Федеральное агентство связи

Ордена трудового Красного Знамени федеральное государственное бюджетное

образовательное учреждение высшего образования

«Московский технический университет связи и информатики»

Кафедра Математической кибернетики и информационных технологий

Отчет по лабораторной работе №5

Синтезирование изображений

по дисциплине «Введение в профессию»

Выполнил: студент группы БВТ1902

Мартынов Николай Владимирович

Проверил: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Москва, 2019

Лабораторная работа №5

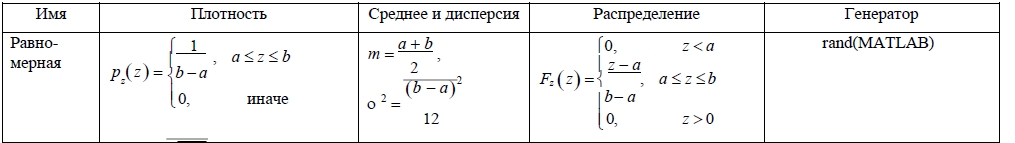
Синтезирование изображений

Цель работы – приобретение навыков синтезирования цифровых изображений, генерации шума различной природы возникновения. Использование базовых арифметических и геометрических операций над цифровым изображением.

Индивидуальное задание:

Вариант №6





Исходный код программы:

clc;

%1

pic = ones(800);

%2

F = noise(pic./2, 'uniform', 800, 800);

%3

shsh = figure('visible','off');

SHUM = histogram(F);

dir = ['lab5'];

if ~exist(dir)

mkdir(dir);

end

cd(dir);

saveas(SHUM, '3', 'png');

cd('../');

close(shsh)

%4

One = im2double(imread('1.png'));

midFig = merge(One, F, 365, 386);

cd(dir);

imwrite(midFig, '4.png');

cd('../');

%5

bil = imresize(One, 2); %Bilinear

midFig = merge(bil, F, 330, 372);

cd(dir);

if ~exist('(5)')

mkdir('(5)');

end

cd('(5)');

imwrite(midFig, 'bil2Mid.png'); %Save bilinear

cd('../');

cd('../');

Cub = imresize3(One, 3, 'cubic'); %Cub

midFig = merge(Cub, F, 295, 358);

cd(dir);

cd('(5)');

imwrite(midFig, 'cub3Mid.png'); %Save cub

cd('../');

cd('../');

Neigh = imresize3(One, 0.5, 'nearest'); %Nearest-neighbor

midFig = merge(Neigh, F, 385, 393);

cd(dir);

cd('(5)');

imwrite(midFig, 'NN0.5Mid.png'); %Save nearest-neighbor

cd('../');

cd('../');

%6

Salty = noise(pic./2, 'salt & pepper', 800, 800);

tri = im2double(imread('2.png'));

cir = im2double(imread('3.png'));

tri = imresize3(tri, 2, 'nearest');

cir = imresize3(cir, 2, 'nearest');

Salty = merge(tri, Salty, 5, 5);

Salty = merge(cir, Salty, 690, 690);

cd(dir);

imwrite(Salty, '6.png');

cd('../');

%7

cd(dir);

imwrite(flip(Salty,2), '7.png');

cd('../');

%8

cd(dir);

imwrite(flip(Salty), '8.png');

cd('../');

%9

cd(dir);

imwrite(imrotate(Salty, 315), '9.png');

cd('../');

%10

cd(dir);

imwrite(imrotate(Salty, 45), '10.png');

cd('../');

%11

ground = imread('lion.png');

%12

cd(dir);

imwrite(imcrop(ground, [200 100 799 799]), '12.png');

cd('../');

%13

cd(dir);

imwrite(imcrop(ground, [200 100 799 799])\*0.25, '13.png');

cd('../');

%14

lion = imcrop(ground, [200 100 799 799])\*0.25;

lion = rgb2gray(lion);

lion = im2double(lion);

lion = merge(tri, lion, 5, 5);

lion = merge(cir, lion, 690, 690);

Safe1 = noise(lion, 'uniform', 800, 800);

cd(dir);

imwrite(Safe1, '14.png');

cd('../');

%15

Safe = Safe1;

for i = 1:size(Safe,1)

for j = 1:size(Safe,2)

Safe(i,j) = 1-Safe(i,j);

end

end

cd(dir);

imwrite(Safe, '15.png');

cd('../');

%16

lion = imcrop(ground, [330 90 799 799])\*0.58;

lion = rgb2gray(lion);

lion = im2double(lion);

lion = merge(One, lion, 500, 350);

Safe2 = noise(lion, 'uniform', 800, 800);

cd(dir);

imwrite(Safe2, '16.png');

cd('../');

%17

cd(dir);

imwrite(Safe2-Safe1, '17.png');

cd('../');

функция для генерации различных шумов на изображении

function R = noise(P, n\_type, M, N, a, b)

if nargin == 1

a = 0;

b = 1;

M = 1;

N = 1;

elseif nargin == 3

a = 0;

b = 1;

end

switch lower(n\_type)

case 'uniform'

if nargin <= 4

a = 0;

b = 1;

end

R = a + (b - a)\*rand(M, N);

R = imlincomb(1, P, 0.2, R);

case 'gaussian'

if nargin <= 4

a = 0;

b = 1;

end

R = a + b\*randn(M, N);

m = max(max(R));

l = min(min(R));

for i = 1:size(R,1)

for j = 1:size(R,2)

R(i,j) = (R(i,j)+abs(l))/(abs(l)+m);

end

end

R = imlincomb(0.9, P, 0.4, R);

for i = 1:size(R,1)

for j = 1:size(R,2)

R(i,j) = R(i,j)-0.1;

end

end

case 'salt & pepper'

if nargin <= 4

a = 0.05;

b = 0.05;

end

if (a + b) > 1

error('The sum Pa + Pb must not exceed 1.')

end

R(1:M, 1:N) = 0.5;

X = rand(M, N);

c = find(X <= a);

R(c) = 0;

u = a + b;

c = find(X > a & X <= u);

R(c) = 1;

R = imlincomb(0.2, P, 1, R);

for i = 1:size(R,1)

for j = 1:size(R,2)

R(i,j) = R(i,j)-0.1;

end

end

case 'lognormal'

if nargin <= 4

a = 0; %1

b = 1;

end

R = exp(b\*randn(M, N) + a);

R = imlincomb(0.2, P, 0.5, R);

case 'rayleigh'

R = a + (-b\*log(1 - rand(M, N))).^0.5;

R = imlincomb(0.2, P, 0.5, R);

case 'exponential'

if nargin <= 4

a = 1;

end

if a <= 0

error('Parameter a must be positive for exponential n\_type.')

end

k = -1/a;

R = k\*log(1 - rand(M, N));

R = imlincomb(0.2, P, 0.5, R);

case 'erlang'

if nargin <= 4

a = 2;

b = 5;

end

if (b ~= round(b) || b <= 0)

error('Param b must be a positive integer for Erlang.')

end

k = -1/a;

R = zeros(M, N);

for j = 1:b

R = R + k\*log(1 - rand(M, N));

end

m = max(max(R));

for i = 1:size(R,1)

for j = 1:size(R,2)

R(i,j) = R(i,j)/m;

end

end

R = imlincomb(0.5, P, 1, R);

otherwise

error('Unknown distribution n\_type.')

end

функция слияния изображений

function F = merge(A, B, y, x)

for i = 1:size(A,1)

for j = 1:size(A,2)

if (A(i,j)<1)

B(i+x, j+y) = A(i,j);

end

end

end

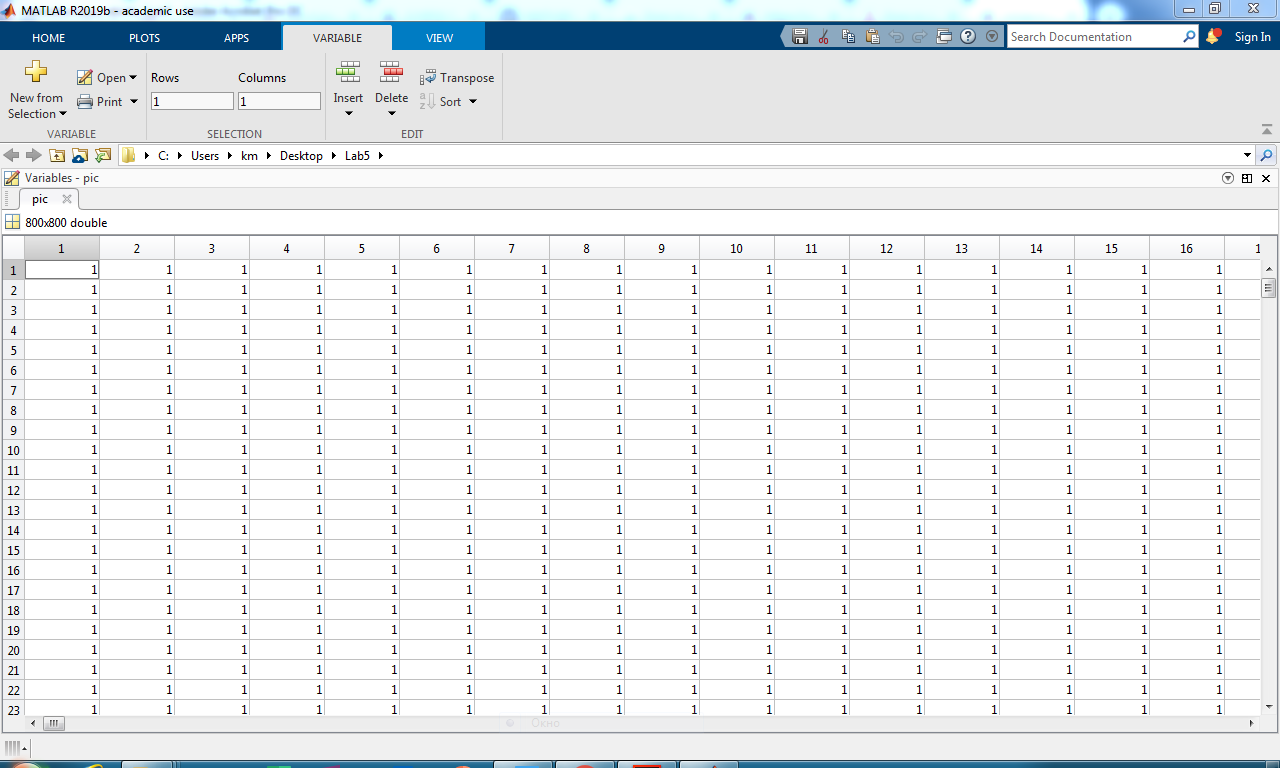
F = B;

end

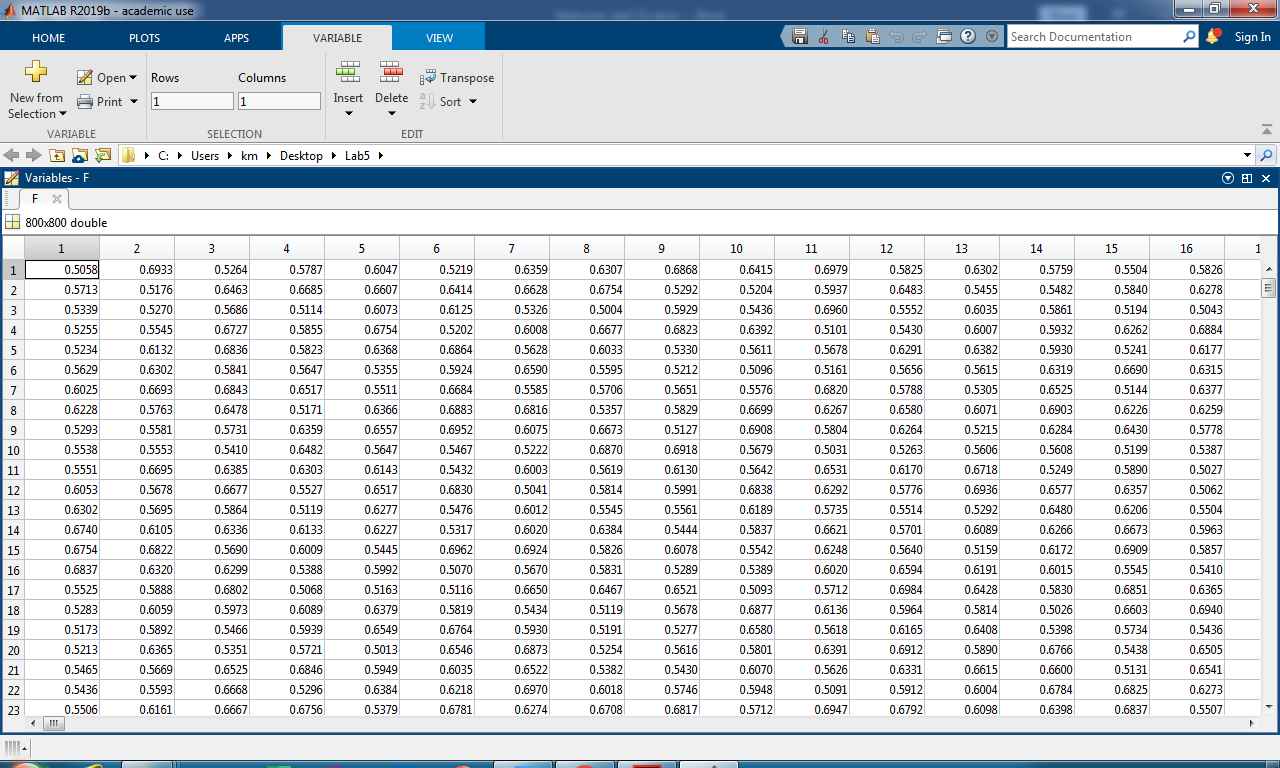
Скриншоты:

1. Синтезировать пустое полутоновое 8-битное изображение размером

800×800 пикселей.

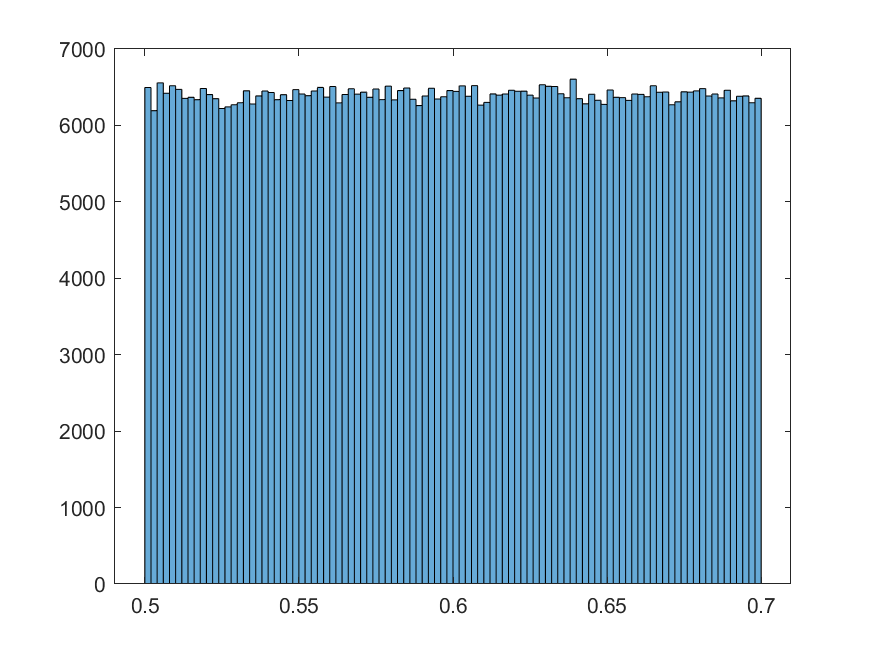


2. Нанести на изображение шум в соответствии с вариантом индивидуального задания по известному закону распределения.

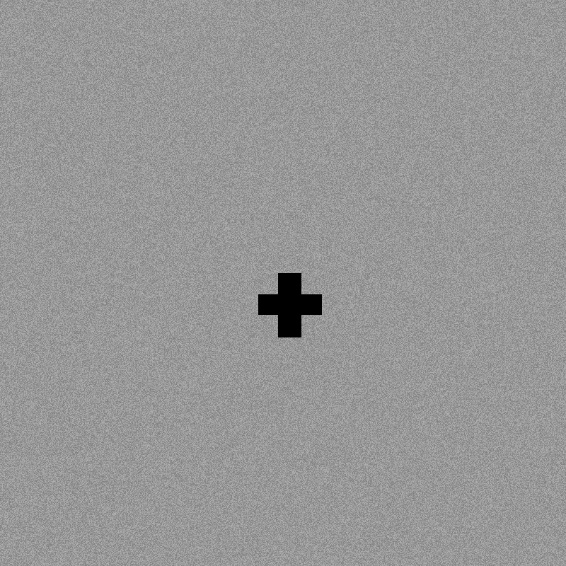
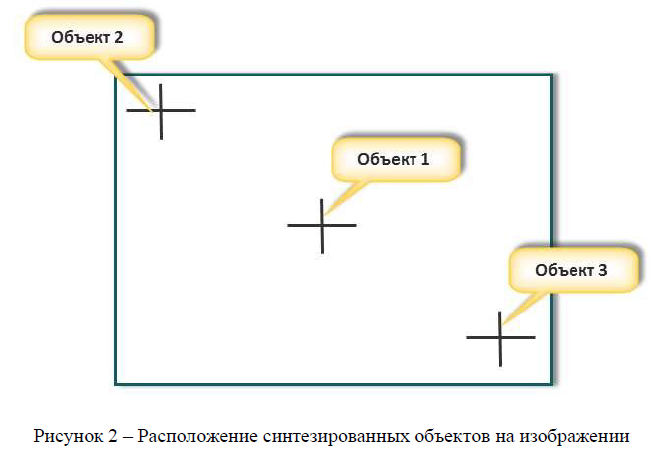


3. Построить гистограмму распределения полученного изображения шума.

Сохранить график гистограммы в директорию «…DIP\Lab2».



4. Синтезировать изображение Объекта 1 в соответствии с вариантом индивидуального задания в центре изображения (см. Рисунок 2).

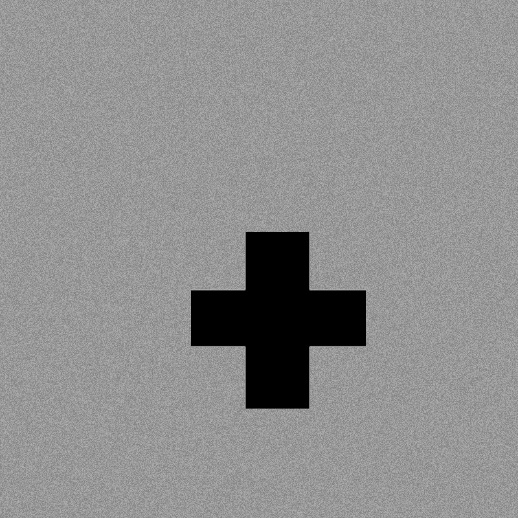


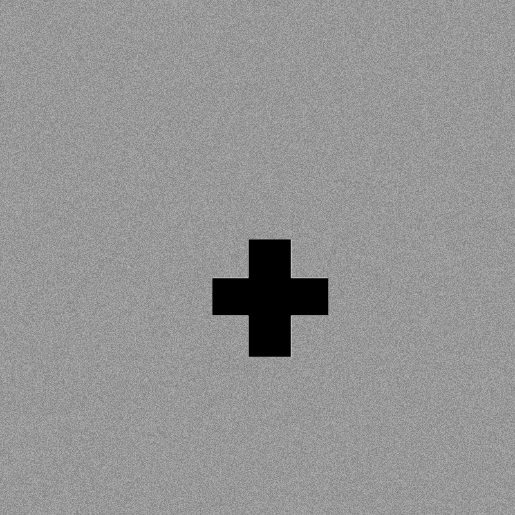
5. Выполнить масштабирование изображение, увеличив и уменьшив размер

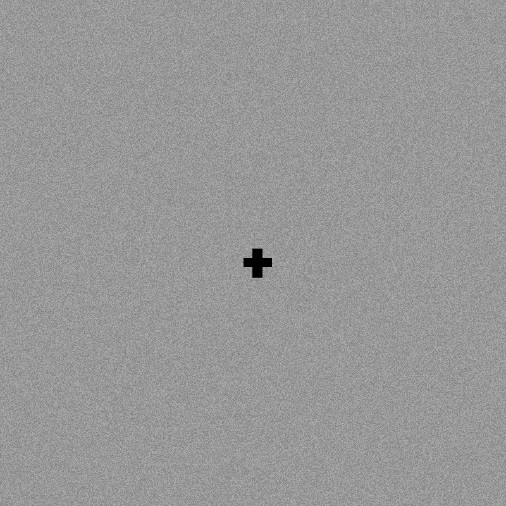
синтезированного объекта в 2 раза разными методами (ближайшего

соседа, билинейной или бикубической интерполяции). Сохранить

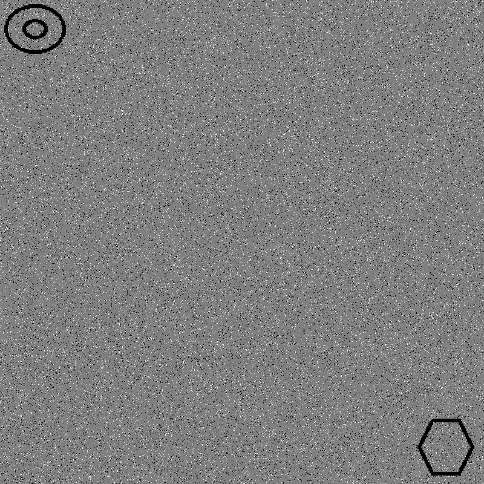
изображения в директорию «…DIP\Lab2».

- билинейная интерполяция

-бикубическая интерполяция

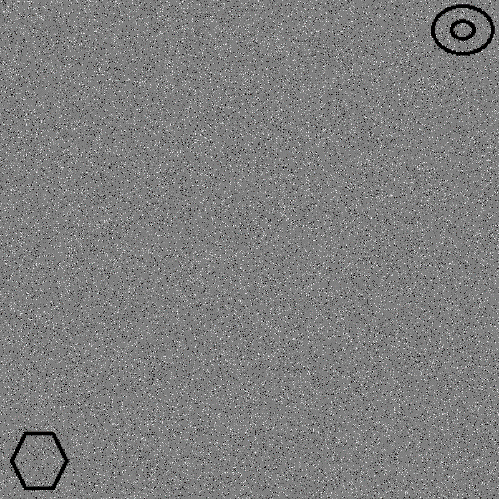
-метод ближайшего соседа

6. Синтезировать новое изображение с шумом и нанести объекты 2 и 3 в соответствии с вариантом индивидуального задания. Объекты должны располагаться в верхнем левом и нижнем правом углах синтезированного изображения (см. Рисунок 2). Размеры объектов подобрать самостоятельно, исходя из того, что наибольшая сторона (или диаметр) должна быть не менее 80 пикселей.



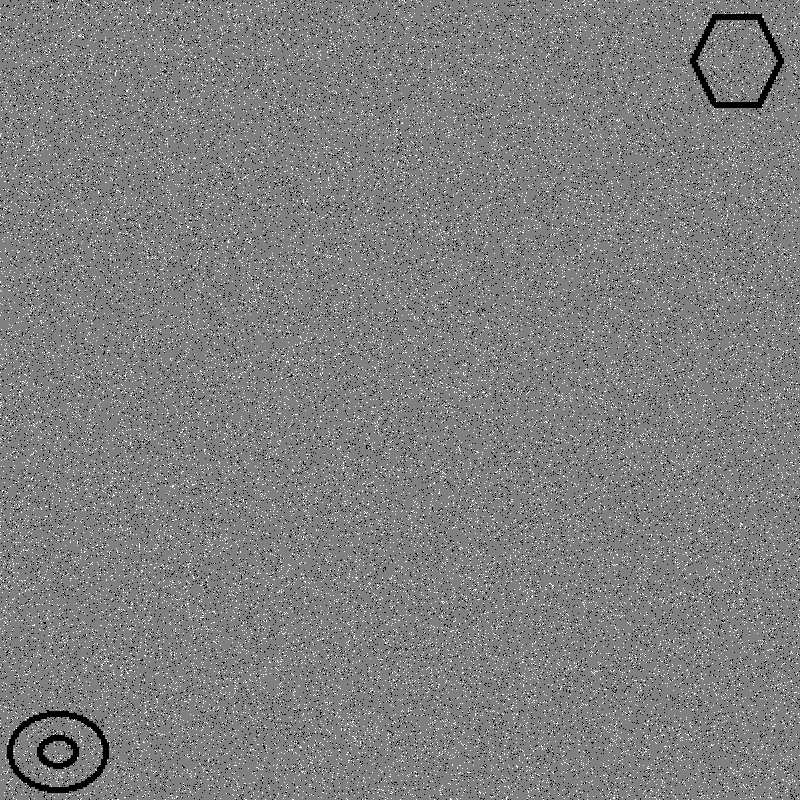
7. Выполнить операцию зеркального отражения по горизонтали. Сохранить

изображение в директорию «…DIP\Lab2».



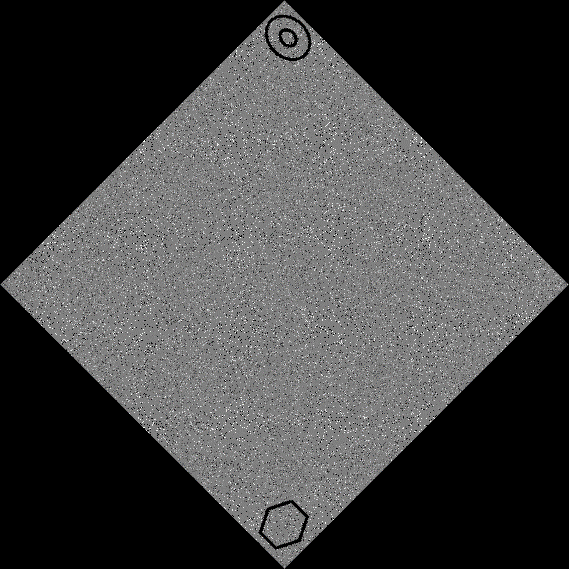
8. Выполнить операцию зеркального отражения по вертикали. Сохранить

изображение в директорию «…DIP\Lab2».



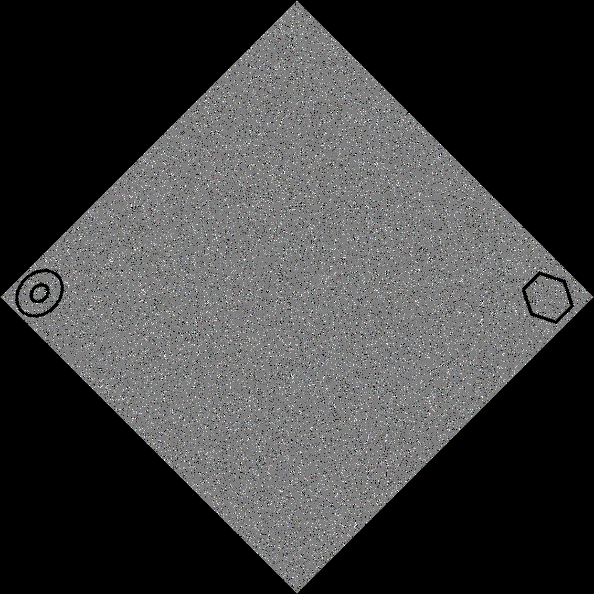
9. Выполнить операцию поворота изображения по часовой стрелке на 45°.

Сохранить изображение в директорию «…DIP\Lab2».



10. Выполнить операцию поворота изображения против часовой стрелки

на 45°. Сохранить изображение в директорию «…DIP\Lab2».



11. Выбрать из папки вспомогательных материалов «…\Labs\Lab2\Fon» любое из предложенных изображений фона.



12. Вырезать из изображения фона любой участок размером 800×800 пикселей.



13. Уменьшить яркость вырезанного фона в 4 раза.



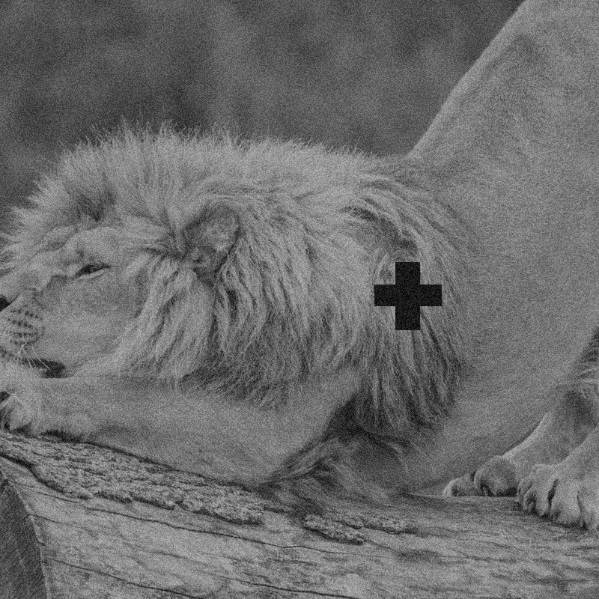
14. Сформировать новое полутоновое изображение из вырезанного фона с уменьшенной яркостью, 2-х объектов и добавить на изображение шум в соответствии с вариантом индивидуального задания. Сохранить полученное изображение в директорию «…DIP\Lab2».



15. Выполнить операцию «Негатив» для синтезированного изображения. Сохранить полученное изображение в директорию «…DIP\Lab2».



16. Сформировать новое полутоновое изображение из вырезанного фона с уменьшенной яркостью, одного объекта и добавить на изображение шум в соответствии с вариантом индивидуального задания. Сохранить полученное изображение в директорию «…DIP\Lab2».



17. Выполнить операцию разности двух изображений, полученных в пунктах 14 и 16. Сохранить полученное изображение в директорию «…DIP\Lab2».



Заключение:

В результате выполнения лабораторной работы, я изучил и использовал основные методы синтезирования цифровых изображений, а также генерацию шума различной природы возникновения. В работе я применил разность над двумя цифровыми изображениями.