

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ ИМЕНИ Н. Э. БАУМАНА
Факультет информатики и систем управления
Кафедра теоретической информатики и компьютерных технологий

Лабораторная работа №1
по курсу «Структуры и алгоритмы обработки больших данных»

«Исследование изображения
с помощью свертки и Фурье-анализа
с использованием MATLAB»

Выполнил:
студент группы ИУ9-21М
Беляев А. В.

Проверила:
Магазов С. С.

Москва 2019

1 Цель работы

Научиться исследовать изображения с помощью операции свертки и Фурье анализа с использованием MatLab.

2 Ход работы. Вариант 3

2.1 Вычислить свертку на интервале $[-30, 30]$, построить график, вычислить максимум

$$g(x) = \frac{1}{\pi(1+x^2)} f(x)$$
$$f(x) = \begin{cases} \exp(-x^2) & x \in [-1, 1] \\ 0 & \text{else} \end{cases}$$

Листинг 1: Вычисление свертки функций

```
1 clear all; % clear workspace
2
3 x = -30:0.1:30;
4 N = length(x);
5
6 for i = 1:N
7     f(i) = func(x(i));
8     g(i) = 1 / (pi*(1 + x(i)^2));
9 end
10
11 w = conv(f, g, 'same');
12 fprintf('max: %f', max(w));
13
14 % draw F, G
15 figure(1)
16 plot(x, g);
17 xlim([-10 10]);
18 hold on % draw multiple plots in same window
19 plot(x, f);
20 legend('g(x)', 'f(x)');
21 hold off
22 saveas(gcf, 'fg.png');
23
24 % draw convolution
25 figure(2)
26 plot(x, w);
27 xlim([-10 10]);
28 saveas(gcf, 'fg-conv.png');
```

```

29
30
31 function y = func(x)
32     if x >= -1 && x <= 1
33         y = exp(-1 * x^2);
34     else
35         y = 0;
36     end
37 end

```

На Рисунке 1 изображены графики функций и свертки функций.

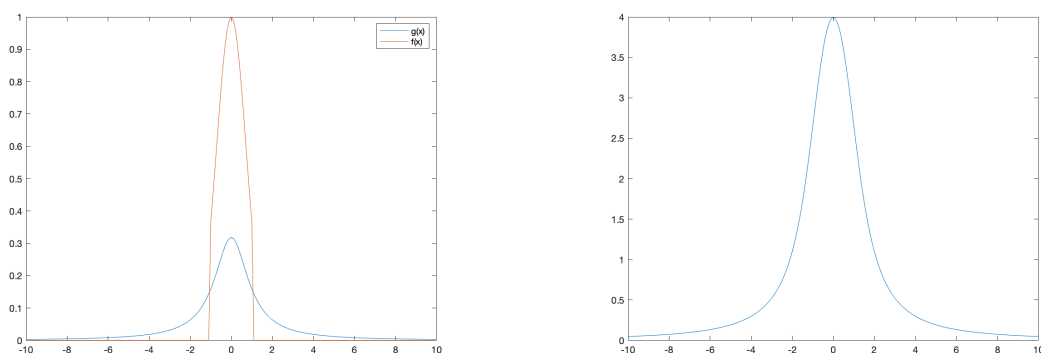


Рис. 1: График функций f и g , график свертки

Максимум функции: 3.995167

2.2 Свертка изображений с ядрами

2.2.1 Вычислить свертку изображения с ядрами. Построить изображение свертки и график свертки

Листинг 2: Вычисление свертки изображения

```

1 import Conv.*
2
3 clear all;
4
5 img = imread('BioID_0003.pgm');
6 [img_h, img_w, dim] = size(img);
7 % imshow(img);
8
9
10 % 3x3 kernels
11 sobelm_x = [ 1 2 1; 0 0 0; -1 -2 -1];
12 sobelm_y = [-1 0 1; -2 0 2; -1 0 1];
13 c1 = Conv('sobel', sobelm_x, sobelm_y);

```

```

14 convolute(c1, img);
15
16 prewit_x = [ -1 1 1; -1 -2 1; -1 1 1];
17 prewit_y = [ 1 1 1; -1 -2 1; -1 -1 1];
18 c2 = Conv('prew', prewit_x, prewit_y);
19 convolute(c2, img);
20
21 kirsch_x = [ -3 -3 5; -3 0 5; -3 -3 5];
22 kirsch_y = [ -3 5 5; -3 0 5; -3 -3 -3];
23 c3 = Conv('kir', kirsch_x, kirsch_y);
24 convolute(c3, img);
25
26 robin3_x = [ -1 0 1; -1 0 1; -1 0 1];
27 robin3_y = [ 0 1 1; -1 0 1; -1 -1 1];
28 c4 = Conv('rob3', robin3_x, robin3_y);
29 convolute(c4, img);
30
31 robin5_x = [ -1 0 1; -2 0 2; -1 0 1];
32 robin5_y = [ 0 1 2; -1 0 1; -2 1 0];
33 c5 = Conv('rob5', robin5_x, robin5_y);
34 convolute(c5, img);

```

Листинг 3: Conv.m

```

1 classdef Conv
2     properties (GetAccess = private)
3         kernel_x;
4         kernel_y;
5     end
6
7     properties
8         name;
9     end
10
11     methods
12         function obj = Conv(m_name, m_x, m_y)
13             obj.name = m_name;
14             obj.kernel_x = m_x;
15             obj.kernel_y = m_y;
16         end
17
18         function convolute(obj, img)
19             w_x = conv2(img, obj.kernel_x);
20             w_y = conv2(img, obj.kernel_y);
21             w = sqrt(double(w_x.^2 + w_y.^2)); % wiki formula
22

```

```

23         % draw image
24         filename = [obj.name '.png'];
25         imwrite(w, gray(256), filename);
26
27         % draw convolution plot
28         figure();
29         mesh(w);
30         filename = [obj.name '-plot.png'];
31         saveas(gcf, filename);
32     end
33 end
34 end

```

На Рисунках 3, 4, 5, 6, 7 представлены изображения после применения свертки и графики свертки.

На Рисунке 2 представлен оригинал изображения.



Рис. 2: Оригинал изображения

2.2.2 Вычислить вручную значение свертки в центральной и угловых точках

Матрица исходного изображения представлена в Таблице 1.

Значение свертки в точке вычисляется по следующей формуле:

$$z * w = \sum_{j=0}^{M-1} \sum_{j=0}^{N-1} z_{i,j} * w_{i,j}$$

$$z = \sqrt{z_x^2 + z_y^2}$$

где w – подматрица 3×3 матрицы изображения (см. Таблицу 1).

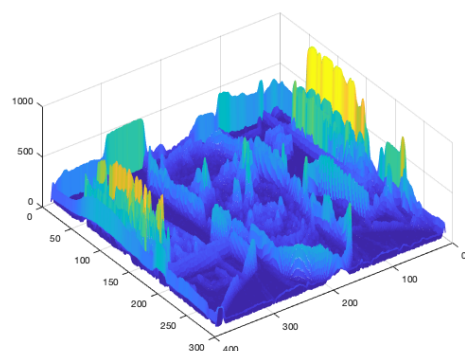
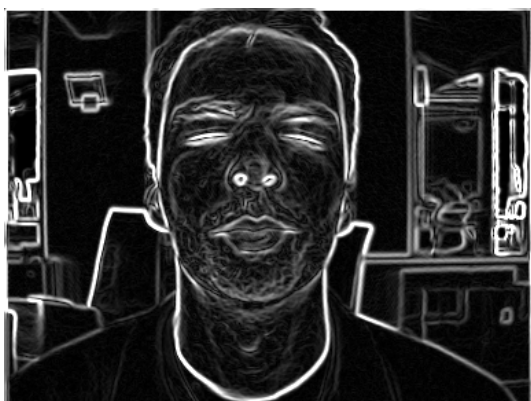


Рис. 3: Фильтр Собеля

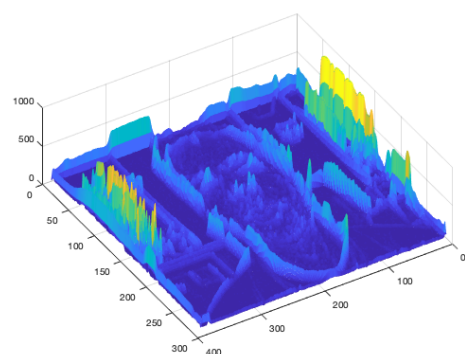
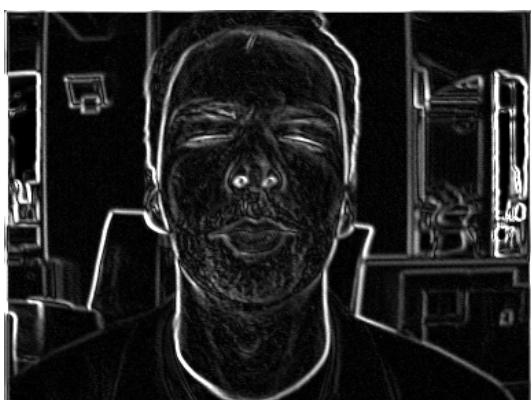


Рис. 4: Фильтр Prewitt

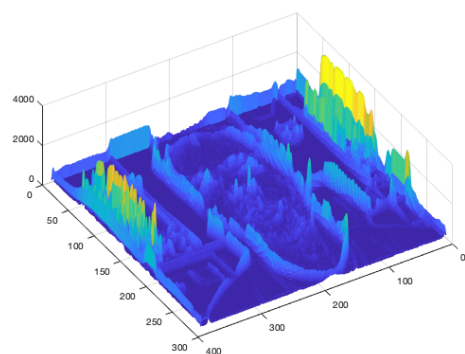


Рис. 5: Фильтр Kirsch

Посчитаем одно значение свертки вручную и вычислим остальные значения по аналогии:

$$Z_x = W_{top-left} * Sobel_x = \begin{pmatrix} 62 & 77 & 68 \\ 62 & 81 & 75 \\ 63 & 82 & 77 \end{pmatrix} * \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix} =$$

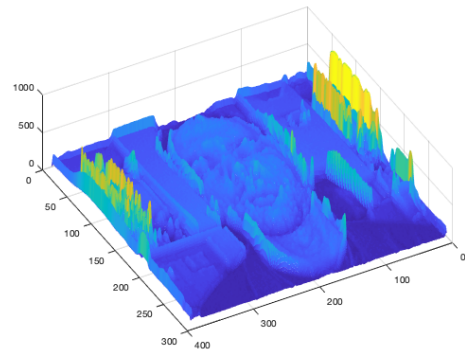
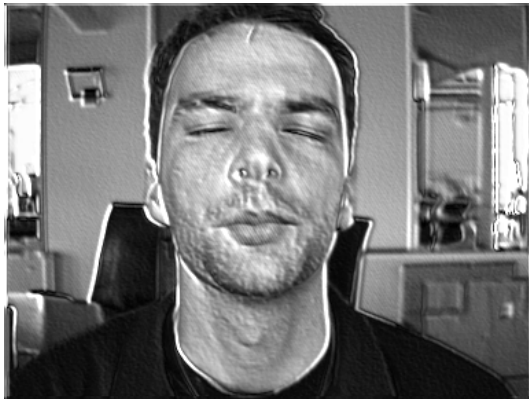


Рис. 6: Фильтр Robinson 3 level

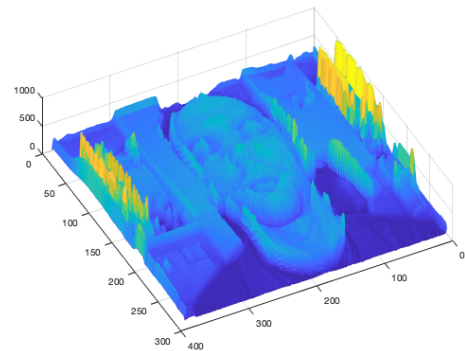
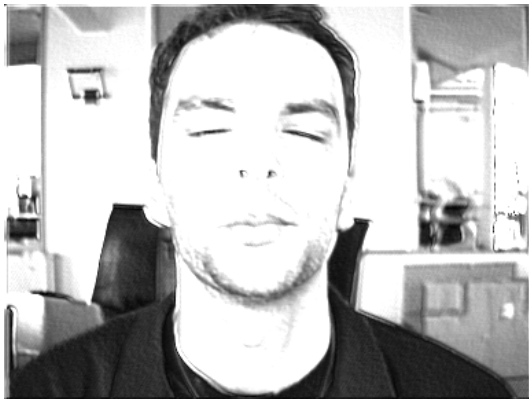


Рис. 7: Фильтр Robinson 5 level

$$= (62*1) + (77*2) + (68*1) + (-1*63) + (-2*82) + (-1*77) = -20$$

$$Z_y = W_{top-left} * Sobel_y = \begin{pmatrix} 62 & 77 & 68 \\ 62 & 81 & 75 \\ 63 & 82 & 77 \end{pmatrix} * \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} =$$

$$= (-1*62) + (1*68) + (-2*62) + (2*75) + (-1*63) + (1*77) = 46$$

$$z = \sqrt{z_x^2 + z_y^2} = \sqrt{(-20)^2 + 46^2} \approx 50$$

Листинг 4: Вычисление свертки изображения в точках вручную

```

1 clc;
2
3 img = imread('BioID_0003.pgm');
4 [img_h, img_w, dim] = size(img);

```

Таблица 1: Матрица исходного изображения

| | | | | | | | | | | |
|----|----|----|---|-----|-----|-----|---|----|----|----|
| 62 | 77 | 68 | | | | | | 57 | 45 | 17 |
| 62 | 81 | 75 | | | | | | 55 | 44 | 17 |
| 63 | 82 | 77 | | | | | | 57 | 43 | 16 |
| | | | . | | | | . | | | |
| | | | | 169 | 158 | 151 | | | | |
| | | | | 167 | 164 | 157 | | | | |
| | | | | 173 | 173 | 166 | | | | |
| | | | . | | | | . | | | |
| 37 | 37 | 21 | | | | | | 51 | 36 | 14 |
| 36 | 36 | 20 | | | | | | 48 | 38 | 15 |
| 37 | 37 | 20 | | | | | | 48 | 38 | 15 |

```

5 |
6 | % 3x3 sobel kernel
7 | sob_x = [ 1 2 1; 0 0 0; -1 -2 -1];
8 | sob_y = [ -1 0 1; -2 0 2; -1 0 1];
9 |
10 | % image submatrices
11 | top_left = img(1:3, 1:3);
12 | top_right = img(1:3, img_w-2:img_w);
13 | bot_left = img(img_h-2:img_h, 1:3);
14 | bot_right = img(img_h-2:img_h, img_w-2:img_w);
15 | center = img(img_h/2-1:img_h/2+1, img_w/2-1:img_w/2+1);
16 |
17 | fprintf('top left: %f\n', convolute(top_left, sob_x, sob_y));
18 | fprintf('top right: %f\n', convolute(top_right, sob_x, sob_y));
19 | fprintf('bot left: %f\n', convolute(bot_left, sob_x, sob_y));
20 | fprintf('bot right: %f\n', convolute(bot_right, sob_x, sob_y));
21 | fprintf('center : %f\n', convolute(center, sob_x, sob_y));
22 |
23 |
24 | function res = conv_mask(m, mask)
25 |     res = double(m(1,1))*mask(1,1) + double(m(1,2))*mask(1,2) +
        double(m(1,3))*mask(1,3) + double(m(2,1))*mask(2,1) +
        double(m(2,2))*mask(2,2) + double(m(2,3))*mask(2,3) +
        double(m(3,1))*mask(3,1) + double(m(3,2))*mask(3,2) +
        double(m(3,3))*mask(3,3);
26 | end
27 |
28 | function res = convolute(m, mask_x, mask_y)
29 |     z_x = conv_mask(m, mask_x);
30 |     z_y = conv_mask(m, mask_y);
31 |     res = sqrt(z_x^2 + z_y^2);

```


Получившиеся значения:

- сверху слева: 50.159745
- сверху справа: 157.079598
- снизу слева: 65.007692
- снизу справа: 136.014705
- по центру: 66.528190

2.3 Вычислить конечную разность для изображения и вывести полученное изображение

Полученное изображение представлено на Рисунке 8

Листинг 5: Вычисление конечной разности

```

1 clear all; % clear workspace
2
3 img = imread('BioID_0003.pgm');
4 [img_h, img_w, dim] = size(img);
5
6 for i = 1:img_h
7     for j = 1:img_w-1
8         img(i,j) = img(i,j) - img(i, j+1);
9     end
10 end
11
12 for i = 1:img_h-1
13     for j = 1:img_w
14         img(i,j) = img(i,j) - img(i+1, j);
15     end
16 end
17
18 imwrite(img, 'finite-difference.png');
```

2.4 Фурье-анализ

2.4.1 Разложение функции в ряд Фурье

Разложение функции $f(x) = \frac{\pi-x}{2}$ в ряд Фурье с периодом $[-\pi, \pi]$. Построить графики для $n = 1, 5, 30, 100$.

$$f(x) = a_0 + \sum_{n=1}^{\infty} \left(a_n \cos \frac{\pi n x}{l} + b_n \sin \frac{\pi n x}{l} \right)$$

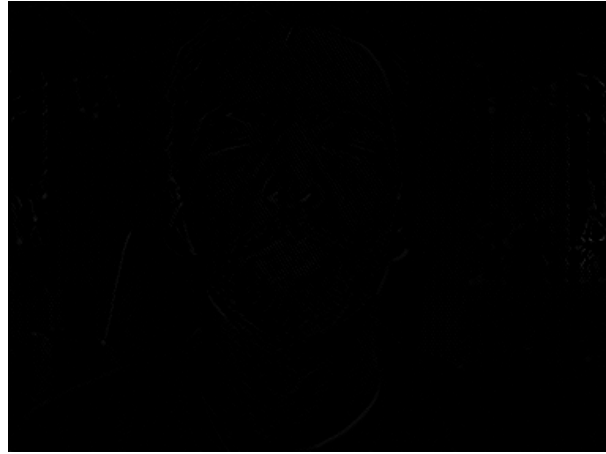


Рис. 8: Конечная разность изображения

$$a_n = \frac{2}{l} \int_{-\frac{l}{2}}^{\frac{l}{2}} f(x) \cos \frac{nx}{2} dx$$

$$b_n = \frac{2}{l} \int_{-\frac{l}{2}}^{\frac{l}{2}} f(x) \sin \frac{nx}{2} dx$$

где $l = 2\pi$. Подставим $f(x)$ и посчитаем коэффициенты:

$$a_n = \frac{1}{\pi} \int_{-\pi}^{\pi} \frac{\pi - x}{2} \cos \frac{nx}{2} dx = \frac{2 \sin(\frac{\pi n}{2})}{n}$$

$$b_n = \frac{1}{\pi} \int_{-\pi}^{\pi} \frac{\pi - x}{2} \sin \frac{nx}{2} dx = \frac{2\pi n \cos(\frac{\pi n}{2}) - 4 \sin(\frac{\pi n}{2})}{\pi n^2}$$

$$a_0 = \frac{\pi x - \frac{x^2}{2}}{2\pi}$$

Листинг 6: Разложение в ряд Фурье

```

1 clear all;
2
3 N = 100;    % steps
4
5 x = -pi;    % from
6 step = 0.01;
7 x_max = pi; % to
8
9 i = 1;
10 while x < x_max
11     ys(i) = fourier(x, N);
12     xs(i) = x;
13
14     x = x + step;
```

```

15     i = i + 1;
16 end
17
18 % draw convolution plot
19 f = figure();
20 plot(xs, ys);
21 filename = ['fr-' num2str(N) '.png'];
22 saveas(gcf, filename);
23
24
25 function sum = fourier(x, N)
26     sum = (pi*x - x^2 / 2) / (2*pi); % a0
27     for n = 1:N
28         an = (2*sin(pi*n / 2)) / n;
29         bn = (2*pi*n*cos(pi*n/2) - 4*sin(pi*n / 2)) / (pi*n^2);
30         sum = sum + (an*cos(n*x) + bn*sin(n*x));
31     end
32 end

```

На Рисунках 9, 10 приведены получившиеся графики преобразований Фурье.

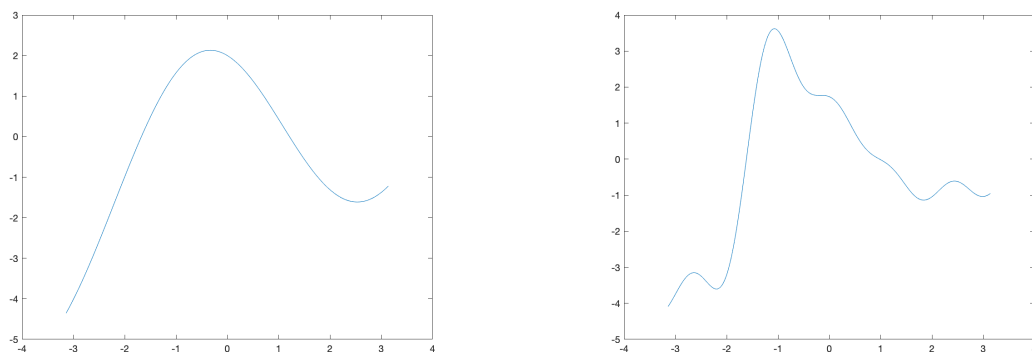


Рис. 9: N=1, N=5

2.4.2 Построить 1D дискретный базис Фурье f^3

Листинг 7: 1D базис

```

1 clear all;
2
3 N = 3;
4 basis = zeros(N);
5
6 for n = 1 : N
7     for k = 1 : N

```

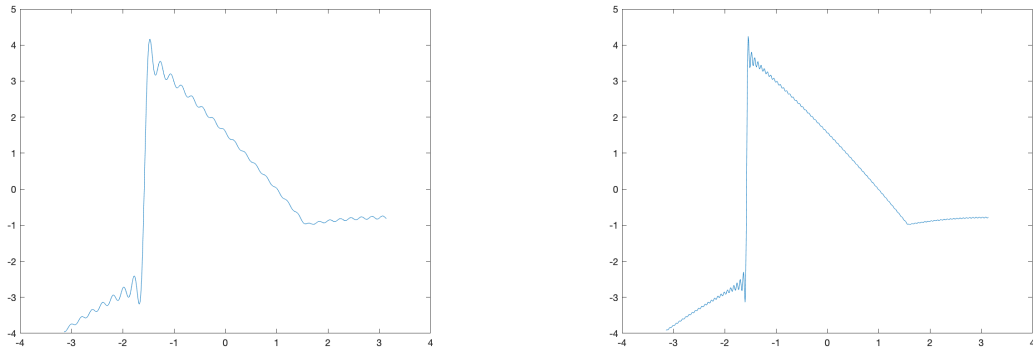


Рис. 10: N=30, N=100

```

8      basis(k, n) = exp(2i * pi * (n-1) * k/N) / sqrt(N);
9  end
10 end

```

Полученный базис представлен в Таблице 2

Таблица 2: 1D базис

| | | |
|-------------------|---------------------------|---------------------------|
| 0,577350269189626 | -0,288675134594813 + 0,5i | -0,288675134594813 - 0,5i |
| 0,577350269189626 | -0,288675134594813 - 0,5i | -0,288675134594813 + 0,5i |
| 0,577350269189626 | 0,577350269189626 | 0,577350269189626 |

2.4.3 Построить 2D дискретный базис Фурье $f^{3 \times 3}$

Листинг 8: 2D базис

```

1  clear all;
2
3  N1 = 3;
4  N2 = 3;
5  basis2D = zeros(N1*N1*N1);
6
7  for n = 1 : N1
8      for m = 1 : N2
9          for l = 1:N
10             for k = 1:N
11                 basis2D(k, l, m, n) = exp(2i * pi * (k*(n-1)/N1
12                     + l*(m-1)/N2)) / sqrt(N1*N2);
13             end
14         end
15     end

```

Базис представлен в Таблице 3.

Таблица 3: 2D базис

| | | |
|----------------|----------------|----------------|
| $0.33333 + 0i$ | $0.33333 + 0i$ | $0.33333 + 0i$ |
| $0.33333 + 0i$ | $0.33333 + 0i$ | $0.33333 + 0i$ |
| $0.33333 + 0i$ | $0.33333 + 0i$ | $0.33333 + 0i$ |