

# 阿洲的程式教學

關於Qt、OpenCV、影像處理演算法

## 開運算、閉運算(morphologyEx、MORPH\_OPEN、MORPH\_CLOSE)

這邊介紹開運算和閉運算，這兩種都是侵蝕和膨脹複合而成，開運算是先侵蝕後膨脹，閉運算是先膨脹後侵蝕。

對於集合I和H，假設使用H對I進行開運算，代表H對I進行侵蝕後膨脹，記作：

$$I \circ H \quad \text{且} \quad I \circ H = (I \ominus H) \oplus H$$

開運算可以使物體輪廓變得光滑，還能使狹窄的連結斷開，以及消除外觀上的毛刺，但在物體大於結構元素的情況下，開運算與侵蝕並不相同，圖像的輪廓並沒有產生整體的收縮，物體位置也沒有發生任何變化，假如我們對一幅影像重複進行開運算，不會產生任何變化，這點和重複進行侵蝕會加強程度的現象不同。

有兩種方式進行開運算，一種是先呼叫`erode()`將影像進行侵蝕，接著將侵蝕後的圖，當作`dilate()`膨脹運算的原始圖，最後的輸出圖即為結果，另外一種方式是使用`morphologyEx()`函式，將其中某個參數輸入MORPH\_OPEN。

### OpenCV開運算

`morphologyEx(const Mat &src, Mat &dst, int op, Mat kernel, Point anchor=Point(-1,-1), int iterations=1)`

- `src`：輸入圖，可以多通道，深度可為CV\_8U、CV\_16U、CV\_16S、CV\_32F或CV\_64F。
- `dst`：輸出圖，和輸入圖尺寸、型態相同。
- `op`：操作種類，決定要進行何種型態學操作，在開運算時輸入MORPH\_OPEN。
- `kernel`：結構元素。
- `anchor`：原點位置，預設為結構元素的中央。
- `iterations`：執行次數，預設為1次。

對於集合**I**和**H**，假設使用**H**對**I**進行閉運算，代表**H**對**I**進行膨脹後侵蝕，記作：

$$I \bullet H \text{ 且 } I \bullet H = (I \oplus H) \ominus H$$

閉運算也可以使物體輪廓變得光滑，和開運算不同，閉運算通常能彌補狹窄的間斷，假如我們對一幅影像重複進行閉運算，不會產生任何變化，這點和重複進行膨脹會加強程度的現象不同。

有兩種方式進行開運算，一種是先呼叫**dilate()**將影像進行膨脹，接著將膨脹後的圖，當作**erode()**侵蝕運算的原始圖，最後的輸出圖即為結果，另外一種方式是使用**morphologyEx()**，將其中某個參數輸入**MORPH\_CLOSE**。

## OpenCV閉運算

**morphologyEx(const Mat &src, Mat &dst, int op, Mat kernel, Point anchor=Point(-1,-1), int iterations=1)**

- **src**：輸入圖，可以多通道，深度可為**CV\_8U**、**CV\_16U**、**CV\_16S**、**CV\_32F**或**CV\_64F**。
- **dst**：輸出圖，和輸入圖尺寸、型態相同。
- **op**：操作種類，決定要進行何種型態學操作，在閉運算時輸入**MORPH\_CLOSE**。
- **kernel**：結構元素。
- **anchor**：原點位置，預設為結構元素的中央。
- **iterations**：執行次數，預設為**1**次。

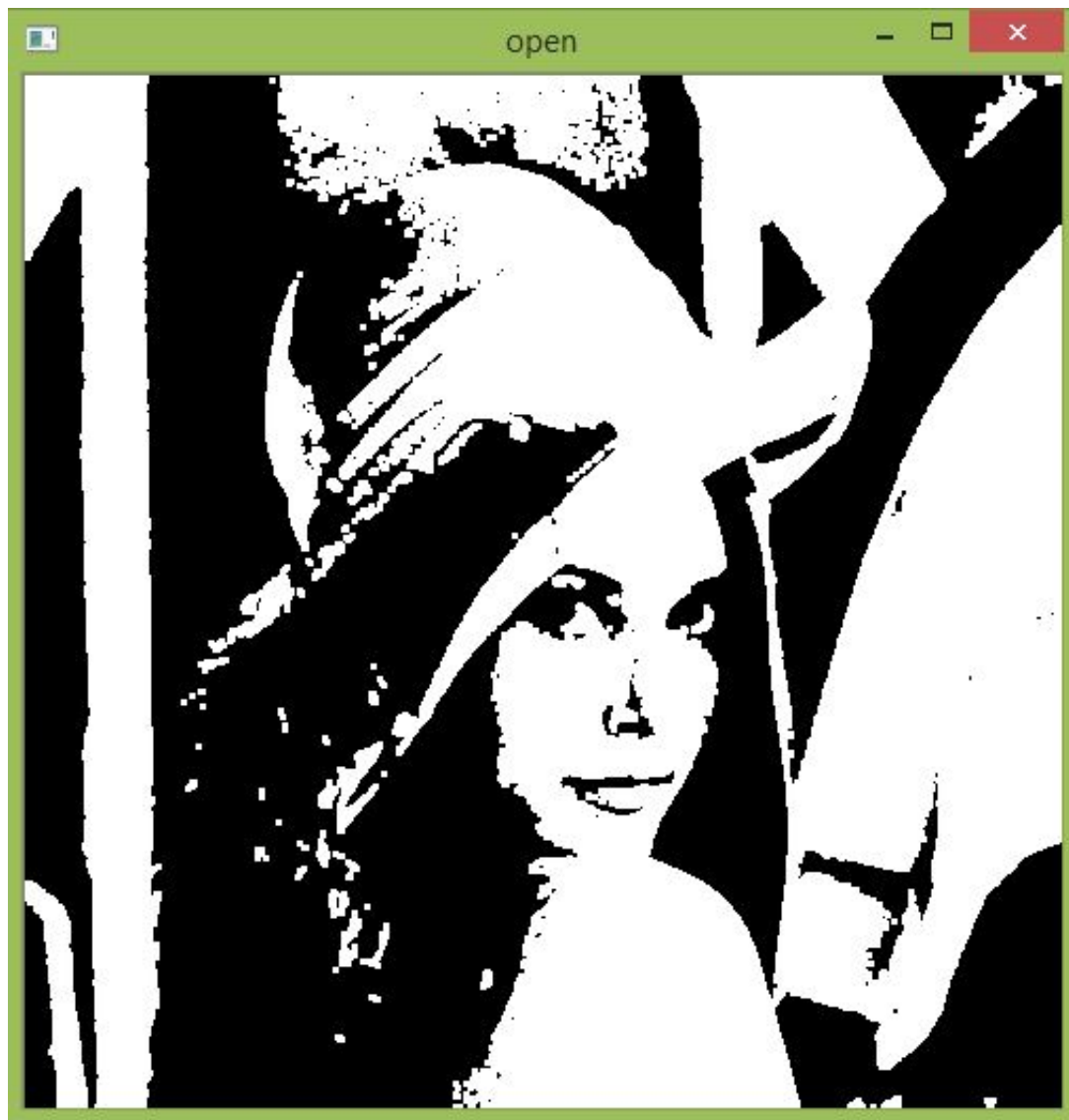
以下程式碼示範**morphologyEx()**的使用，模板參數可輸入**Mat()**，此時會用**3×3**的矩形模板，也可以用**getStructuringElement()**來得到想要的大小和形狀的模板，我們可以看出二值化的圖形，在進行開運算或閉運算後，影像變化的樣子：

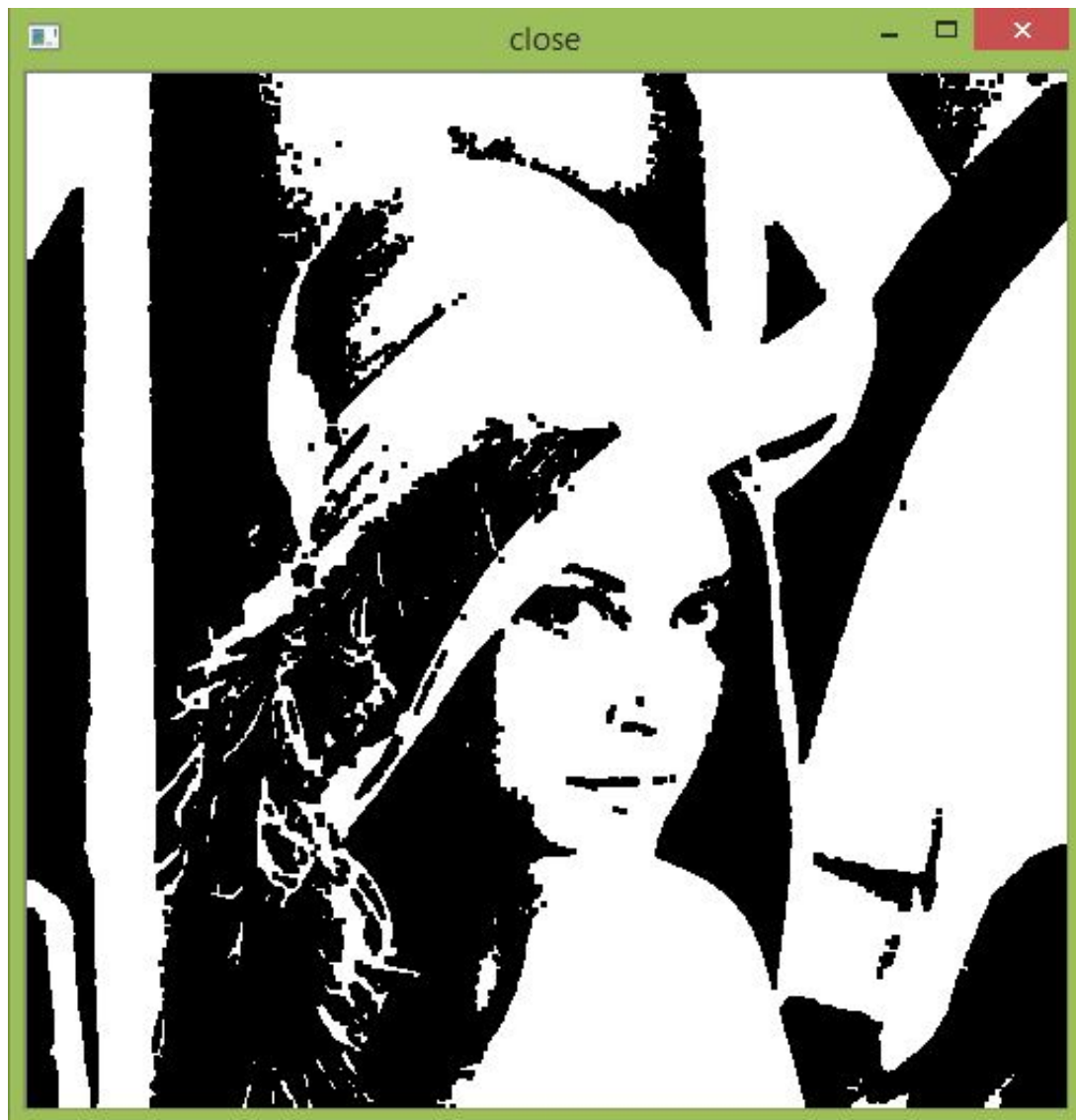
```
#include <cstdio>
#include <opencv2/opencv.hpp>
using namespace cv;

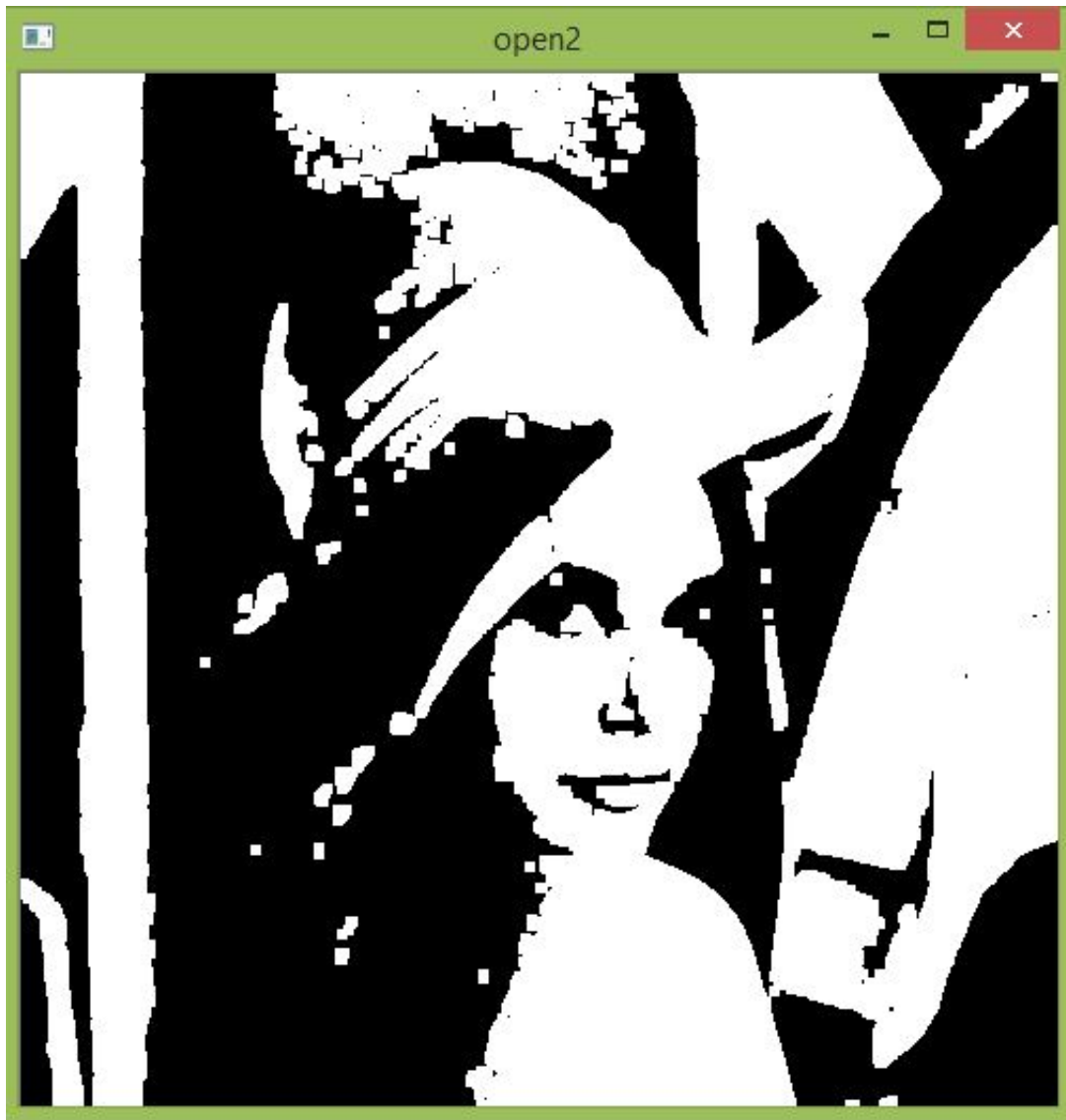
int main(){
    Mat src = imread("lena.jpg",CV_LOAD_IMAGE_GRAYSCALE);
    Mat src2;
    threshold(src,src2,120,255,THRESH_BINARY);
    Mat dst1;
    Mat dst2;
    Mat dst3;
    morphologyEx(src2, dst1, MORPH_OPEN, Mat());
    morphologyEx(src2, dst2, MORPH_CLOSE, Mat());
```

```
Mat erodeStruct = getStructuringElement(MORPH_RECT,Size(5,5));  
morphologyEx(src2, dst3, MORPH_OPEN, erodeStruct);  
imshow("origin", src2);  
imshow("open", dst1);  
imshow("close", dst2);  
imshow("open2", dst3);  
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, 型態學    💡 morphologyEx, 閉運算, 開運算

0 Comments

猴子遇到0與1! 程式學習筆記

 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