

A decorative graphic on the left side of the slide consisting of two overlapping parallelograms. The front one is blue and the back one is light green. They are positioned diagonally, with the blue one partially covering the green one.

# Box filter

Вадим Володин



# Описание приложения

Используется `opencv` для чтения / записи `png`

Два класса - `BoxFilterSimple` и `BoxFilterOptimized`, соответствующие им функции

Выводится время, затраченное на исполнение функции

Пример запуска: `BoxFilter lena.png lena_filtered.png`




# Вычислительная сложность

Размер картинки  $N \times M$  и размер фильтра  $S$  (в данном случае равен 5)

Простой алгоритм:

Для каждого пикселя считаем сумму пикселей в его окрестности, ее делим на площадь фильтра



	Простой алгоритм	Оптимизированный алгоритм
Асимптотика	$O(N * M * S^2)$	$O(N * M)$
Дополнительная память	$O(1)$	$O(S * M)$
Деления	$55 * NM = (5 + 2 * S * S) * NM$	$N * M$
Умножения	1	1
Суммирования	$77 * NM = (2 + 3 * S * S) * NM$	$4 * NM > 3 + 3M + N * (1 + S + 4M)$
Прибавление единицы	$31 * NM < N + NM * (1 + S + S^2)$	$5 * NM > 3M + 2N + 4 * NM$
Разница	$2 * NM$	$5 * NM < 3 + N + 5 * NM$
Сравнения	$35 * NM < N + NM * (5 + S + S^2)$	$5 * NM > 3 * (N + M) + 4 * N * M$
Время выполнения	0.1338 сек	0.0098 сек



# Оптимизированный алгоритм

Для каждой строки для каждого элемента считаем сумму  $s[i][j]$  всех элементов до этого в данной строке включительно

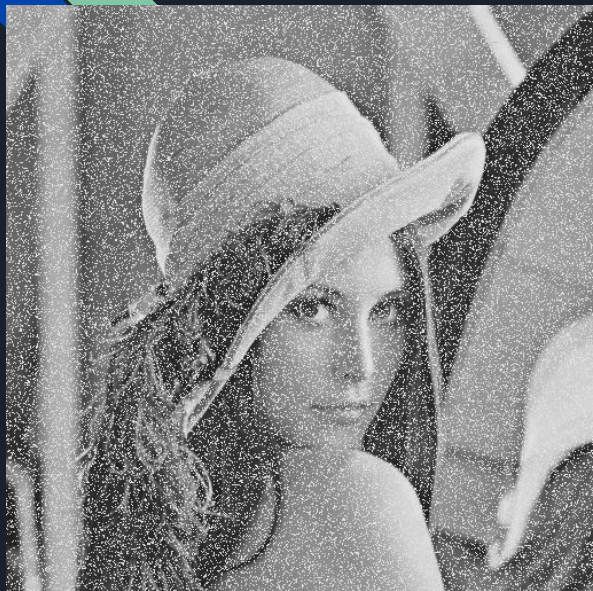
Сумма элементов в строке  $i$  от элемента  $l$  до элемента  $r$  будет равна  $s[i][r] - s[i][l - 1]$

Поддерживаем такие суммы для последних пяти строк

Также поддерживаем суммарную строку  $t$  для всех строк ( $s[0] + s[1] + s[2] + s[3] + s[4]$ )

Сумма элементов в окрестности пикселя с номером  $x$  будет равна  $t[x + 2] - t[x - 3]$

Таким образом для пикселя мы считаем сумму в его окрестности за  $O(1)$



Original



Box filter



Median filter