

Санкт-Петербургский политехнический университет Петра Великого
Институт машиностроения, материалов и транспорта
Высшая школа автоматизации и робототехники

Отчёт

по лабораторной работе №2

Дисциплина: Техническое зрение

Тема: Фильтрация изображения с использованием библиотеки OpenCV

Студент гр. 3331506/70401

Самарин А.С.

Преподаватель

Варлашин В.В.

« » _____ 2020 г.

Санкт-Петербург

2020

Задание

Реализовать фильтр Гаусса с ядром 5x5 и якорной точкой в центре. Использовать заполнение border wrap.

Описание

В фильтре Гаусса используется матрица коэффициентов, заполненных по формуле:

$$G(x, y) = \frac{1}{2 \cdot \pi \cdot \sigma^2} \cdot \exp \frac{-x^2 + y^2}{2 \cdot \sigma^2}$$

Ядро 5x5 заполненное при $\sigma = 1$ имеет вид:

$$\begin{pmatrix} G(2,2) & G(1,2) & G(0,2) & G(1,2) & G(2,2) \\ G(1,2) & G(1,1) & G(0,1) & G(1,1) & G(1,2) \\ G(0,2) & G(1,0) & G(0,0) & G(0,0) & G(0,2) \\ G(1,2) & G(1,1) & G(0,1) & G(1,1) & G(1,2) \\ G(2,2) & G(1,2) & G(0,2) & G(1,2) & G(2,2) \end{pmatrix}$$

$$\begin{pmatrix} 0.003 & 0.013 & 0.022 & 0.013 & 0.003 \\ 0.013 & 0.059 & 0.097 & 0.059 & 0.013 \\ 0.022 & 0.097 & 0.159 & 0.097 & 0.022 \\ 0.013 & 0.059 & 0.097 & 0.059 & 0.013 \\ 0.003 & 0.013 & 0.022 & 0.013 & 0.003 \end{pmatrix}$$

При изменении параметра σ будет меняться заполнение матрицы и степень размытия изображения.

Алгоритм

Алгоритм работы программы:

- Создание увеличенного изображения;
- Заполнение нового изображения (border wrap);
- Создание ядра;
- Проход ядром по новому изображению;
- Вывод полученного изображения.

Реализация

Для реализации фильтра гаусса был создан класс *gauss_blur*. Для применения фильтра необходимо вызвать функцию *gaussBlur*.

В функцию *gaussBlur* передается два изображения и параметр σ . Первый параметр *inputImage* – исходное изображение, которое необходимо подвергнуть фильтрации. Второй параметр *outputImage* – изображение для сохранения результата.

```
int gauss_blur::gaussBlur(Mat inputImage, Mat outputImage, float sigma)
{
    //создаем изображение, которое больше оригинала на половину ядра во все стороны
    Mat newInputImage(Size(inputImage.cols + 4, inputImage.rows + 4), inputImage.type(), Scalar(0, 0, 0));

    //Увеличиваем и обрабатываем края
    borderWrap(newInputImage, inputImage);

    //проходи ядром
    gaussFiltr(newInputImage, outputImage, sigma);

    return 0;
}
```

Сперва создается новое изображение, которое больше исходного на размер ядра (якорная точка в центре) по двум осям. Далее вызывается функция обработки краев. В конце идет вызов функции, в которой идет проход по изображению ядром.

Обработка краев

Для обработки краев используется функция *borderWrap*. В данной функции происходит 5 операций:

- Копирование оригинального изображения в центр;
- Заполнение левой границы нового изображения;
- Заполнение правой границы;
- Заполнение нижней границы;
- Заполнение верхней границы.

Проход ядра

Для применения ядра к изображению используется функция *gaussFiltr*. Сначала создается матрица с нормальным распределением, далее эту матрицу прикладывают к каждому пикселю.

Каждый коэффициент матрицы умножается на значение в пикселе. Далее происходит суммирование и нормирование результата. Результат записывается в якорную точку на новом изображении.

Заполнение матрицы и проход ядром реализован с помощью циклов.

Сравнение с функцией *opencv GaussianBlur*

Для сравнения функций использовались одноканальные изображения и функция *absdiff*.

В таблице 1 представлено время работы и среднеквадратичная погрешность для черно-белых изображений.

Таблица 1

размер изображения	отладка, с		релиз, с		отклонение
	моя функция	библиотечная	моя функция	библиотечная	
200x200	0,115	0,011	0,002	0,025	0,56
300x300	0,269	0,014	0,003	0,029	0,55
400x400	0,46	0,016	0,005	0,031	0,56
500x500	0,776	0,018	0,008	0,037	0,55

В таблице 2 представлено время работы и среднеквадратичная погрешность для изображений в оттенках серого.

Таблица 2

размер изображения	отладка, с		релиз, с		отклонение
	моя функция	библиотечная	моя функция	библиотечная	
800x600	1,495	0,028	0,018	0,023	0,58
1000x800	2,416	0,026	0,028	0,023	0,58
1200x800	2,982	0,031	0,031	0,024	0,58
1400x1000	4,282	0,043	0,05	0,024	0,59

Среднеквадратичная погрешность обусловлена разным округлением результатов библиотечной функции и функции *gaussFiltr*. Проанализировав результаты, можно сделать вывод, что написанный фильтр работает корректно. Но так как он не оптимизирован, что в режиме отладки он проигрывает по времени.