In [61]:

```python
############# HW2-1
import cv2
import numpy as np
from PIL import Image
###(1)
img = cv2.imdecode(np.fromfile(r'C:\Users\apple\OneDrive\桌面\影像\HW2\text-broken.tif',

# 二值化處理
_, binary_img = cv2.threshold(img, 127, 255, cv2.THRESH_BINARY)

# 定義結構元素
kernel = cv2.getStructuringElement(cv2.MORPH_RECT, (3, 3))

# 進行放大操作
dilated_img = cv2.dilate(binary_img, kernel, iterations=1)

# 進行縮小操作
eroded_img = cv2.erode(dilated_img, kernel, iterations=1)

# 修復後
repaired_img = eroded_img

cv2.imshow('Original Image', img)
cv2.imshow('Repaired Image', repaired_img)
cv2.waitKey(0)
cv2.destroyAllWindows()

#####(2)
img2 = cv2.erode(repaired_img, kernel, iterations=1)
bound = repaired_img - img2
cv2.imshow('boundary', bound)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

In [62]:

```python
############# hw2-2
gray_img = cv2.imdecode(np.fromfile(r'C:\Users\apple\OneDrive\桌面\影像\HW2\aerialview-wa
min_val = np.min(gray_img)
max_val = np.max(gray_img)
stretched_img = (gray_img - min_val) * (255.0 / (max_val - min_val))
stretched_img = stretched_img.astype(np.uint8)
cv2.imshow('stretched', stretched_img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

In [63]:

```python
############# hw2-4
img = cv2.imdecode(np.fromfile(r'C:\Users\apple\OneDrive\桌面\影像\HW2\aerialview-washedo
median = int(np.median(img))
histogram_1 = [0] * (median + 1)
histogram_2 = [0] * (256 - (median + 1))
cumulative_hist_1 = [0] * (median + 1)
cumulative_hist_2 = [0] * (256 - (median + 1))
for i in range(img.shape[0]):
    for j in range(img.shape[1]):
        pixel_value = img[i, j]
        if pixel_value <= median:
            histogram_1[pixel_value] += 1
        else:
            histogram_2[pixel_value - (median + 1)] += 1
cumulative_hist_1[0] = histogram_1[0]
for i in range(1, len(histogram_1)):
    cumulative_hist_1[i] = cumulative_hist_1[i - 1] + histogram_1[i]

cumulative_hist_2[0] = histogram_2[0]
for i in range(1, len(histogram_2)):
    cumulative_hist_2[i] = cumulative_hist_2[i - 1] + histogram_2[i]
equalized_hist_1 = [0] * (median + 1)
equalized_hist_2 = [0] * (256 - (median + 1))

total_pixels_1 = img.shape[0] * img.shape[1]
total_pixels_2 = total_pixels_1

for i in range(len(equalized_hist_1)):
    equalized_hist_1[i] = round((cumulative_hist_1[i] / total_pixels_1) * 255)

for i in range(len(equalized_hist_2)):
    equalized_hist_2[i] = round((cumulative_hist_2[i] / total_pixels_2) * 255)
equalized_img = np.zeros_like(img)

for i in range(img.shape[0]):
    for j in range(img.shape[1]):
        pixel_value = img[i, j]
        if pixel_value <= median:
            equalized_img[i, j] = equalized_hist_1[pixel_value]
        else:
            equalized_img[i, j] = equalized_hist_2[pixel_value - (median + 1)]
cv2.imshow('equalized', equalized_img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```
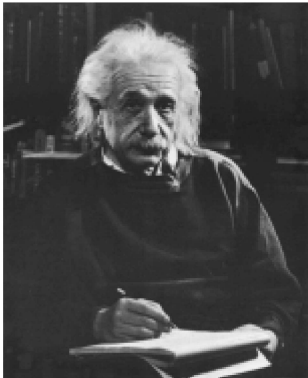
In [49]:

```python
###### hw2-3
import matplotlib.pyplot as plt
img = cv2.imdecode(np.fromfile(r'C:\Users\apple\OneDrive\桌面\影像\HW2\einstein-low-contr
histogram, bins = np.histogram(img.flatten(), 256, [0, 256])
cdf = histogram.cumsum()
cdf_normalized = cdf * histogram.max() / cdf.max()   # 歸一化到0-255範圍
equalized_img = cdf_normalized[img]
plt.subplot(1, 2, 1)
plt.imshow(img, cmap='gray')
plt.title('Original Image')
plt.axis('off')

plt.subplot(1, 2, 2)
plt.imshow(equalized_img, cmap='gray')
plt.title('Equalized Image')
plt.axis('off')

plt.show()
```


Original Image   Equalized Image