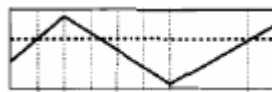# Binary image

The core idea of binarization is to set a threshold, and the values greater than the threshold are 0 (black) or 255 (white), making the image a black and white image. The threshold can be fixed or adaptive.

The adaptive threshold is generally a comparison between a pixel at a point and the average value of the pixels in the region with this point as the center or the weighted sum of the Gaussian distribution, where a difference can be set or not.

Global Threshold:

Python-OpenCV provides a threshold function: 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.

cv2.THRESH_BINARY: The grayscale value of the pixel point that is greater than the threshold is set to maxValue (such as the maximum 8-bit grayscale value is 255), and the grayscale value of the pixel point that is less than the threshold is set to 0.

cv2.THRESH_BINARY_INV : The grayscale value of pixels greater than the threshold is set to 0, while those less than the threshold are set to maxValue.

cv2.THRESH_TRUNC: If the gray value of a pixel is less than the threshold, it will not change. If the gray value of a pixel is greater than the threshold, it will be set to the threshold.
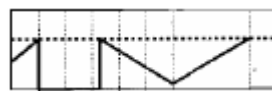
cv2.THRESH_TOZERO: The grayscale values of pixels that are less than the threshold value are not changed, while the grayscale values of pixels that are greater than the threshold value are all changed to 0.

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

Code path:

```python
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()
```