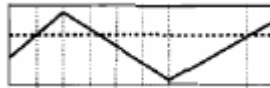


Binary image

The core idea of binarization is to set a threshold, and the value greater than the threshold is 0 (black) or 255 (white), making the image called a black and white image. The threshold can be fixed or adaptive. The adaptive threshold is generally a comparison of a point pixel with the mean value of regional pixels or the weighted sum of Gaussian distribution at this point, in which a difference value may or may not be set.

Global threshold:

The threshold function is provided in Python-OpenCV: `cv2.threshold (src, threshold, maxValue, method)`



src original image: the dashed line is the value that will be thresholded; the dotted line is the threshold value



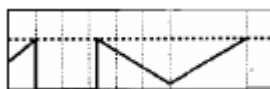
`cv2.THRESH_BINARY`: The gray value of pixels greater than the threshold is set to `maxValue` (for example, the maximum 8-bit gray value is 255), and the gray value of pixels whose gray value is less than the threshold is set to 0.



`cv2.THRESH_BINARY_INV`: The gray value of pixels greater than the threshold is set to 0, and the gray value of pixels less than the threshold is set to `maxValue`.



`cv2.THRESH_TRUNC`: The gray value of the pixel is less than the threshold and does not change. The pixel with a gray value greater than the threshold is set to the threshold.



`cv2.THRESH_TOZERO`: If the gray value of the pixel is less than the threshold, no change will be made, and for the part greater than the threshold, the gray value will all become 0.



`cv2.THRESH_TOZERO_INV`: If the gray value of the pixel is greater than the threshold, no change will be made. If the gray value of the pixel is less than the threshold, all the gray values will be changed to 0.

- Start Docker

After entering the Raspberry Pi 5 desktop, open a terminal and run the following command to start the container corresponding to Dofbot:

```
./Docker_Ros.sh
```

Access Jupyter Lab within Docker:

```
IP:9999 // Example: 192.168.1.11:9999
```

Code path:/root/Dofbot/4.opencv/3.draw_picture/02_binarization.ipynb

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
img = cv2.imread('yahboom.jpg',1)
GrayImage = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY) #Convert to grayscale image

ret,thresh1=cv2.threshold(GrayImage,10,255,cv2.THRESH_BINARY)
ret,thresh2=cv2.threshold(GrayImage,10,255,cv2.THRESH_BINARY_INV)
ret,thresh3=cv2.threshold(GrayImage,10,255,cv2.THRESH_TRUNC)
ret,thresh4=cv2.threshold(GrayImage,10,255,cv2.THRESH_TOZERO)
ret,thresh5=cv2.threshold(GrayImage,10,255,cv2.THRESH_TOZERO_INV)
titles = ['Gray Image','BINARY','BINARY_INV','TRUNC','TOZERO','TOZERO_INV']
images = [GrayImage, thresh1, thresh2, thresh3, thresh4, thresh5]
for i in range(6):
    plt.subplot(2,3,i+1),plt.imshow(images[i],'gray')
    plt.title(titles[i])
    plt.xticks([],plt.yticks([]))
plt.show()
```

