DIP_Hw5_粒子大小分布_苏智龙

一、白帽子操作

对原图进行不同尺寸的白帽子操作,结构元为大小从2到52的正方形,部分效果图见图1.

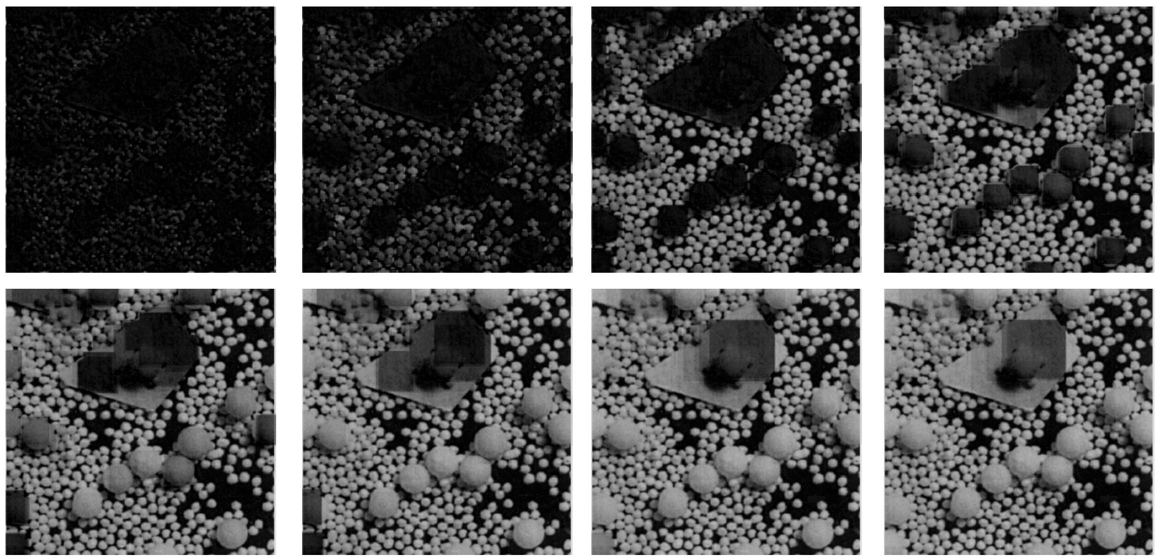

图1 结构元大小分别为5,7,11,20,28,35,46,52的白帽子操作

二、二值化

对白帽子之后的图进行二值化,经过对比,阈值选择40的效果比较好,二值化后的部分图见图2.

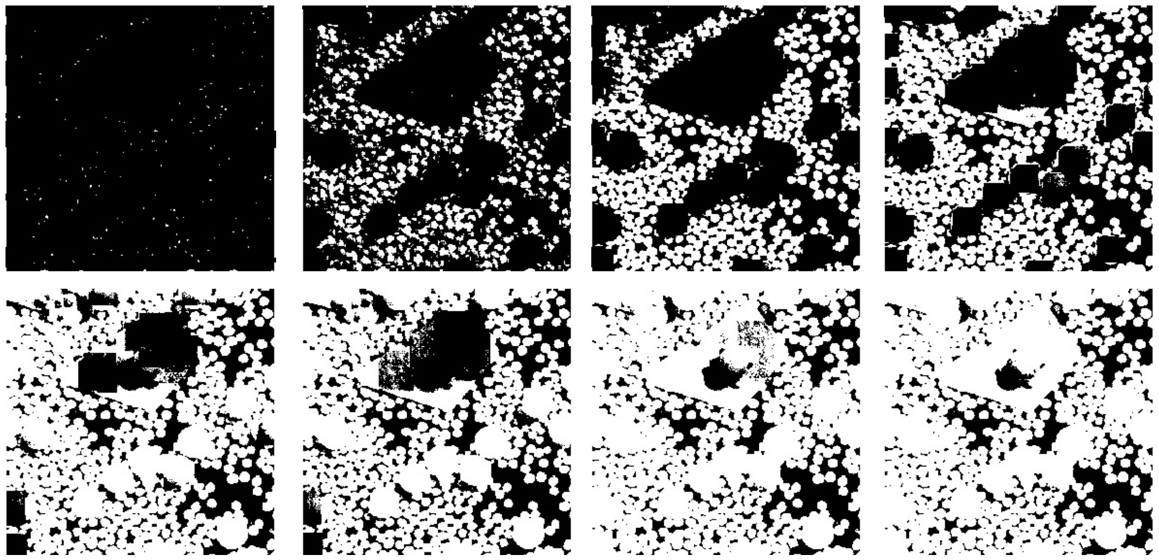

图2 结构元大小分别为5,7,11,20,28,35,46,52白帽子操作后阈值40为二值化

三、单一尺度亮细节

WTH(n+1)-WTH(n-1),就能得到直径为n的亮细节,即尺寸为n的粒子.图3为白帽子操作过后的灰度图相减之后的部分结果,为了视觉效果,已做二值化处理.

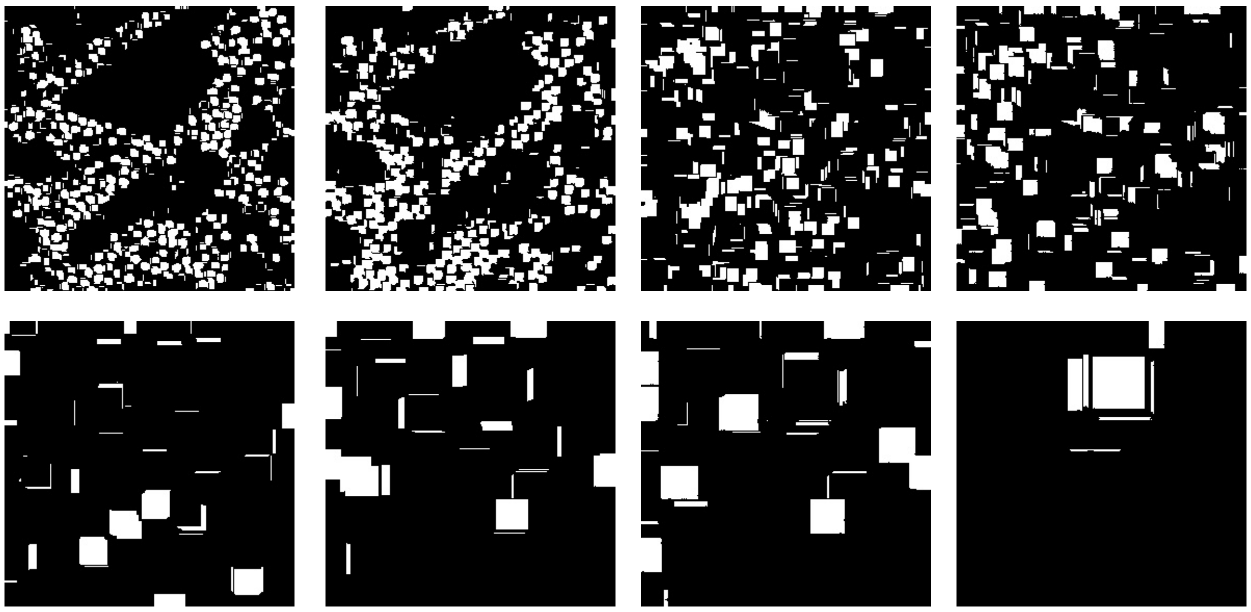
图3 减完之后大小分别为6,9,12,15,24,30,33,51的亮点

四、统计个数

对每一个尺度的二值图的像素进行亮点统计,用亮度为255的像素点数,除以结构元的像素点数,即得到相应尺度的粒子个数.统计图件图4.

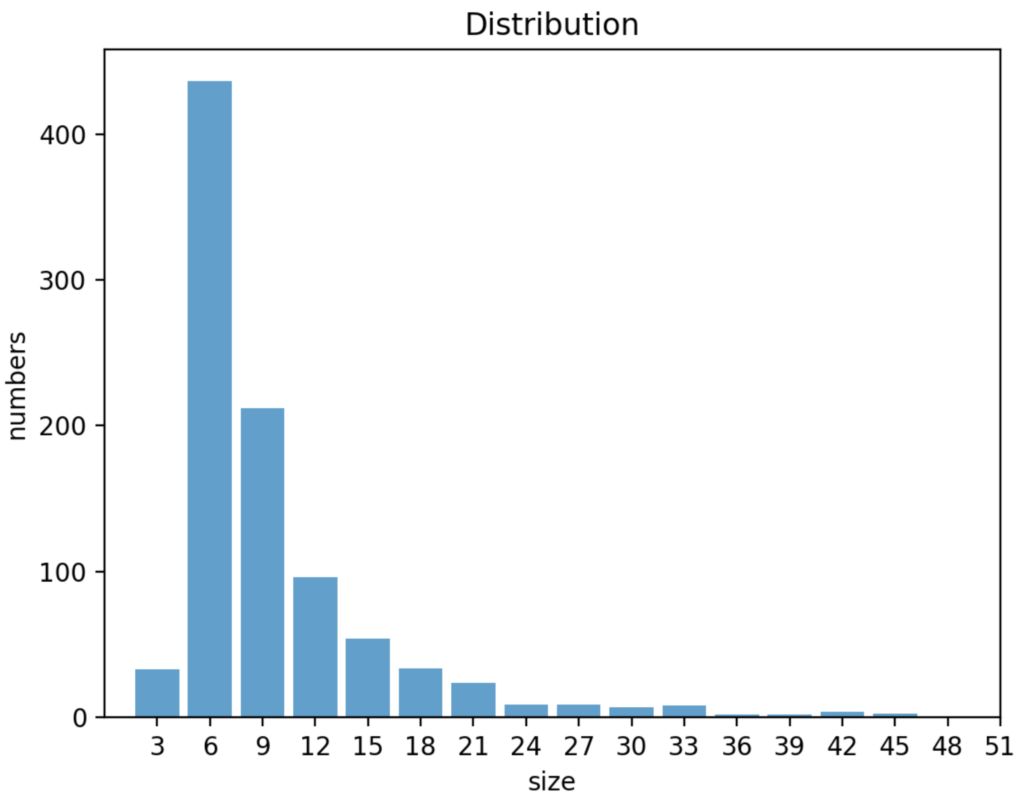

图4 粒子大小分布统计图

五、代码

一、二、三使用c++,四用c++计算后把数据保存成txt文件,再用python画出分布图.代码如下:

1)c++代码:

```
#include <opencv2/core/core.hpp>
#include <opencv2/highgui/highgui.hpp>
#include <opencv2/imgproc/imgproc.hpp>
#include <iostream>
#include <fstream>

#define PATH “/Users/sousic/code/cpp/DIP/Hw5/"

cv::Mat subWTH(cv::Mat srcImg,int kerSize)
{//得到kerSize的WTH相减后的图片
    cv::Mat WTHImg, binWTHImg;
    cv::Mat WTHImg2, binWTHImg2;
    cv::Mat result;

    //kerSize-1
    cv::Mat kernel1 = cv::getStructuringElement(cv::MORPH_RECT, 
cv::Size(kerSize-1, kerSize-1));
    morphologyEx(srcImg, WTHImg, cv::MORPH_TOPHAT, kernel1);//WTH of 
origin
    cvtColor(WTHImg, WTHImg, CV_BGR2GRAY);//灰度化
    threshold(WTHImg, binWTHImg, 15, 255, CV_THRESH_BINARY);//灰度阈值处理

    //kerSize+1
    cv::Mat kernel2 = cv::getStructuringElement(cv::MORPH_RECT, 
cv::Size(kerSize+1, kerSize+1));
    morphologyEx(srcImg, WTHImg2, cv::MORPH_TOPHAT, kernel2);//WTH of 
origin
    cvtColor(WTHImg2, WTHImg2, CV_BGR2GRAY);//灰度化
    threshold(WTHImg2, binWTHImg2, 15, 255, CV_THRESH_BINARY);//灰度阈值处
理
    //sub
    cv::subtract(WTHImg2,WTHImg,result);
    threshold(result, result, 20, 255, CV_THRESH_OTSU);//灰度阈值处理
    //cv::imshow(std::to_string(kerSize+2)+"-
"+std::to_string(kerSize),result);
    return result;
}

void getSubWTH(bool wr=false){//得到不同尺寸的subWTH
    std::string path = PATH;
    cv::Mat srcImg = cv::imread(path+"Chapter5_1.bmp");
    for (int kerSize = 3; kerSize < 52; kerSize+=3)
    {
        cv::Mat result = subWTH(srcImg,kerSize);

        if (wr==true) {//save
            std::string folder="/opt/WTH/";
            cv::imwrite(PATH + folder + 
std::to_string(kerSize)+".jpg",result);
        }
    }
}

void getWTH(){
    std::string path = PATH;
    cv::Mat srcImg = cv::imread(path+"Chapter5_1.bmp");
    cv::Mat WTHImg1, binWTHImg1;
    cv::Mat WTHImg2, binWTHImg2;
    for (int kerSize=3; kerSize<52; kerSize+=3) {
        cv::Mat kernel1 = cv::getStructuringElement(cv::MORPH_RECT, 
cv::Size(kerSize-1, kerSize-1));
        cv::Mat kernel2 = cv::getStructuringElement(cv::MORPH_RECT, 
cv::Size(kerSize+1, kerSize+1));
        morphologyEx(srcImg, WTHImg1, cv::MORPH_TOPHAT, kernel1);//WTH 
of origin
        cvtColor(WTHImg1, WTHImg1, CV_BGR2GRAY);//灰度化
        threshold(WTHImg1, binWTHImg1, 60, 255, CV_THRESH_BINARY);//灰度
阈值处理
        morphologyEx(srcImg, WTHImg2, cv::MORPH_TOPHAT, kernel2);//WTH 
of origin
        cvtColor(WTHImg2, WTHImg2, CV_BGR2GRAY);//灰度化
        threshold(WTHImg2, binWTHImg2, 40, 255, CV_THRESH_BINARY);//灰度
阈值处理
        std::string orifolder="/opt/oriWTH/";
        cv::imwrite(PATH + orifolder + std::to_string(kerSize-
1)+".jpg",WTHImg1);
        cv::imwrite(PATH + orifolder + 
std::to_string(kerSize+1)+".jpg",WTHImg2);
        std::string binfolder="/opt/binWTH/";
        cv::imwrite(PATH + binfolder + std::to_string(kerSize-
1)+".jpg",binWTHImg1);
        cv::imwrite(PATH + binfolder + 
std::to_string(kerSize+1)+".jpg",binWTHImg2);
    }
}
int getBalls(cv::Mat img,int kerSize){
    //计算亮点的数量
    int n = 0;//亮点的数量
    int row = img.rows;            //获取行数目;
    int col = img.cols;            //获取列数目;
    //int chan = img.channels();  //获取通道数目;
    for (int i = 0; i<row; i++) {
        for (int j = 0; j<col; j++) {
            int v =img.at<uchar>(i,j);
            if (v>128) {
                n++;
            }
        }
    }
    //kernel的面积
    int s = kerSize*kerSize;
    //亮像素点总数除以kernel的面积得到球的数量
    int balls = n / s;
    return balls;
}
int main()
{
    getWTH();//保存原WTH的灰度图和二值图

    getSubWTH(true);//保存相减后的WTH图
    std::string path = PATH;
    cv::Mat srcImg = cv::imread(path+"Chapter5_1.bmp");
    std::ofstream writer;
    std::string filename = "opt/ballsDis.txt";
    writer.open(PATH+filename);
    cv::Mat subImg;
    for (int kerSize=3; kerSize<52; kerSize+=3) {
        //白帽子相减
        subImg = subWTH(srcImg,kerSize);
        //计算球的数量
        int balls = getBalls(subImg, kerSize);
        //std::cout <<kerSize<<":"<<balls<<std::endl;
        writer << kerSize<<":"<<balls <<"\n";
    }
    writer.close();
    return 0;
}
```

2)python代码:

```
# -*-coding:utf-8 -*-
import codecs
from matplotlib import pyplot as plt

def main():
     fileName = "/Users/sousic/code/cpp/DIP/Hw5/opt/ballsDis.txt"
     reader = codecs.open(fileName,'r',encoding='utf-8')
     lines = reader.readlines()

     n = []
     balls = []

     for i in range(len(lines)):
          n.append(int(lines[i].split(':')[0]))
          balls.append(int(lines[i].split(':')[1]))

     plt.bar(left=n, height=balls, width=2.5, alpha=0.7, label="")
     plt.xlim(0, 50)     # y轴取值范围
     plt.xlabel("size")
     plt.ylabel("numbers")
     plt.xticks([index for index in n], n)
     plt.title('Distribution')
     plt.show()

if __name__ == '__main__':
     main()
```