

影像視訊處理實驗作業結報

班 級	電機	學 號	M11107S08	姓 名	王維澤
實驗題目	影像中物體的分割				

實驗內容

*請勿抄襲，否則視為未交

一、 實驗簡介(至少 200 字)：

相似性可定義在不同的特徵上；例如，像素的灰階或色彩值、小區塊像素的紋理或幾何結構、區塊的移動或變形、等。我們將區塊分割法分成三類：

1. 以灰階數量分佈圖為基礎的分割法(threshold-based segmentation)
2. 以區塊為基礎的分割法(region-based segmentation)
3. 以動態資訊為基礎的分割法(motion-based segmentation)。

二、 實驗動機及其解決方法(至少 500 字)：

設置 Threshold 做門檻值法的動作，稱為門檻值分割，簡稱分割。門檻值法的目的通常都是要擷取 (extract) 出影像中的物件或瑕疵區域，只要這些物件或瑕疵區域的灰階或顏色與背景有差異，我們就可以用門檻值法將這些物件或瑕疵上的點從背景分離出來。

三、 程式碼(須附註解說明，截圖即可)：

```
1 def Threshold(img):
2     Thresholding=130
3     ret, thresh1 = cv.threshold(img, Thresholding, 255, cv.THRESH_BINARY)
4     ret, thresh2 = cv.threshold(img, Thresholding, 255, cv.THRESH_BINARY_INV)
5     ret, thresh3 = cv.threshold(img, Thresholding, 0, cv.THRESH_TRUNC)
6     ret, thresh4 = cv.threshold(img, Thresholding, 0, cv.THRESH_TOZERO)
7     ret, thresh5 = cv.threshold(img, Thresholding, 255, cv.THRESH_TOZERO_INV)
8     titles = ['Original Image', 'BINARY', 'BINARY_INV', 'TRUNC', 'TOZERO', 'TOZERO_INV']
9     images = [img, thresh1, thresh2, thresh3, thresh4, thresh5]
10    for i in range(len(images)):
11        if(i == 0):
12            cv.imwrite('Original Image.jpg', images[i])
13        if(i > 0):
14            cv.imwrite('thresh'+str(i)+'.jpg', images[i])
15    for i in range(len(images)):
16        plt.subplot(3,2,i+1), plt.imshow(images[i], 'gray', vmin=0, vmax=255)
17        plt.title(titles[i])
18        plt.xticks([], plt.yticks([]))
19    plt.savefig('Image_Threshold_Result.jpg')
20    plt.show()
```

原本 Threshold 的函式

```

1 def Threshold2(img, number):
2     Thresholding=130
3     ret, thresh1 = cv.threshold(img, Thresholding, 255, cv.THRESH_BINARY)
4     ret, thresh2 = cv.threshold(img, Thresholding, 255, cv.THRESH_BINARY_INV)
5     ret, thresh3 = cv.threshold(img, Thresholding, 0, cv.THRESH_TRUNC)
6     ret, thresh4 = cv.threshold(img, Thresholding, 0, cv.THRESH_TOZERO)
7     ret, thresh5 = cv.threshold(img, Thresholding, 255, cv.THRESH_TOZERO_INV)
8     titles = ['Original Image', 'BINARY', 'BINARY_INV', 'TRUNC', 'TOZERO', 'TOZERO_INV']
9     images = [img, thresh1, thresh2, thresh3, thresh4, thresh5]
10    for i in range(len(images)):
11        if(i == 0):
12            cv.imwrite('Original Image.jpg', images[i])
13        if(i > 0):
14            cv.imwrite('thresh'+str(i)+'.jpg', images[i])
15    plt.title(titles[number])
16    plt.xticks([], plt.yticks([]))
17    plt.savefig('Image_Threshold_Result.jpg')
18    plt.imshow(images[number], 'gray', vmin=0, vmax=255)

```

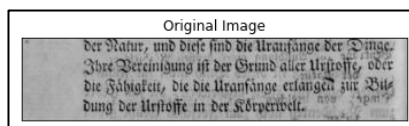
為了方便可以依序顯示輸出的圖片及好解析度所以寫 Threshold2



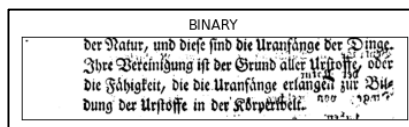
輸出呈現如下

四、實驗結果(貼圖與簡述說明)：

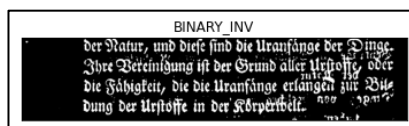
Original image:原圖



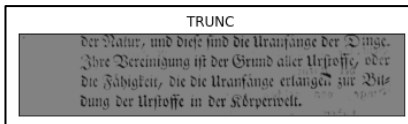
BINARY:大於門檻值的灰階值設為最大灰階值，小於門檻值的值設為 0



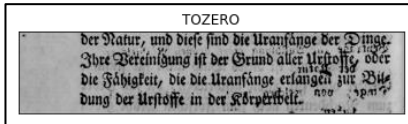
BINARY_INV：大於門檻值的灰階值設為 0，其他值設為最大灰階值。



TRUNC: 大於門檻值的灰階值設為門檻值，小於門檻值的值保持不變。



TOZERO：小於門檻值的灰階值設為 0，大於門檻值的值保持不變。



TOZERO_INV：大於門檻值的灰階值設為 0，小於門檻值的值保持不變。



(可自行增頁)

*內容字級為 12 字級、中文為標楷體、英文為 Times New Roman