

【图像处理】-011 顶帽和底帽变换

顶帽变换和底帽变换属于图像形态学处理的一种，可用于校正不均匀光照的影响。

1 理论依据

2 底帽变换应用

3 OpenCV实现

4 注意

1 理论依据

图像相减与开操作和闭操作相结合，会产生Top-hat(顶帽)变换和bottom-hat(底帽)变换。灰度级图像 f 的顶帽变换定义为 f 减去其开操作：

$$T_{hat}(f) = f - f(f \circ b) \quad (1)$$

类似的， f 的底帽变换定义为 f 的闭操作减去 f ：

$$B_{hat}(f) = (f \cdot b) - f \quad (2)$$

这些变换的主要应用之一是，用一个结构元通过开操作或闭操作从一幅图像中删除物体，而不是拟合被删除的物体。然后，差操作得到一幅仅保留已删除分量的图像。顶帽变换用于在暗背景上的亮物体，而底帽变换用于相反的情况。常常被称为白顶帽变换和黑底帽变换。顶帽变换的一个重要用途是校正不均匀光照的影响。

2 底帽变换应用

这里，我通过一个实际应用来说明底帽变换的应用。这是一幅在网上找到的图片，版权归原作者所有。右下角的水印不是版权的声明。从图中可以看出，这幅图的左右两边的光照是不均匀的。我们的目标是从图中把圆环分割出来。

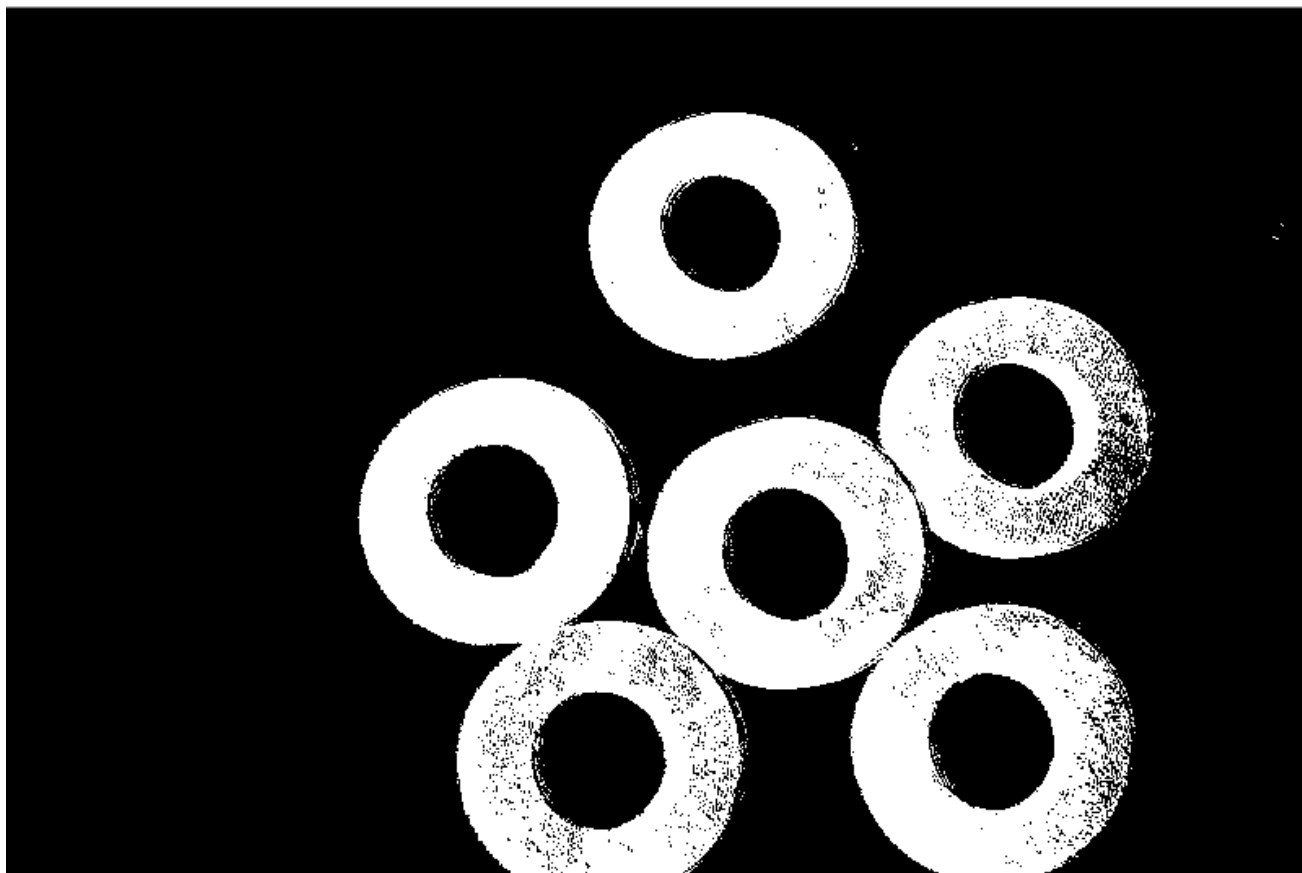
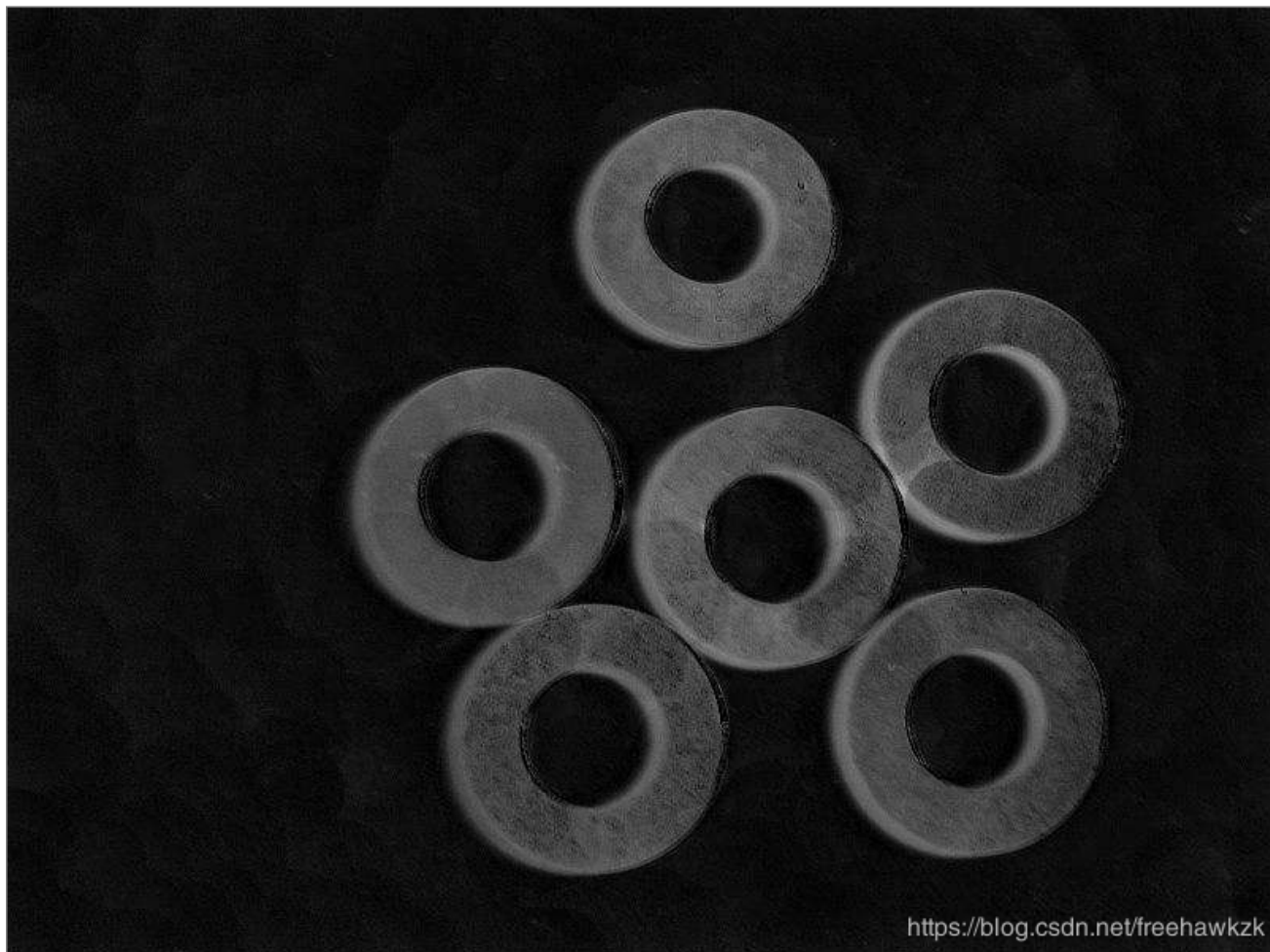


图像灰度化之后直接OSTU二值化结果如下：



可以看出，虽然圆环被分割为目标区域，但左侧背景区域有很大一部分也被分割为目标，并且该区域面积较大，难以通过开操作或闭操作去除。

接下来我们通过底帽操纵来进行处理。首先对原图进行底帽处理，然后再对底帽处理的结果进行OSTU二值化，最后对处理结果进行一轮闭操作。



从上图中可以看出，圆环大体上已经被分割出来，但在圆环表面还有很多小小的黑洞。这可以通过一轮闭操作来去除。



3 OpenCV实现

```
#include "../include/importOpenCV.h"//请根据您的环境配置OpenCV环境
#include "../include/baseOps.h"//将工作目录设置到当前EXE所在目录
#include <iostream>
int main()
{
    //将工作目录设置到EXE所在的目录。
    SetCurrentDirectoryToExePath();

    cv::Mat src = cv::imread("../images/64.jpg");
    cv::cvtColor(src, src, cv::COLOR_BGR2GRAY);
    cv::imshow("原图", src);

    cv::Mat bSrc;
    cv::threshold(src, bSrc, 150, 255, cv::THRESH_BINARY_INV | cv::THRESH_OTSU);
    cv::imshow("原图OSTU二值化", bSrc);
```

```
cv::Mat bHat;
cv::Mat element = cv::getStructuringElement(cv::MORPH_ELLIPSE, cv::Size(55, 55));
cv::morphologyEx(src, bHat, cv::MORPH_BLACKHAT, element);
cv::imshow("黑帽处理", bHat);
cv::Mat bbDst;
cv::threshold(bHat, bbDst, 50, 255, cv::THRESH_OTSU | cv::THRESH_BINARY);
cv::imshow("黑帽处理之后OSTU二值化", bbDst);

cv::Mat element1 = cv::getStructuringElement(cv::MORPH_RECT, cv::Size(3, 3));
cv::morphologyEx(bbDst, bbDst, cv::MORPH_CLOSE, element1);
cv::imshow("黑帽处理后OSRT二值化再闭操作", bbDst);

cv::waitKey();
cv::destroyAllWindows();
return 0;
}
```

4 注意

在进行底帽处理的时候，选择的模板大小将影响底帽变换的结果。该模板结构的大小需要和目标的尺寸相匹配。同时该尺寸的大小，将影响操作执行的速度。