

# Работа К. фильтрация изображений

---

автор: Хонер П.Д.

дата: 2022-05-16T23:40:10

[https://github.com/pavelkhoner/OpenCV/tree/master/khoner\\_p\\_d/prj.labs/labk](https://github.com/pavelkhoner/OpenCV/tree/master/khoner_p_d/prj.labs/labk)

## Задание

0. текст, иллюстрации и подписи отчета придумываем самостоятельно

1. нарисовать

- одноканальное изображение
- поле 2x3 из квадратных клеток 150x150px черного, белого и серого (127) цвета (соседние цвета разные)
- в центре клеток - круг другого цвета (все сочетания перебрать)

2. отфильтровать и визуализировать I1 (фильтр вида)

1 0

0 -1

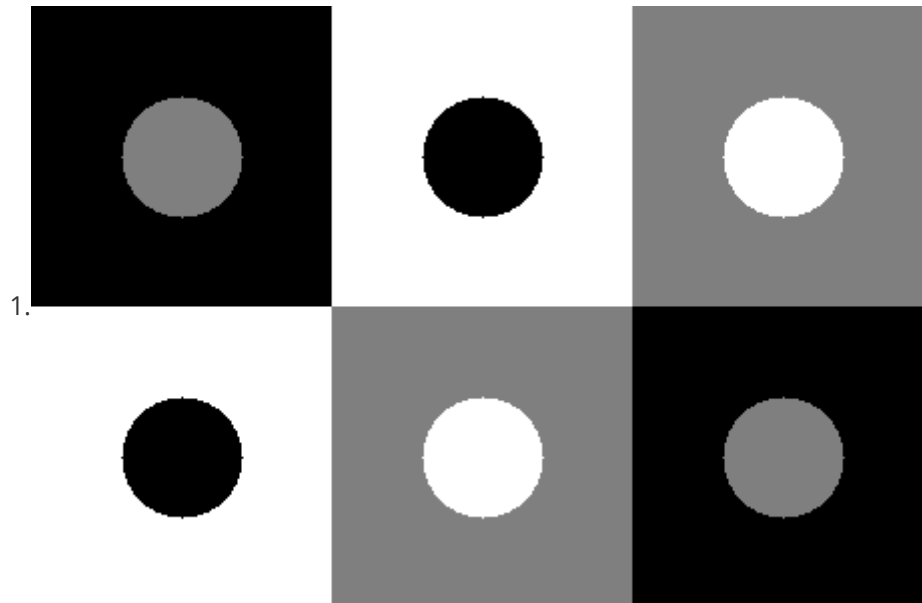
3. отфильтровать и визуализировать I2 (фильтр вида)

0 1

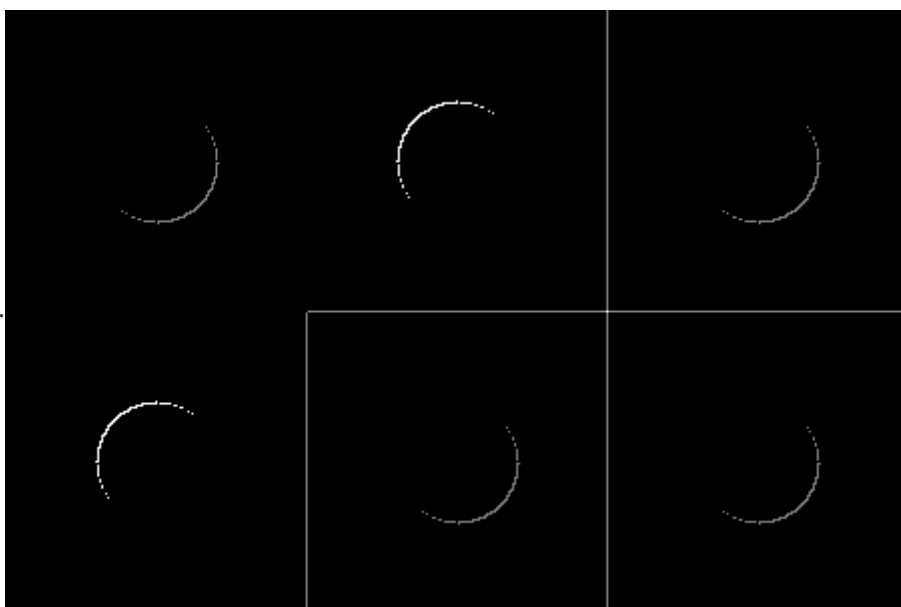
-1 0

4. вычислить и визуализировать геометрическое среднее (корень из суммы квадратов I1 и I2)

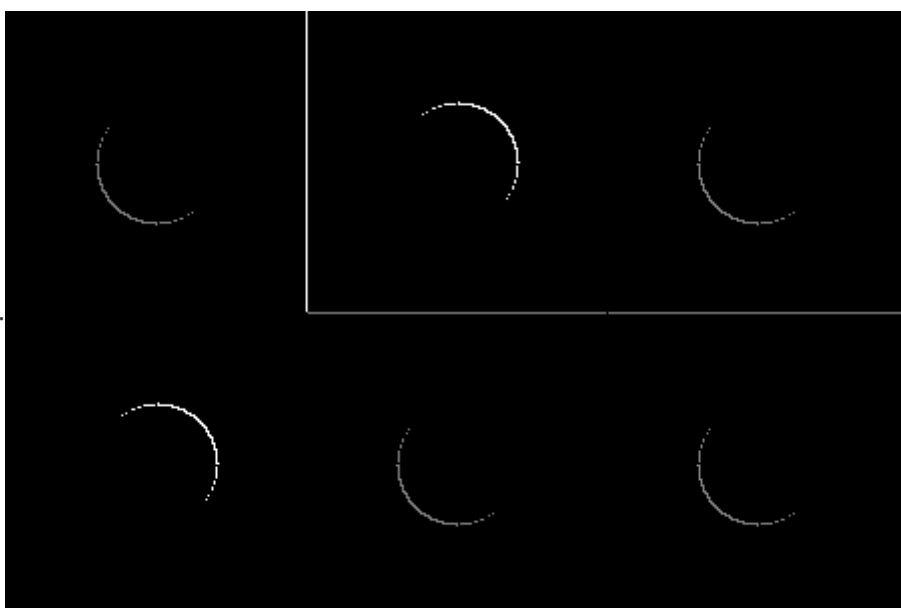
## Результаты



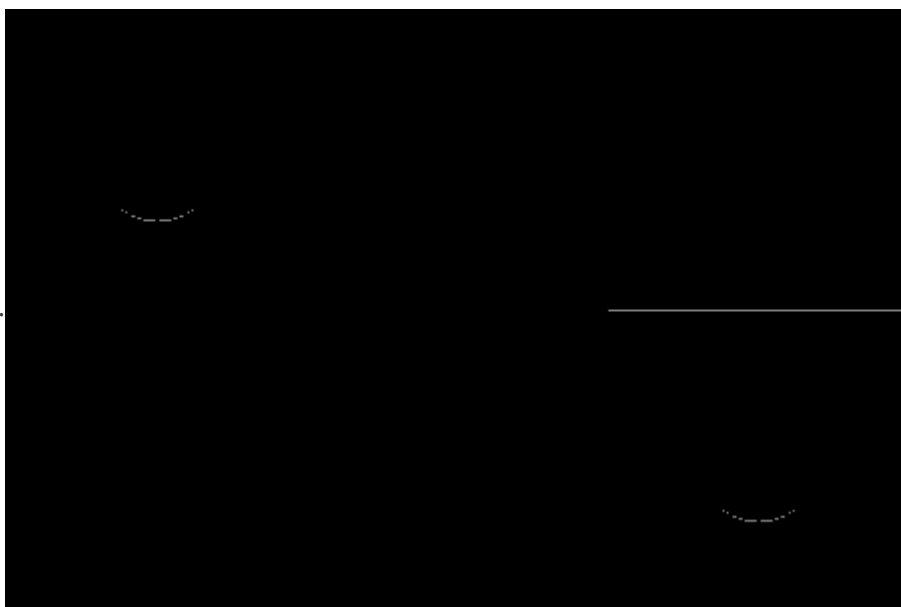
2.

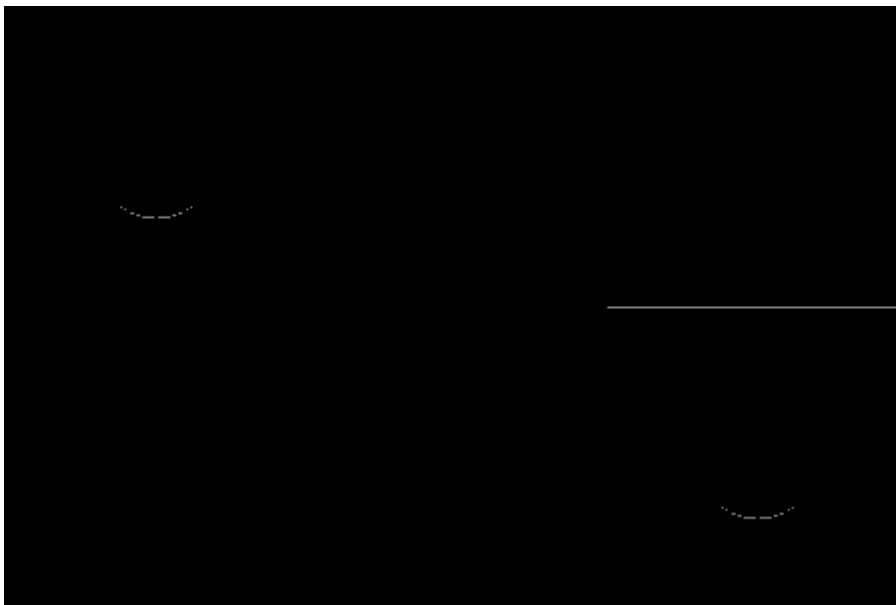


3.



4.





## Текст программы

```
#include <opencv2/opencv.hpp>

int main() {
    cv::Mat img(300, 450, CV_8UC1);
    // draw dummy image
    img = 0;
    //1.rectangles
    //chor
    for (int i = 0; i < 150; i++)
        for (int j = 0; j < 150; j++)
            img.at<char>(i, j) = 0;
    //bel
    for (int i = 0; i < 150; i++)
        for (int j = 150; j < 300; j++)
            img.at<char>(i, j) = 255;
    //ser
    for (int i = 0; i < 150; i++)
        for (int j = 300; j < 450; j++)
            img.at<char>(i, j) = 127;
    //belniz
    for (int i = 150; i < 300; i++)
        for (int j = 0; j < 150; j++)
            img.at<char>(i, j) = 255;
    ///serniz
    for (int i = 150; i < 300; i++)
        for (int j = 150; j < 300; j++)
            img.at<char>(i, j) = 127;
    ///krujochki
    int r = 30;
    ///perviy
    cv::circle(img, cv::Point(75, 75), r, 127, -1);
    //vtoroy
    cv::circle(img, cv::Point(225, 75), r, 0, -2);
    //tretiy
    cv::circle(img, cv::Point(375, 75), r, 255, -3);
    //perviyniz
    cv::circle(img, cv::Point(75, 225), r, 0, -4);
```

```

//vtoroyniz
cv::circle(img, cv::Point(225, 225), r, 255, -5);
//poslednniz
cv::circle(img, cv::Point(375, 225), r, 127, -6);

cv::imwrite("kr.png", img);
///2.filtr I1
cv::Mat I1;
float kerI1[4]{ 1,0,0,-1 };
cv::Mat kerI1_matrix(2, 2, CV_32FC1, kerI1);
cv::filter2D(img, I1, -1, kerI1_matrix);
cv::imwrite("kr1.png", I1);

///3.filtrI2
cv::Mat I2;
float kerI2[4]{ 0,1,-1,0 };
cv::Mat kerI2_matrix(2, 2, CV_32FC1, kerI2);
cv::filter2D(img, I2, -1, kerI2_matrix);
cv::imwrite("kr2.png", I2);
///4.
cv::Mat getsr(img);
for (int i = 0; i < img.cols; i++) {
    for (int j = 0; j < img.rows; j++) {
        getsr.at<char>(j, i) = sqrt(I1.at<char>(j, i) * I2.at<uchar>(j, i));
    }
}
cv::imwrite("kr3.png", getsr);
cv::imwrite("kr4.png", img);
return 0;
}
@cpp_source@

```