开运算

```
开运算 (Opening Operation) , 其实就是先腐蚀后膨胀的操作。 \mathbf{dst} = \mathbf{open}(\mathbf{src}, \mathbf{element}) = \mathbf{dilate}(\mathbf{erode}(\mathbf{src}, \mathbf{element}))
```

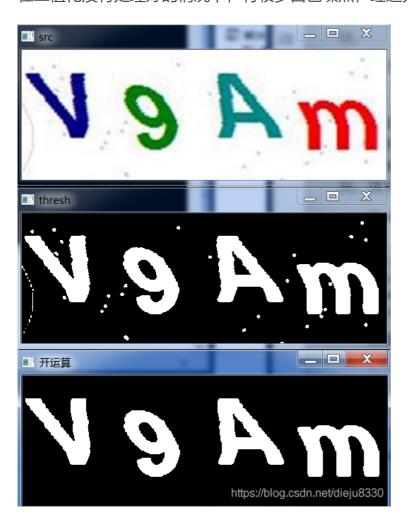
作用:

- 去除噪声,消除小物体
- 在纤细点处分离物体
- 平滑较大物体的边界的同时并不明显改变其面积

简单测试:

```
#include<opencv2/core/core.hpp>
#include<opencv2/highgui/highgui.hpp>
#include<opencv2/imgproc/imgproc.hpp>
#include<iostream>
using namespace std;
using namespace cv;
void main()
{
    Mat srcImg = imread("F:\\opencv_re_learn\\test.jpg");
    if (!srcImg.data){
        cout << "failed to read" << endl;</pre>
        system("pause");
        return;
    imshow("src", srcImg);
    Mat srcGray;
    cvtColor(srcImg, srcGray, CV_BGR2GRAY);
    Mat thresh;
    threshold(srcGray, thresh, 230, 255, CV_THRESH_BINARY_INV);
    imshow("thresh", thresh);
    //自定义核
    Mat element = getStructuringElement(MORPH_RECT,
        Size(5, 5));
    //开运算
    Mat open_result;
    morphologyEx(thresh, open_result, MORPH_OPEN,element);
    imshow("开运算", open_result);
    waitKey(0);
}
```

在二值化没有处理好的情况下,有较多白色噪点,经过开运算可去除



闭运算

闭运算: 操作为先膨胀后腐蚀

dst = close(src, element) = erode(dilate(src, element))

作用:

- 排除小型空洞 (指黑色区域)
- 平滑物体轮廓
- 弥合(连接)窄的间断点,沟壑
- 填补轮廓线断裂

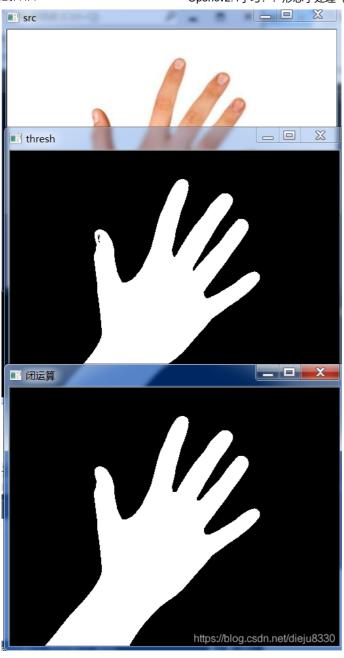
简单测试:

#include<opencv2/core/core.hpp>
#include<opencv2/highgui/highgui.hpp>
#include<opencv2/imgproc/imgproc.hpp>

```
| #include<iostream>
                       using namespace std;
 using namespace cv;
 void main()
 {
     Mat srcImg = imread("F:\\opencv_re_learn\\hand.jpg");
     if (!srcImg.data){
         cout << "failed to read" << endl;</pre>
         system("pause");
         return;
     imshow("src", srcImg);
     Mat srcGray;
     cvtColor(srcImg, srcGray, CV_BGR2GRAY);
     threshold(srcGray, thresh, 200, 255, CV_THRESH_BINARY_INV|
         CV_THRESH_OTSU);
     imshow("thresh", thresh);
     //自定义核
     Mat element = getStructuringElement(MORPH_RECT,
         Size(5, 5));
     //闭运算
     Mat close_result;
     morphologyEx(thresh, close_result, MORPH_CLOSE, element);
     imshow("闭运算", close_result);
     waitKey(0);
 }
```

测试结果:

使用OTSU算法对图片进行二值化,从第二张图可以看到拇指处有小部分黑色空洞,第三张图经过闭运算后,填补了黑色空洞部分。



核心函数

const Scalar& borderValue=morphologyDefaultBorderValue

一般来说,看前四个参数就行了,后面的就用默认值

- 第一个参数输入
- 第二个参数 输出
- 第三个参数 操作类型

1. MORTH_OPEN 函数做开运算

2. MORTH_CLOSE 函数做闭运算

3. MORTH GRADIENT 函数做形态学梯度运算

4. MORTH_TOPHAT 函数做顶帽运算

5. MORTH_BLACKHAT 函数做黑帽运算

6. MORTH_DILATE 函数做膨胀运算

7. MORTH_ERODE 函数做腐蚀运算

• 第四个参数 内核类型 用getStructuringElement函数得到