# Анализ изображений и видео: Cеминар 2

# Методы обработки изображений

1. Линейное градационное преобразование вида T(r)=a\*r + b
   1. Рейлизуйте самостоятельно функцию градационного преобразование вида T(r)=a\*r + b (обход всех пикселей изображения). Примените релизованную функцию к изображению Lena.jpg (из папки OPENCV\_INSTALL \_DIR\samples\cpp) с параметрами a = 2, b= 100. Примените к исходному изображению стандартную функцию *convertTo* с теми же параметрами преобразования. Постройте разность полученных результатов, убедитесь что результаты совпадают – найдите максимальный элемент полученной разности (для всех цветовых каналов) и выведите его.
   2. Примените градационное линейное преобразование T(r)=a\*r + b к изображению Lena.jpg со следующими наборами параметров:
      1. a = 0.5, b=0
      2. a = 1, b = -125
      3. a = 2, b = 0
      4. a = 2, b = -255
      5. a = 1, b = 125.

Состыкуйте (при помощи конкатенации по всем трем каналам) результаты для вариантов 1) и 2) и для вариантов 3), 4), 5). Сохраните два полученных состыкованных изображения.

<http://docs.opencv.org/2.4/doc/tutorials/core/basic_linear_transform/basic_linear_transform.html>

1. Загрузите изображение Lena.jpg. Переведите изображение в градации серого. Примените к серому изображению операцию линейного растяжения диапазона яркостей (*normalize*) и эквализации гистограммы (*equalizeHist*). Постройте разность полученных результатов. Состыкуйте в одно изображение в градациях серого результат линейного растяжения, эквализации и их разности. Сохраните полученное состыкованное изображение.

<http://docs.opencv.org/doc/tutorials/imgproc/histograms/histogram_equalization/histogram_equalization.html>

1. Реализуйте функцию добавления шума «соль и перец» для одноканального изображения. Входные параметры: src – исходная матрица Mat с элементами типа CV\_8UC1, p – вероятность замещения исходных значений на 0, q – вероятность замещения исходных значений на 255. Загрузите изображение Lena.jpg, выделите каналы изображения в отдельные матрицы (*split*), добавьте шум «соль и перец» в каждый канал при помощи реализованной Вами функции с параметрами 1) p=q=0.05, 2) p=q=0.1, 3) p=q=0.15. Сохраните полученные изображения.
2. Реализуйте функцию добавления аддитивного белого гауссова шума для одноканального изображения. Входные параметры: src – исходная матрица Mat с элементами типа CV\_8UC1, mean – математическое ожидание генерируемого шума, stedev – стандартное отклонение генерируемого шума. Для генерации шума можно воспользоваться функцией *randn*. Загрузите изображение Lena.jpg, выделите каналы изображения в отдельные матрицы (*split*), добавьте белый гауссов шум в каждый канал при помощи реализованной Вами функции с параметрами 1) mean=0, stddev=30, 2) mean=0, stddev=60, 3) mean=50, stddev=30. Сохраните полученные изображения.
3. Для каждого изображения, полученного в заданиях 3 и 4, примените следующие сглаживающие фильтры с размерами ядра 3х3, 5х5 и 7х7:
   1. Линейное усреднение (normalized box filter) – функция *blur*
   2. Гауссов фильтр – функция *GaussianBlur*
   3. Медианный фильтр – функция *medianBlur*

Для каждого исходного изображения состыкуйте в одно изображение полученные девять результатов фильтрации. Расположите результаты фильтрации различными фильтрами по столбцам, а результаты фильтрации с различными размерами ядра по строкам (см таблицу внизу). Сохраните состыкованные изображения для каждого исходного изображения (должно получится 6 состыкованных изображений).

|  |  |  |
| --- | --- | --- |
| Blur, 3х3 | Blur, 5x5 | Blur, 7x7 |
| GaussianBlur, 3x3 | GaussianBlur, 5x5 | GaussianBlur, 7x7 |
| MedianBlur, 3x3 | MedianBlur, 5x5 | MedianBlur, 7x7 |

<http://docs.opencv.org/doc/tutorials/imgproc/gausian_median_blur_bilateral_filter/gausian_median_blur_bilateral_filter.html>

<http://docs.opencv.org/doc/tutorials/imgproc/imgtrans/filter_2d/filter_2d.html>

1. Загрузите изображение Lena.jpg (512х512). Переведите изображение в градации серого. Произведите фильтрацию данного изображения в частотной области идеальным фильтром низких частот (фильтр, обрезающий все высокочастотные составляющие фурье-образа, находящиеся на большем расстоянии от начала координат, чем некоторое заданное расстояние D), идеальным фильтром высоких частот (фильтр, обрезающий все низкокочастотные составляющие фурье-образа, находящиеся на меньшем расстоянии от начала координат, чем некоторое заданное расстояние D) со значениями радиуса фильтров 1) 5, 2) 15, 3) 30, 4) 255, 5) 380. Для каждого фильтра постройте состыкованное изображение из следующих: исходное изображение в градациях серого, изображение Фурье-спектра (после логарифмической коррекции для лучшей визуализации), изображение Фурье-спектра после фильтрации, изображение в пространственной области после фильтрации. На выходе должно получиться 10 изображений (5 параметров фильтров х 2 типа фильтров).

Функции *dft, mulSpectrums, idft*

<http://paulbourke.net/miscellaneous/imagefilter/>