

РЕАЛИЗАЦИЯ АЛГОРИТМА ВОССТАНОВЛЕНИЯ «ФИЛЬТР ВИНЕРА» НА PYTHON

Рассматривается алгоритм восстановления «Фильтр Винера» и его реализация на языке Python.

ВВЕДЕНИЕ

Во время получения изображений они подвергаются различным дефектам. Проблема дефокусировки и потери качества изображений является часто встречаемой при работе как с 2D, так и с 3D графикой. Восстановление изображений является актуальной научно-практической проблемой, по причине применения изображений при построении математических, информационных и графических моделей.

I. АЛГОРИТМ «ФИЛЬТР ВИНЕРА»

Существуют подходы, учитывающие наличие шума на изображении. Один из самых известных подходов основан на применении фильтра Винера. Фильтр Винера приближает обратный фильтр в тех значениях частот, для которых отношение сигнал/шум принимает большие значения и подавляет частоты с малым отношением, базируется на нахождении такой функции $F(u,v)$, которая минимизирует средний квадрат отклонения оценки от неискаженного изображения. Оптимальную оценку в частотной области $F(u,v)$ находят по формуле [1].

$$\frac{G(u,v)|H(u,v)|^2}{H(u,v)|H(u,v)|^2 + S_n(u,v)/S_f(u,v)} \quad (1)$$

где $S_n(u,v)$ — спектральная плотность шума; $S_f(u,v)$ — спектральная плотность исходного изображения. В случае отсутствия шума его спектральная плотность равна нулю, тогда этот метод эквивалентен инверсной фильтрации.

Цветное изображение с разрешением $M \times N$ воспринимается в данном случае как дискретная случайная величина, где каждый пиксель характеризуется интенсивностью красной, зеленой и синей составляющих от 0 до 1.

II. РЕАЛИЗАЦИЯ НА PYTHON

Для технической реализации алгоритма «Фильтр Винера» на Python необходимо подключение библиотеки для одномерного дискрет-

ного преобразования Фурье `numpy.fft`, библиотеки инструментов для обработки сигналов и изображений `scipy.signal` и подмодуль `ruplot` библиотеки `matplotlib`.

Для визуального представления работы фильтра Винера с различными дефектами изображения были написаны функции `blur(imgg, kernelSize)`, `addGaussianNoise(imgg, sigma)`, `rgb2gray(rgb)`, `gaussianKernel(kernelSize)` и `wienerFilter(imgg, kernel, K)`.

Функция `blur` применяет размытие к изображению, используя заданное ядро размера `kernelSize`. Функция `addGaussianNoise` генерирует гауссовский шум с заданным параметром `sigma` и добавляет его к изображению. Функция `rgb2gray` преобразует цветное изображение в оттенки серого. Функция `gaussianKernel` создает гауссовское ядро заданного размера. Функция `wienerFilter` применяет фильтр Винера к изображению, имеющему дефекты, с использованием заданного ядра и параметра `K`. В основной части кода происходит загрузка исходного изображения, конвертирование в оттенки серого, затем применяются последовательно размытие, добавление гауссовского шума и фильтр Винера с последующим выводом результатов на экран.

Восстановленное изображение отличается от идеального исходного изображения, но заметно улучшение качества исходного изображения с размытием и шумом.

III. ВЫВОДЫ

Полученные результаты показали эффективность и быстроту работы алгоритма «Фильтр Винера» в улучшении качества изображения. Это открывает новые перспективы для применения данного метода в различных областях, где необходимо поддерживать высокое качество изображений.

1. Bovik, The Essential Guide to Image Processing / A. Bovik// – 2009. – С. 170-236.
2. Billauer, Python Digital Signal Processing / E. Billauer// – 2017. – С. 146.

Вербицкая Вероника Игоревна, студент 3 курса факультета информационных технологий и управления БГУИРа, veron.itgame@gmail.com.

Коршикова Дарья Валерьевна, ассистент кафедры вычислительных методов и программирования БГУИР, korshikova@bsuir.by.

Купчина Екатерина Валерьевна, инженер кафедры вычислительных методов и программирования БГУИР, e.kupchina@bsuir.by.

Научный руководитель: Кукин Дмитрий Петрович, заведующий кафедры вычислительных методов и программирования БГУИР, кандидат технических наук, доцент kukin@bsuir.by.