

# 阿洲的程式教學

關於Qt、OpenCV、影像處理演算法

---

## 固定閾值(threshold)

所謂的二值化是將影像進行區分，分成我們感興趣的部分(前景)，以及不感興趣的部分(背景)，通常將某個強度當作分割的標準，這個強度稱作閾值(threshold)，通常以強度超過閾值的像素當作前景，反之則為背景。

閾值的算法主要分兩類：

- 固定閾值：程式或使用者直接給定一個灰階值當閾值，再用這個閾值進行二值化。
- 自適應閾值：輸入影像，程式依這影像計算出較合適的閾值，再用這個閾值進行二值化。

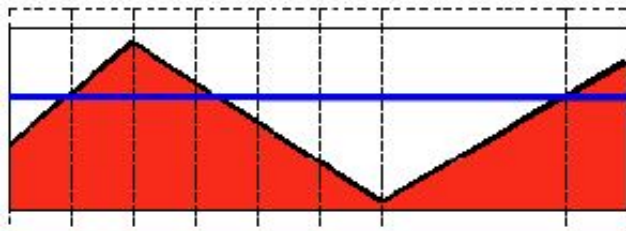
這邊介紹如何用OpenCV的threshold()，輸入固定閾值來進行二值化。

### OpenCV固定閾值二值化

double threshold(InputArray src, OutputArray dst, double thresh, double maxval, int type)

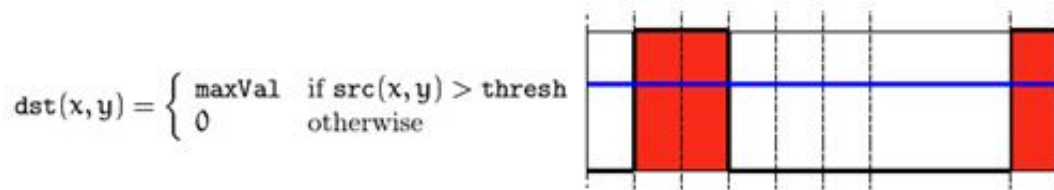
- src：輸入圖，只能輸入單通道，8位元或32位元浮點數影像。
- dst：輸出圖，尺寸大小、深度會和輸入圖相同。
- thresh：閾值。
- maxval：二值化結果的最大值。
- type：二值化操作型態，共有THRESH\_BINARY、THRESH\_BINARY\_INV、THRESH\_TRUNC、THRESH\_TOZERO、THRESH\_TOZERO\_INV五種。

以下為進行二值化前的原始圖，假設黑線為影像的各個像素強度，藍線為閾值：

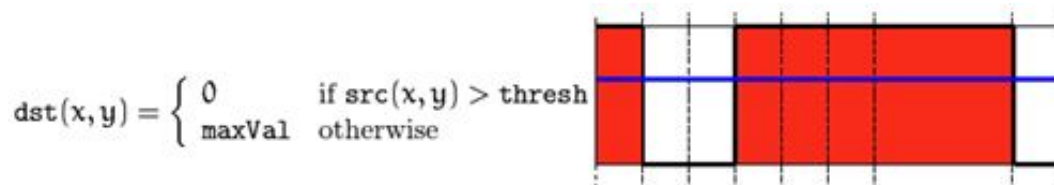


這邊分別介紹以這五種操作型態進行二值化後，影像像素值的變化。

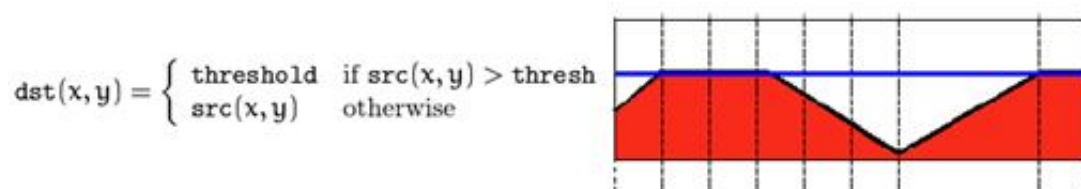
**THRESH\_BINARY**：超過閾值的像素設為最大值(maxval)，小於閾值的設為0。



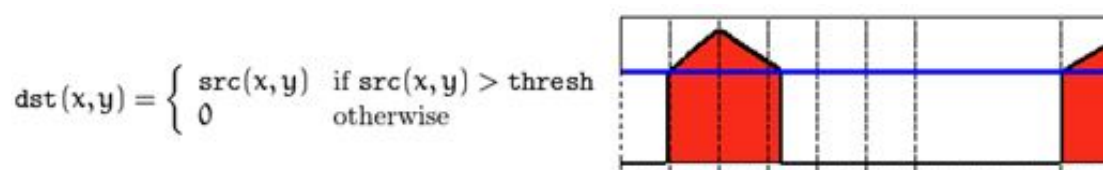
**THRESH\_BINARY\_INV**：超過閾值的像素設為0，小於閾值的設為最大值(maxval)。



**THRESH\_TRUNC**：超過閾值的像素設為閾值，小於閾值的設為0。

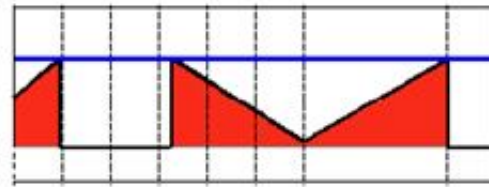


**THRESH\_TOZERO**：超過閾值的像素值不變，小於閾值的設為0。



**THRESH\_TOZERO\_INV**：超過閾值的像素值設為0，小於閾值的不變。

$$\text{dst}(x,y) = \begin{cases} 0 & \text{if } \text{src}(x,y) > \text{thresh} \\ \text{src}(x,y) & \text{otherwise} \end{cases}$$



以下示範threshold()的用法，將灰階值小於150的設為0，大於150的設為255：

```
#include <stdio>
#include <opencv2/opencv.hpp>
using namespace cv;

int main(){
    Mat src = imread("lena.jpg",CV_LOAD_IMAGE_GRAYSCALE);
    Mat dst;
    threshold(src, dst, 150, 255, THRESH_BINARY);
    imshow("origin", src);
    imshow("threshold", dst);
    waitKey(0);

    return 0;
}
```





[回到首頁](#)

[回到OpenCV教學](#)

參考資料：

[OpenCV 教程](#)

📅 2015-11-26    👤 阿宅    📁 OpenCV, 二值化    🔑 threshold, 二值化, 固定閾值

**0 Comments**

猴子遇到0與1! 程式學習筆記

**1** Login ▾ Recommend Share

Sort by Best ▾



Start the discussion...

Be the first to comment.

**ALSO ON** 猴子遇到0與1! 程式學習筆記**Qt主窗口(Top Level Window)**

1 comment • 6 months ago

mike — 喔喔

**文件對話框(QFileDialog)**

1 comment • 6 months ago

楊政穎 — dialog.cpp 裡面的 QString s  
=  
QFileDialog::getOpenFileName(this, tr

 Subscribe Add Disqus to your site Add Disqus Add Privacy

自豪的採用 WordPress