

Санкт-Петербургский Национально Исследовательский Университет
информационных технологий, механики и оптики

Основы технического зрения

Отчет по лабораторной работе №1
Гистограммы, профили и проекции
вариант: 5

Выполнили:
Суздалев О.
Караваев А.А.
Группа: R3335
Преподаватель:
Шаветов С.В.

Санкт-Петербург
2019

Содержание

1. Цель Работы	2
2. Теоретическое обоснование применяемых методов и функции геометрических преобразований	2
2.1. Гистограмма изображения	2
2.1.1. Растяжение динамического диапазона	2
2.1.2. Равномерное преобразование	2
2.1.3. Экспоненциальное преобразование	2
2.1.4. Преобразование по закону Рэлея	2
2.1.5. Преобразование по закону степени $2/3$	2
2.1.6. Гиперболическое преобразование	2
2.2. Профиль изображения	3
2.3. Проекция изображения	3
3. Ход выполнения работы	4
3.1. Гистограмма	4
3.1.1. Исходные изображения	4
3.1.2. Листинги программных реализации	4
3.1.3. Результирующие изображения	7
3.2. Профили	8
3.2.1. Исходные изображения	8
3.2.2. Листинги программных реализации	8
3.2.3. Результирующие изображения	8
3.3. Проекции	9
3.3.1. Исходные изображения	9
3.3.2. Листинги программных реализации	9
3.3.3. Результирующие изображения	10
4. Выводы о проделанной работе	10

1. Цель Работы

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

2. Теоретическое обоснование применяемых методов и функции геометрических преобразований

2.1. Гистограмма изображения

Гистограмма — это распределение частоты встречаемости пикселей одинаковой яркости на изображении.

2.1.1. Растяжение динамического диапазона

Если интенсивности пикселей областей интереса находятся в узком динамическом диапазоне, то можно растянуть этот диапазон. Подобные преобразования выполняются согласно следующему выражению:

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

2.1.2. Равномерное преобразование

$$I_{new} = (I_{max} - I_{min}) \cdot P(I) + I_{min},$$

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

2.1.3. Экспоненциальное преобразование

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

2.1.4. Преобразование по закону Рэля

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

2.1.5. Преобразование по закону степени 2/3

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

2.1.6. Гиперболическое преобразование

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

2.2. Профиль изображения

Профилем изображения вдоль некоторой линии называется функция интенсивности изображения, распределенного вдоль данной линии.

Простейшим случаем профиля изображения является профиль строки:

$$\text{Profile } i(x) = I(x, i)$$

Профиль столбца изображения:

$$\text{Profile } j(y) = I(j, y)$$

2.3. Проекция изображения

Проекцией изображения на некоторую ось называется сумма интенсивностей пикселей изображения в направлении, перпендикулярном данной оси. Простейшим случаем проекции двумерного изображения являются вертикальная проекция на ось Ox , представляющая собой сумму интенсивностей пикселей по столбцам изображения:

$$\text{Proj } X(y) = \sum_{y=0}^{\dim Y - 1} I(x, y)$$

Горизонтальная проекция на ось Oy , представляющая собой сумму интенсивностей пикселей по строкам изображения:

$$\text{Proj } Y(x) = \sum_{x=0}^{\dim X - 1} I(x, y)$$

Запишем выражение для проекции на произвольную ось. Допустим, что направление оси задано единичным вектором с координатами (e_x, e_y) . Тогда проекция изображения на ось Oe определяется следующим выражением:

$$\text{Proj } E(t) = \sum_{xe_x + ye_y = t} I(x, y)$$

3. Ход выполнения работы

3.1. Гистограмма

3.1.1. Исходные изображения



3.1.2. Листинги программных реализации

```
1 from skimage import data, exposure
2 from skimage.util import img_as_ubyte, img_as_float
3 import matplotlib.pyplot as plt
4 import numpy as np
5
6
7 def convertToUint(array):
8     array[array > 255] = 255
9     array[array < 0] = 0
10    return array.astype('uint8')
11
12 def toGraph(array, row):
13     for i in range(array.shape[2]):
14         hist, hist_centers = exposure.histogram(array[:, :, i])
15         axs[row, 1].plot(hist_centers, hist)
16
17
18 # original image
19 im = data.chelsea()
20
21
22 #-----Арифметические операции-----
23 im_arithm = data.chelsea().astype('float')
24 to_add = 50
25 im_arithm += to_add
26 im_arithm = convertToUint(im_arithm)
27
28
29 #-----Нелинейное растяжение динамического диапазона-----
30 im_nonLinear = img_as_float(data.chelsea())
31 nonLinear_alpha = 0.5
32 for i in range(im_nonLinear.shape[2]):
33     min = im_nonLinear[:, :, i].min()
34     max = im_nonLinear[:, :, i].max()
35     im_nonLinear[:, :, i] = ((im_nonLinear[:, :, i] - min) / (max - min)) **
    ↪ nonLinear_alpha
```

```

36 im_nonLinear = img_as_ubyte(im_nonLinear)
37
38 #-----Равномерное преобразование-----
39 im_uniformly = img_as_float(data.chelsea())
40 im_uniformly_index = data.chelsea()
41 for k in range(im_uniformly.shape[2]):
42     min = im_uniformly[:, :, k].min()
43     max = im_uniformly[:, :, k].max()
44     hist, hist_centers = exposure.histogram(im_uniformly[:, :, k])
45     cs = np.cumsum(hist) / (im_uniformly.shape[0] * im_uniformly.shape[1])
46     for i in range(im_uniformly.shape[0]):
47         for j in range(im_uniformly.shape[1]):
48             index = im_uniformly_index[i, j, k]
49             im_uniformly[i, j, k] = ((max - min) * cs[index] + min)
50 im_uniformly = img_as_ubyte(im_uniformly)
51
52 #-----Экспоненциальное преобразование-----
53 im_exponent = img_as_float(data.chelsea())
54 im_exponent_index = data.chelsea()
55 exponent_alpha = 1
56 for k in range(im_exponent.shape[2]):
57     min = im_exponent[:, :, k].min()
58     hist, hist_centers = exposure.histogram(im_exponent[:, :, k])
59     cs = np.cumsum(hist) / (im_exponent.shape[0] * im_exponent.shape[1])
60     for i in range(im_exponent.shape[0]):
61         for j in range(im_exponent.shape[1]):
62             im_exponent[i, j, k] = min - (1 / exponent_alpha) * np.log10(1 - cs[
        ↪ im_exponent_index[i, j, k]])
63 im_exponent[im_exponent > 1.0] = 0.0
64 im_exponent = img_as_ubyte(im_exponent)
65
66
67 #-----Преобразование по закону Рэлея-----
68 im_rayleigh = img_as_float(data.chelsea())
69 im_rayleigh_index = data.chelsea()
70 rayleigh_alpha = 1
71 for k in range(im_rayleigh.shape[2]):
72     min = im_rayleigh[:, :, k].min()
73     hist, hist_centers = exposure.histogram(im_rayleigh[:, :, k])
74     cs = np.cumsum(hist) / (im_rayleigh.shape[0] * im_rayleigh.shape[1])
75     for i in range(im_rayleigh.shape[0]):
76         for j in range(im_rayleigh.shape[1]):
77             im_rayleigh[i, j, k] = min + np.sqrt(2 * rayleigh_alpha ** 2 * np.
        ↪ log10((1 / (1 - cs[im_rayleigh_index[i, j, k]]))) ** (1/2)
78
79
80 #-----Преобразование по закону степени 2/3-----
81 im_twoThree = img_as_float(data.chelsea())
82 im_twoThree_index = data.chelsea()
83 for k in range(im_twoThree.shape[2]):
84     hist, hist_centers = exposure.histogram(im_twoThree[:, :, k])
85     cs = np.cumsum(hist) / (im_twoThree.shape[0] * im_twoThree.shape[1])
86     for i in range(im_twoThree.shape[0]):
87         for j in range(im_twoThree.shape[1]):
88             im_twoThree[i, j, k] = cs[im_twoThree_index[i, j, k]] ** 2/3
89
90
91 #-----Гиперболическое преобразование-----
92 im_hyperbolic = img_as_float(data.chelsea())
93 im_hyperbolic_index = data.chelsea()

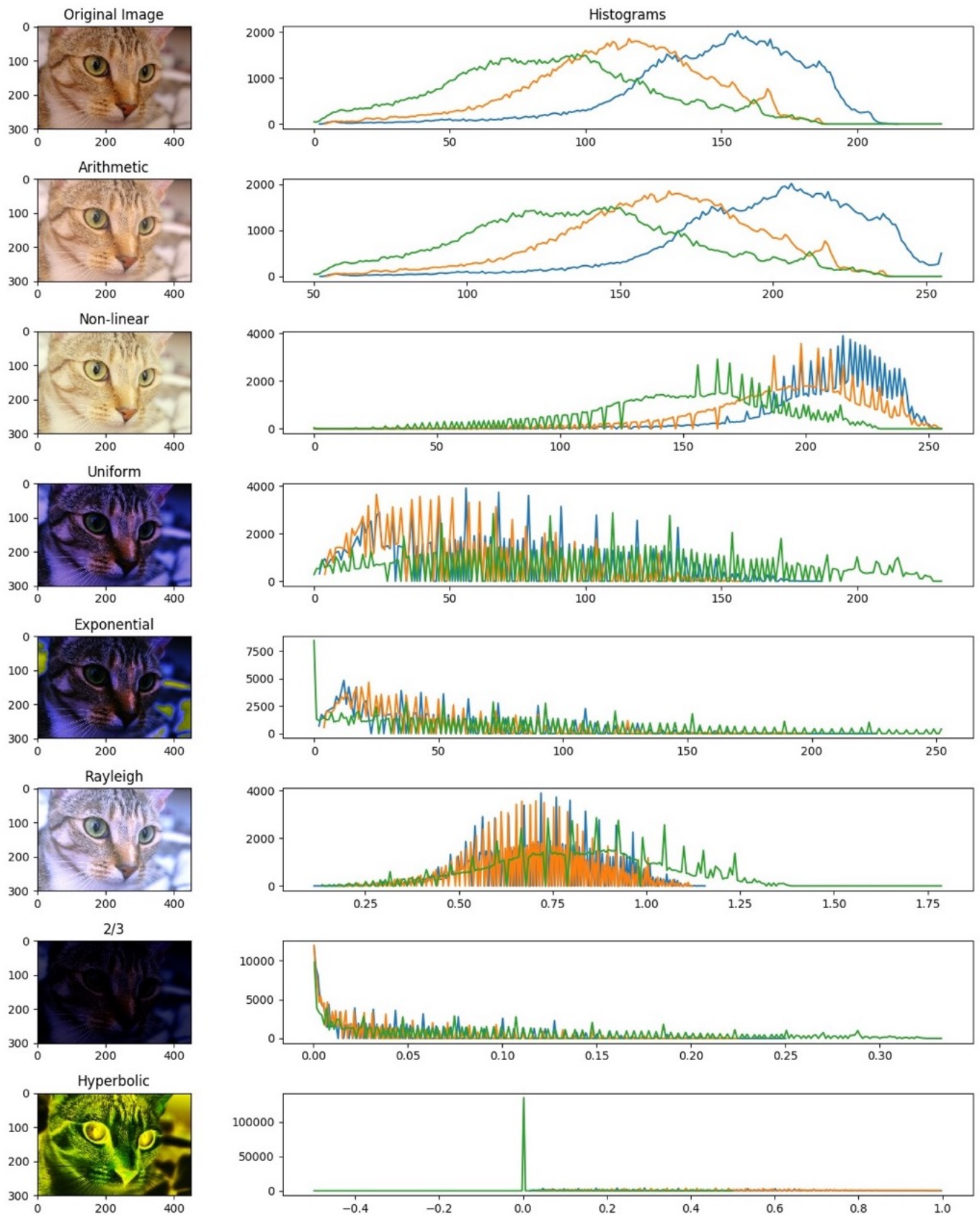
```

```

94 for k in range(im_hyperbolic.shape[2]):
95     min = im_hyperbolic[:, :, k].min()
96     hist, hist_centers = exposure.histogram(im_hyperbolic[:, :, k])
97     cs = np.cumsum(hist) / (im_hyperbolic.shape[0] * im_hyperbolic.shape[1])
98     for i in range(im_hyperbolic.shape[0]):
99         for j in range(im_hyperbolic.shape[1]):
100             im_hyperbolic[i, j, k] = min ** cs[im_hyperbolic_index[i, j, k]]
101
102
103 #-----Графики-----
104 fig, axs = plt.subplots(8, 2, figsize=(15,15))
105
106 axs[0, 0].set_title('Original Image')
107 axs[0, 0].imshow(im)
108 axs[0, 1].set_title('Histograms')
109 toGraph(im, 0)
110
111 axs[1, 0].set_title('Arithmetic')
112 axs[1, 0].imshow(im_arithm)
113 toGraph(im_arithm, 1)
114
115 axs[2, 0].set_title('Non-linear')
116 axs[2, 0].imshow(im_nonLinear)
117 toGraph(im_nonLinear, 2)
118
119 axs[3, 0].set_title('Uniform')
120 axs[3, 0].imshow(im_uniformly)
121 toGraph(im_uniformly, 3)
122
123 axs[4, 0].set_title('Exponential')
124 axs[4, 0].imshow(im_exponent)
125 toGraph(im_exponent, 4)
126
127 axs[5, 0].set_title('Rayleigh')
128 axs[5, 0].imshow(im_rayleigh)
129 toGraph(im_rayleigh, 5)
130
131 axs[6, 0].set_title('2/3')
132 axs[6, 0].imshow(im_twoThree)
133 toGraph(im_twoThree, 6)
134
135 axs[7, 0].set_title('Hyperbolic')
136 axs[7, 0].imshow(im_hyperbolic)
137 toGraph(im_hyperbolic, 7)
138
139 fig.tight_layout()
140
141 plt.show()

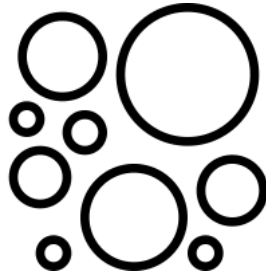
```

3.1.3. Результирующие изображения



3.2. Профили

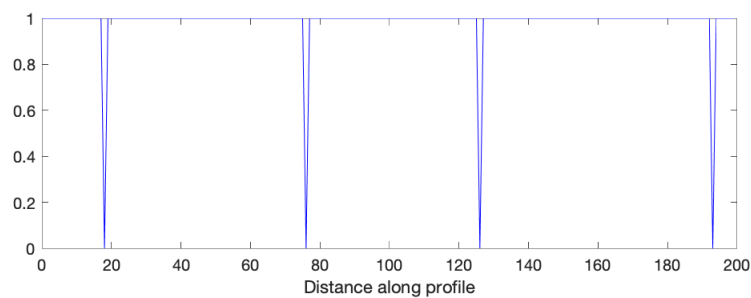
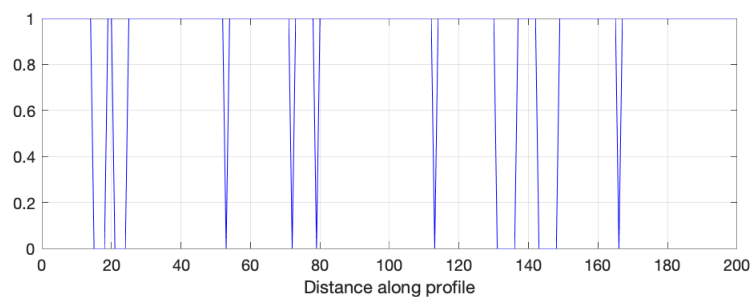
3.2.1. Исходные изображения



3.2.2. Листинги программных реализации

```
1 I = imread("shape.png");
2 [numRows , numCols , Layers] = size(I);
3
4 % X axis
5 subplot(2,1,1);
6 x = [1 numCols];
7 y = [ceil(numRows/2) ceil(numRows/2)];
8 improfile(I,x,y);
9 grid on;
10
11 subplot(2,1,2);
12 x = [ceil(numCols/2) ceil(numCols/2)];
13 y = [1 numRows];
14 improfile(I,x,y);
```

3.2.3. Результирующие изображения



3.3. Проекции

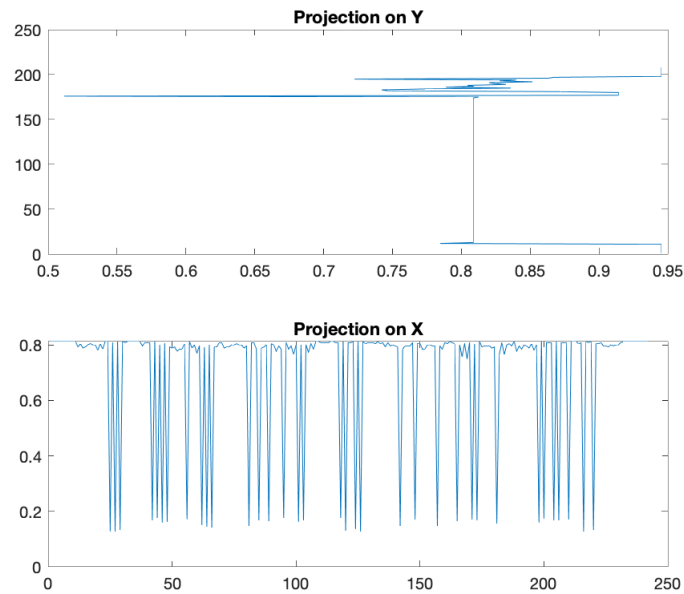
3.3.1. Исходные изображения



3.3.2. Листинги программных реализации

```
1 I = imread("barcode.png");
2 I = imbinarize(I);
3 I = not(I);
4 [numRows , numCols , Layers] = size(I);
5
6 for i=1:1:numRows
7     Proj(i,1) = sum(I(i,:)) / 256;
8 end
9
10 subplot(2,1,1);
11 plot(Proj, (1:1:numRows));
12 title('Projection on Y');
13
14 for i=1:1:numCols
15     Proj(i,1) = sum(I(:,i)) / 256;
16 end
17
18 subplot(2,1,2);
19 plot((1:1:numCols), Proj);
20 title('Projection on X');
```

3.3.3. Результирующие изображения



4. Выводы о проделанной работе

В данной работе были изучены основные яркостные и геометрические характеристики изображений и их использование для анализа изображений. Использовали разные преобразования для растяжения динамического диапазона, построили профили и проекции изображения.