**Обработка изображений: повышение четкости**

Язык программирования: С++

Алгоритм: вычитание из изображения оператора Лапласа

Описание входа и выхода: 120 изображений в формате RGB(8-bit) jpg (75 Mb).

Метод сравнения качества изображений: визуальный.

Описание алгоритма:

Ссылка на репозиторий с кодом: <https://github.com/VeronikaVasilyeva/Laplacian>

Оптимизация кода: что сделано, зачем нужна, как добавить оптимизацию, как узнать, что оптимизация прошла

Методы оптимизации:

1. Изменение программного кода
2. Добавление ключей оптимизации

**делайте обход по изображениям в правильном порядке**

Двумерные изображения хранятся в памяти в виде одномерных массивов. Обычно они записываются построчно: сначала идёт 0-я строка, затем 1-я и т.д. Последовательный доступ к памяти осуществляется быстрее, чем произвольный. Поэтому обход по изображению нужно делать так, чтобы доступ к памяти был последовательный: во внешнем цикле производится обход по вертикали, а во внутреннем — по горизонтали:

for (int y = 0; y < image.Height(); y++)  
  for (int x = 0; x < image.Width(); x++)  
    ...

Также стоит быть аккуратным при использовании двумерных массивов в C# (а лучше вообще их не использовать, написав класс-обёртку для одномерных массивов): в них первый индекс — это строка (Y-координата), а вторая — столбец (X).

**Результаты запуска:**

1. Плохой и хороший код решает задачу корректно. На снимках визуально заметно увеличение качества изображения, объекты становятся более яркими, линии становятся четче. Результаты можно посмотреть в папке DataSet, каждое изображение именовано соответствующим именем программы, также приложены исходные изображения. Сравнив качество работы моих двух программ с результатом OpenCV, результат OpenCV иногда лучше.
2. Среднее время работы программы:

|  |  |
| --- | --- |
| Плохой код | Хороший код |
| sec | sec |