Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра вычислительных технологий**

**ЛАБОРАТОРНАЯ РАБОТА №3**

**Дисциплина: Алгоритмы цифровой обработки мультимедиа**

Работу выполнил: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Маркарян А. С.

Направление подготовки: 02.03.02 Фундаментальная информатика и информационные технологии

Преподаватель: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Крамаренко А. А.

Лабораторная работа №3

Методы фильтрации и размытия изображения.

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 отличаться. Если же так происходит, например, вокруг пикселя яркость 240, а в самом пикселе 12, то мы предполагаем, что данный пиксель – это шум. И размытие картинки – один из главных способов нивелировать эти шумы. Базовый подход к фильтрации строится на том, что значение пикселя должно измениться под влиянием окружающих пикселей. Для начала переведем изображение в черно-белый формат, в этом случае каждый пиксель представляет собой только одно целое неотрицательное значение длиной 8 бит – яркость. Далее рассмотрим изображение как двумерную матрицу размерности ℎ × 𝑤:

Теперь рассмотрим произвольный пиксель данной матрицы 𝑎𝑖𝑗:

Значение яркости данного пикселя будем менять с учетом всех указанных пикселей вокруг заданного. Для этого необходимо выполнить операцию свертки. Операция свертки заключается в преобразовании исходной матрицы B размерности 𝑛 × 𝑛 в числовое значение с помощью специальной матрицы ker размерности 𝑛 × 𝑛, называемой ядром свертки:

Матрица ker берется с различными значениями для разных типов фильтрации, стандартное условие – данная матрица должна иметь сумму элементов близкую к единице.

Гауссовский фильтр построен на ядре свертки, полученной как значения функции Гаусса (плотность распределения) для двумерной случайной величины:

Здесь 𝑎, 𝑏 – математическое ожидание двумерной случайной величины.

Для построения матрицы в качестве этих значений выбираем координаты центрального элемента матрицы, например, для матрицы 5 × 5 значения математического ожидания будут следующими: 𝑎 = 3, 𝑏 = 3. В качестве 𝑥, 𝑦 выбираем все индексы всех элементов матрицы поочередно. Значение 𝜎 будем считать параметром размытия по Гауссу. При каждом новом значении 𝜎 результат размытия будет отличаться.

Размерность матрицы свертки – это другой возможный параметр фильтрации. Далее применяем операцию свертки ко всем внутренним пикселям изображения и записываем вычисленное по формуле (1) число в качестве нового значения яркости пикселя.

Итого алгоритм кратко будет выглядеть следующим образом:

1. установить размер матрицы свертки и среднее квадратичное отклонение;
2. заполнить матрицу свертки значениями функции Гаусса с мат.
3. ожиданием, равным координатам центра матрицы;
4. нормировать матрицу так, чтобы сумма элементов равнялась 1;
5. создать копию изображения;
6. для каждого внутреннего пикселя копии рассчитать новое значение
7. насыщенности пикселя по формуле (1) и ЗАПИСАТЬ это значение в пиксель нового изображения

**Ход работы.**

**Задание 1**. Выполнить пункты 1 и 2 алгоритма, то есть построить

матрицу Гаусса. Просмотреть итоговую матрицу для размерностей 3, 5, 7.

**Задание 2.** Нормировать полученную матрицу Гаусса. Протестировать

результаты на матрицах из предыдущего пункта.

**Задание 3**. Реализовать фильтр Гаусса средствами языка python.

**Задание 4.** Применить данный фильтр для двух разных значений

среднего квадратичного отклонения и двух разных размерностей матрицы свертки, сравнить результаты для ОДНОГО изображения.

**Задание 5.** Реализовать размытие Гаусса встроенным методом библиотеки OpenCV, сравнить результаты с Вашей реализацией.

В ходе выполнения задания 1 и 2 было необходимо построить матрицу свёртки и нормировать её, итоговая матрица для размерности 3 представлена на рисунке 1. Нормированная матрица свёртки представлена на рисунке 2.

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

Рисунок 1 – матрица свёртки

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

Рисунок 2 – нормированная матрица

В ходе выполнения 3 и 4 задания было необходимо реализовать фильтр Гаусса, тестирование на разных параметрах размерности матрицы и среднего квадратичного отклонения. В качестве примера были взяты следующие значения: (5, 3) и (3, 2), где первый параметр – размерность матрицы, а второй – среднее квадратичное отклонение. Исходное изображение представлено на рисунке 3, а результаты – на рисунках 4 и 5 соответственно.



Рисунок 3 – исходное изображение

Изображение выглядит как природа, луна, Астрономический объект, Небесное явление

Автоматически созданное описание

Рисунок 4 – пример №1

Изображение выглядит как природа, Астрономический объект, луна, Небесное явление

Автоматически созданное описание

Рисунок 5 – пример №2

В ходе выполнения задания 5 было необходимо реализовать гауссовское размытие с помощью встроенного метода библиотеки OpenCV. Итоговое изображение представлено на рисунке 5.

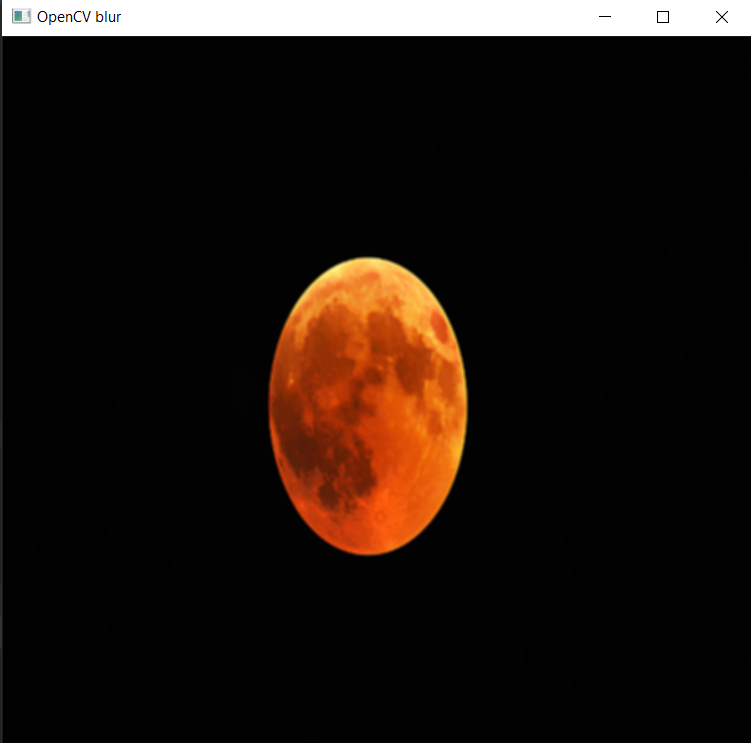


Рисунок 5 – Размытие Гаусса с помощью OpenCV

В задании 6 необходимо было реализовать размытие Гаусса средствами другого языка программирования. Выбор пал на Ruby, результат работы программы представлен на рисунке 6

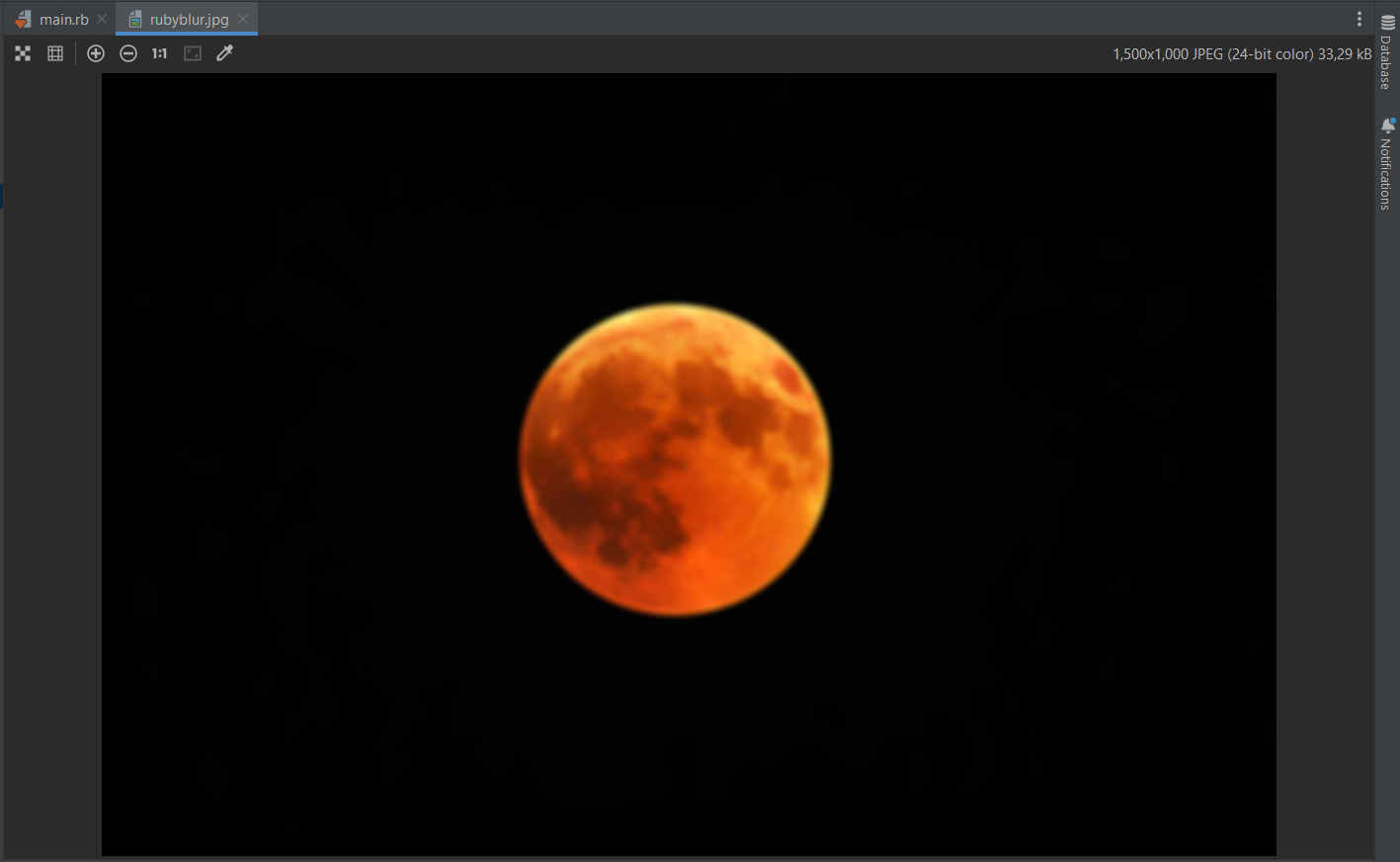


Рисунок 6 – Размытие Гаусса на Ruby