

The background is a dark blue gradient with a subtle pattern of white dots. Overlaid on this are several faint, light blue geometric elements: concentric circles, arcs, and degree markings. A large arc on the left side is marked with degrees from 140 to 260 in increments of 10. Other smaller arcs and circles are scattered across the frame, some with arrows indicating a clockwise direction.

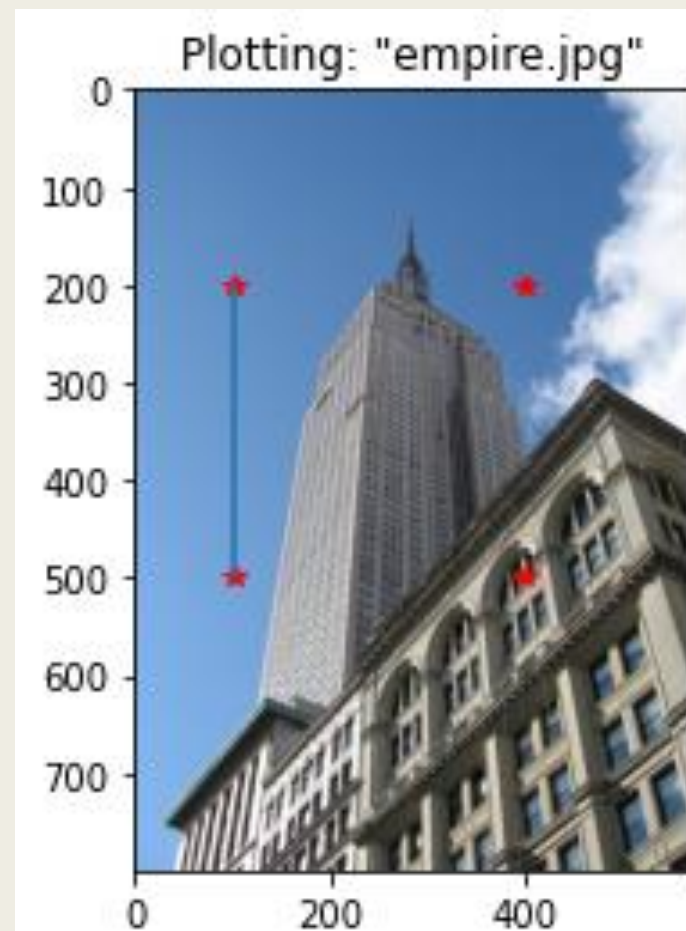
# 電腦視覺實務期中考重點

# 考試例題

1. `ch01_fig1-2_matplot.py`
2. `ch01_fig1-3_contour.py`
3. `ch01_fig1-5_graylevel-transforms.py`
4. `ch01_fig1-6(7)_histeq.py`
5. `ch01_fig1-8_pca_graylevel.py`
6. `ch01_fig1-12_morphology.py`
7. `ch02_sift.py`
8. `ch02_sift_match.py`
9. `ch2_harris_corners.py`
10. `ch2_harris_matching.py`

# 1.ch01\_fig1-2\_matplotlib.py

- Matplotlib應用
- 繪製圖像、點和線



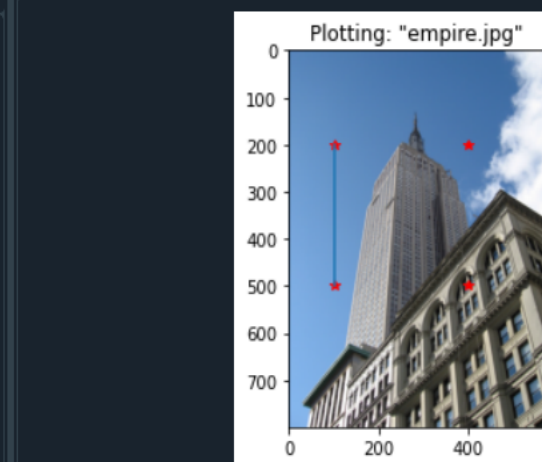
```
1 from PIL import Image
2 from pylab import *
3
4 im = array(Image.open('../data/empire.jpg'))
5
6 imshow(im)
7
8 x = [100, 100, 400, 400]    #x座標的點
9 y = [200, 500, 200, 500]   #y座標的點
10 plot(x, y, 'r*')           #用紅色繪製點
11
12 plot(x[:2], y[:2])
13
14 #axis('off')
15
16 title('Plotting: "empire.jpg"')
17 show()
18
```

表 1-2：用 PyLab 庫繪圖的基本線型格式命令

線型	
'-'	實線
'--'	虛線
'.'	點線

表 1-3：用 PyLab 庫繪圖的基本繪製標記格式命令

標記	
'.'	點
'o'	圓圈
's'	正方形
'*'	星形
'+'	加號
'x'	叉號

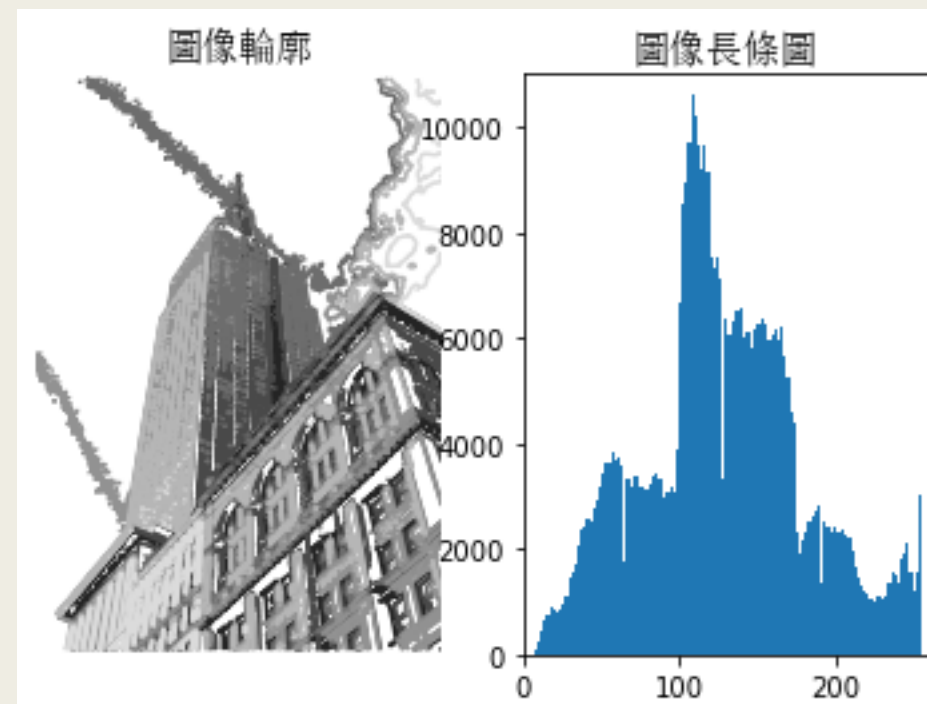


```
In [7]: runfile('C:/Users/20200925/Desktop/KSU/109-1/電腦視覺實務/pcv-book-code-master/ch01/ch01_fig1-2_matplot.py', wdir='C:/Users/20200925/Desktop/KSU/109-1/電腦視覺實務/pcv-book-code-master/ch01')
```

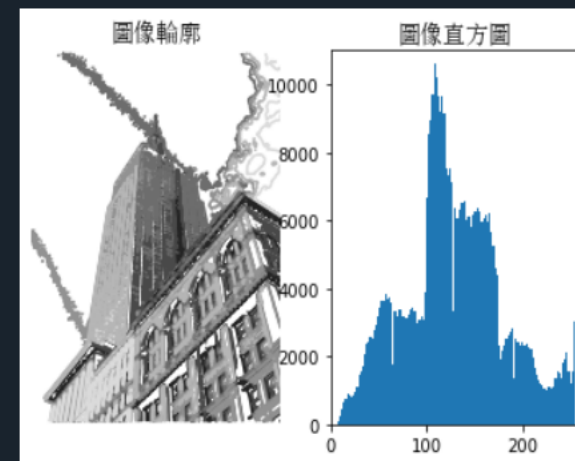
```
In [8]:
```

## 2.ch01\_fig1-3\_contour.py

- 圖像輪廓和長條圖
- 先將圖片轉為灰階圖，再繪製長條圖



```
1  # -*- coding: utf-8 -*-
2  from PIL import Image
3  from pylab import *
4
5  # 添加中文字体支持
6  from matplotlib.font_manager import FontProperties
7  font = FontProperties(fname=r"c:\windows\fonts\SimSun.ttc", size=14)
8  im = array(Image.open('../data/empire.jpg').convert('L')) # 打開圖像轉為灰階
9
10 figure()           #新建一個圖像
11 subplot(121)
12 gray()             #不使用顏色資訊
13 contour(im, origin='image') # 在原點的左上角顯示輪廓圖像
14 axis('equal')
15 axis('off')
16 title(u'圖像輪廓', fontproperties=font)
17
18 subplot(122)
19 hist(im.flatten(), 128) #長條圖寬度
20 title(u'圖像直方圖', fontproperties=font)
21 plt.xlim([0,260])      #設定x軸最大值
22 plt.ylim([0,11000])    #設定y軸最大值
23
24 show()
```



Variable explorer Help Plots Files

Console 5/A ×

```
ch01_fig1-3_contour.py', wdir='C:/Users/20200925/
Desktop/KSU/109-1/ 電腦視覺實務/pcv-book-code-master/
Project')
```

Figures now render in the Plots pane by default. To make them also appear inline in the Console, uncheck "Mute Inline Plotting" under the Plots pane options menu.

In [2]:

IPython console History



# 3.ch01\_fig1-5\_graylevel-transforms.py

- 圖像陣列表示
- 灰度變換

$$f(x) = 255 - x$$



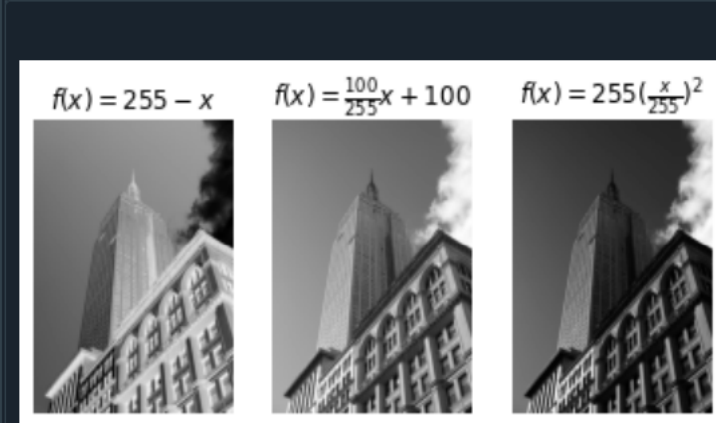
$$f(x) = \frac{100}{255}x + 100$$



$$f(x) = 255\left(\frac{x}{255}\right)^2$$



```
5
6 im = array(Image.open('../data/empire.jpg').convert('L'))
7 print (int(im.min()), int(im.max()))
8
9 im2 = 255 - im #對圖像進行反相處理
10 print (int(im2.min()), int(im2.max()))
11
12 im3 = (100.0/255) * im + 100 #將圖像圖元值變換到 100...200 區間
13 print (int(im3.min()), int(im3.max()))
14
15 im4 = 255.0 * (im/255.0)**2 #對圖像圖元值求平方後得到的圖像
16 print (int(im4.min()), int(im4.max()))
17
18 figure()
19 gray()
20 subplot(1, 3, 1)
21 imshow(im2)
22 axis('off')
23 title(r'$f(x)=255-x$')
24
25 subplot(1, 3, 2)
26 imshow(im3)
27 axis('off')
28 title(r'$f(x)=\frac{100}{255}x+100$')
29
30 subplot(1, 3, 3)
31 imshow(im4)
32 axis('off')
33 title(r'$f(x)=255(\frac{x}{255})^2$')
34 show()
```



```
ch01_fig1-5_graylevel-transforms.py', wdir='C:/
Users/20200925/Desktop/KSU/109-1/電腦視覺實務/pcv-
book-code-master/Project')
```

```
3 255
0 252
101 200
0 255
```

Figures now render in the Plots pane by default. To make them also appear inline in the Console, uncheck "Mute Inline Plotting" under the Plots pane options menu.



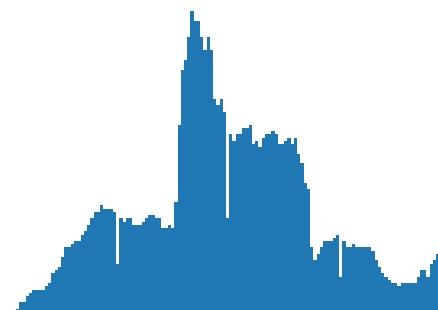
# 4.ch01\_fig1-6(7)\_histeq.py

## ■ 長條圖均衡化

原始圖像



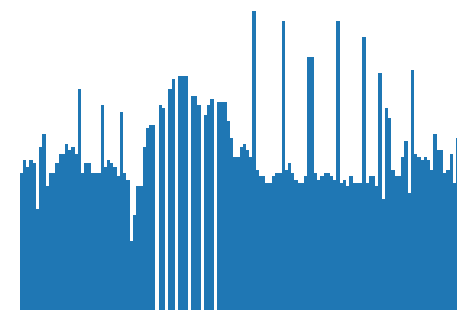
原始直方圖



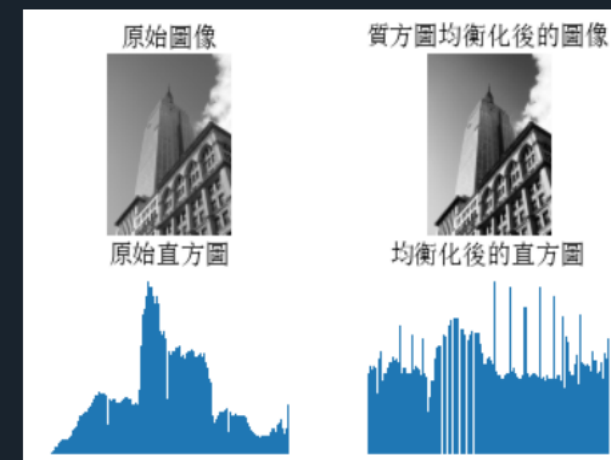
質方圖均衡化後的圖像



均衡化後的直方圖



```
11 #im = array(Image.open('../data/AquaTermi_lowcontrast.JPG').convert('L'))
12 im2, cdf = imtools.histeq(im)
13
14 figure()
15 subplot(2, 2, 1)
16 axis('off')
17 gray()
18 title(u'原始圖像', fontproperties=font)
19 imshow(im)
20
21 subplot(2, 2, 2)
22 axis('off')
23 title(u'質方圖均衡化後的圖像', fontproperties=font)
24 imshow(im2)
25
26 subplot(2, 2, 3)
27 axis('off')
28 title(u'原始直方圖', fontproperties=font)
29 #hist(im.flatten(), 128, cumulative=True, density=True)
30 hist(im.flatten(), 128, density=True)
31
32 subplot(2, 2, 4)
33 axis('off')
34 title(u'均衡化後的直方圖', fontproperties=font)
35 #hist(im2.flatten(), 128, cumulative=True, density=True)
36 hist(im2.flatten(), 128, density=True)
37
38 show()
39
```



Variable explorer Help Plots Files

Console 7/A ×

The arguments will be removed in a future version of numpy.

```
imhist, bins =  
histogram(im.flatten(), nbr_bins, normed=True)
```

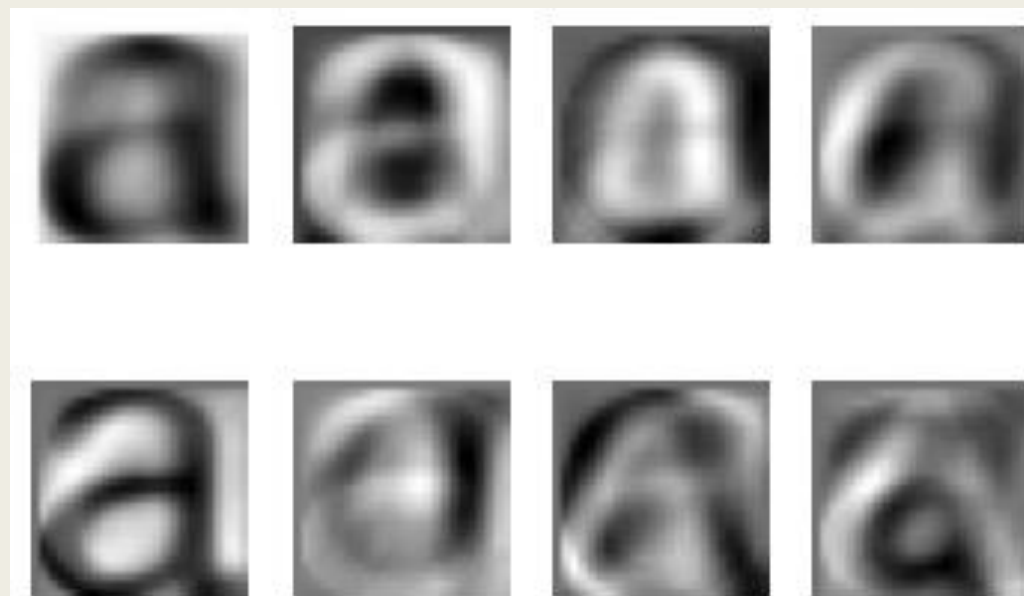
Figures now render in the Plots pane by default. To make them also appear inline in the Console, uncheck "Mute Inline Plotting" under the Plots pane options menu.

In [2]:

IPython console History

# 5.ch01\_fig1-8\_pca\_graylevel.py

## ■ 圖像的主成分分析 ( PCA )



```
14 m, n = im.shape[:2] # get the size of the images
15 imnbr = len(imlist) # get the number of images
16 print ("The number of images is %d" % imnbr)
17
18 # Create matrix to store all flattened images
19 immatrix = array([array(Image.open(imname)).flatten() for imname in imlist], 'f')
20
21 # PCA降维
22 V, S, immean = pca.pca(immatrix)
23
24 # 保存均值和主成分
25 f = open('../data/font_pca_modes.pkl', 'wb')
26 pickle.dump(immean, f)
27 pickle.dump(V, f)
28 f.close()
29
30 # Show the images (mean and 7 first modes)
31 # This gives figure 1-8 (p15) in the book.
32 figure()
33 gray()
34 subplot(2, 4, 1)
35 axis('off')
36 imshow(immean.reshape(m, n))
37 for i in range(7):
38     subplot(2, 4, i+2)
39     imshow(V[i].reshape(m, n))
40     axis('off')
41 show()
42
```



Variable explorer Help Plots Files

Console 8/A

20200925/Desktop/KSU/109-1/電腦視覺實務/pcv-book-code-master/ch01')

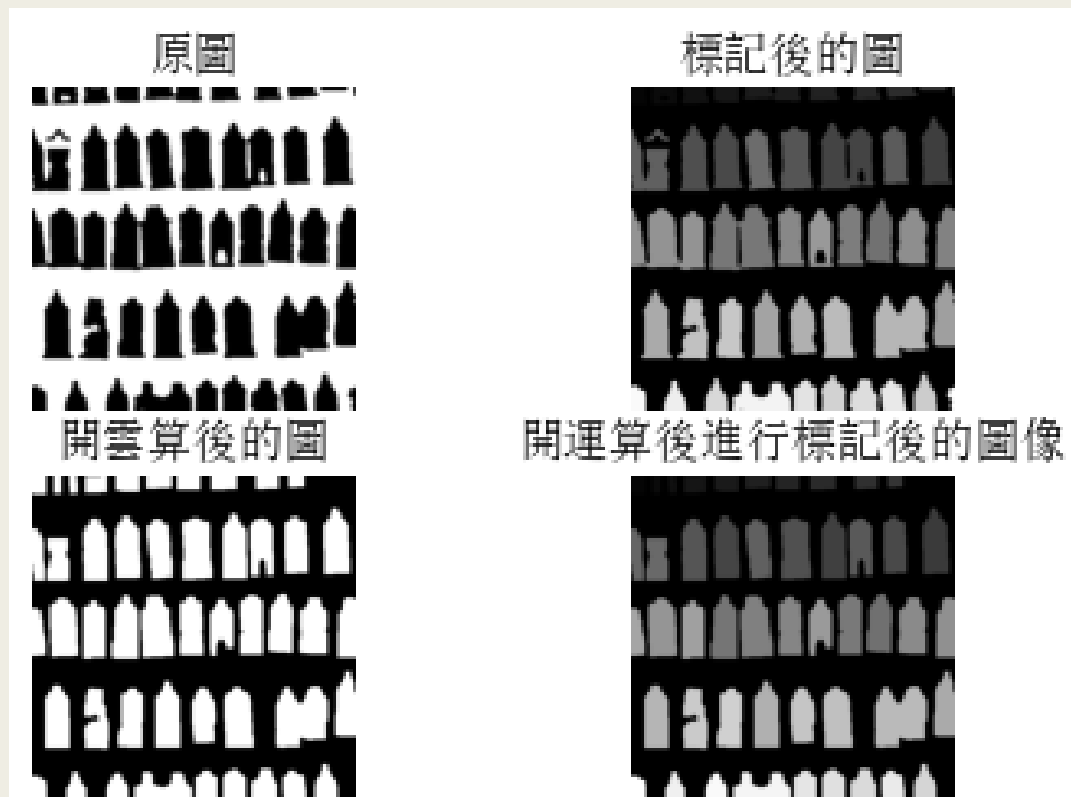
The number of images is 2359

Figures now render in the Plots pane by default. To make them also appear inline in the Console, uncheck "Mute Inline Plotting" under the Plots pane options menu.

In [2]:

# 6.ch01\_fig1-12\_morphology.py

## ■ 形態學：物件計數



Spyder (Python 3.7)

File Edit Source Run Debug Consoles Projects Tools View Help

C:\Users\20200925\Desktop\KSU\109-1\電腦視覺實務\pcv-book-code-master\Project

C:\Users\20200925\Desktop\KSU\109-1\電腦視覺實務\pcv-book-code-master\Project\ch01\_fig1-12\_morphology.py

```
14 figure()
15 gray()
16 im = array(Image.open('../data/houses.png').convert('L'))
17 subplot(221)
18 imshow(im)
19 axis('off')
20 title(u'原圖', fontproperties=font)
21 im = (im < 128)
22
23 labels, nbr_objects = measurements_label(im)
24 print ("Number of objects:", nbr_objects)
25
26 subplot(222)
27 imshow(labels)
28 axis('off')
29 title(u'標記後的圖', fontproperties=font)
30
31 # morphology - opening to separate objects better
32 im_open = morphology.binary_opening(im, ones((9, 5)), iterations=2)
33 subplot(223)
34 imshow(im_open)
35 axis('off')
36 title(u'開雲算後的圖', fontproperties=font)
37
38 labels_open, nbr_objects_open = measurements_label(im_open)
39 print ("Number of objects:", nbr_objects_open)
40
41 subplot(224)
42 imshow(labels_open)
43 axis('off')
44 title(u'開運算後進行標記後的圖像', fontproperties=font)
```

Figure 1: Morphology processing results. The figure consists of four subplots arranged in a 2x2 grid. The top-left subplot shows the original grayscale image of a row of houses. The top-right subplot shows the image after binary thresholding (im < 128). The bottom-left subplot shows the image after binary opening (im\_open). The bottom-right subplot shows the image after binary opening and labeling (labels\_open). The subplots are titled in Chinese: '原圖' (Original Image), '標記後的圖' (Image after labeling), '開雲算後的圖' (Image after opening), and '開運算後進行標記後的圖像' (Image after opening and labeling).

Console 9/A

```
code-master/Project')
Number of objects: 45
Number of objects: 48
```

Figures now render in the Plots pane by default. To make them also appear inline in the Console, uncheck "Mute Inline Plotting" under the Plots pane options menu.

In [2]:

IPython console History

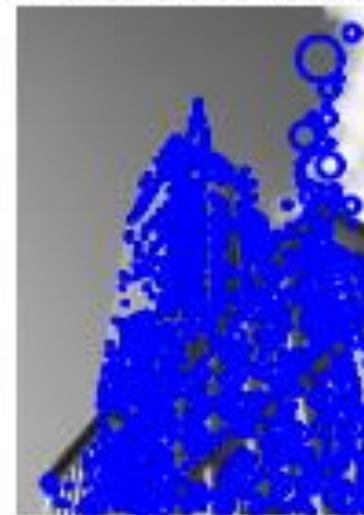
LSP Python: ready conda: Steven (Python 3.7.7) Line 1, Col 1 UTF-8 LF RW Mem 45%



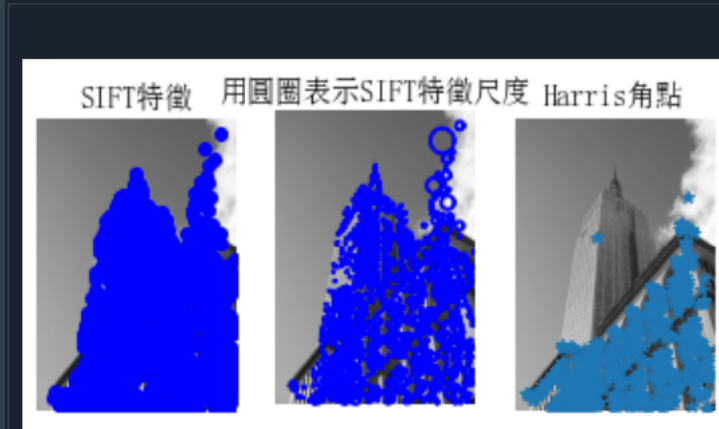
# 7.ch02\_sift.py

## ■ SIFT特徵尺度

SIFT特徵 用圓圈表示SIFT特徵尺度 Harris角點



```
7 # 添加中文字体支持
8 from matplotlib.font_manager import FontProperties
9 font = FontProperties(fname=r"c:\windows\fonts\SimSun.ttc", size=14)
10
11 imname = '../data/empire.jpg'
12 im = array(Image.open(imname).convert('L'))
13 sift.process_image(imname, 'empire.sift')
14 l1, d1 = sift.read_features_from_file('empire.sift')
15
16 figure()
17 gray()
18 subplot(131)
19 sift.plot_features(im, l1, circle=False)
20 title(u'SIFT特徵', fontproperties=font)
21 subplot(132)
22 sift.plot_features(im, l1, circle=True)
23 title(u'用圓圈表示SIFT特徵尺度', fontproperties=font)
24
25 # 检测harris角点
26 harrisim = harris.compute_harris_response(im)
27
28 subplot(133)
29 filtered_coords = harris.get_harris_points(harrisim, 6, 0.1)
30 imshow(im)
31 plot([p[1] for p in filtered_coords], [p[0] for p in filtered_coords], '*')
32 axis('off')
33 title(u'Harris角點', fontproperties=font)
34
35 show()
```



Variable explorer Help Plots Files

Console 10/A

processed tmp.pgm to empire.sift

'sift' 不是內部或外部命令、可執行的程式或批次檔。

Figures now render in the Plots pane by default. To make them also appear inline in the Console, uncheck "Mute Inline Plotting" under the Plots pane options menu.

In [2]:

IPython console History

# 8.ch02\_sift\_match.py(HW3)

- 在相似圖像間尋找對應點



```
9 else:
10     im1f = '../data/com01.jpg'
11     im2f = '../data/com02.jpg'
12     # im1f = '../data/crans_1_small.jpg'
13     # im2f = '../data/crans_2_small.jpg'
14     # im1f = '../data/climbing_1_small.jpg'
15     # im2f = '../data/climbing_2_small.jpg'
16 im1 = array(Image.open(im1f))
17 im2 = array(Image.open(im2f))
18
19 #sift.process_image(im1f, 'out_sift_1.txt')
20 l1, d1 = sift.read_features_from_file('out_sift_1.txt')
21 figure()
22 gray()
23 subplot(121)
24 sift.plot_features(im1, l1, circle=False)
25
26 #sift.process_image(im2f, 'out_sift_2.txt')
27 l2, d2 = sift.read_features_from_file('out_sift_2.txt')
28 subplot(122)
29 sift.plot_features(im2, l2, circle=False)
30
31 #matches = sift.match(d1, d2)
32 matches = sift.match_twosided(d1, d2)
33 print ('{} matches'.format(len(matches.nonzero()[0])))
34
35 figure()
36 gray()
37 sift.plot_matches(im1, im2, l1, l2, matches, show_below=True)
38 show()
```



Variable explorer Help Plots Files

Console 11/A ×

```
Desktop/KSU/109-1/電腦視覺實務/pcv-book-code-
master/ch02')
1375 matches
```

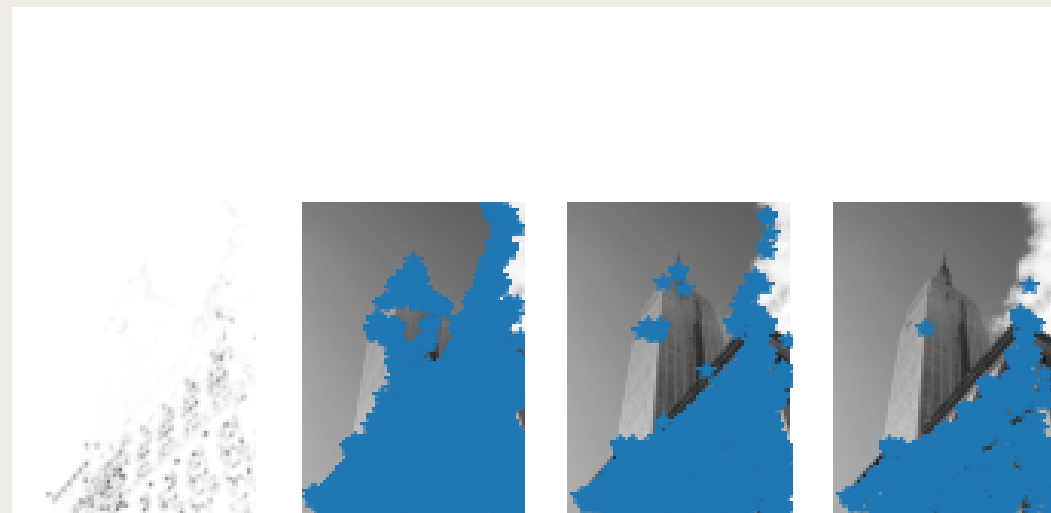
Figures now render in the Plots pane by default. To make them also appear inline in the Console, uncheck "Mute Inline Plotting" under the Plots pane options menu.

In [2]:

IPython console History

# 9.ch2\_harris\_corners.py

## ■ Harris角點檢測器



Spyder (Python 3.7)

File Edit View Source Run Debug Consoles Projects Tools View Help

C:\Users\20200925\Desktop\KSU\109-1\電腦視覺實務\pcv-book-code-master\Project

C:\Users\20200925\Desktop\KSU\109-1\電腦視覺實務\pcv-book-code-master\Project\ch2\_harris\_corners.py

ch2\_harris\_corners.py

```
7 Example of detecting Harris corner points (Figure 2-1 in the book).
8 """
9
10 # 讀入圖像
11 im = array(Image.open('../data/empire.jpg').convert('L'))
12
13 # 檢測harris角點
14 harrisim = harris.compute_harris_response(im)
15
16 # Harris回應函數
17 harrisim1 = 255 - harrisim
18
19 figure()
20 gray()
21
22 #畫出Harris回應圖
23 subplot(141)
24 imshow(harrisim1)
25 print(harrisim1.shape)
26 axis('off')
27 axis('equal')
28
29 threshold = [0.01, 0.05, 0.1]
30 for i, thres in enumerate(threshold):
31     filtered_coords = harris.get_harris_points(harrisim, 6, thres)
32     subplot(1, 4, i+2)
33     imshow(im)
34     print(im.shape)
35     plot([p[1] for p in filtered_coords], [p[0] for p in filtered_coords], '*')
36     axis('off')
```

Variable explorer Help Plots Files

Console 11/A

```
In [3]: runfile('C:/Users/20200925/Desktop/KSU/109-1/電腦視覺實務/pcv-book-code-master/Project/ch2_harris_corners.py', wdir='C:/Users/20200925/Desktop/KSU/109-1/電腦視覺實務/pcv-book-code-master/Project')
(800, 569)
(800, 569)
(800, 569)
(800, 569)

In [4]:
```

IPython console History

LSP Python: ready conda: Steven (Python 3.7.7) Line 1, Col 24 UTF-8 LF RW Mem 48%

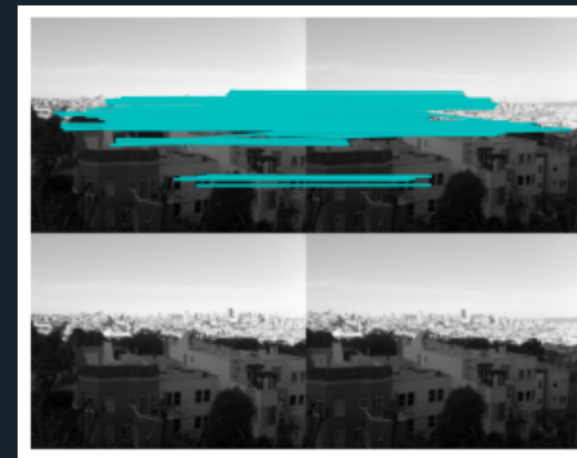


# 10.ch2\_harris\_matching.py

## ■ Harris角點檢測器



```
10  """
11
12  # Figure 2-2上面的图
13  #im1 = array(Image.open("../data/crans_1_small.jpg").convert("L"))
14  #im2 = array(Image.open("../data/crans_2_small.jpg").convert("L"))
15
16  # Figure 2-2下面的图
17  im1 = array(Image.open("../data/sf_view1.jpg").convert("L"))
18  im2 = array(Image.open("../data/sf_view2.jpg").convert("L"))
19
20  # resize to make matching faster
21  im1 = imresize(im1, (int(im1.shape[1]/2), int(im1.shape[0]/2)))
22  im2 = imresize(im2, (int(im2.shape[1]/2), int(im2.shape[0]/2)))
23
24  wid = 5
25  harrisim = harris.compute_harris_response(im1, 5)
26  filtered_coords1 = harris.get_harris_points(harrisim, wid+1)
27  d1 = harris.get_descriptors(im1, filtered_coords1, wid)
28
29  harrisim = harris.compute_harris_response(im2, 5)
30  filtered_coords2 = harris.get_harris_points(harrisim, wid+1)
31  d2 = harris.get_descriptors(im2, filtered_coords2, wid)
32
33  print('starting matching')
34  matches = harris.match_twosided(d1, d2)
35
36  figure()
37  gray()
38  harris.plot_matches(im1, im2, filtered_coords1, filtered_coords2, matches)
39  show()
```



Variable explorer Help Plots Files

Console 11/A ×

```
In [5]: runfile('C:/Users/20200925/Desktop/KSU/
109-1/ 電腦視覺實務/pcv-book-code-master/Project/
ch2_harris_matching.py', wdir='C:/Users/20200925/
Desktop/KSU/109-1/ 電腦視覺實務/pcv-book-code-
master/Project')
starting matching
```

```
In [6]:
```

IPython console History

LSP Python: ready

conda: Steven (Python 3.7.7)

Line 1, Col 1

UTF-8

LF

RW

Mem 47%

# 作業練習

1. ch01\_P002-003\_PIL.py

Spyder (Python 3.7)

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\20200925\Desktop\KSU\109-1\電腦視覺實務\pcv-book-code-master\ch01

C:\Users\20200925\Desktop\KSU\109-1\電腦視覺實務\pcv-book-code-master\ch01\_P002-003\_PIL.py

ch02\_sift\_match.py · ch02 × ch2\_harris\_corners.py × ch2\_harris\_matching.py × ch03\_fig3-1\_warping.py × ch03\_fig3-2\_placing.py × warp.py × ch01\_P002-003\_PIL.py ×

```
10 # 顯示原圖
11 pil_im = Image.open('../data/empire.jpg')
12 print (pil_im.mode, pil_im.size, pil_im.format)
13 subplot(231)
14 title(u'原圖', fontproperties=font)
15 axis('off')
16 imshow(pil_im)
17
18 # 顯示灰度圖
19 pil_im = Image.open('../data/empire.jpg').convert('L')
20 gray()
21 subplot(232)
22 title(u'灰度圖', fontproperties=font)
23 axis('off')
24 imshow(pil_im)
25
26 # 拷貝粘貼區域
27 pil_im = Image.open('../data/empire.jpg')
28 box = (100,100,400,400)
29 region = pil_im.crop(box)
30 region = region.transpose(Image.ROTATE_180)
31 pil_im.paste(region,box)
32 subplot(233)
33 title(u'拷貝粘貼區域', fontproperties=font)
34 axis('off')
35 imshow(pil_im)
36
37 # 縮略圖
38 pil_im = Image.open('../data/empire.jpg')
39 size = 128, 128
```

原圖 灰度圖 拷貝粘貼區域

縮略圖 調整尺寸後的圖像 旋轉45°後的圖像

Variable explorer Help Plots Files

Console 12/A ×

```
In [1]: runfile('C:/Users/20200925/Desktop/KSU/109-1/電腦視覺實務/pcv-book-code-master/ch01/ch01_P002-003_PIL.py', wdir='C:/Users/20200925/Desktop/KSU/109-1/電腦視覺實務/pcv-book-code-master/ch01')
RGB (91, 128) JPEG
(91, 128)
(128, 128)
```

Figures now render in the Plots pane by default. To make them also appear inline in the Console, uncheck "Mute Inline Plotting" under the Plots

IPython console History

LSP Python: ready conda: Steven (Python 3.7.7) Line 67, Col 1 UTF-8 LF RW Mem 49%

Spyder (Python 3.7)

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\20200925\Desktop\KSU\109-1\電腦視覺實務\pcv-book-code-master\ch01

C:\Users\20200925\Desktop\KSU\109-1\電腦視覺實務\pcv-book-code-master\ch01\_P002-003\_PIL.py

ch02\_sift\_match.py - ch02 × ch2\_harris\_corners.py × ch2\_harris\_matching.py × ch03\_fig3-1\_warping.py × ch03\_fig3-2\_placing.py × warp.py × ch01\_P002-003\_PIL.py ×

```
37 # 縮略圖
38 pil_im = Image.open('../data/empire.jpg')
39 size = 128, 128
40 pil_im.thumbnail(size)
41 print (pil_im.size)
42 subplot(234)
43 title(u'縮略圖', fontproperties=font)
44 axis('off')
45 imshow(pil_im)
46 pil_im.save('../data/empire.jpg') #保存縮略圖
47
48 # 調整圖像尺寸
49 pil_im = Image.open('../data/empire.jpg')
50 pil_im = pil_im.resize(size)
51 print (pil_im.size)
52 subplot(235)
53 title(u'調整尺寸後的圖像', fontproperties=font)
54 axis('off')
55 imshow(pil_im)
56
57 # 旋轉圖像45°
58 pil_im = Image.open('../data/empire.jpg')
59 pil_im = pil_im.rotate(45)
60 subplot(236)
61 title(u'旋轉45°後的圖像', fontproperties=font)
62 axis('off')
63 imshow(pil_im)
64
65 show()
66
```

原圖 灰度圖 拷貝粘貼區域

縮略圖 調整尺寸後的圖像 旋轉45°後的圖像

Variable explorer Help Plots Files

Console 12/A ×

```
In [1]: runfile('C:/Users/20200925/Desktop/KSU/109-1/電腦視覺實務/pcv-book-code-master/ch01/ch01_P002-003_PIL.py', wdir='C:/Users/20200925/Desktop/KSU/109-1/電腦視覺實務/pcv-book-code-master/ch01')
RGB (91, 128) JPEG
(91, 128)
(128, 128)
```

Figures now render in the Plots pane by default. To make them also appear inline in the Console, uncheck "Mute Inline Plotting" under the Plots

IPython console History