

Министерство образования Республики Беларусь
Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Кафедра электронных вычислительных средств

Лабораторная работа № 3
«Преобразование яркости изображений и пространственная фильтрация»

Проверил:
Рыбенков Е.В.

Выполнил:
ст. гр. 850701
Филиппцов Д. А.

Минск 2021

Цель работы

Изменение яркости, синтез и использование цифровых фильтров в задачах обработки изображений.

Ход работы

Задание на эквализацию гистограммы:

- a) Подсказка: функции `unique`, `cumsum`.
- b) Загрузить все изображения (1_1.tif – 1_4.tif) с помощью команды `imread`.
- c) Преобразовать изображение в тип `double`. Приводить изображения к диапазону `[0;1]` ненужно. Задавать диапазон входных яркостей изображения от 0 до 255.
- d) Рассчитать гистограмму изображения в соответствии с уравнением (1.1).
- e) Рассчитать значения sk по формуле (1.2) и сохранить их в массиве `T`.
- f) Выполнить эквализацию гистограммы $sk=T(rk)$.
- g) Отобразить изображения и их гистограммы после эквализации с помощью команд `imshow`.

MATLAB-код:

```
% преобразование исходных изображений к типу double [0 255]
I1 = double(imread('1_1.tif'));
I2 = double(imread('1_2.tif'));
I3 = double(imread('1_3.tif'));
I4 = double(imread('1_4.tif'));

scale = 256;

gist1 = gist(I1);
gist2 = gist(I2);
gist3 = gist(I3);
gist4 = gist(I4);

figure('Name','Before'); % вывод оригинальных изображений и их гистограмм
subplot(4, 2, 1); imshow(I1, [0 (scale - 1)]);
subplot(4, 2, 2); plot(gist1, 'o');
subplot(4, 2, 3); imshow(I2, [0 (scale - 1)]);
subplot(4, 2, 4); plot(gist2, 'o');
subplot(4, 2, 5); imshow(I3, [0 (scale - 1)]);
subplot(4, 2, 6); plot(gist3, 'o');
subplot(4, 2, 7); imshow(I4, [0 (scale - 1)]);
subplot(4, 2, 8); plot(gist4, 'o');

% эквализация гистограмм
[T1, Ieq1] = ekval(gist1, I1);
[T2, Ieq2] = ekval(gist2, I2);
[T3, Ieq3] = ekval(gist3, I3);
[T4, Ieq4] = ekval(gist4, I4);

figure('Name','After'); % вывод изображений после эквализации и их гистограмм
subplot(4, 2, 1); imshow(Ieq1, [0 (scale - 1)]);
subplot(4, 2, 2); plot(gist(Ieq1), 'o');
subplot(4, 2, 3); imshow(Ieq2, [0 (scale - 1)]);
```

```
subplot(4, 2, 4); plot(gist(Ieq2), 'o');
subplot(4, 2, 5); imshow(Ieq3, [0 (scale - 1)]);
subplot(4, 2, 6); plot(gist(Ieq3), 'o');
subplot(4, 2, 7); imshow(Ieq4, [0 (scale - 1)]);
subplot(4, 2, 8); plot(gist(Ieq4), 'o');
```

Код функции gistag:

```
function [gistag] = gist(imag)
%GIST making histogram from image 2-dimension matrix

gistag = zeros(1, 256);
sizeImag = size(imag);
for a = 1:(sizeImag(1)*sizeImag(2))
    gistag(imag(a) + 1) = gistag(imag(a) + 1) + 1;
end
gistag = gistag ./ (sizeImag(1) * sizeImag(2));

end
```

Код функции ekval:

```
function [T, Ieq] = ekval(gistag, I)
%EKVAL making histogram and image equalization

scale = size(gistag);
T = zeros(scale);
for k = 1:scale(2)
    T(k) = round( (scale(2) - 1) * sum(gistag(1:k)) );
end

sizeI = size(I);
Ieq = zeros(sizeI);
for a = 1:(sizeI(1)*sizeI(2))
    Ieq(a) = T(I(a)+1);
end

end
```

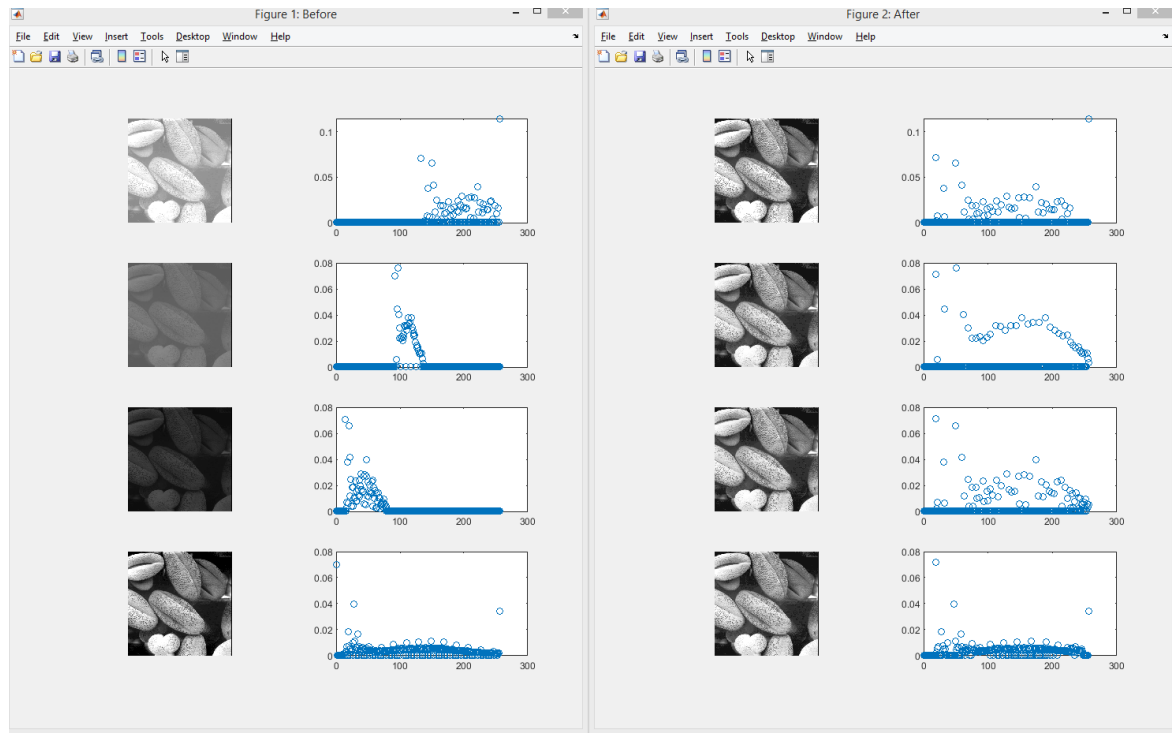


Рисунок 1 – Исходные изображения с их гистограммами (слева) и изображения после эквализации (справа)

Задание на локальную эквализацию гистограммы:

- Загрузить изображение (2_1.tif) с помощью команды `imread`.
- Преобразовать изображение в тип `double`. Приводить изображения к диапазону `[0;1]` не нужно. Задавать диапазон входных яркостей изображения от 0 до 255.
- Дополнить исходное изображение нулями для правильной работы на границе изображения (`wextend`).
- Выполнить локальную гистограммную обработку для области размером `3x3` и `5x5` (см. теор. Часть).
- Отобразить изображения с помощью команд `imshow`.

Код программы:

```
I2_1 = wextend('2d', 'zpd', double(imread('2_1.tif')), 2); sizeI2_1 = size(I2_1);
figure('Name','Before');
imshow(I2_1, [0 (scale - 1)]);

% локальная эквализация окрестностью 3x3
Ieq2_1_1 = zeros(sizeI2_1);
for a = 3:(sizeI2_1(1)-2)
    for b = 3:(sizeI2_1(2)-2)
        tempI = I2_1(a-1:a+1, b-1:b+1);
        [~, Ieq2_1_1(a-1:a+1, b-1:b+1)] = ekval(gist(tempI), tempI);
    end
end

% локальная эквализация окрестностью 5x5
Ieq2_1_2 = zeros(sizeI2_1);
```

```

for a = 3:(sizeI2_1(1)-2)
    for b = 3:(sizeI2_1(2)-2)
        tempI = I2_1(a-2:a+2, b-2:b+2);
        [T, Ieq2_1_2(a-2:a+2, b-2:b+2)] = ekval(gist(tempI), tempI);
    end
end

figure('Name','After');
subplot(1, 2, 1); imshow(Ieq2_1_1, [0 (scale - 1)]);
subplot(1, 2, 2); imshow(Ieq2_1_2, [0 (scale - 1)]);

```

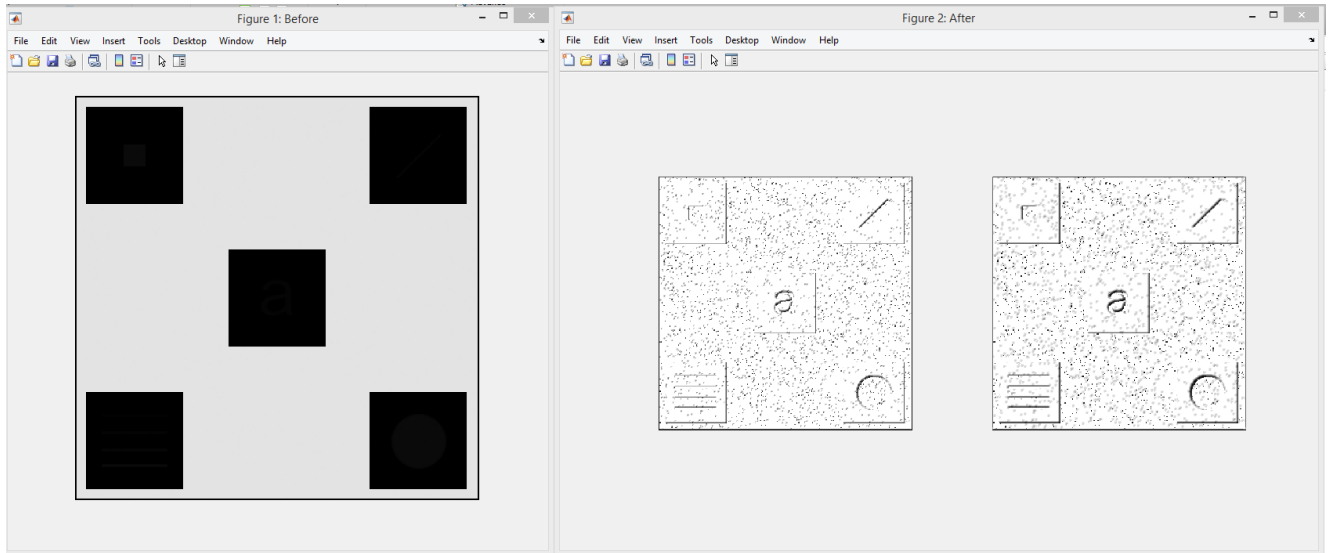


Рисунок 2 – Исходное изображение (слева) и изображения, обработанные по алгоритму локальной эквализации области 3x3 (по центру) и 5x5 (справа)

Задание на пространственную фильтрацию:

- Загрузить изображение (3_1.tif) с помощью команды `imread`.
- Преобразовать изображение в тип `double`. Приводить изображения к диапазону `[0;1]` не нужно. Задавать диапазон входных яркостей изображения от 0 до 255.
- Написать функцию расчета 2D свертки `my_filt2d` в соответствии с уравнением (1.3).
- Рассчитать ядро сглаживающего фильтра размером 15x15 по формуле (1.4).
- Выполнить сглаживание тестового изображения.
- Отобразить изображение с помощью команд `imshow`.
- Выполнить пороговую обработку сглаженного изображения путем присвоения значения 255 всем пикселям большим 64, иначе присвоить значение 0.
- Отобразить изображение с помощью команд `imshow`.

Код программы:

```

m = 15;
I3_1 = wextend('2d', 'zpd', double(imread('3_1.tif')), (m-1)/2);
figure('Name','Before');

```

```

imshow(I3_1, [0 (scale - 1)]);

Ifiltr3_1 = my_filt2d(I3_1, my_filt(m, m)); % фильтрация сглаживающим фильтром
15 x 15
figure('Name','After');
subplot(1, 2, 1); imshow(Ifiltr3_1, [0 (scale - 1)]);

for x = 1:size(Ifiltr3_1, 1) % пороговая фильтрация
    for y = 1:size(Ifiltr3_1, 2)
        if Ifiltr3_1(x, y) > 64
            Ifiltr3_1(x, y) = 255;
        else
            Ifiltr3_1(x, y) = 0;
        end
    end
end
subplot(1, 2, 2); imshow(Ifiltr3_1, [0 (scale - 1)]);

```

Код функции my_filt:

```

function [w] = my_filt(m, n)
%MY_FILT сглаживающий фильтр m x n

a = (m-1)/2;
b = (n-1)/2;
w = zeros(m, n);

for s = -a:a
    for t = -b:b
        w(s+a+1, t+b+1) = 1/(m*n);
    end
end

end

```

Код функции my_filt2d:

```

function [Ifiltr] = my_filt2d(I, w)
%MY_FILT2D filtering

sizeI = size(I);
Ifiltr = zeros(sizeI);
[m, n] = size(w);
a = (m-1)/2;
b = (n-1)/2;
for x = (1 + a):(sizeI(1) - a)
    for y = (1 + b):(sizeI(2) - b)
        sum = 0;

        for s = -a:a
            for t = -b:b
                sum = sum + (w(s+a+1, t+b+1) * I(x - s, y - t));
            end
        end

        Ifiltr(x, y) = round(sum);
    end
end

end

```

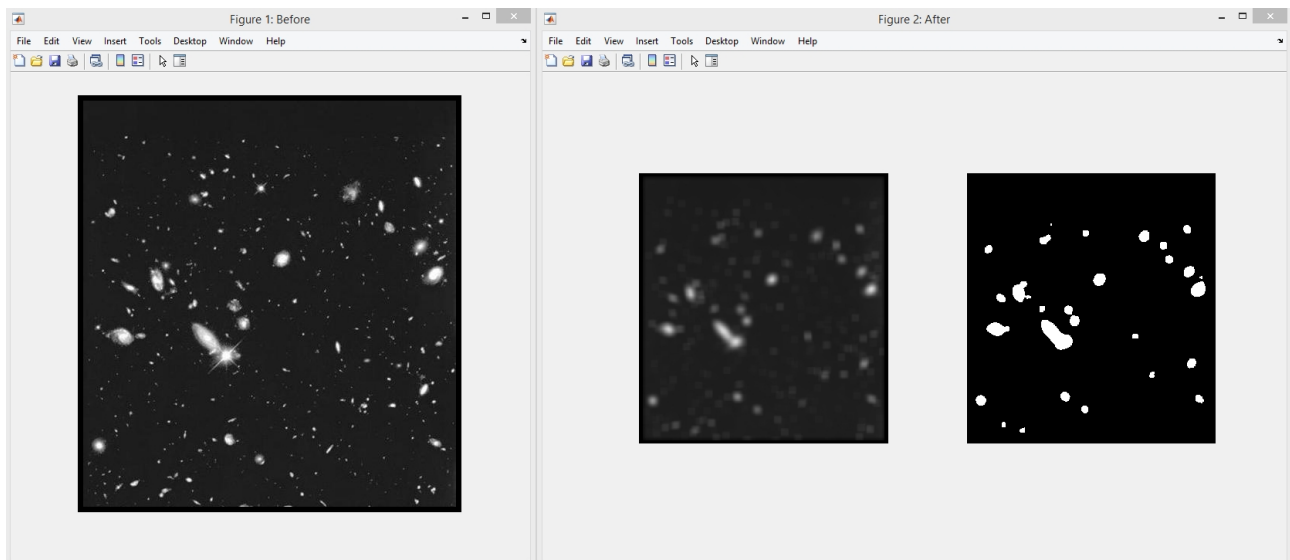


Рисунок 3 – Исходное изображение (слева) и изображения, обработанные по алгоритму сглаживания (по центру) и пороговой фильтрации впоследствии (справа)

Задание на медианную фильтрацию:

- а) Загрузить изображение (4_1.tif) с помощью команды `imread`.
- б) Преобразовать изображение в тип `double`. Приводить изображения к диапазону `[0;1]` не нужно. Задавать диапазон входных яркостей изображения от 0 до 255.
- в) Рассчитать ядро сглаживающего фильтра размером 3×3 по формуле (1.4).
- г) Выполнить сглаживание тестового изображения.
- д) Выполнить медианную фильтрацию тестового изображения.
- е) Отобразить изображения с помощью команд `imshow`.

Код программы:

```
m = 3;
step = (m-1)/2;
I4_1 = wextend('2d', 'zpd', double(imread('4_1.tif')), step);
figure('Name','Before');
imshow(I4_1, [0 (scale - 1)]);

Ifiltr4_1 = my_filt2d(I4_1, my_filt(m, m)); % фильтрация сглаживающим фильтром 3x3
figure('Name','After');
subplot(1, 2, 1); imshow(Ifiltr4_1, [0 (scale - 1)]);

Temp = zeros(m, m);
Temp_s = zeros(m^2, 1);
for x = (1 + step):(size(Ifiltr4_1, 1) - step) % медианная фильтрация
    for y = (1 + step):(size(Ifiltr4_1, 2) - step)
        Temp = Ifiltr4_1((x - step):(x + step), (y - step):(y + step));
        Temp_s = sort(Temp(:));
        Ifiltr4_1(x, y) = Temp_s((m^2+1)/2);
    end
end
```

```
subplot(1, 2, 2); imshow(Ifiltr4_1, [0 (scale - 1)]);
```

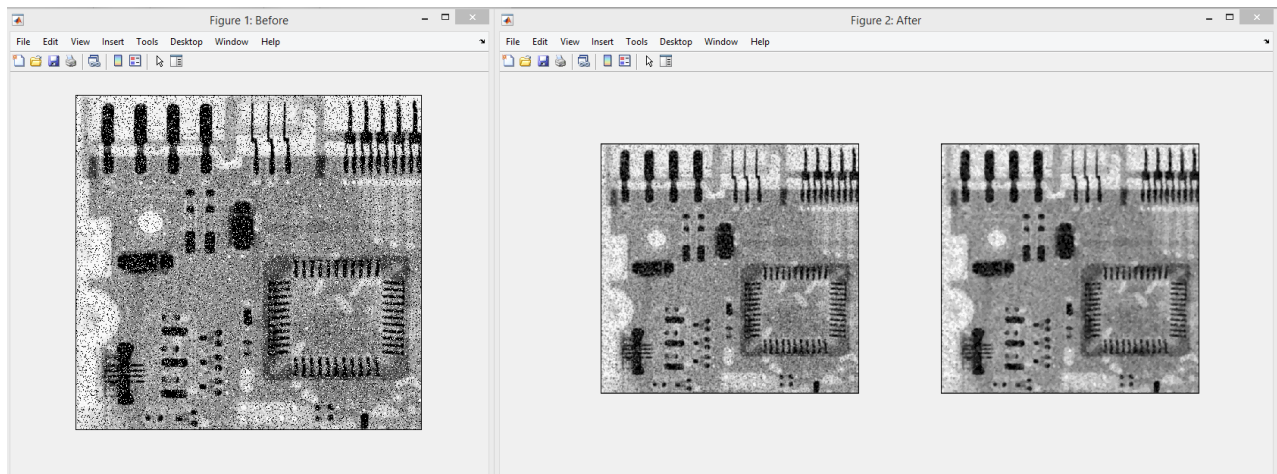


Рисунок 4 – Исходное изображение (слева) и изображения, обработанные по алгоритму сглаживания (по центру) и медианной фильтрации впоследствии (справа)

Задание на повышение резкости изображения:

- a) Загрузить изображение (5_1.tif) с помощью команды `imread`.
- b) Преобразовать изображение в тип `double`. Приводить изображения к диапазону `[0;1]` не нужно. Задавать диапазон входных яркостей изображения от 0 до 255.
- c) Рассчитать ядро лапласиана размером `3x3` по формуле (1.5).
- d) Выполнить фильтрацию тестового изображения тестового изображения с помощью ядра фильтра.
- e) Отобразить изображение с помощью команд `imshow`.
- f) Добавить фон на изображение по формуле (1.6).
- g) Отобразить изображение с помощью команд `imshow`.

Код программы:

```
m = 3;
step = (m-1)/2;
I5_1 = wextend('2d', 'zpd', double(imread('5_1.tif')), step);
figure('Name','Before');
imshow(I5_1, [0 (scale - 1)]);

Ifiltr5_1 = laplasian(I5_1); % расчёт лапласиана 3 x 3 для оригинального
изображения
figure('Name','After');
subplot(1, 2, 1); imshow(Ifiltr5_1, [0 (scale - 1)]);

Ifiltr5_1 = I5_1 - Ifiltr5_1; % добавление фона на изображение
subplot(1, 2, 2); imshow(Ifiltr5_1, [0 (scale - 1)]);
```


Код функции laplasian:

```
function [Iret] = laplasian(I)
%LAPLASIAN Compiting laplassian 3 x 3

m = 3;
step = (m-1)/2;
Iret = zeros(size(I));
for x = (1 + step):(size(I, 1) - step)
    for y = (1 + step):(size(I, 2) - step)
        Iret(x, y) = I(x+1, y) + I(x-1, y) + I(x, y+1) + I(x, y-1) - 4*I(x, y);
    end
end
end
```

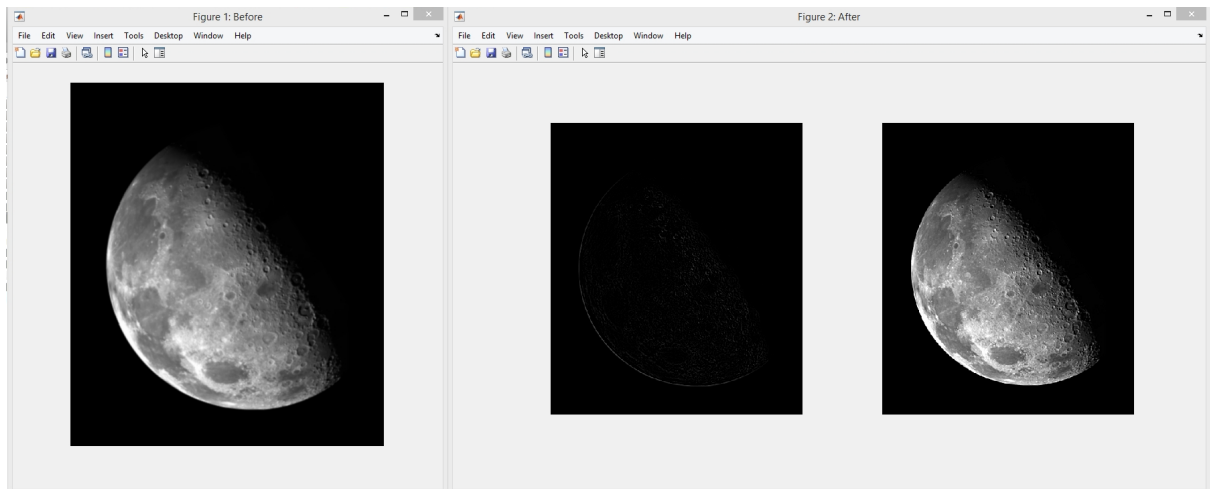


Рисунок 4 – Исходное изображение (слева) и изображения, обработанные по алгоритму лапласиана размером 3x3 (по центру) и после добавления фона на изображение (справа)

Вывод

В ходе лабораторной работы были произведены расчёт гистограммы изображений, их глобальная и локальная эквализации, пространственная пороговая и медианная фильтрации изображений, увеличение резкости изображения при помощи оператора Лапласа, были подробно изучены данные алгоритмы.