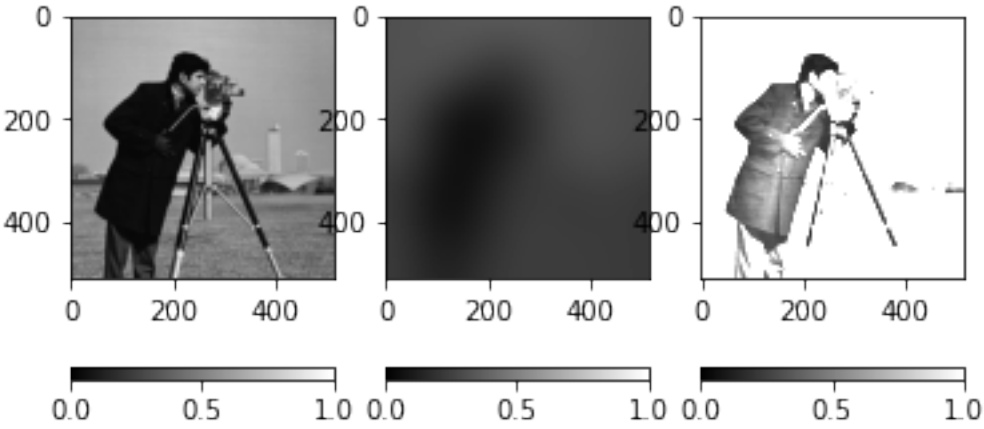


[ ]:
```
im0 = img_as_float(data.camera())
im1 = gaussian(im0, sigma=50)/2
imd = np.zeros_like(im0)
imd[im1!=0] = im0[im1!=0] / im1[im1!=0]

fig = plt.figure()
ax = fig.add_subplot(1, 3, 1)
p = plt.imshow(im0, cmap='gray')
c = plt.colorbar(orientation='horizontal')
plt.clim(0, 1)

ax = fig.add_subplot(1, 3, 2)
p = plt.imshow(im1, cmap='gray')
c = plt.colorbar(orientation='horizontal')
plt.clim(0, 1)

ax = fig.add_subplot(1, 3, 3)
p = plt.imshow(imd, cmap='gray')
c = plt.colorbar(orientation='horizontal')
plt.clim(0, 1)
```

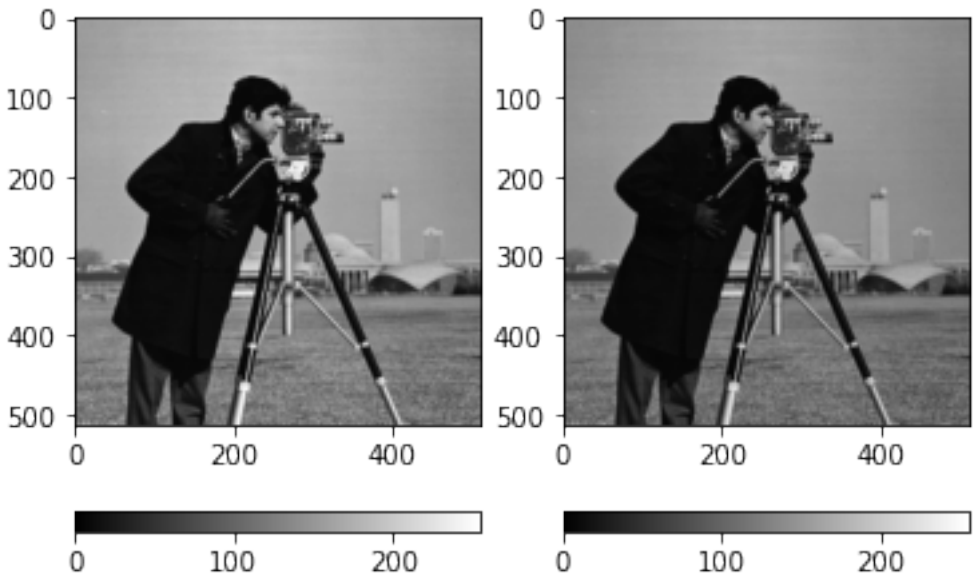


7 Mutliplication

[ ]:
```
im = data.camera()
imm = 0.3 * im
imm = 3 * imm
imm = imm.astype(np.uint8)

fig = plt.figure()
ax = fig.add_subplot(1, 2, 1)
p = plt.imshow(im, cmap='gray')
c = plt.colorbar(orientation='horizontal')
plt.clim(0, 255)

ax = fig.add_subplot(1, 2, 2)
p = plt.imshow(imm, cmap='gray')
c = plt.colorbar(orientation='horizontal')
plt.clim(0, 255)
```



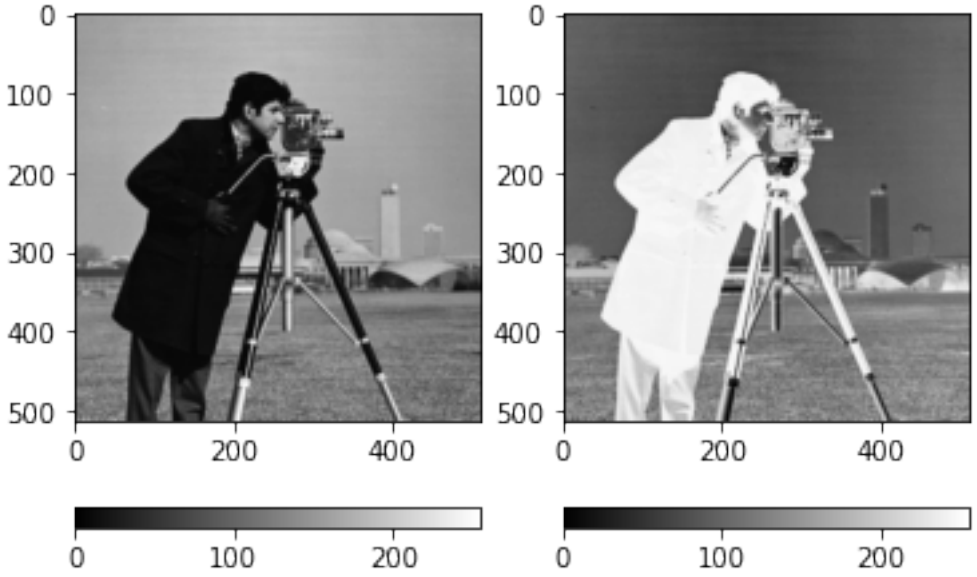
NOT

[ ]:
```
im = data.camera()
imn = invert(im)

fig = plt.figure()
```

```
ax = fig.add_subplot(1, 2, 1)
p = plt.imshow(im, cmap='gray')
c = plt.colorbar(orientation='horizontal')
plt.clim(0, 255)

ax = fig.add_subplot(1, 2, 2)
p = plt.imshow(imn, cmap='gray')
c = plt.colorbar(orientation='horizontal')
plt.clim(0, 255)
```



8 AND

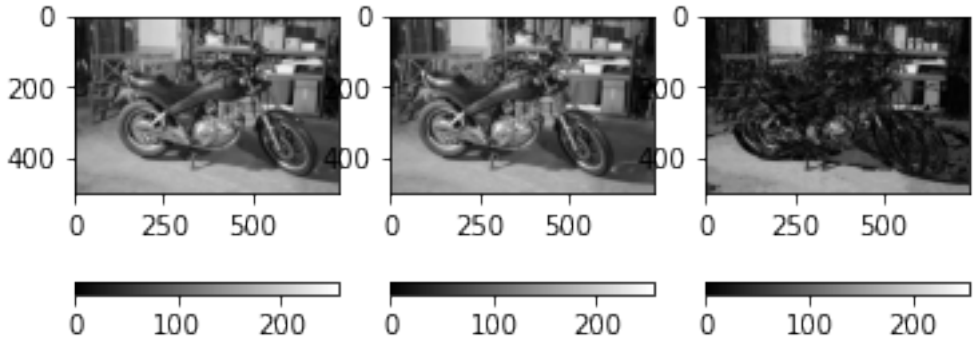
[ ]:
```
im0, im1, _ = data.stereo_motorcycle()
im0 = rgb2gray(im0)
im1 = rgb2gray(im1)
im0 = img_as_ubyte(im0)
im1 = img_as_ubyte(im1)
im = np.bitwise_and(im0, im1)

fig = plt.figure()
ax = fig.add_subplot(1, 3, 1)
p = plt.imshow(im0, cmap='gray')
c = plt.colorbar(orientation='horizontal')
plt.clim(0, 255)

ax = fig.add_subplot(1, 3, 2)
```

```
p = plt.imshow(im1, cmap='gray')
c = plt.colorbar(orientation='horizontal')
plt.clim(0, 255)

ax = fig.add_subplot(1, 3, 3)
p = plt.imshow(im, cmap='gray')
c = plt.colorbar(orientation='horizontal')
plt.clim(0, 255)
```

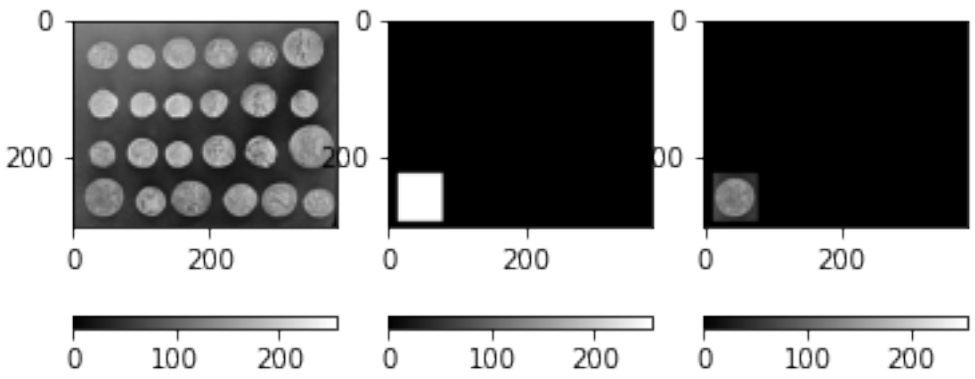


[ ]:
```
im = data.coins()
imm = np.zeros_like(im)
imm[225:295,15:80] = 255
ima = im.copy()
ima[imm==0] = 0

fig = plt.figure()
ax = fig.add_subplot(1, 3, 1)
p = plt.imshow(im, cmap='gray')
c = plt.colorbar(orientation='horizontal')
plt.clim(0, 255)

ax = fig.add_subplot(1, 3, 2)
p = plt.imshow(imm, cmap='gray')
c = plt.colorbar(orientation='horizontal')
plt.clim(0, 255)

ax = fig.add_subplot(1, 3, 3)
p = plt.imshow(ima, cmap='gray')
c = plt.colorbar(orientation='horizontal')
plt.clim(0, 255)
```



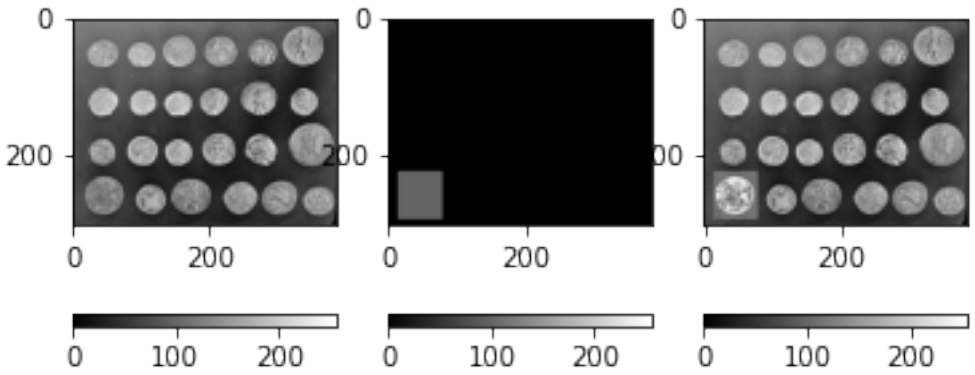
9 OR

[ ]:
```
im = data.coins()
imm = np.zeros_like(im)
imm[225:295,15:80] = 100
imor = np.bitwise_or(im, imm)

fig = plt.figure()
ax = fig.add_subplot(1, 3, 1)
p = plt.imshow(im, cmap='gray')
c = plt.colorbar(orientation='horizontal')
plt.clim(0, 255)

ax = fig.add_subplot(1, 3, 2)
p = plt.imshow(imm, cmap='gray')
c = plt.colorbar(orientation='horizontal')
plt.clim(0, 255)

ax = fig.add_subplot(1, 3, 3)
p = plt.imshow(imor, cmap='gray')
c = plt.colorbar(orientation='horizontal')
plt.clim(0, 255)
```



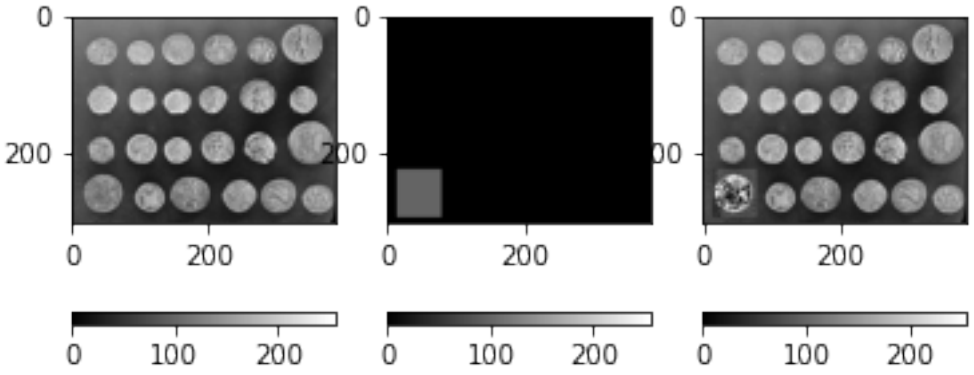
10 XOR

[ ]:
```
im = data.coins()
imm = np.zeros_like(im)
imm[225:295,15:80] = 100
imxor = np.bitwise_xor(im, imm)

fig = plt.figure()
ax = fig.add_subplot(1, 3, 1)
p = plt.imshow(im, cmap='gray')
c = plt.colorbar(orientation='horizontal')
plt.clim(0, 255)

ax = fig.add_subplot(1, 3, 2)
p = plt.imshow(imm, cmap='gray')
c = plt.colorbar(orientation='horizontal')
plt.clim(0, 255)

ax = fig.add_subplot(1, 3, 3)
p = plt.imshow(imxor, cmap='gray')
c = plt.colorbar(orientation='horizontal')
plt.clim(0, 255)
```


0
200
0
200
0
100
200
0
200
0
200
0
100
200
0
200
0
200
0
100
200