

Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»

Отчет по лабораторной работе №1
«Гистограммы, профили и проекции»
По дисциплине «Техническое зрение»

Выполнил: Курчавый В. В.,

студент группы R3338

Преподаватель: Шаветов С. В.,

канд. техн. Наук, доцент ФСУ и Р

Санкт-Петербург, 2022

Цель работы

Освоение основных яркостных и геометрических характеристик изображений и их использование для анализа изображений.

Теоретическое обоснование применимых методов

? объяснить почему равномерное преобразование выдает линейную кумулятивную гистограмму.

Ход выполнения работы

1. Гистограммы

1.1 Исходные данные

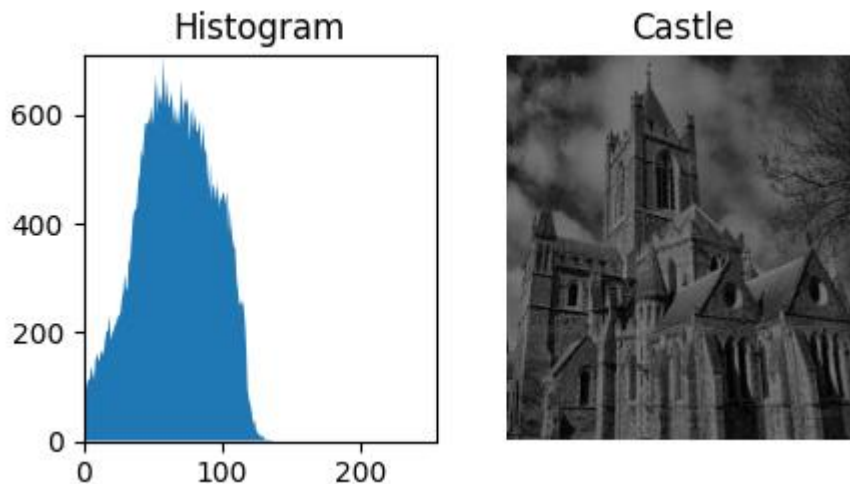


Рисунок 1 Исходное изображение

Контрастность изображения:

$$K = I_{max} - I_{min} = 137 - 0 = 137$$

Изображение темное с небольшим контрастом, из-за этого гистограмма смещена влево.

Листинг 1 Считывание изображения и построение гистограммы

```
img = cv.imread(PATH_TO_PHOTO, cv.IMREAD_GRAYSCALE)
hist = cv.calcHist([img], [0], None, [256], [0, 256])
```

Листинг 2 Совместное изображение гистограммы и изображения

```
fig, (h, i) = plt.subplots(1, 2, figsize=(5, 2.5))
h.set_title('Histogram')
h.set_xlim([0, 255])
h.set_ylim([0, int(np.max(hist))])
h.fill_between(list(range(256)), 0, list(map(int, hist)))
i.set_title('Castle')
i.imshow(img, cmap='gray', vmin=0, vmax=255, aspect='auto')
i.axis('off')
plt.show()
```

1.2 Линейное преобразование

Увеличим каждое значение интенсивности на 60, чтобы изображение стало более светлым.

$$I_{new}(x, y) = I(x, y) + 60$$

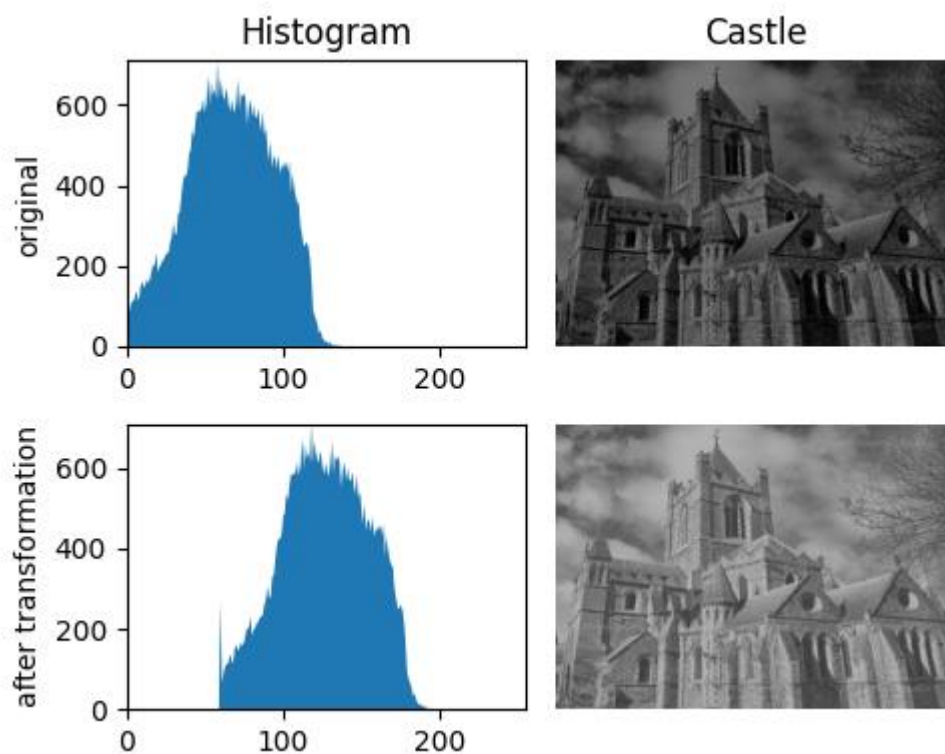


Рисунок 2 Влияние линейного сдвига

Как и ожидалось изображение стало более светлым, но при этом не осталось очень темных тонов.

Листинг 3 Функция линейного сдвига

```
def linear_transform(img, shift=0):
    return img + shift
```

1.3 Растяжение динамического диапазона.

Применим растяжение динамического диапазона с параметром с различными α .

$$I_{new} = \left(\frac{I - I_{min}}{I_{max} - I_{min}} \right)^\alpha$$

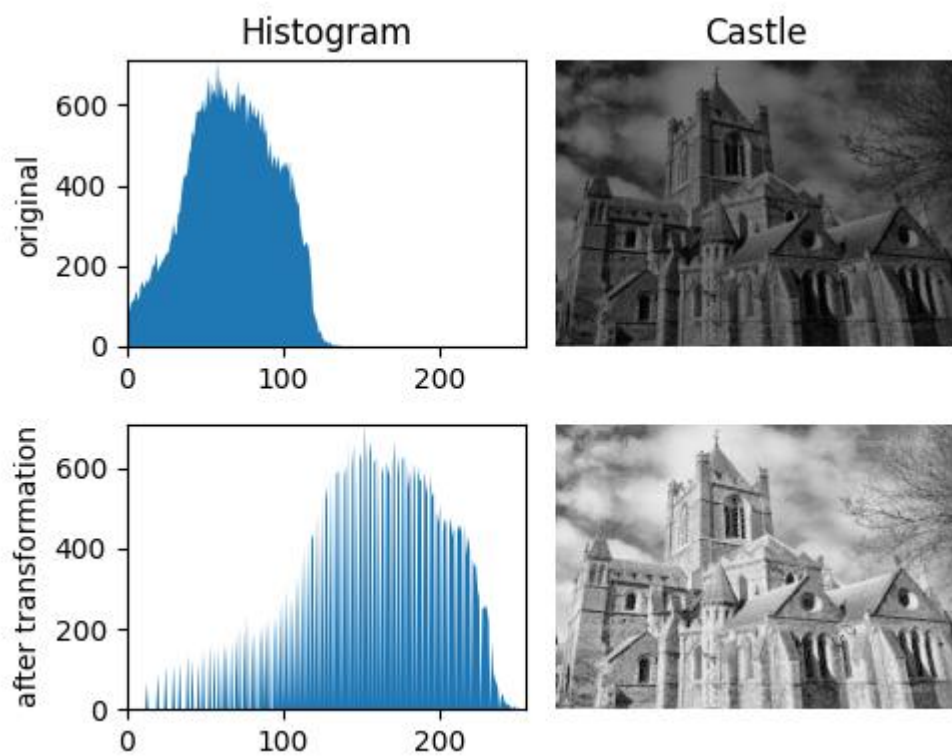


Рисунок 3 Влияние динамического растяжения с $\alpha = 0.6$

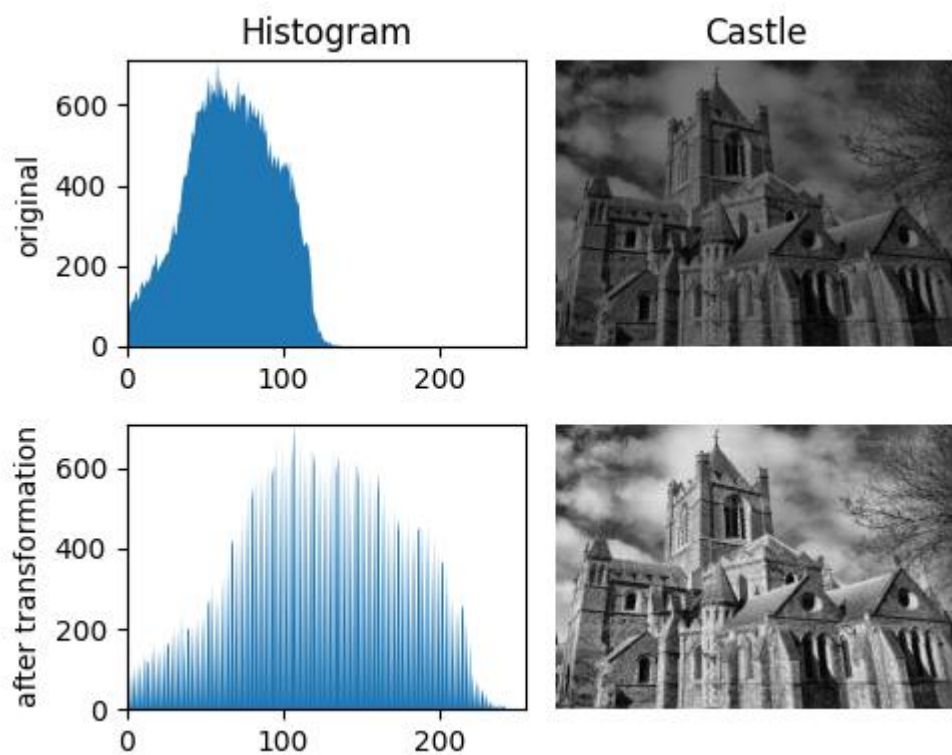


Рисунок 4 Влияние динамического растяжения с $\alpha = 1$

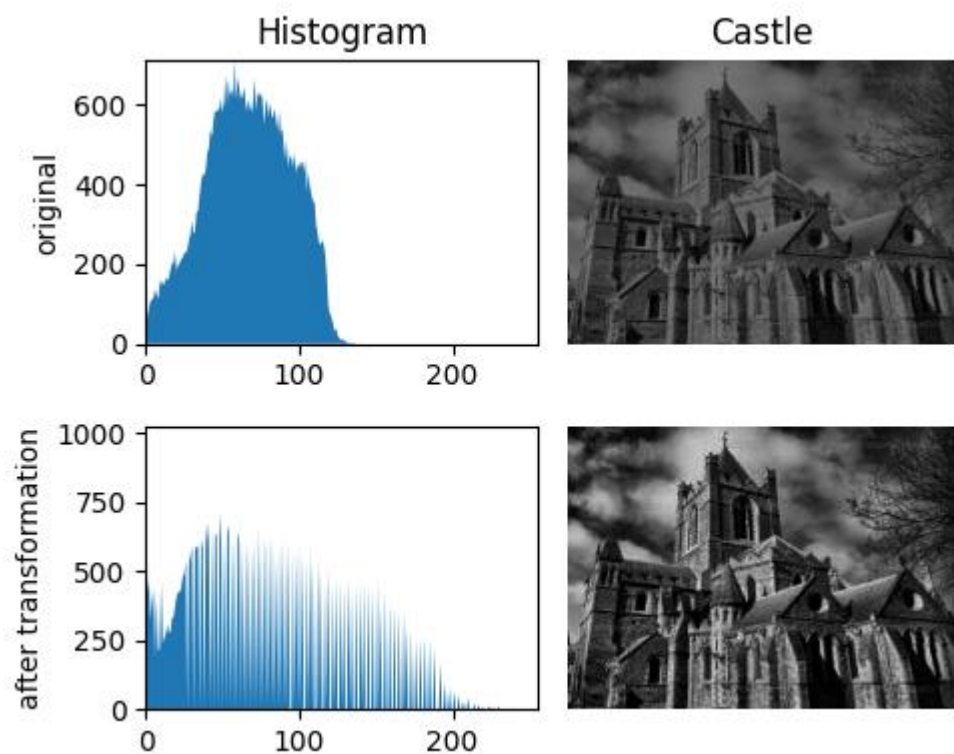


Рисунок 5 Влияние динамического растяжения с $\alpha = 1.9$

Можно заметить, что при увеличении α картинка становится темнее, так как значения интенсивностей становятся меньше. При этом изображения становятся контрастнее.

Листинг 4 Функция динамического растяжения

```
def stretching_tranform(img: ndarray, a: float = 1) -> ndarray:
    i_max = np.max(img)
    i_min = np.min(img)
    return (255*(np.power((img - i_min)/(i_max - i_min),
a))).astype(np.uint8)
```

1.4 Равномерное преобразование

Применим равномерное преобразование для исходного изображения:

$$I_{new} = (I_{max} - I_{min}) * P(I) + I_{min}$$

$$P(I) = \sum_{m=0}^I Hist(I)$$

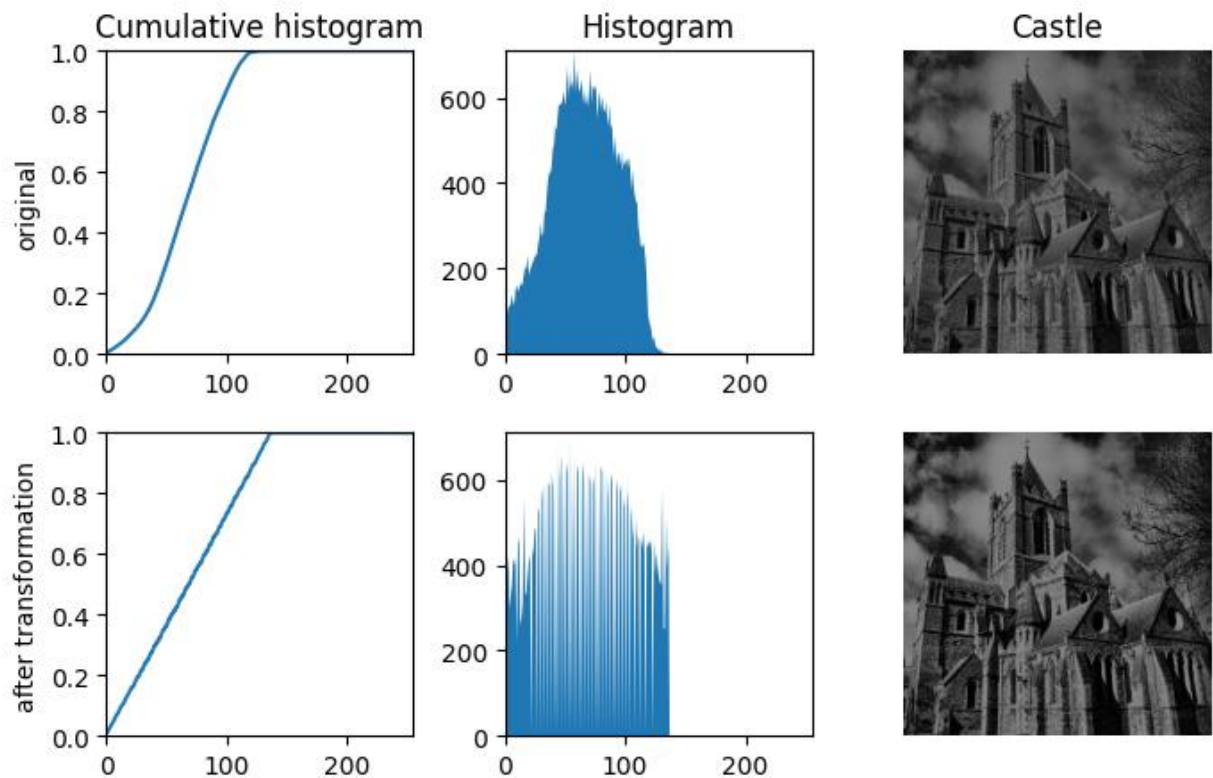


Рисунок 6 Влияние равномерного преобразования

Можно заметить, что кумулятивная гистограмма стала линейно расти, а количество различных значений интенсивностей стало менее разбросанным. Контраст изображения не поменялся.

Листинг 5 Функция расчета нормированной кумулятивной гистограммы

```
def cum_histogram(hist, num_rows, num_column):
    return np.cumsum(hist) / (num_rows * num_column)
```

Листинг 6 Функция расчета равномерного преобразования.

```
def uniform_transform(img: ndarray, cum_hist: ndarray) -> ndarray:
    i_max, i_min = np.max(img), np.min(img)
    new_img = ndarray(img.shape)
    for x in range(img.shape[0]):
        for y in range(img.shape[1]):
            new_img[x][y] = (i_max - i_min) * cum_hist[img[x][y]] + i_min
    return new_img.astype(np.uint8)
```

1.5 Экспоненциальное преобразование

Вычислим новое значение интенсивностей по формуле:

$$I_{new} = I_{min} - \frac{1}{\alpha} \ln(1 - P(I))$$

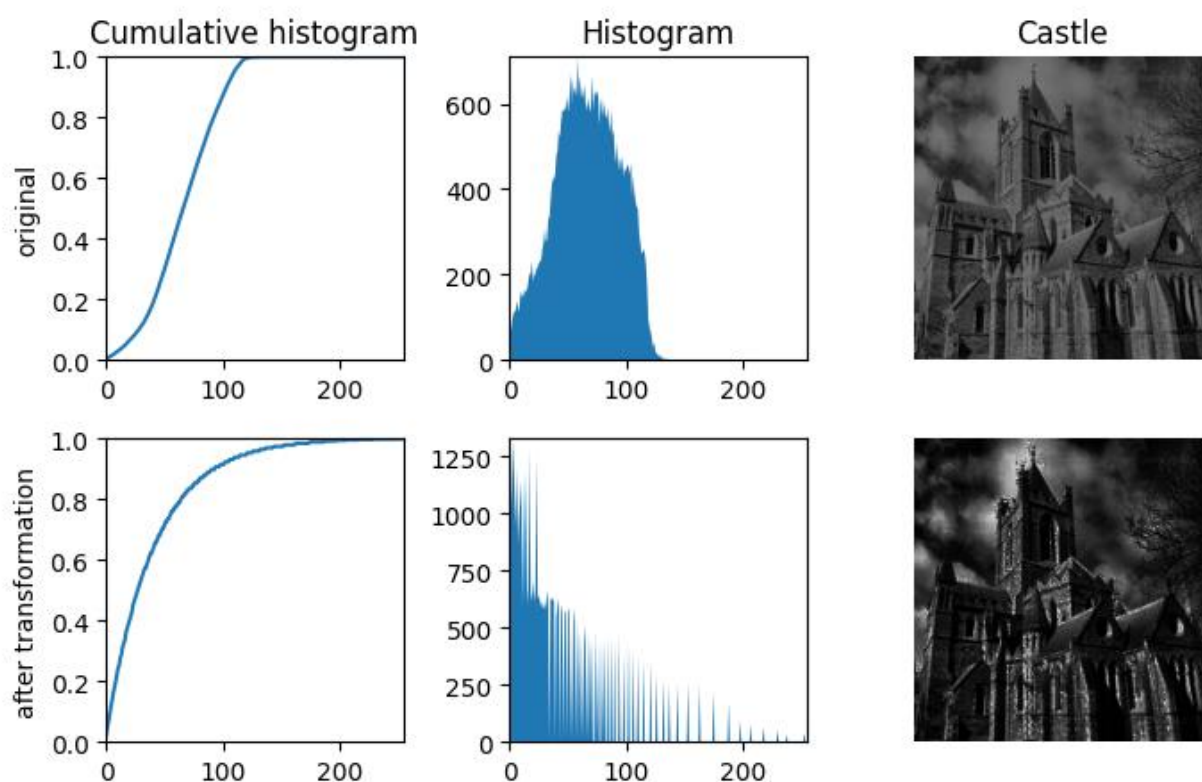


Рисунок 7 Влияние экспоненциального преобразования при $\alpha = 0.025$

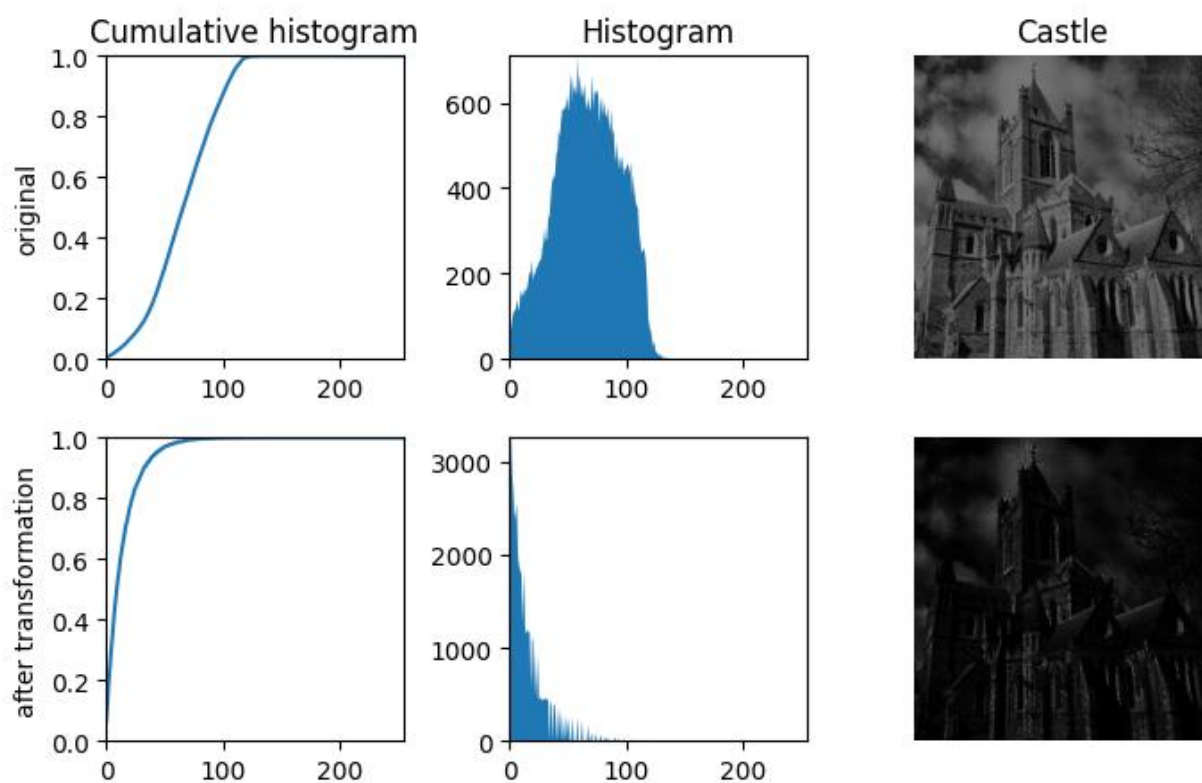


Рисунок 8 Влияние экспоненциального преобразования при $\alpha = 0.07$

Можно заметить, что, чем меньше α , тем с более меньшей скоростью растет кумулятивная гистограмма. Кумулятивная гистограмма растет по экспоненте. Значения на обычной

гистограмме сосредоточены близко к нулю, поэтому фотографии получаются в темных тонах.

Листинг 7 Функция расчета экспоненциального преобразования.

```
def exponential_transform(img: ndarray, cum_hist: ndarray, a) -> ndarray:
    i_min = np.min(img)
    new_img = ndarray(img.shape)
    for x in range(img.shape[0]):
        for y in range(img.shape[1]):
            new_img[x][y] = i_min - 1/a * np.log(1 - cum_hist[img[x][y]])
    return new_img.astype(np.uint8)
```

1.6 Преобразование по закону Рэля

Вычислим новое значение интенсивностей по формуле:

$$I_{new} = I_{min} + \left(2\alpha^2 \ln \frac{1}{1 - P(I)} \right)^{\frac{1}{2}}$$

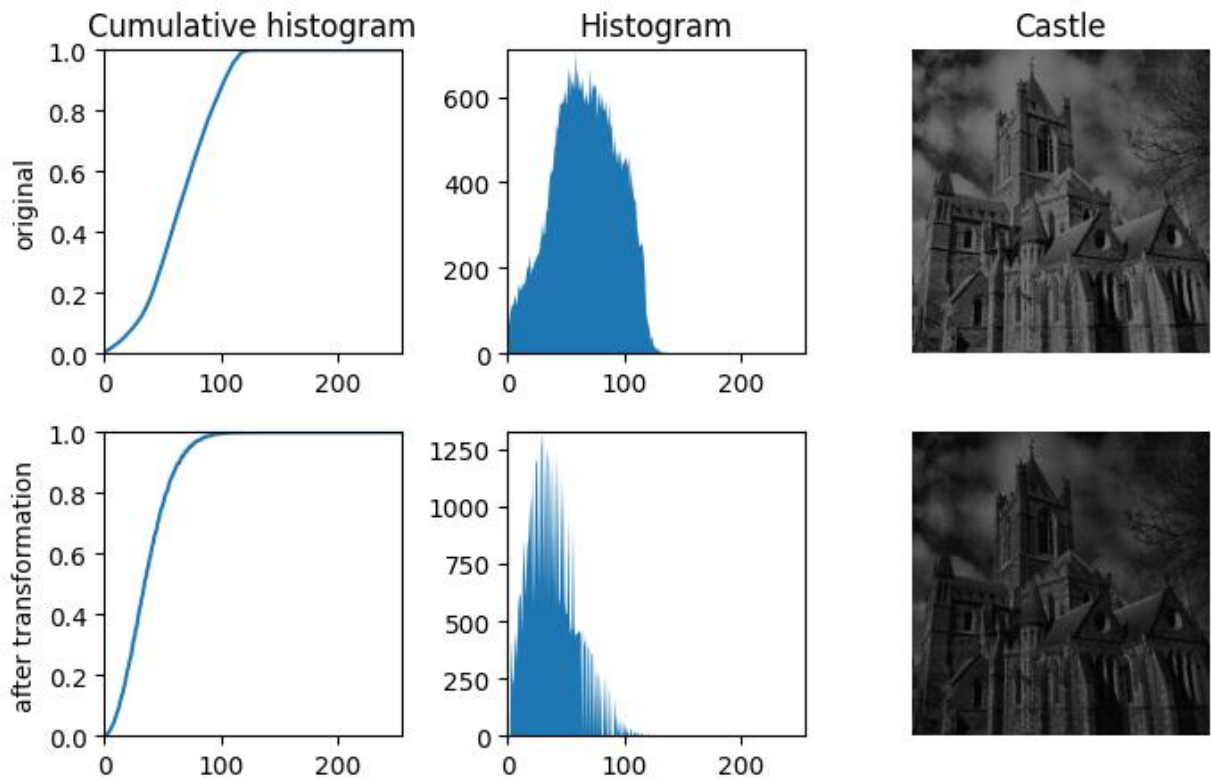


Рисунок 9 Влияние преобразования по закону Рэля при $\alpha = 20$

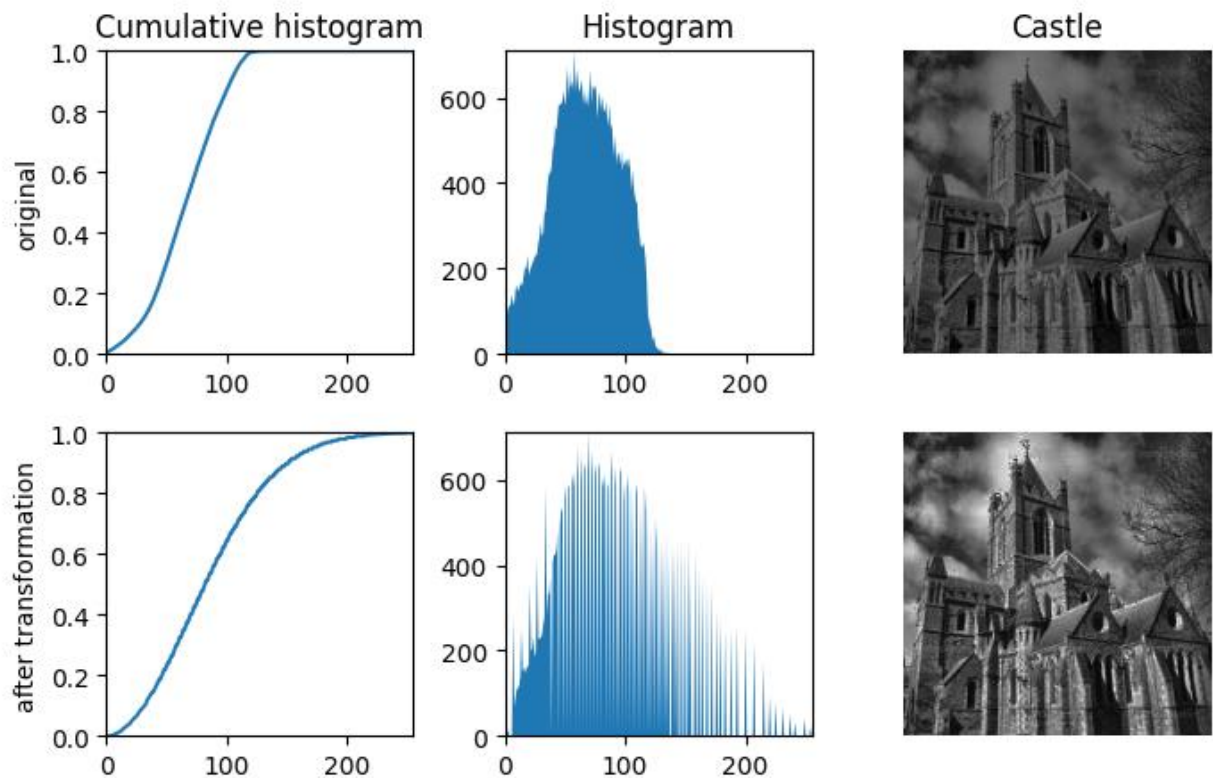


Рисунок 10 Влияние преобразования по закону Рэля при $\alpha = 70$

Можно заметить, что при увеличении α увеличивается контраст изображения, но только до определенного значения α , после которого, начинает теряться информация об изображении.

Листинг 8 Функция расчета преобразования по закону Рэля.

```
def rayleigh_low_transform(img: ndarray, cum_hist, a: float = 100):
    i_min = np.min(img)
    new_img = ndarray(img.shape)
    for x in range(img.shape[0]):
        for y in range(img.shape[1]):
            new_img[x][y] = i_min + np.power(2*np.power(a, 2) * np.log(1 / (1 - cum_hist[img[x][y]])), 0.5)
    return new_img.astype(np.uint8)
```

1.7 Преобразование по закону $2/3$

Вычислим новое значение интенсивностей по формуле:

$$I_{new} = (P(I))^{2/3}$$

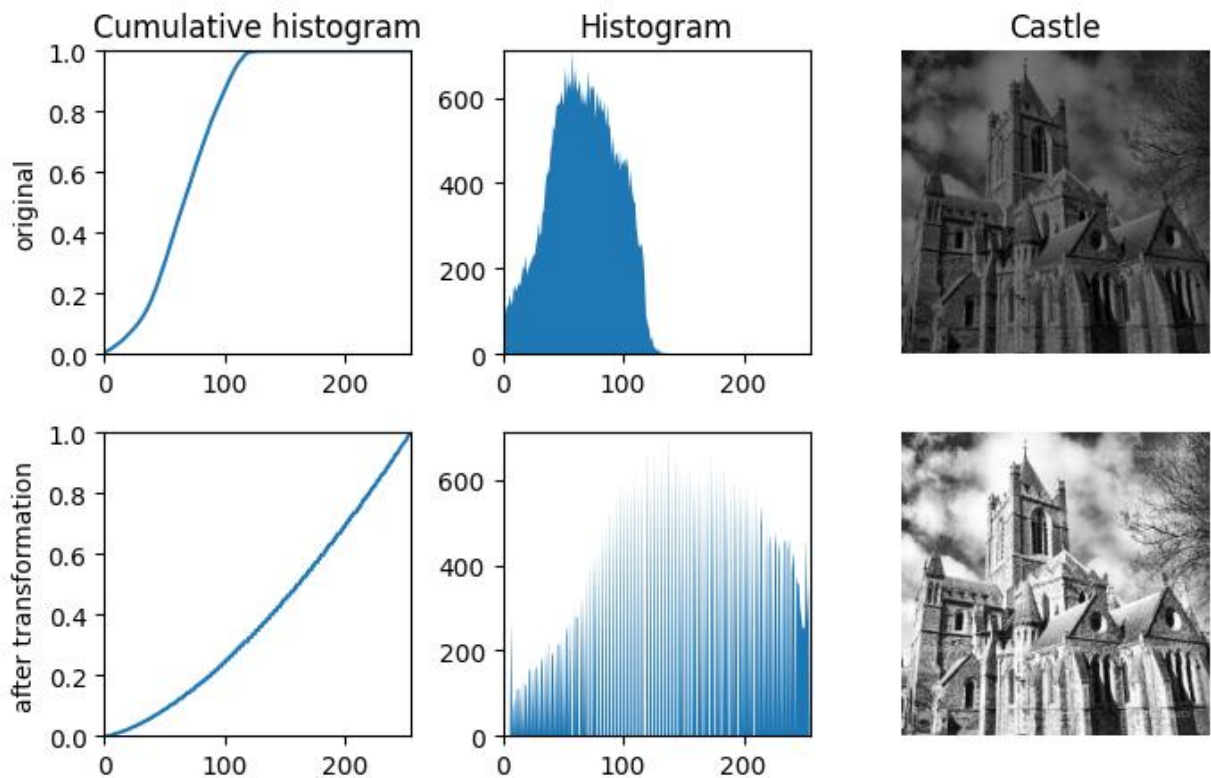


Рисунок 11 Влияние преобразования по закону $2/3$

Можно заметить, что после преобразования изображение стало сильно контрастным с преобладанием светлых тонов.

Листинг 7 Функция преобразования по закону $2/3$.

```
def two_thirds_low_transform(img: ndarray, cum_hist: ndarray):
    new_img = ndarray(img.shape)
    for x in range(img.shape[0]):
        for y in range(img.shape[1]):
            print(cum_hist[img[x][y]])
            new_img[x][y] = 255*np.power((cum_hist[img[x][y]]), 2/3)
    return new_img.astype(np.uint8)
```

1.8 Преобразование по гиперболическому закону

Вычислим новое значение интенсивностей по формуле:

$$I_{new} = \alpha^{P(I)}$$

α часто выбирают равным I_{min} .

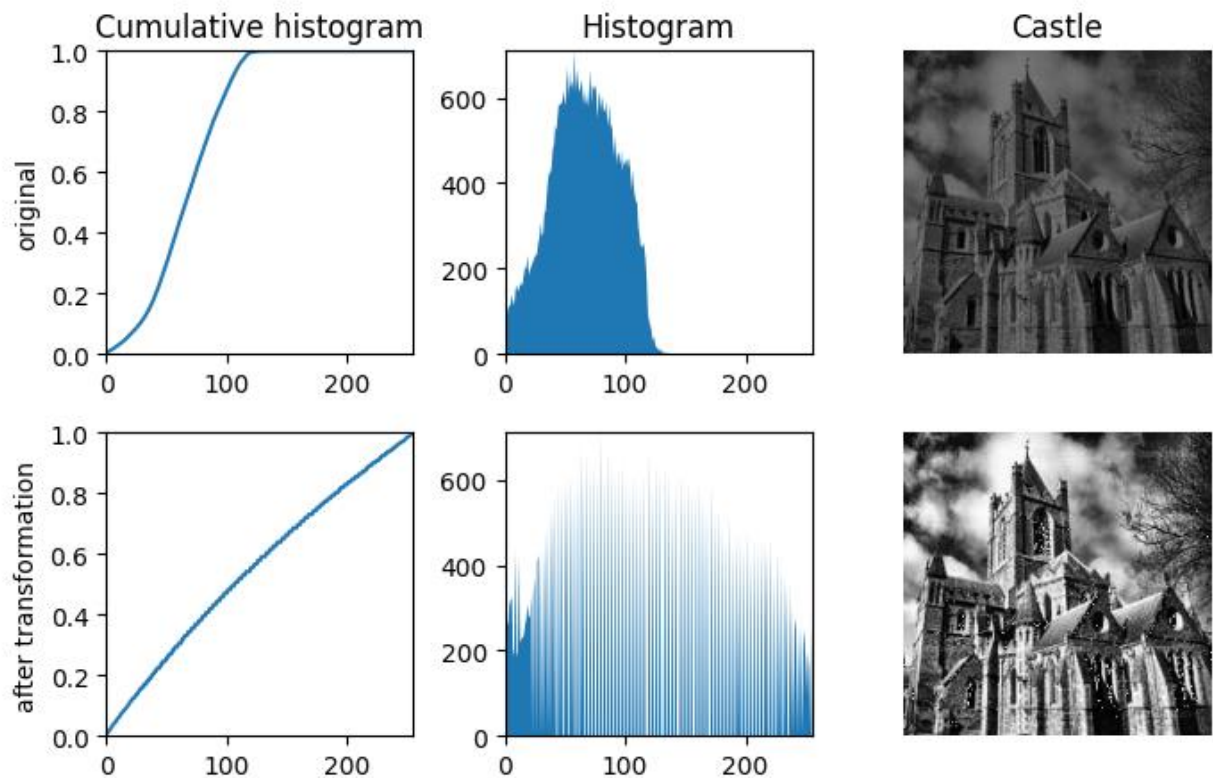


Рисунок 12 Влияние преобразования по гиперболическому закону

Можно заметить, что после преобразования изображение стало сильно контрастным.

Листинг 8 Функция преобразования по гиперболическому закону.

```
def hyperbolic_transform(img: ndarray, cum_hist: ndarray, a=None):
    if a is None:
        a = np.min(img)
    if a == 0 or a == 1:
        a = 2
    new_img = ndarray(img.shape)
    for x in range(img.shape[0]):
        for y in range(img.shape[1]):
            print(cum_hist[img[x][y]])
            new_img[x][y] = 255*np.power(a, cum_hist[img[x][y]])
    return new_img.astype(np.uint8)
```

1.9 Преобразование с помощью встроенных в OpenCV функций

Воспользуемся функцией `equalizeHist()` из OpenCV

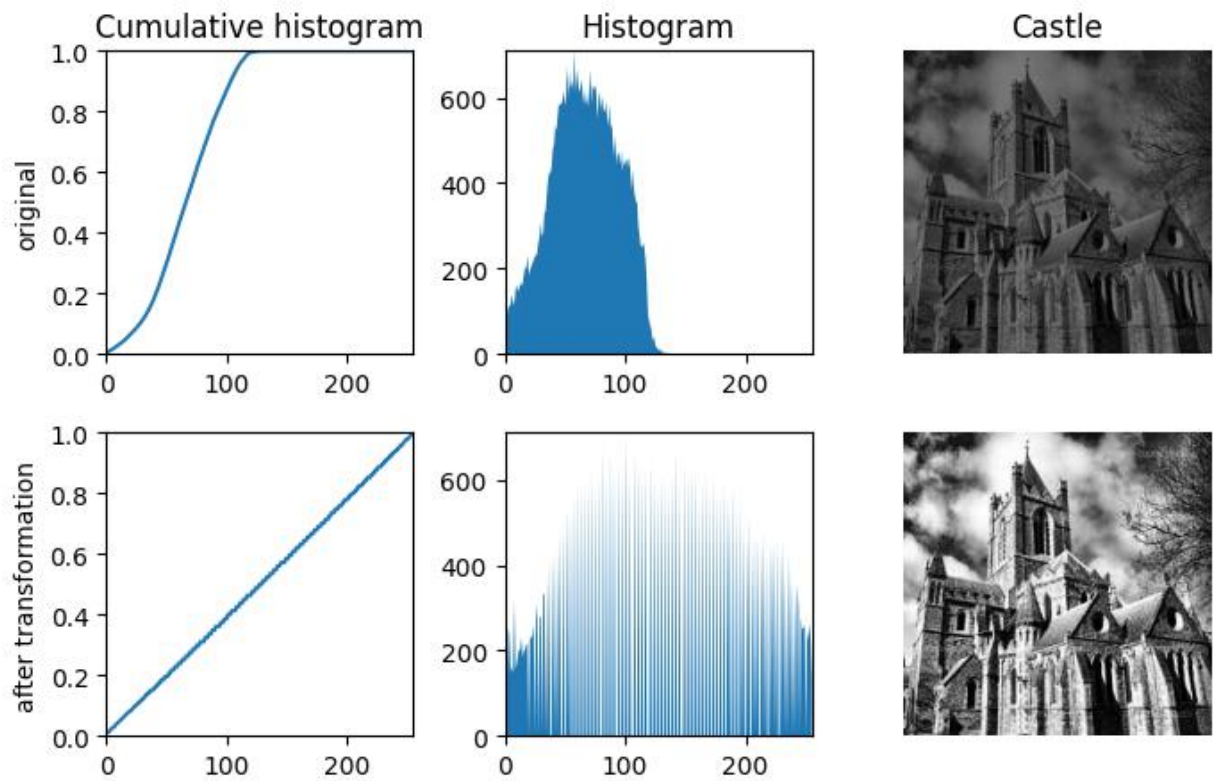


Рисунок 13 Влияние преобразования с помощью функции `equalizeHist()`

Преобразованное изображение контрастное и при этом с “нормальным” распределением интенсивностей.

Воспользуемся функцией `createCLAHE()` из OpenCV