

Digital Image Processing 2024 Fall

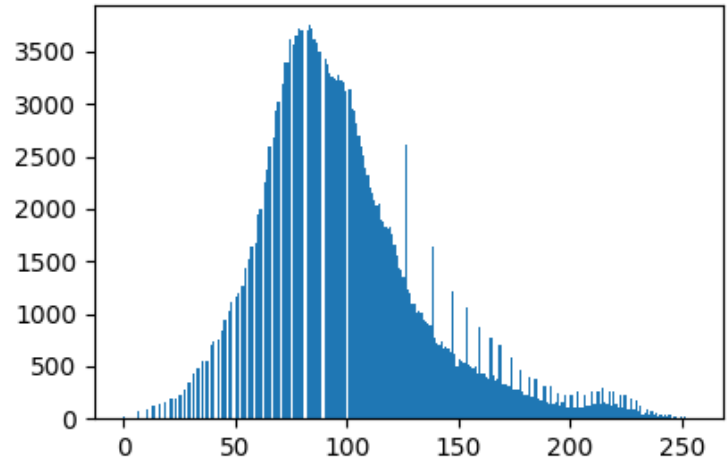
HW3-1

1. Please depict the histogram and graph of the assigned image “aerial_view.tif”, and print out the source code? (10)

Original Image



Histogram



Source code:

```
1 import cv2
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import bisect
```

```
1 # 讀取圖片
2 img = cv2.imread('aerial_view.tif', cv2.IMREAD_GRAYSCALE)
3 img
```

```
1 def show_plot(image, title):
2     plt.figure(figsize=(10, 3))
3     plt.subplot(121)
4     plt.axis('off')
5     plt.imshow(image, cmap='gray')
6     plt.title(title)
7     plt.subplot(122)
8     plt.hist(image.ravel(), bins=256, range=(0, 256))
9     plt.title('Histogram')
10    plt.show()
```

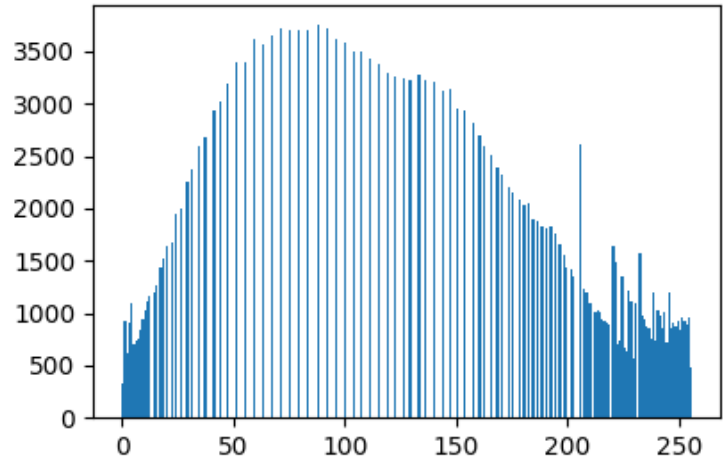
```
1 show_plot(img, 'Original Image')
```

2. Please plot the histogram and graph of the image after Histogram Equalization, and print out the source code? (30)

Histogram-equalized Image



Histogram



Source code:

```
1 import cv2
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import bisect
```

```
1 # 讀取圖片
2 img = cv2.imread('aerial_view.tif', cv2.IMREAD_GRAYSCALE)
3 img
```

```
1 def show_plot(image, title):
2     plt.figure(figsize=(10, 3))
3     plt.subplot(121)
4     plt.axis('off')
5     plt.imshow(image, cmap='gray')
6     plt.title(title)
7     plt.subplot(122)
8     plt.hist(image.ravel(), bins=256, range=(0, 256))
9     plt.title('Histogram')
10    plt.show()
11
12 def histEqualization(image):
13     hist, _ = np.histogram(image.flatten(), 256, [0, 255])
14     pdf = hist / (image.shape[0] * image.shape[1])
15     cdf = pdf.cumsum()
16     s_k = np.round(255 * cdf)
17     image_new = s_k[image]
18     return image_new.astype(int)
```

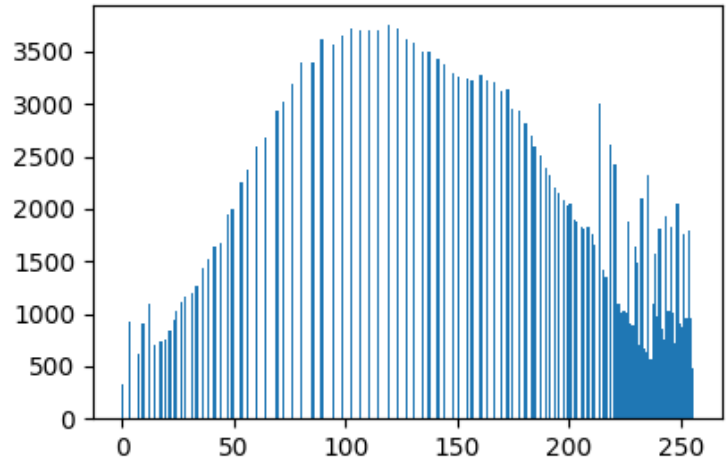
```
1 equalized_img = histEqualization(img)
2 show_plot(equalized_img, 'Histogram-equalized Image')
```

3. Please plot the histogram and graph of the image after Histogram Matching (specification) by $p_z(z_q) = c \cdot z_q^{0.4}$, and print out the source code? (NOTE: the parameter, c, needs to calculate in advance) (40)

Histogram-matching Image



Histogram



Source code:

```
1 import cv2
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import bisect
```

```
1 # 讀取圖片
2 img = cv2.imread('aerial_view.tif', cv2.IMREAD_GRAYSCALE)
3 img
```

```
1 def show_plot(image, title):
2     plt.figure(figsize=(10, 3))
3     plt.subplot(121)
4     plt.axis('off')
5     plt.imshow(image, cmap='gray')
6     plt.title(title)
7     plt.subplot(122)
8     plt.hist(image.ravel(), bins=256, range=(0, 256))
9     plt.title('Histogram')
10    plt.show()
```

```
20 def histMatching(image):
21     z = np.arange(256)
22     sum_z = np.sum(z**0.4)
23     c = 1 / sum_z
24
25     target_pdf = c * (z**0.4)
26     target_cdf = target_pdf.cumsum()
27     target_s_k = np.round(255 * target_cdf)
28
29     original_hist, _ = np.histogram(image.flatten(), 256, [0, 255])
30     original_cdf = (original_hist / (image.shape[0] * image.shape[1])).cumsum()
31     original_s_k = np.round(255 * original_cdf)
```

```

33 def find_closest_index(sorted_list, value):
34     # 在sorted_list中找到接近value的位置
35     idx = bisect.bisect_left(sorted_list, value)
36
37     if idx == 0:
38         return 0 # 如果value小於等於sorted_list中最小的值，返回0
39     elif idx == len(sorted_list):
40         return len(sorted_list) - 1 # 如果value大於sorted_list中最大的值，返回最後一個索引
41     else:
42         # 如果左右有值，選擇離value較近的值
43         prev_val = sorted_list[idx - 1]
44         next_val = sorted_list[idx]
45         if abs(prev_val - value) <= abs(next_val - value):
46             return idx - 1
47         else:
48             return idx
49
50 def map_indexes(original_s_k, target_s_k):
51     mapping = [0] * len(original_s_k) # 初始化一個與original_s_k長度相同的列表
52     for i, val in enumerate(original_s_k):
53         closest_idx = find_closest_index(target_s_k, val)
54         mapping[i] = closest_idx
55     return mapping

```

```

57 mapping = np.array(map_indexes(original_s_k, target_s_k))
58 image_new = mapping[image]
59 return image_new.astype(int)

```

```

1 matched_img = histMatching(img)
2 show_plot(matched_img, 'Histogram-matching Image')

```

4. Please comment the original, the histogram-equalized and the histogram-matching images? (20)

- (1) 根據原圖的 histogram，可以看出 pixel 值大多落在 100 左右的範圍，因此圖片偏暗。
- (2) 做完 histogram-equalization 的 histogram 分布較原圖均勻，對比度強使得亮暗明顯，可以看到更多細節。
- (3) 與 equalize 相比，histogram-matching 的 histogram 在 pixel 值 200 以上的分布更集中、更靠近 255，因此圖片比 equalized image 來得亮。

