Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования «Кубанский государственный университет»

Кафедра вычислительных технологий

**ОТЧЕТ**

о выполнении лабораторной работы № 3

по дисциплине «Алгоритмы цифровой обработки мультимедиа»

Выполнил:

Вавакин В. О.

Проверил:

ст. преподаватель

Крамаренко А. А.

Краснодар

2025

**Ход работы**

1. Выполним построение матрицы Гаусса по формуле:

где – математическое ожидание двумерной случайной величины, в качестве которого берутся координаты центрального элемента матрицы, – индексы элементов матрицы, а – параметр размытия по Гауссу. В качестве данного параметра возьмём значение 5 и проверим построение матрицы для размерностей 3x3, 5x5 и 7x7:

*Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Содержимое, созданное искусственным интеллектом, может быть неверным.*

Рисунок 1 – Матрицы Гаусса трёх разных размерностей

2. Нормируем полученные матрицы, поделив все значения на суммы элементов матриц, а также для проверки выведем суммы элементов нормированных матриц:

Изображение выглядит как текст, снимок экрана, дизайн

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 2 – Нормированные матрицы Гаусса

3. Далее реализуем фильтр Гаусса средствами языка Python. Для этого выделим построение матрицы Гаусса с заданным параметром размытия в отдельную функцию и воспользуемся формулой операции свёртки:

где – значение яркости отдельного пикселя изображения, – матрица пикселей, в центре которой находится рассматриваемый пиксель, – ядро свёртки, в данном случае представленное матрицей Гаусса.

Проверим выполнение фильтрации Гаусса на изображении для двух различных размерностей матриц и параметров размытия:

Изображение выглядит как черно-белый, Человеческое лицо, человек, в помещении

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 3 – Оригинальное чёрно-белое изображение

Изображение выглядит как черно-белый, человек, Человеческое лицо, в помещении

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 4 – Размытие с размерностью ядра 3x3 и параметром размытия 2

Изображение выглядит как человек, Человеческое лицо, черно-белый, очки

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 5 – Размытие с размерностью ядра 7x7 и параметром размытия 20

4. Воспользуемся встроенной функцией GaussianBlur библиотеки OpenCV и сравним результаты с полученными ручным способом на тех же параметрах (размерность ядра 7x7, параметр размытия 20):

Изображение выглядит как человек, Человеческое лицо, черно-белый, очки

Содержимое, созданное искусственным интеллектом, может быть неверным.Изображение выглядит как человек, Человеческое лицо, черно-белый, очки

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 6 – Сравнение размытия Гаусса средствами Python (слева) и OpenCV (справа)

Для подсчёта относительной погрешности ручного вычисления найдём разность между средними значениями пикселей каждого из двух вариантов и поделим на среднее значение пикселей, полученных через встроенную функцию GaussianBlur:

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 7 – Относительная погрешность ручного метода по сравнению с встроенным методом GaussianBlur

**Ответы на вопросы**

1. Опишите, в чём заключается принцип операции размытия изображения и зачем её применяют?

Размытие – один из базовых алгоритмов обработки изображения. Размытие строится из предположения, что каждый пиксель картинки по своему значению связан с предыдущим и не должен заметно от него отличатся. Если же так происходит, например, вокруг пикселя яркость 240, а в самом пикселе 12, то мы предполагаем, что данный пиксель – это шумы. И размытие картинки – один из главных способов нивелировать эти шумы. Базовый подход к фильтрации строится на том, что значение пикселя должно изменится под влиянием окружающих пикселей.

2. Что такое операция свёртки матрицы?

Операция свёртки заключается в преобразовании исходной матрицы размерности в числовое значение с помощью специальной матрицы размерности , называемой ядром свёртки:

3. Каким образом строится матрица свёртки в размытии по Гауссу?

В размытии по Гаусса матрица свёртки строится по формуле:

где – математическое ожидание двумерной случайной величины, в качестве которого берутся координаты центрального элемента матрицы, – индексы элементов матрицы, а – параметр размытия по Гауссу. При разных значениях размытие будет отличаться.

4. Опишите алгоритм размытия по Гауссу.

Алгоритм размытия по Гауссу состоит из следующих шагов:

1) Установить размер матрицы свёртки и среднее квадратичное отклонение;

2) Заполнить матрицу свёртки значениями функции Гаусса с мат. ожиданием, равным координатам центра матрицы;

3) нормировать матрицу так, чтобы сумма элементов равнялась 1;

4) создать копию изображения;

5) для каждого внутреннего пикселя копии рассчитать новое значение насыщенности пикселя по формуле операции свёртки и записать это значение в пиксель нового изображения

5. Укажите известные вам параметры размытия по Гауссу.

Главными параметрами размытия по Гауссу являются размер ядра свёртки и стандартное отклонение . При их увеличении размытие становится более сильным. Стандартное отклонение может вычисляться по-разному для осей X и Y. Также на результат размытия может влиять способ обработки краёв изображения, где ядро свёртки выходит за пределы. Для этого поле с длиной, равной радиусу ядра свёртки, может заполняться константами либо отражать или копировать краевые пиксели.

6. Каким образом производить фильтрацию по Гауссу средствами библиотеки OpenCV?

В библиотеке OpenCV фильтрация по Гауссу осуществляется с помощью метода GaussianBlur, принимающего следующие параметры:

* src – исходное изображение
* ksize – размер ядра свёртки (ширина, высота)
* sigmaX – среднеквадратичное отклонение по оси X
* sigmaY – среднеквадратичное отклонение по оси Y (необязательно)
* borderType – тип обработки границ изображения, где ядро свёртки выходит за его пределы