

image_filtering

November 28, 2022

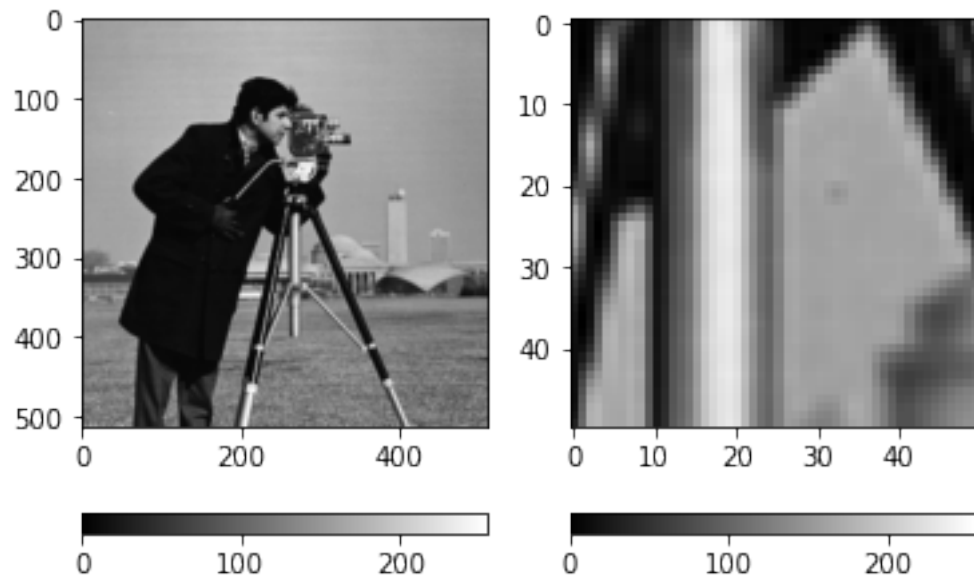
```
[1]: import numpy as np
import matplotlib.pyplot as plt
from skimage.util import random_noise
from skimage.morphology import square
from skimage import io, data, img_as_ubyte, img_as_float
from skimage.filters import gaussian, laplace, rank
from skimage.restoration import denoise_bilateral, denoise_nl_means, estimate_sigma
```

1 Noise - zoom

```
[ ]: im = data.camera()
imn = im[250:300, 250:300]

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)
```

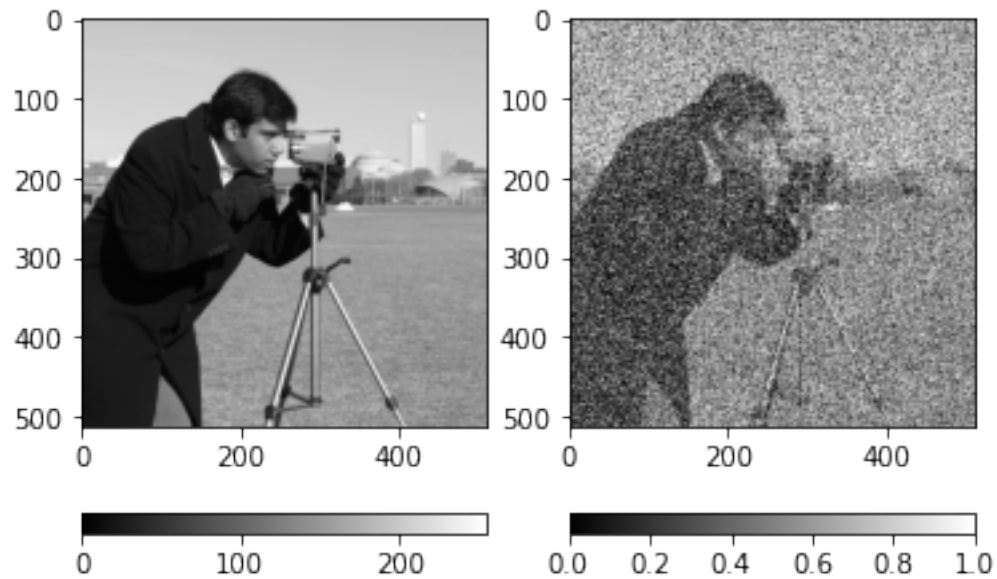


2 Noise - s&p

```
[3]: im = data.camera()
     imn = random_noise(im, seed=42, mode='s&p', amount=0.5, salt_vs_pepper=0.5)

     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, 1)
```

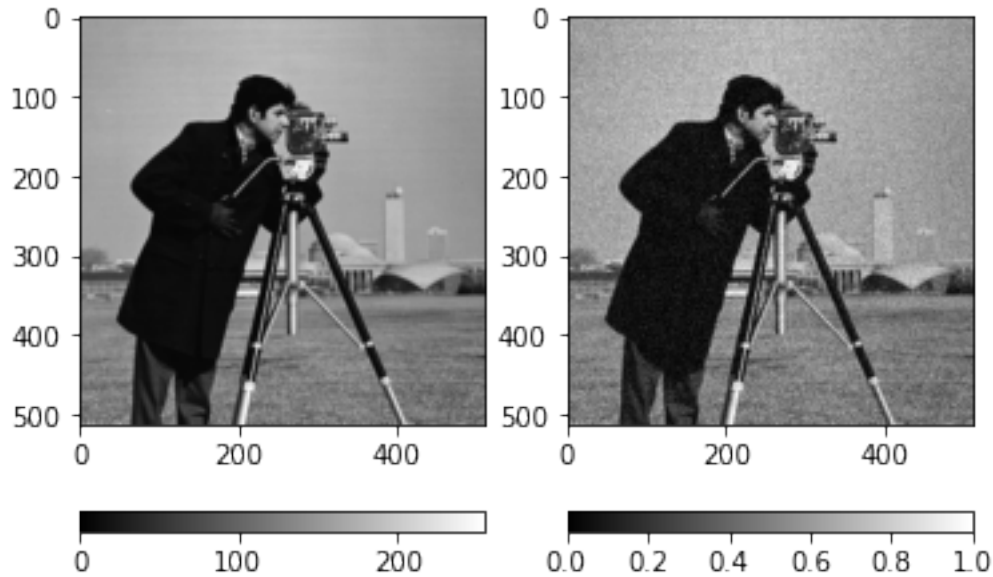


3 Noise - Gaussian

```
[ ]: im = data.camera()
     imn = random_noise(im, seed=42, mode='gaussian')

     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, 1)
```



4 Mean - Gaussian

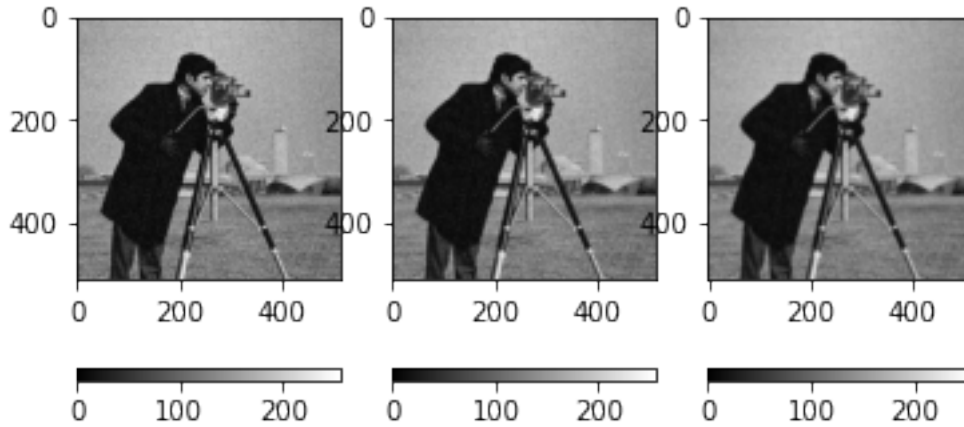
```
[ ]: im = data.camera()
imn = random_noise(im, seed=42, mode='gaussian')
imn = img_as_ubyte(imn)

se = square(3)
imm3 = rank.mean(imn, selem=se)
se = square(5)
imm5 = rank.mean(imn, selem=se)

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

ax = fig.add_subplot(1, 3, 2)
p = plt.imshow(imm3, cmap='gray')
c = plt.colorbar(orientation='horizontal')
plt.clim(0, 255)

ax = fig.add_subplot(1, 3, 3)
p = plt.imshow(imm5, cmap='gray')
c = plt.colorbar(orientation='horizontal')
plt.clim(0, 255)
```



5 Mean - Gaussian & SP

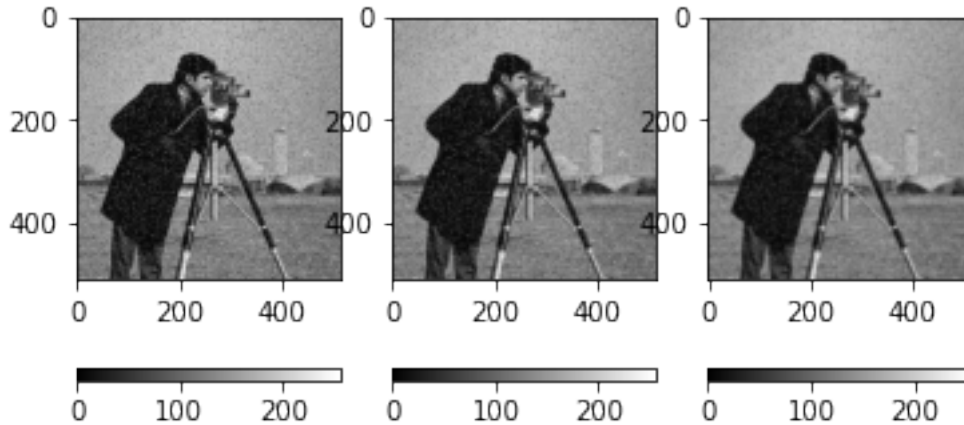
```
[ ]: im = data.camera()
      imn = random_noise(im, seed=42, mode='s&p', amount=0.1, salt_vs_pepper=0.5)
      imn = img_as_ubyte(imn)

      se = square(3)
      imm3 = rank.mean(imn, selem=se)
      se = square(5)
      imm5 = rank.mean(imn, selem=se)

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

      ax = fig.add_subplot(1, 3, 2)
      p = plt.imshow(imm3, cmap='gray')
      c = plt.colorbar(orientation='horizontal')
      plt.clim(0, 255)

      ax = fig.add_subplot(1, 3, 3)
      p = plt.imshow(imm5, cmap='gray')
      c = plt.colorbar(orientation='horizontal')
      plt.clim(0, 255)
```



6 Median - Gaussian & SP

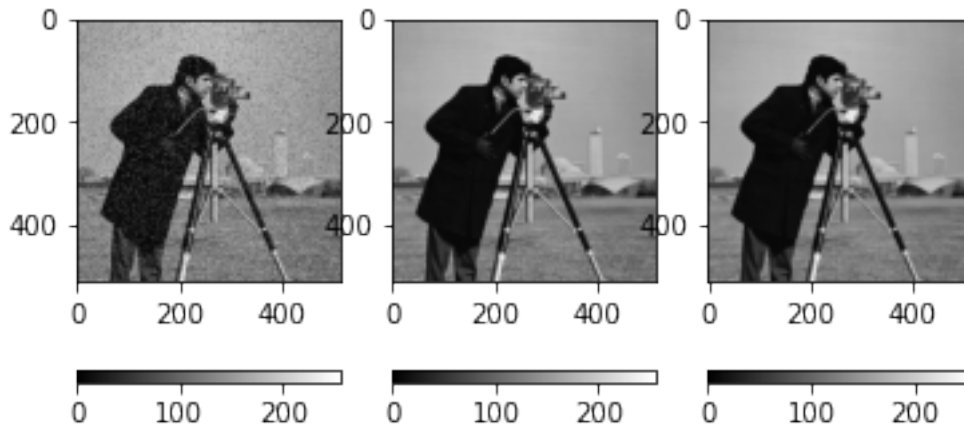
```
[ ]: im = data.camera()
      imn = random_noise(im, seed=42, mode='s&p', amount=0.1, salt_vs_pepper=0.5)
      imn = img_as_ubyte(imn)

      se = square(3)
      imm3 = rank.median(imn, selem=se)
      se = square(5)
      imm5 = rank.median(imn, selem=se)

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

      ax = fig.add_subplot(1, 3, 2)
      p = plt.imshow(imm3, cmap='gray')
      c = plt.colorbar(orientation='horizontal')
      plt.clim(0, 255)

      ax = fig.add_subplot(1, 3, 3)
      p = plt.imshow(imm5, cmap='gray')
      c = plt.colorbar(orientation='horizontal')
      plt.clim(0, 255)
```



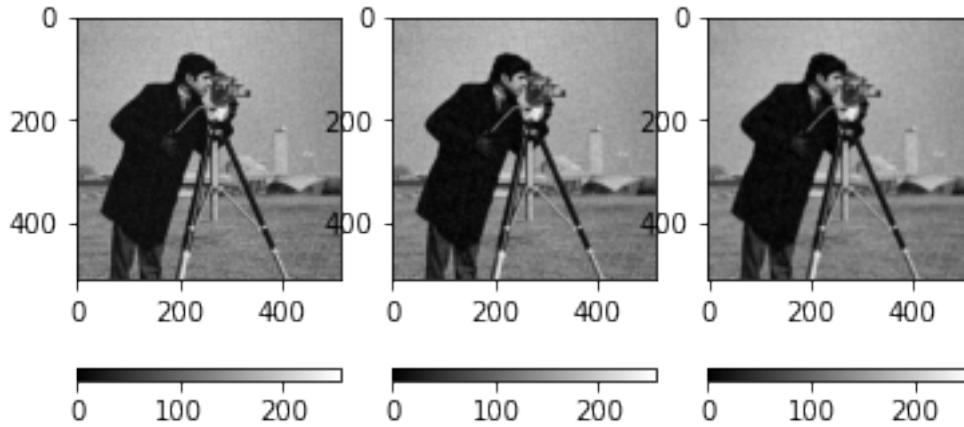
```
[ ]: im = data.camera()
immn = random_noise(im, seed=42, mode='gaussian')
immn = img_as_ubyte(imn)

se = square(3)
imm3 = rank.median(imn, selem=se)
se = square(5)
imm5 = rank.median(imn, selem=se)

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

ax = fig.add_subplot(1, 3, 2)
p = plt.imshow(imm3, cmap='gray')
c = plt.colorbar(orientation='horizontal')
plt.clim(0, 255)

ax = fig.add_subplot(1, 3, 3)
p = plt.imshow(imm5, cmap='gray')
c = plt.colorbar(orientation='horizontal')
plt.clim(0, 255)
```



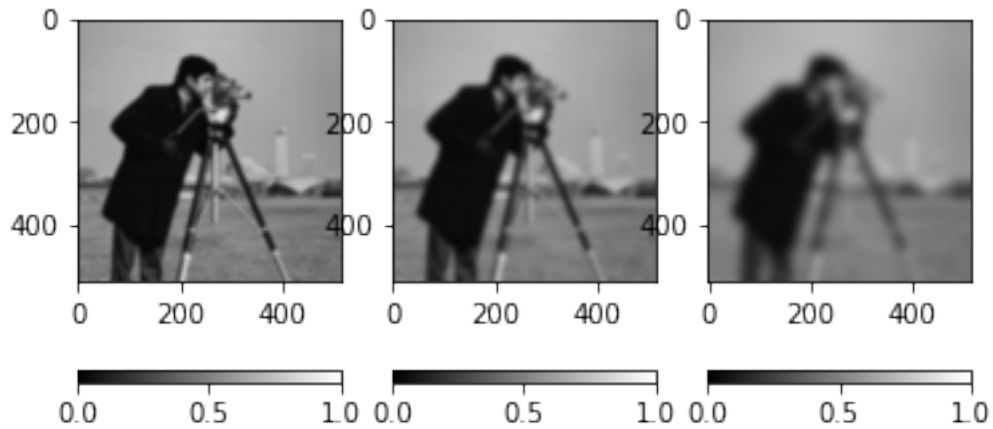
7 Gaussian

```
[ ]: im = data.camera()
img3 = gaussian(im, sigma=3)
img5 = gaussian(im, sigma=5)
img10 = gaussian(im, sigma=10)

fig = plt.figure()
ax = fig.add_subplot(1, 3, 1)
p = plt.imshow(img3, cmap='gray')
c = plt.colorbar(orientation='horizontal')
plt.clim(0, 1)

ax = fig.add_subplot(1, 3, 2)
p = plt.imshow(img5, cmap='gray')
c = plt.colorbar(orientation='horizontal')
plt.clim(0, 1)

ax = fig.add_subplot(1, 3, 3)
p = plt.imshow(img10, cmap='gray')
c = plt.colorbar(orientation='horizontal')
plt.clim(0, 1)
```

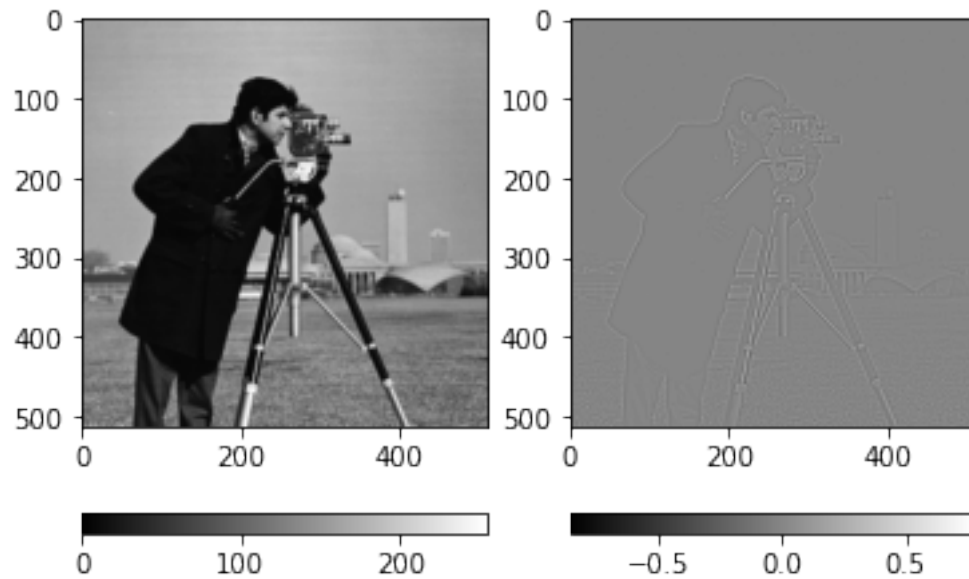



8 Laplacian

```
[ ]: im = data.camera()
imf = laplace(im, ksize=3)

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')
```

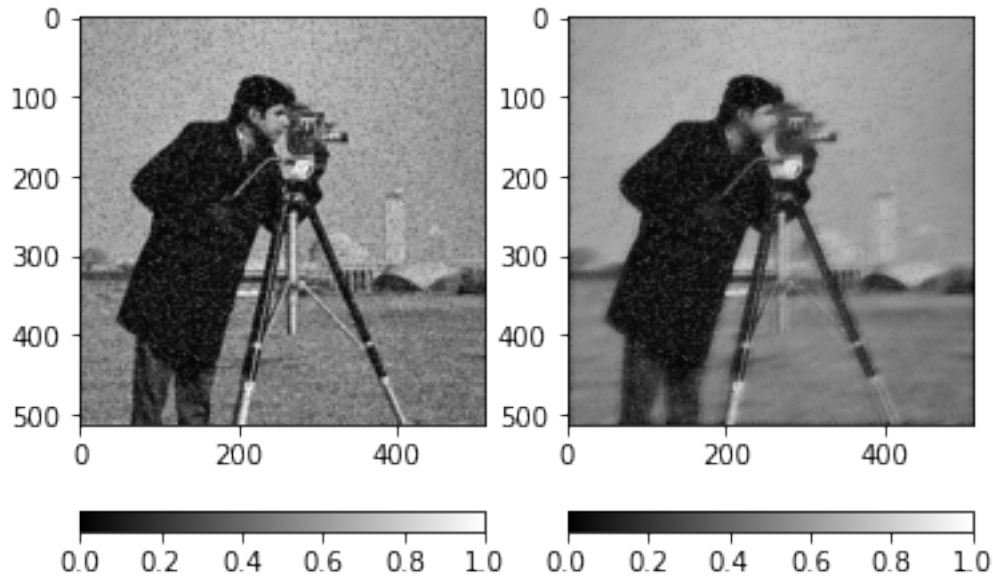


9 Bilateral

```
[ ]: im = data.camera()
seed = 42
imn = random_noise(im, seed=42, mode='s&p', amount=0.1, salt_vs_pepper=0.5)
imf = denoise_bilateral(imn, sigma_spatial=3)

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

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



10 Non-local Means

```
[ ]: im = img_as_float(data.camera())
     imn = random_noise(im, seed=42, mode='gaussian')
     imf = gaussian(imn, sigma=3)

     sigma_est = np.mean(estimate_sigma(imn, multichannel=True))
     imd = denoise_nl_means(imn, h=0.6 * sigma_est, sigma=sigma_est, fast_mode=True)

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

     ax = fig.add_subplot(1, 3, 2)
     p = plt.imshow(imf, 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)
```

