Работа К. фильтрация изображений

автор: Конев Т.В. дата: 2022-05-17T21:15:50 хранилище: https://github.com/timakon/KonevCV/tree/main/prj.labs/task

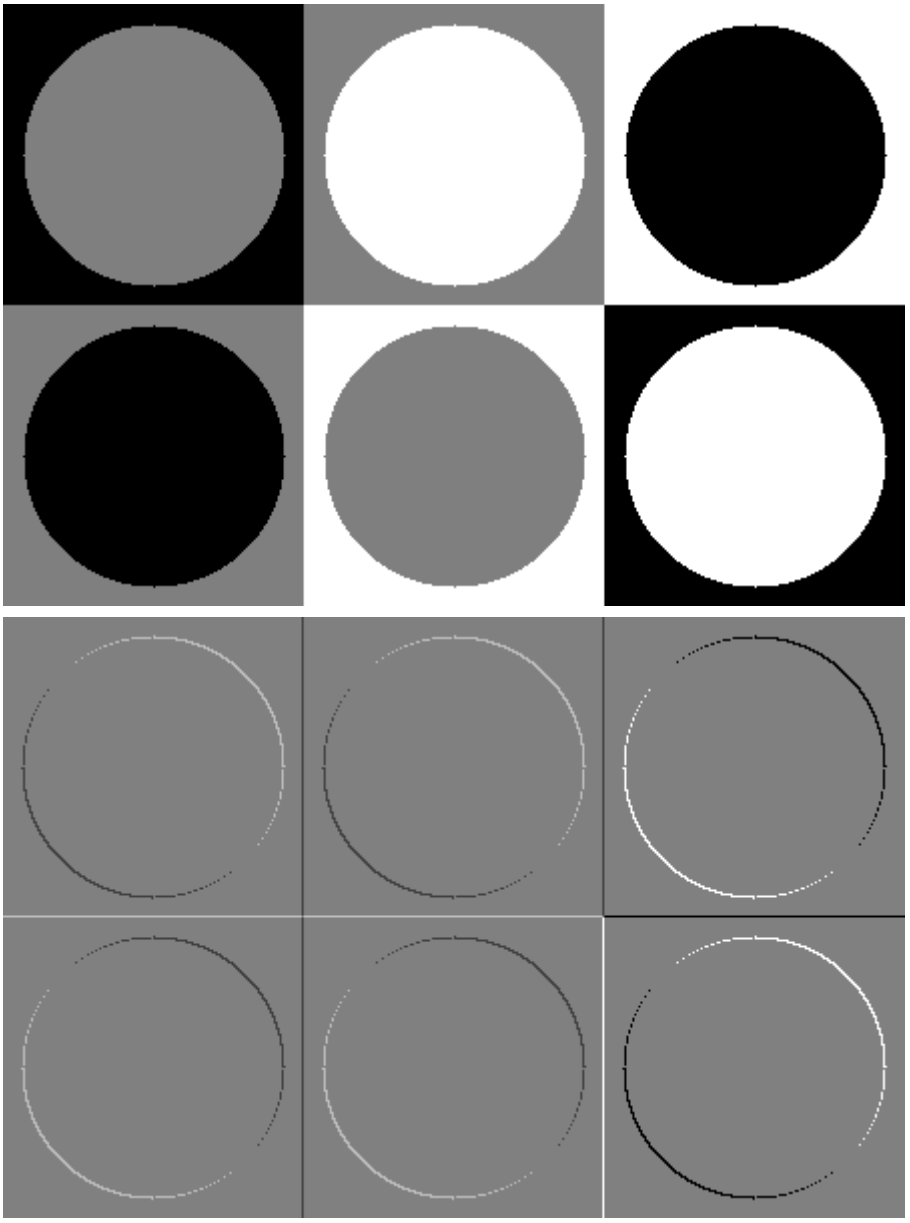
Задание

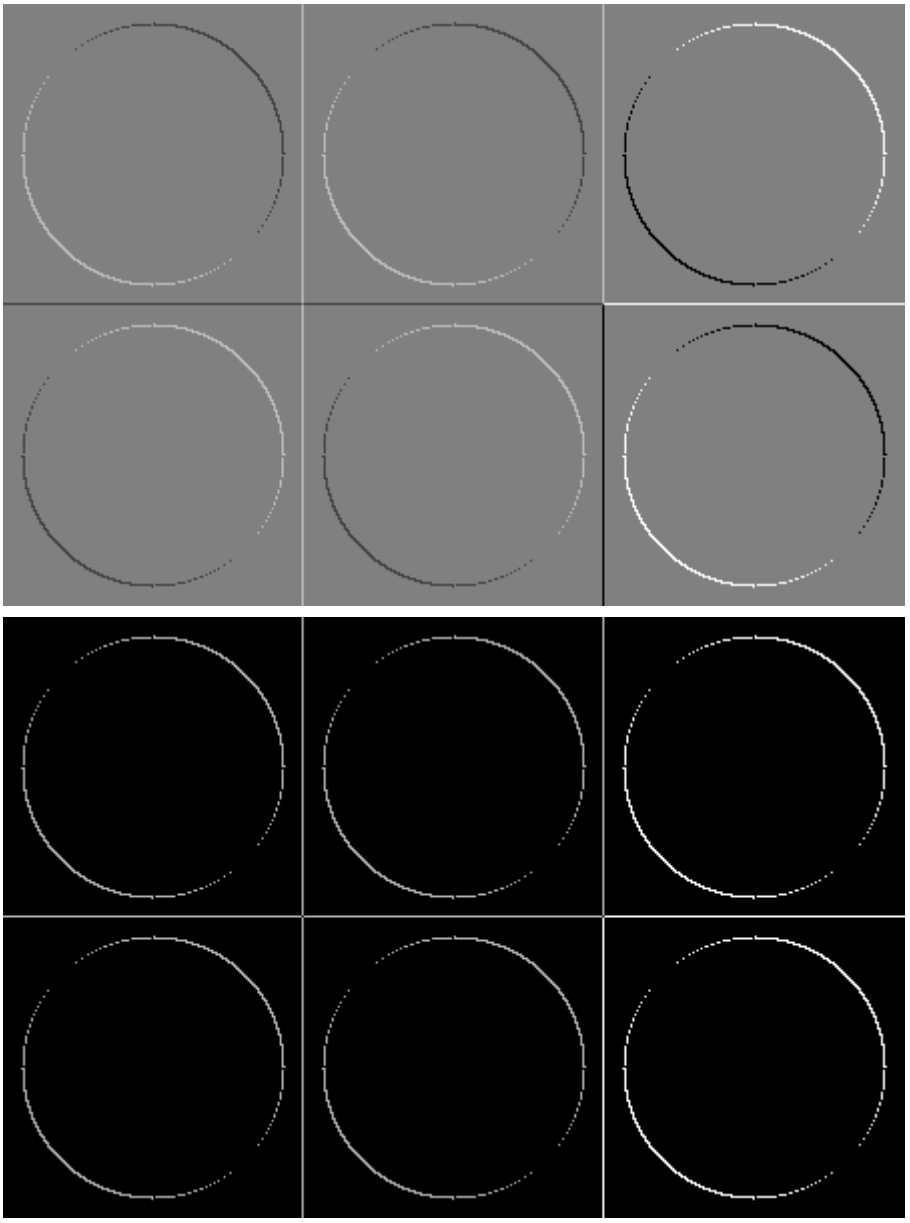
0. текст, иллюстрации и подписи отчета придумываем самостоятельно
1. нарисовать

- одноканальное изображение
- поле 2x3 из квадратных клеток 150x150px черного, белого и серого (127) цвета (соседние цвета разные)
- в цетре 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 визуалиировать геометрическое среднее (корень из суммы квадратов I1 и I2)

Результаты





Текст программы

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


int main() {
    cv::Mat screen(300, 450, CV_32FC1);
    std::vector <std::vector<float>> color = { {0, 127, 255}, {127,  255, 0} };
    int radius = 75;
    int width = 150;
    cv::Rect2d rect(0, 0, width, width);

    for (int i = 0; i < 2; i++) {
        rect.y = i * width;
        for (int j = 0; j < 3; j++) {
            rect.x = j * width;
            cv::Mat tmp = screen(rect);
            tmp = color[i][j] / 255;
            cv::circle(screen, cv::Point(rect.x + radius, rect.y + radius), radius-10, color[1 - i]
        }
    }

    cv::imshow("screen", screen);
```

```
    cv::imwrite("screen.png", screen * 255);

    cv::Mat I1(2, 2, CV_32FC1);
    I1 = 0;
    I1.at<float>(0, 1) = 1;
    I1.at<float>(1, 0) = -1;
    cv::Mat I1_filtered;
    cv::filter2D(screen, I1_filtered, -1, I1, cv::Point(0, 0));

    cv::Mat I2(2, 2, CV_32FC1);
    I2 = 0;
    I2.at<float>(1, 0) = 1;
    I2.at<float>(0, 1) = -1;
    cv::Mat I2_filtered;
    cv::filter2D(screen, I2_filtered, -1, I2, cv::Point(0, 0));

    cv::Mat middle(300, 450, CV_32FC1);
    for (int i = 0; i < middle.rows; i++) {
        for (int j = 0; j < middle.cols; j++) {
            middle.at<float>(i, j) = std::sqrt(std::pow(I1_filtered.at<float>(i, j),2)  + std::pow(
        }
    }

    I1_filtered = (I1_filtered + 1) / 2; //-1 0 1 -> 0 1 2 /2 -> 0 0.5 1
    cv::imshow("I1_filtered", I1_filtered);
    cv::imwrite("I1_filtered.png", I1_filtered * 255);

    I2_filtered = (I2_filtered + 1) / 2;
    cv::imshow("I2_filtered", I2_filtered);
    cv::imwrite("I2_filtered.png", I2_filtered * 255);

    cv::imshow("middle", middle);
    cv::imwrite("middle.png", middle * 255);

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