F74062109 許嘉倫 資訊110

程式語言: python

import cv2

import numpy as np

# histogram equalization

def histogram\_equalization(img):

    h = np.zeros((256,), dtype=np.int) #calculate histogram

    s = np.zeros((256,), dtype=np.int) #calculate s = T(r)

    Sum = np.zeros((256,)) # calculate p[0]~p[k]的總和

    height, width = img.shape # picture's height width

    for x in range(width): #calculate histogram

        for y in range(height):

            h[img[y][x]] = h[img[y][x]] + 1

    p = h/(height\*width) #calculate p[r] = h[r]/MN

    for k in range(256): # calculate p[0]~p[k]的總和

        if k==0:

            Sum[k] = p[k]

        else:

            Sum[k] = Sum[k-1]+p[k]

    for k in range(256): #calculate s = T(r) = 255\*Sum[k]  , 0.5是拿來四捨五入

        s[k] = int(255\*Sum[k]+0.5)

    for x in range(width): # according to s[r] 改變原本的gray level

        for y in range(height):

            img[y][x] = s[img[y][x]]

    return img

def sobel(img):

    height, width = img.shape

    img\_sobel = img.copy()

    gx = np.array([[-1,-2,-1],

                   [ 0, 0, 0],

                   [ 1, 2, 1]])

    gy = np.array([[-1, 0, 1],

                   [-2, 0, 2],

                   [-1, 0, 1]])

    for x in range(1,width-1):

        for y in range(1,height-1):

            dx = sum(sum(gx\*img[y-1:y+2,x-1:x+2]))

            dy = sum(sum(gy\*img[y-1:y+2,x-1:x+2]))

            img\_sobel[y][x] = abs(dx) + abs(dy)

    return img\_sobel

image = cv2.imread("1.jpg",0)

cv2.imshow("origin", image)

cv2.waitKey(0)

cv2.destroyAllWindows()

img\_histogram = histogram\_equalization(image.copy())

cv2.imshow("histogram", np.hstack((image, img\_histogram)))

cv2.waitKey(0)

cv2.imwrite("1\_histogram.jpg", img\_histogram)

cv2.destroyAllWindows()

img\_sobel = sobel(img\_histogram.copy())

cv2.imshow("sobel", img\_sobel)

cv2.waitKey(0)

cv2.imwrite("1\_sobel.jpg", img\_sobel)

cv2.destroyAllWindows()

image = cv2.imread("2.jpg",0)

cv2.imshow("origin", image)

cv2.waitKey(0)

cv2.destroyAllWindows()

img\_histogram = histogram\_equalization(image.copy())

cv2.imshow("compare", np.hstack((image, img\_histogram)))

cv2.waitKey(0)

cv2.imwrite("2\_histogram.jpg", img\_histogram)

cv2.destroyAllWindows()

img\_sobel = sobel(img\_histogram.copy())

cv2.imshow("sobel", img\_sobel)

cv2.waitKey(0)

cv2.imwrite("2\_sobel.jpg", img\_sobel)

cv2.destroyAllWindows()

image = cv2.imread("1.jpg",0)

img\_sobel = sobel(image.copy())

cv2.imshow("sobel", img\_sobel)

cv2.waitKey(0)

cv2.imwrite("origin1\_sobel.jpg", img\_sobel)

cv2.destroyAllWindows()

image = cv2.imread("2.jpg",0)

img\_sobel = sobel(image.copy())

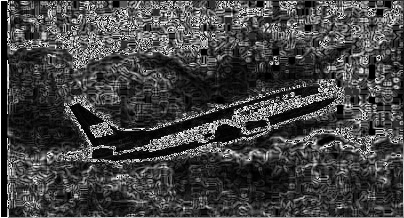
cv2.imshow("sobel", img\_sobel)

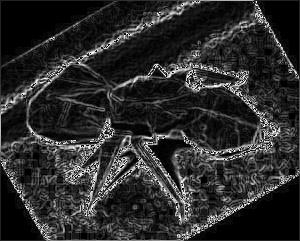
cv2.waitKey(0)

cv2.imwrite("origin2\_sobel.jpg", img\_sobel)

cv2.destroyAllWindows()

origin histogram equalization sobel

不做histogram\_equalization的sobel較好

