

Федеральное государственное автономное образовательное учреждение высшего
образования «Национальный исследовательский университет ИТМО»
Факультет программной инженерии и компьютерной техники

Отчёт

по Лабораторной работе №2
по дисциплине «Алгоритмы Компьютерной Графики»

Выполнил: Захарченко Роман Владимирович, группа Р3331

Преподаватель: Игорь Станиславович Потёмин

Санкт-Петербург 2025

Цель работы

Целью данной работы является расширение функциональности ядра операционной системы xv6 путем добавления новых системных вызовов для отладки процессов, а также создание пользовательской программы для демонстрации межпроцессного взаимодействия с использованием каналов(pipes).

Фильтры

1. GrayScale - преобразует цветное изображение (в модели RGB) в однотонное (серое), убирая информацию о цвете, но сохраняя информацию о яркости каждого пикселя.

Алгоритм:

$$Y=0.299R+0.587G+0.114B$$

Здесь:

- R - даёт вклад 29,9%
- G - 58,7% (самый важный, потому что глаз наиболее чувствителен к зелёному)
- B - 11,4%
- Y - это значение яркости пикселя в сером изображении.

2. Invert - преобразует каждый пиксель изображения так, что его цвет становится противоположным на цветовом круге

Алгоритм:

$$R'=255-R \quad G'=255-G \quad B'=255-B$$

Где:

- R',G',B' — новые (инвертированные) значения каналов,
- 255 — максимум яркости (в 8-битных изображениях).

3. Seria - преобразует цветное изображение в монохромное, но не чёрно-белое, а с оттенками коричнево-золотого (теплого спектра).

Алгоритм:

1) Используем алогитм GrayScale для преобразования изображения в оттенки серого

2) $R'=\min(255,Y\times0.393+G\times0.769+B\times0.189)$

$G'=\min(255,Y\times0.349+G\times0.686+B\times0.168)$

$B'=\min(255,Y\times0.272+G\times0.534+B\times0.131)$

Где:

- R',G',B' — новые значения каналов в тёплом свете

4. Posterize - уменьшает количество тоновых уровней (градаций яркости или цвета) на изображении.

Алгоритм:

Каждый канал изображения содержит значения от 0 до 255. Фильтр разделяет диапазон этих значений на несколько уровней, например 4, 8, 16 и т. д. Затем все пиксели “округляются” до ближайшего уровня. По формуле:

Где:

$$P' = \text{round} \left(\frac{P \times (L - 1)}{255} \right) \times \frac{255}{L - 1}$$

- P — исходное значение канала (0–255)
- L — количество уровней (например, 4, 8, 16)
- P' — новое (постеризованное) значение

5. Brightness - изменяет интенсивность света каждого пикселя.

Алгоритм:

$$R' = \text{clip}(R + d, 0, 255) \quad G' = \text{clip}(G + d, 0, 255) \quad B' = \text{clip}(B + d, 0, 255)$$

где:

- d — изменение яркости (может быть положительным или отрицательным),
- clip — ограничение результата в диапазоне 0–255.
- R', G', B' - новые значения каналов.

6. Contrast - изменяет динамический диапазон изображения — то есть разницу между светлыми и тёмными пикселями.

Алгоритм:

$$P' = (P - 128) \times \text{factor} + 128$$

где:

- P — исходное значение канала (0–255),
- P' — новое значение,
- factor — коэффициент контраста.

7. Blur - фильтр, который сглаживает изображение, уменьшая резкие переходы между пикселями.

Алгоритм:

Для начала мы изем матрицу весов т. е. Гауссово ядро с помощью формулы \/. Далее делим все значения на сумму ядра чтобы их сумма была равна 1.

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

Далее пробегаемся по всему ядру и получаем новое значение интенсивности пикселя по данной формуле

$$I'(x, y) = \sum_{i=-r}^r \sum_{j=-r}^r I(x + i, y + j) \cdot G(i, j)$$

В данных формулах:

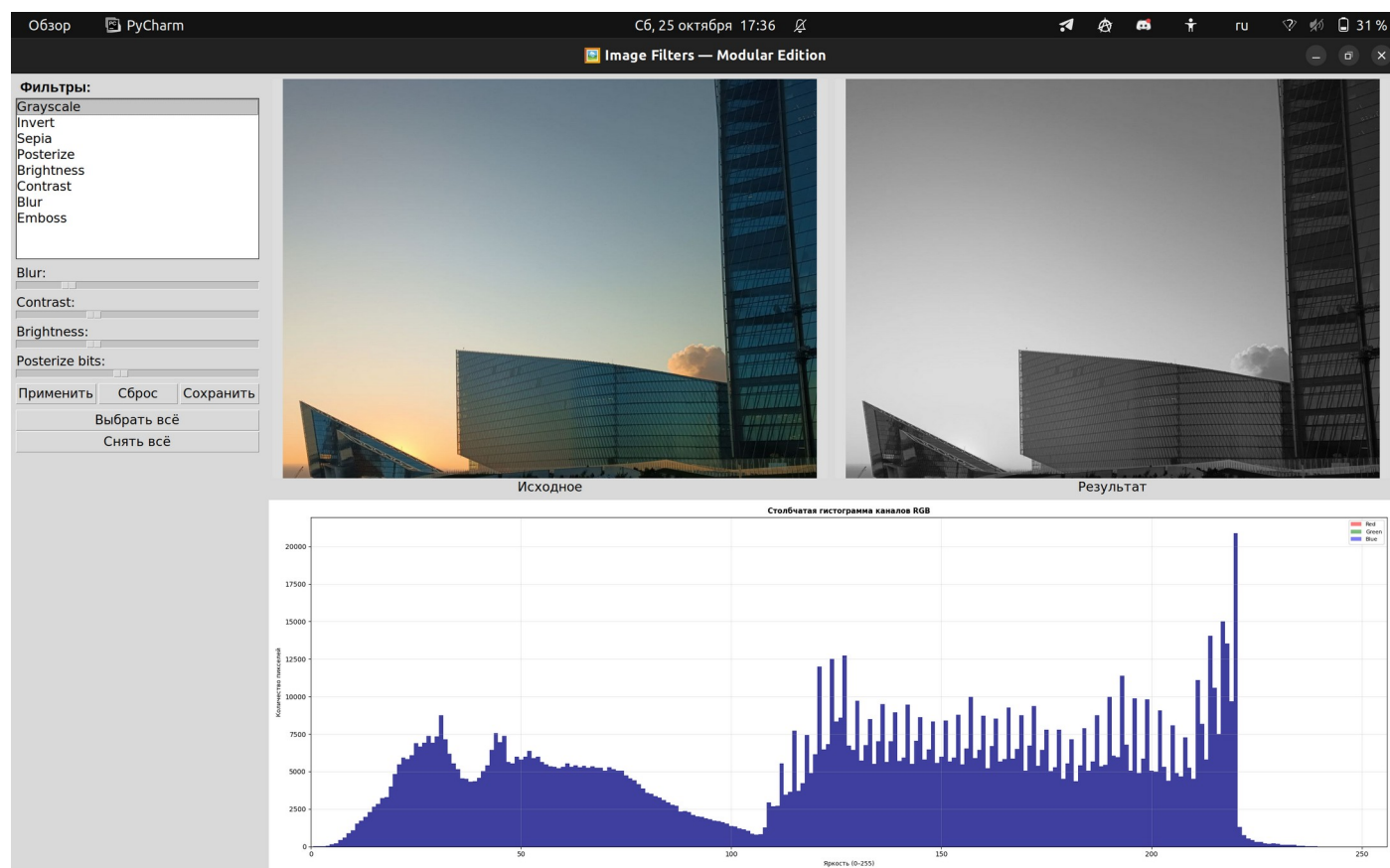
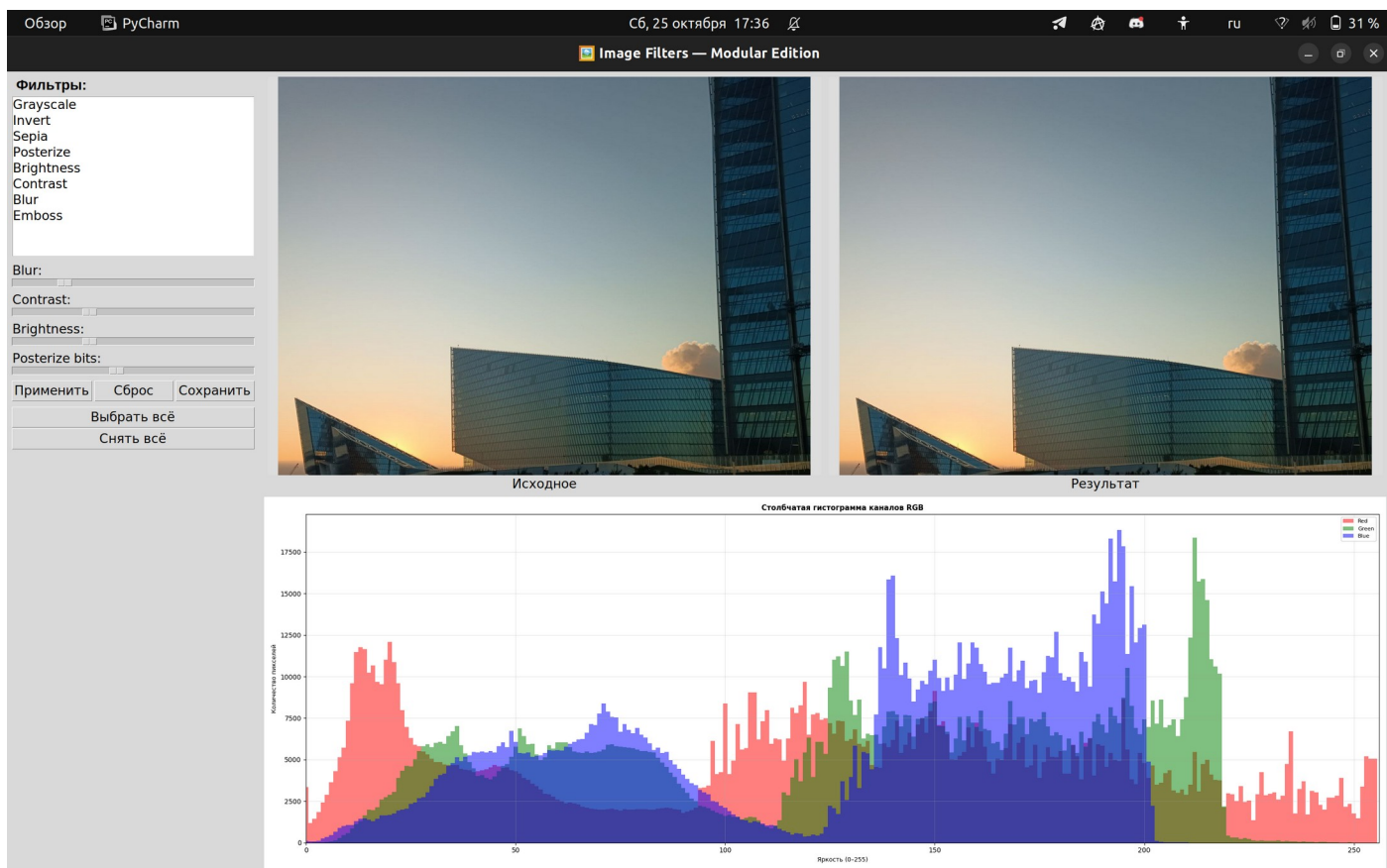
- x, y — координаты пикселя относительно центра,
- σ — сила размытия,
- n — размер окна (нечётное число)
- $r = 0.5 * (n - 1)$

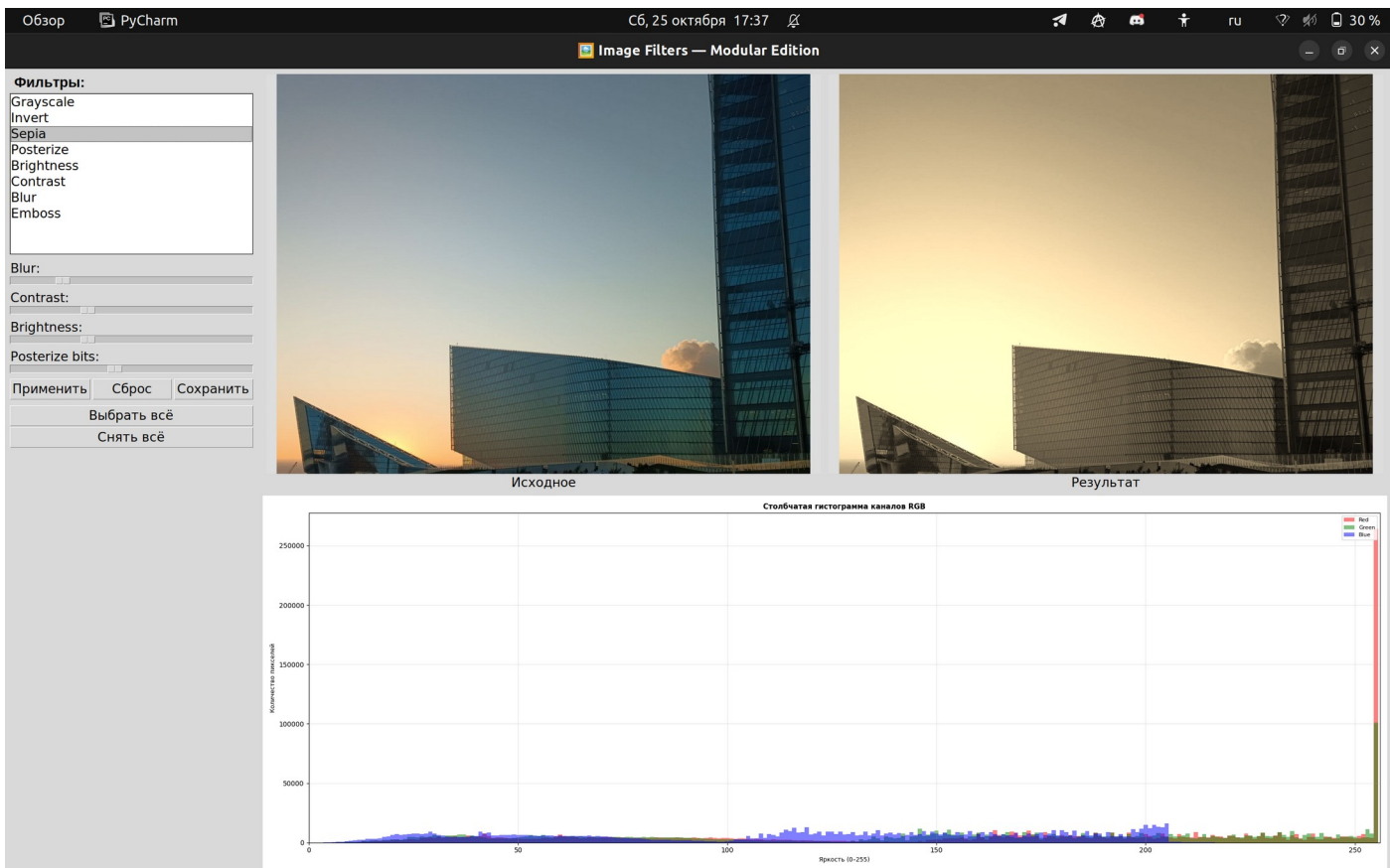
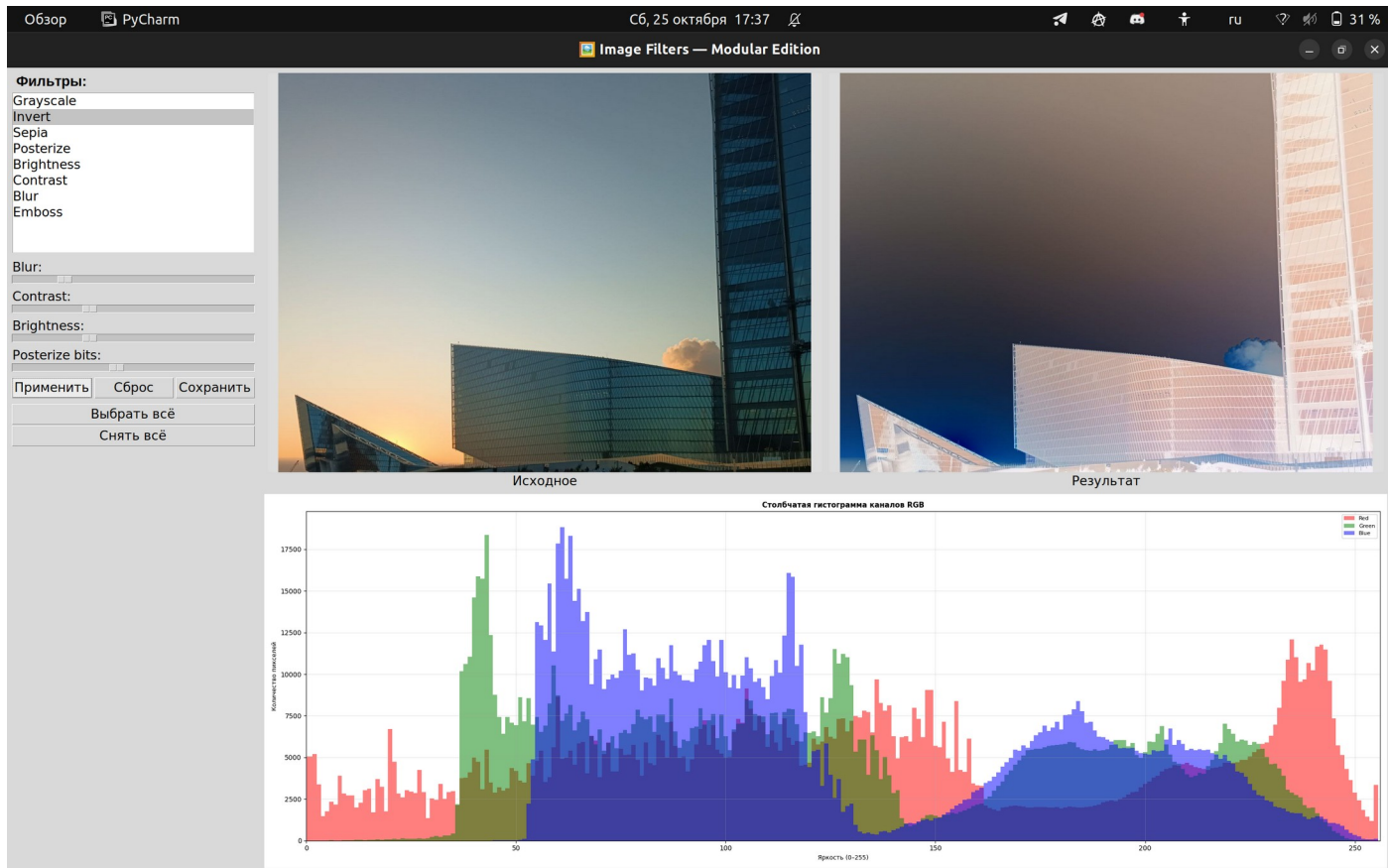
8. Emboss — этот фильтр делает из изображения барельеф, где свет падает под углом, и видны "выпуклости" и "вмятины"

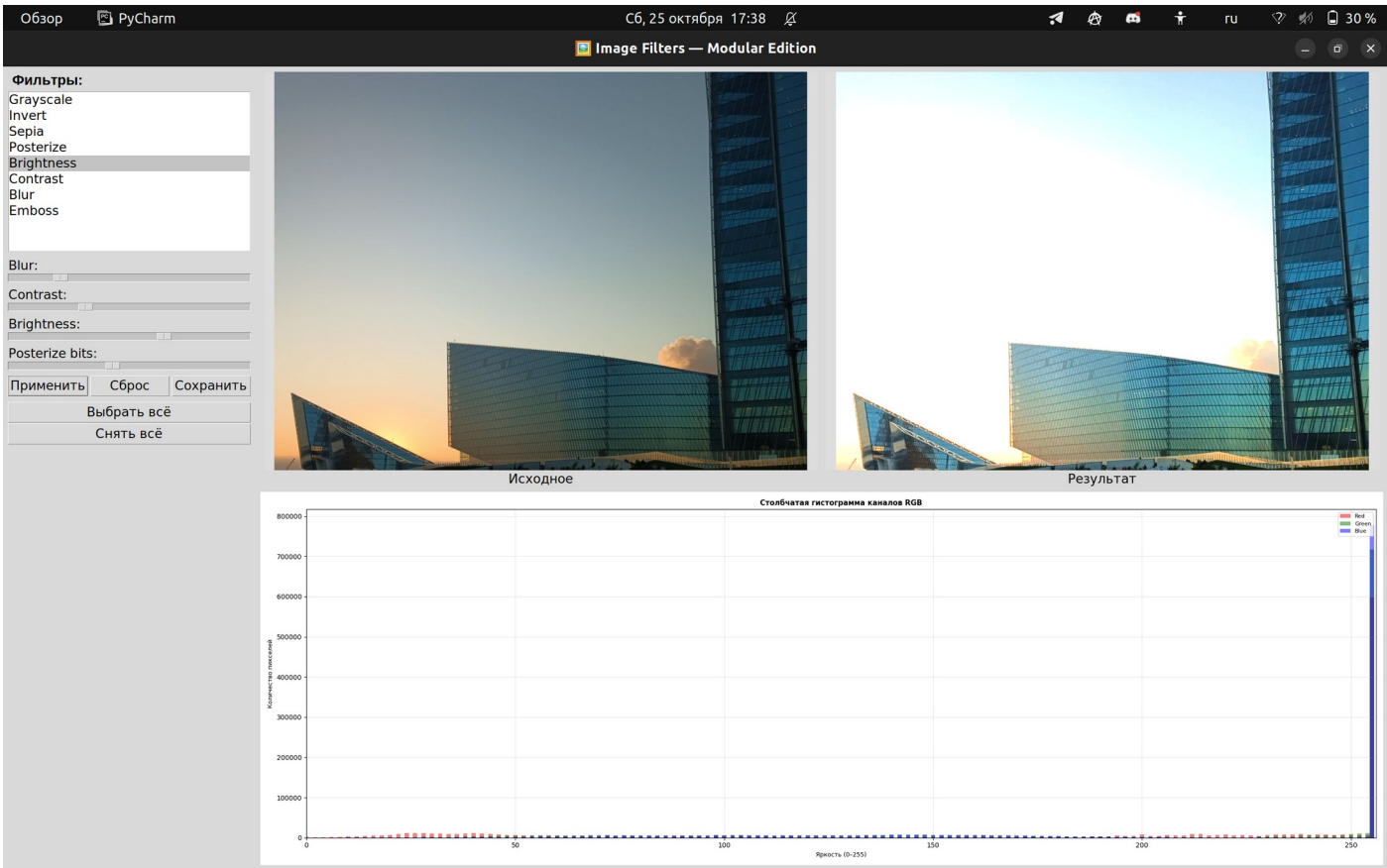
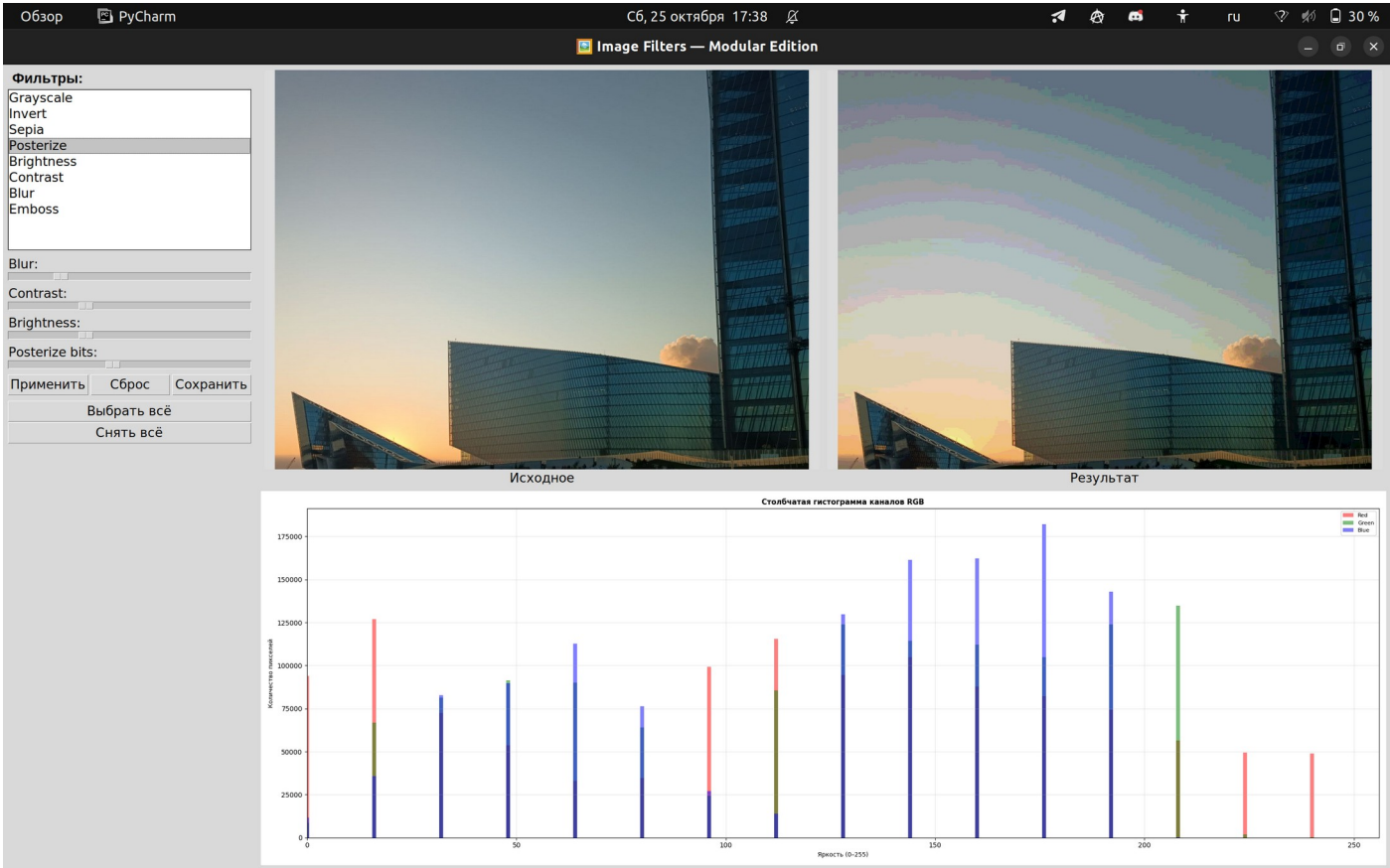
Алгоритм: аналогичен Blur только в качестве ядра используется матрица 3 на 3 которая задаёт направление отсвещения где -1 минимальный свет 1 максимальный свет

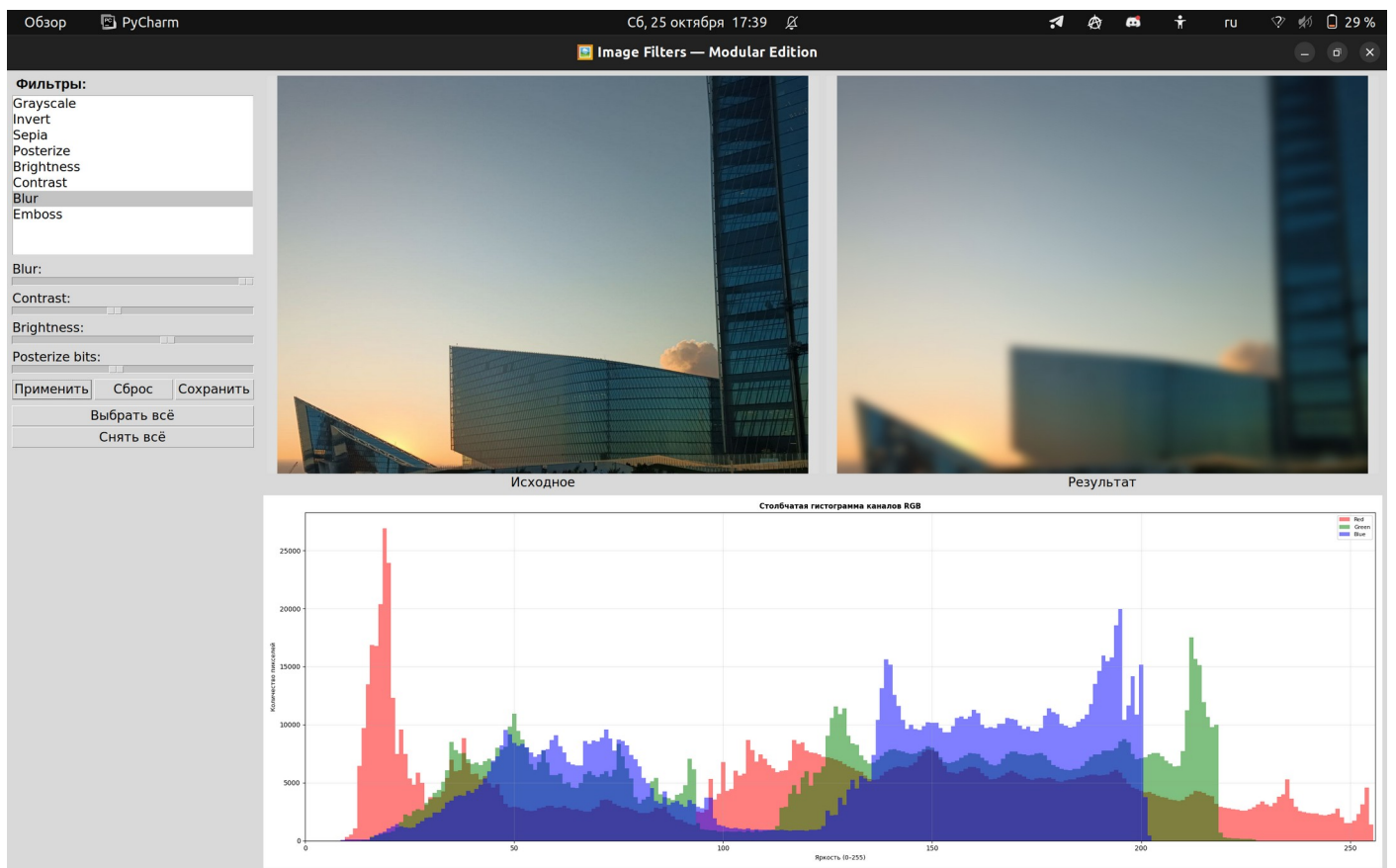
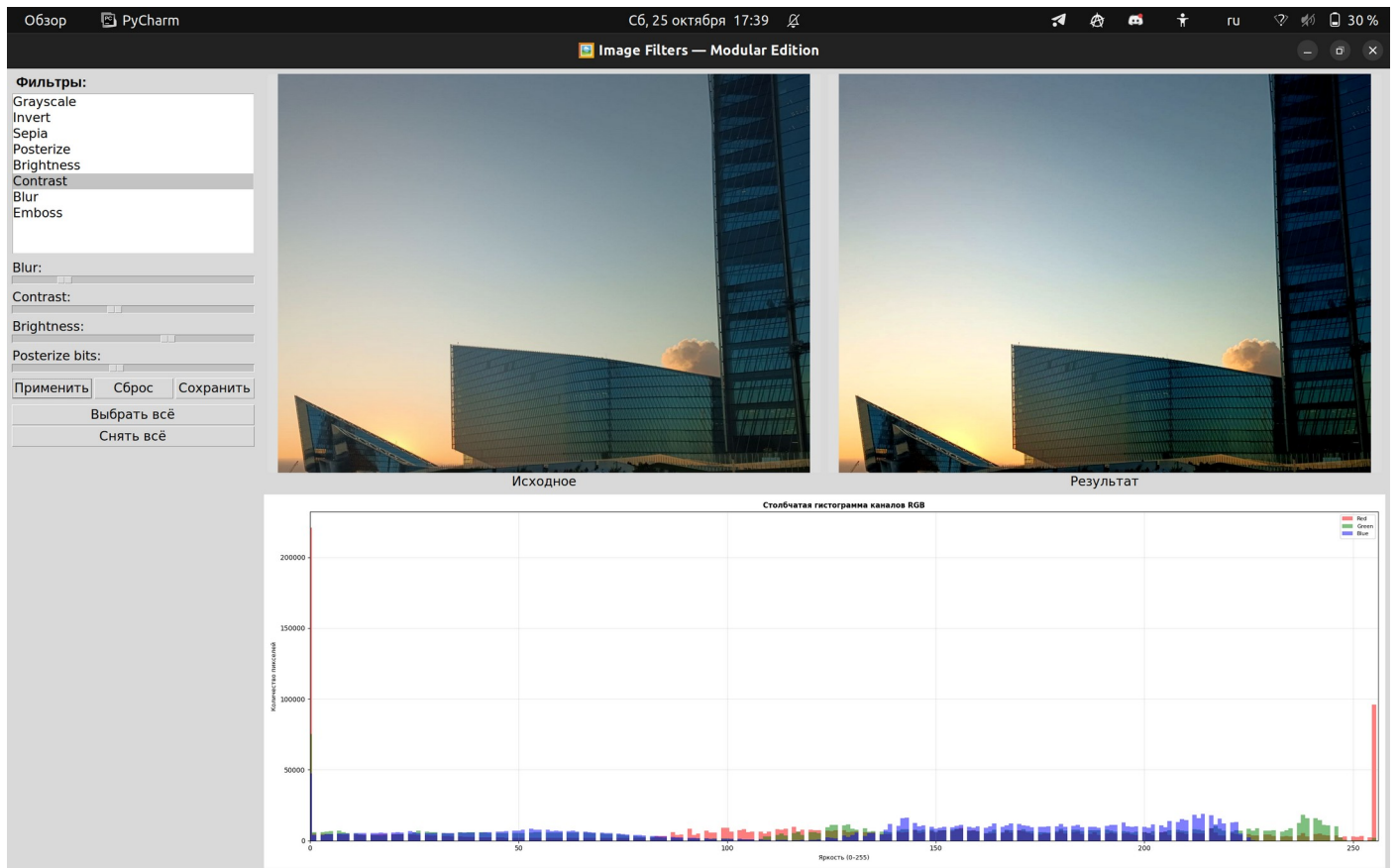
Код и результат его работы

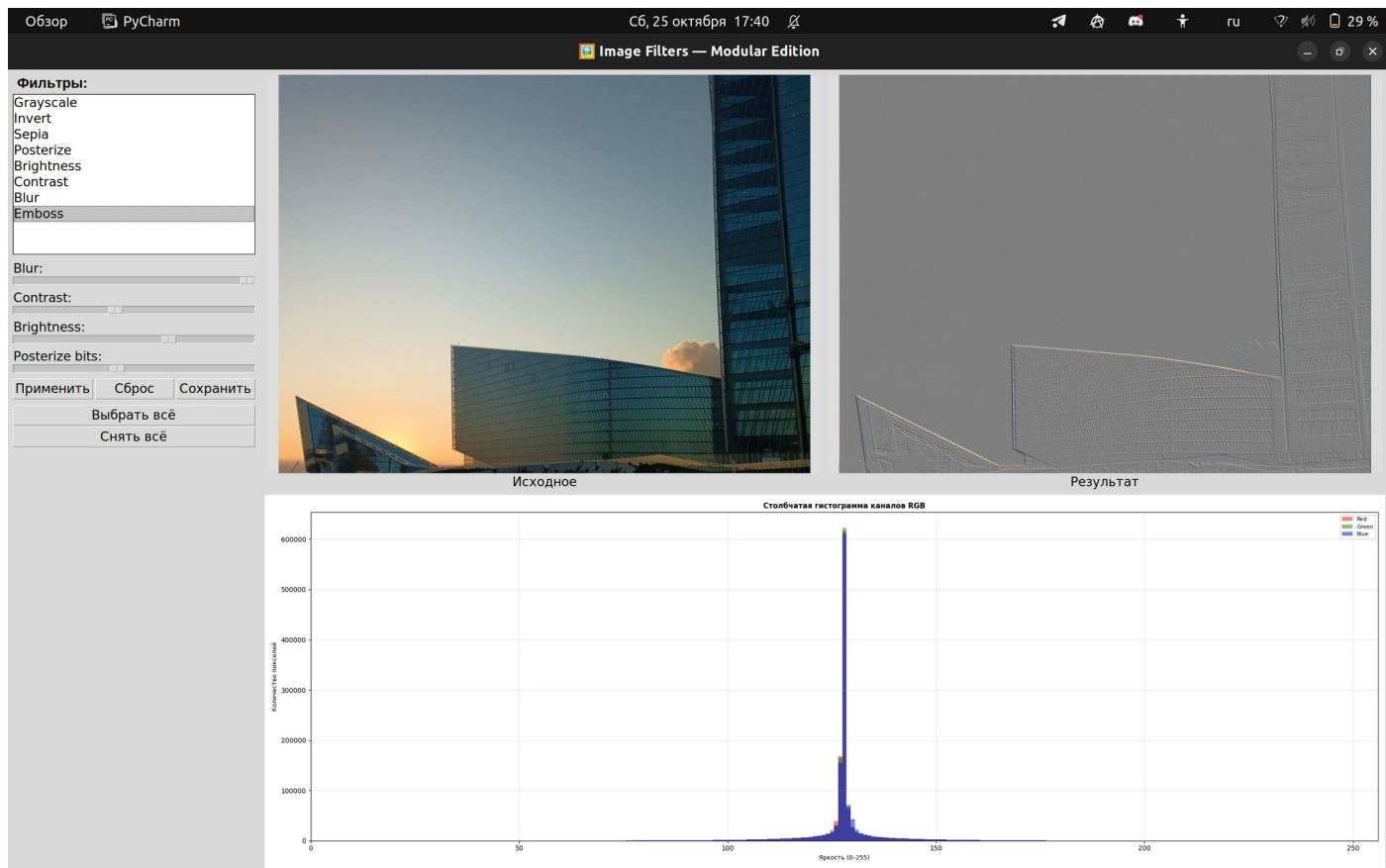
[Код программы можно посмотреть тут](#)











Вывод:

В ходе лабораторной работы узнал алгоритмы и реализовал несколько примитивных фильтров.