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. **Исходные данные**

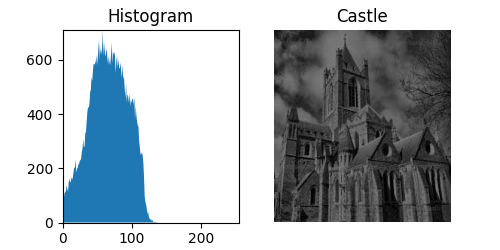


Рисунок 1 Исходное изображение

Контрастность изображения:

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([image], [**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. **Линейное преобразование**

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

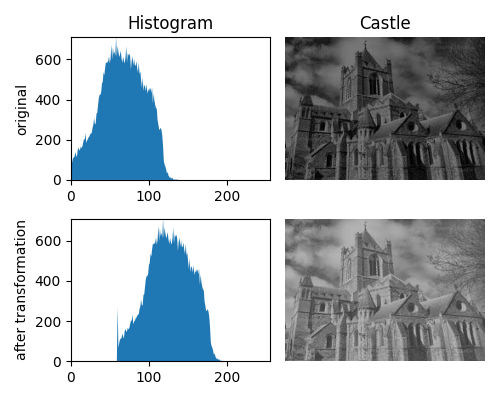


Рисунок 2 Влияние линейного сдвига

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. **Растяжение динамического диапазона.**

Применим растяжение динамического диапазона с параметром с различными .

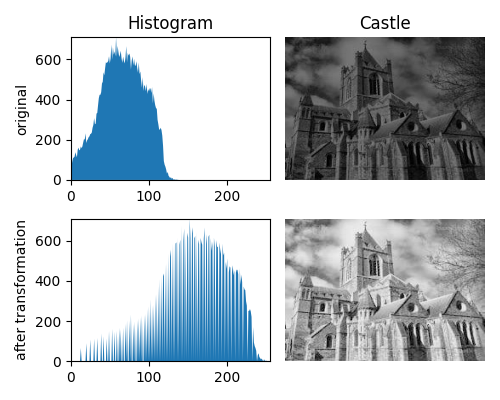
**

Рисунок 3 Влияние динамического растяжение с =0.6

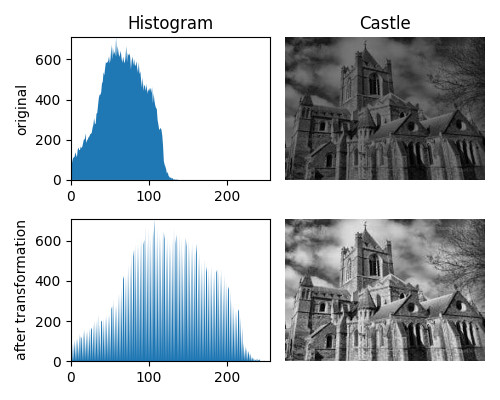
**

Рисунок 4 Влияние динамического растяжение с =1

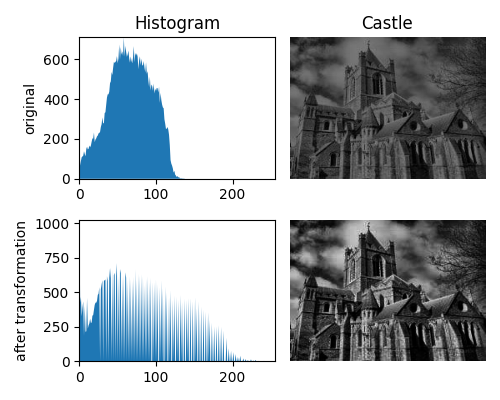


Рисунок 5 Влияние динамического растяжение с =1.9

Можно заметить, что при увеличении 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. **Равномерное преобразование**

Применим равномерное преобразование для исходного изображения:

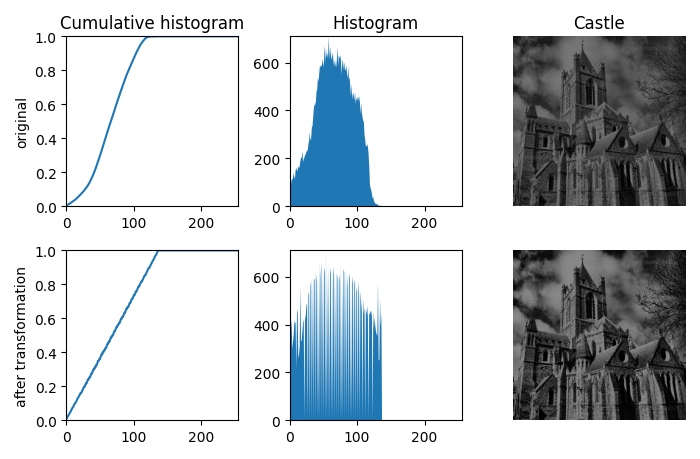


Рисунок 6 Влияние равномерного преобразования

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 проебразования.

**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. **Экспоненциальное преобразование**

Вычислим новое значение интенсивностей по формуле:

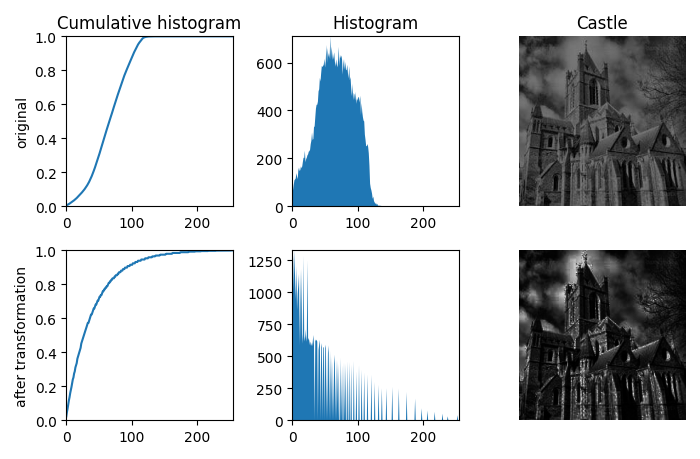


Рисунок 7 Влияние экспоненциального преобразования при

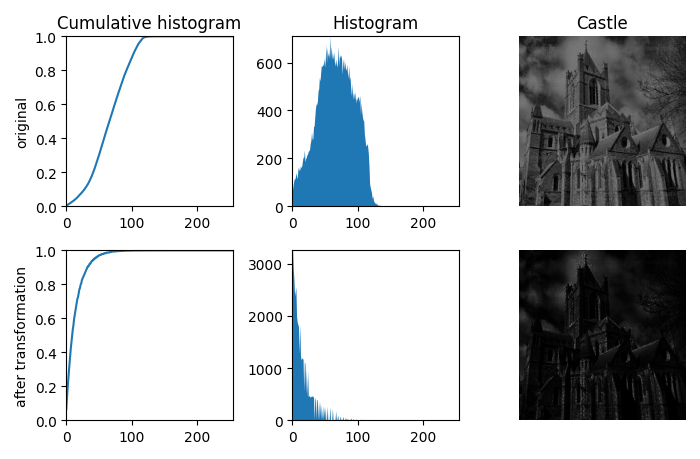


Рисунок 8 Влияние экспоненциального преобразования при α=0.07

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)