Санкт-Петербургский национальный исследовательский университет информационных технологий механики и оптики

Цифровая обработка сигналов

**Лабораторная работа №7**

**Студент:**

Черезов Игорь Юрьевич

**Группа:**

P3400

**Преподаватель:**

Тропченко Андрей Александрович

Санкт-Петербург

2020

# Цель работы

* получение навыков сокрытия данных в цифровых полутоновых изображениях методами стеганографии;
* получение навыков самостоятельного программирования алгоритмов обработки изображений в системе Matlab.

# Задание

**Задание №1.**

1. Используя соответствующие функции пакета Matlab, напишите программу, реализующую алгоритм сокрытия данных в младшем разрядном срезе изображения.
2. Проведите эксперименты по сокрытию данных и их извлечению для различных изображений и типов скрываемых данных (бинарных изображений, данных типа .txt, .dat).
3. Определите количественные характеристики, определяющие соответствие между пустым и заполненным стегоконтйнером.

**Задание №2**

1. Самостоятельно реализуйте алгоритм сокрытия данных (текста) в псевдо-белых и псевдо-черных пикселах изображения и с использованием средств пакета Matlab напишите программу, реализующую данный метод.
2. Выполните п.п. 2 и 3 задания №1.
3. Сравните емкость стегоконтейнеров для обоих методов.

Самостоятельно провести подобные манипуляции с несколькими различными изображениями. Привести вид исходного стегоконтейнера и гистограмму яркости пикселов подлежащего сокрытию бинарного изображения, стегоконтейнера после помещения в него скраваемой информации (с гистограммой) и вид восстановленного бинарного изображения (для каждого из изображений). Проанализировать недостатки и достоинства методов и вариант модификации для помещения скрываемой текстовой (символьной) информации.

# Выполнение

## Задание 1

Код

function g = hide\_data(img, data)

con = bitand(img, 254);

data = data/256;

con(1:200, 1:200) += data;

g = con;

endfunction

function retval = reveal\_data(img)

buf = bitand(img, 1);

buf = buf\*256;

retval = buf;

endfunction

img = imread("inconspicuous\_cat\_bw.jpg");

data = imread("qr.jpg");

hidden = hide\_data(img, data);

figure, imshow(img);

figure, imshow(hidden);

imwrite(hidden, "hidden.bmp")

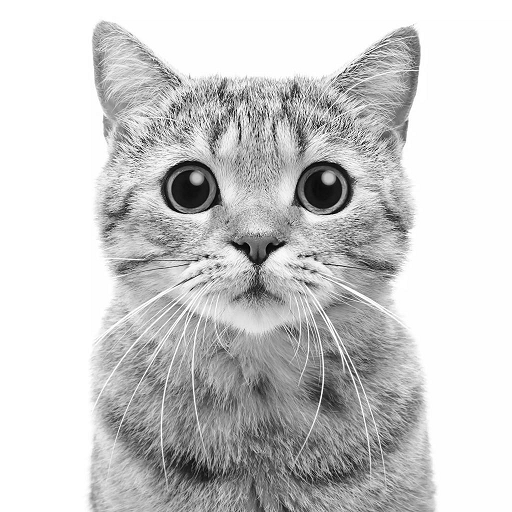
cor = corr2(img, hidden);

disp(["last bits corr " num2str(cor)]);

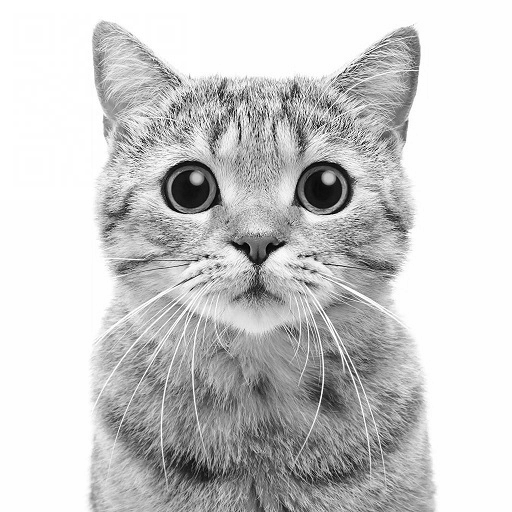
revealed = reveal\_data(hidden);

figure, imshow(revealed);

imwrite(revealed, "revealed.bmp")



Изображение 1 Оригинальное изображение



Изображение 2 Изображение со скрытым сообщением



Изображение 3 Извлечённое зашифрованное изображение

Коэффициент корреляции 0.99996.

## Задание 2

function g = hide\_pseudo(img, data)

con = img;

%prepare the container

%Compute the amount of data in bits which can be stored

%If data is too big for container return a corresponding message

m = size(con)(1,1);

n = size(con)(1,2);

count = 0;

for i=1:m

for j=1:n

byte = bitand(con(i, j), 0b11110000);

if (byte == 0 || byte == 0b11110000)

con(i, j) = bitand(con(i, j), 0b11110000);

count++;

endif

endfor

endfor

count \*= 4;

data\_size = size(data)(1,1) \* size(data)(1,2);

if data\_size > count

disp("Data too big for file");

disp(["Max data\_size is " num2str(count)]);

disp(["Current data\_size is " num2str(data\_size)]);

return

endif

%convert data into an array of 0b0000xxxx-like nums

data = data/256;

adapted\_data = zeros(data\_size/4)(1, :);

m = size(data)(1,1);

n = size(data)(1,2);

order = 0;

order = uint16(order);

for i=1:m

for j=1:n

adapted\_data(idivide(order, 4, "floor") + 1) += data(i, j) \* uint8(2^(mod(order, 4)));

order++;

endfor

endfor

%add adapted data to the container

m = size(con)(1,1);

n = size(con)(1,2);

a = size(adapted\_data)(1,2);

count = 1;

for i=1:m

for j=1:n

byte = bitand(con(i, j), 0b11110000);

if (byte == 0 || byte == 0b11110000)

con(i,j) += adapted\_data(count);

count++;

endif

if count > a

break;

endif

endfor

if count > a

break;

endif

endfor

g = con;

endfunction

function g = reveal\_pseudo(img, data\_size\_x)

%get adapted data hidden in image

con = img;

a = data\_size\_x^2/4;

adapted\_data = zeros(a)(1,:);

m = size(con)(1,1);

n = size(con)(1,2);

count = 1;

for i=1:m

for j=1:n

byte = bitand(con(i, j), 0b11110000);

if (byte == 0 || byte == 0b11110000)

adapted\_data(count) = bitand(con(i,j), 0b00001111);

count++;

endif

if count > a

break;

endif

endfor

if count > a

break;

endif

endfor

%transform adapted\_data into a pic

data = zeros(data\_size\_x);

m = data\_size\_x;

n = data\_size\_x;

order = 0;

order = uint16(order);

for i=1:m

for j=1:n

adapted\_data(idivide(order, 4, "floor") + 1) += data(i, j) \* uint8(2^(mod(order, 4)));

data(i,j) = bitand(adapted\_data(idivide(order, 4, "floor") + 1),uint8(2^(mod(order, 4))));

if data(i, j) > 0

data(i, j) /= data(i, j);

endif

order++;

endfor

endfor

g = data\*256;

endfunction



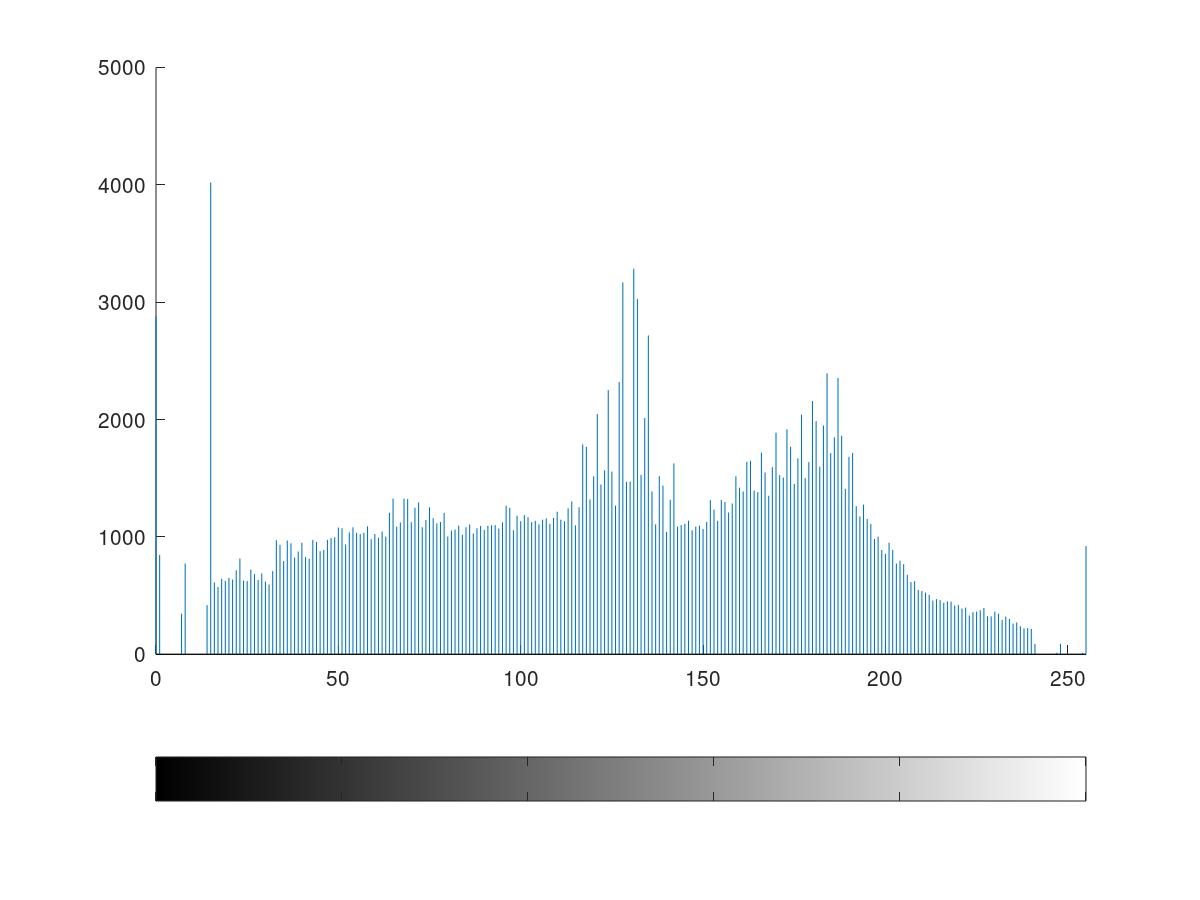
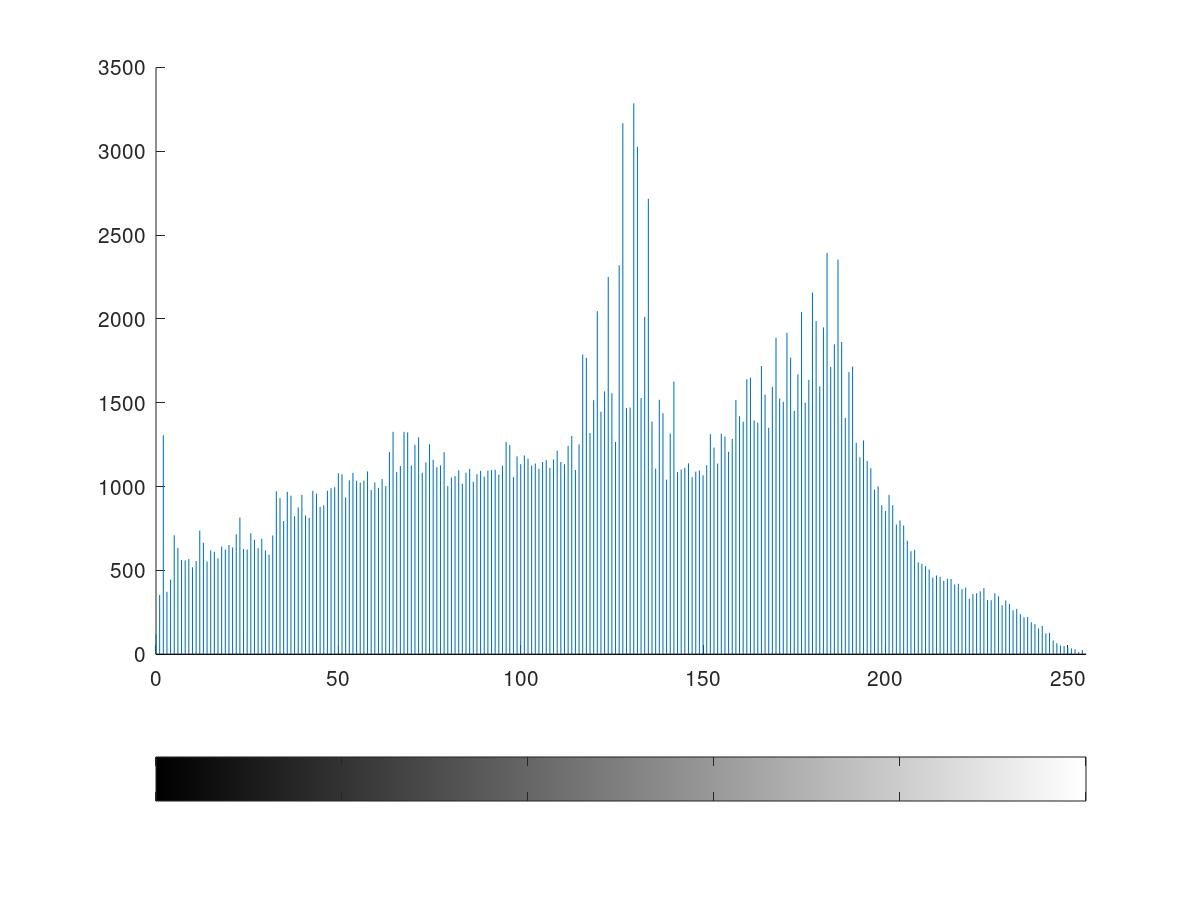
Изображение 4 Исходное изображение



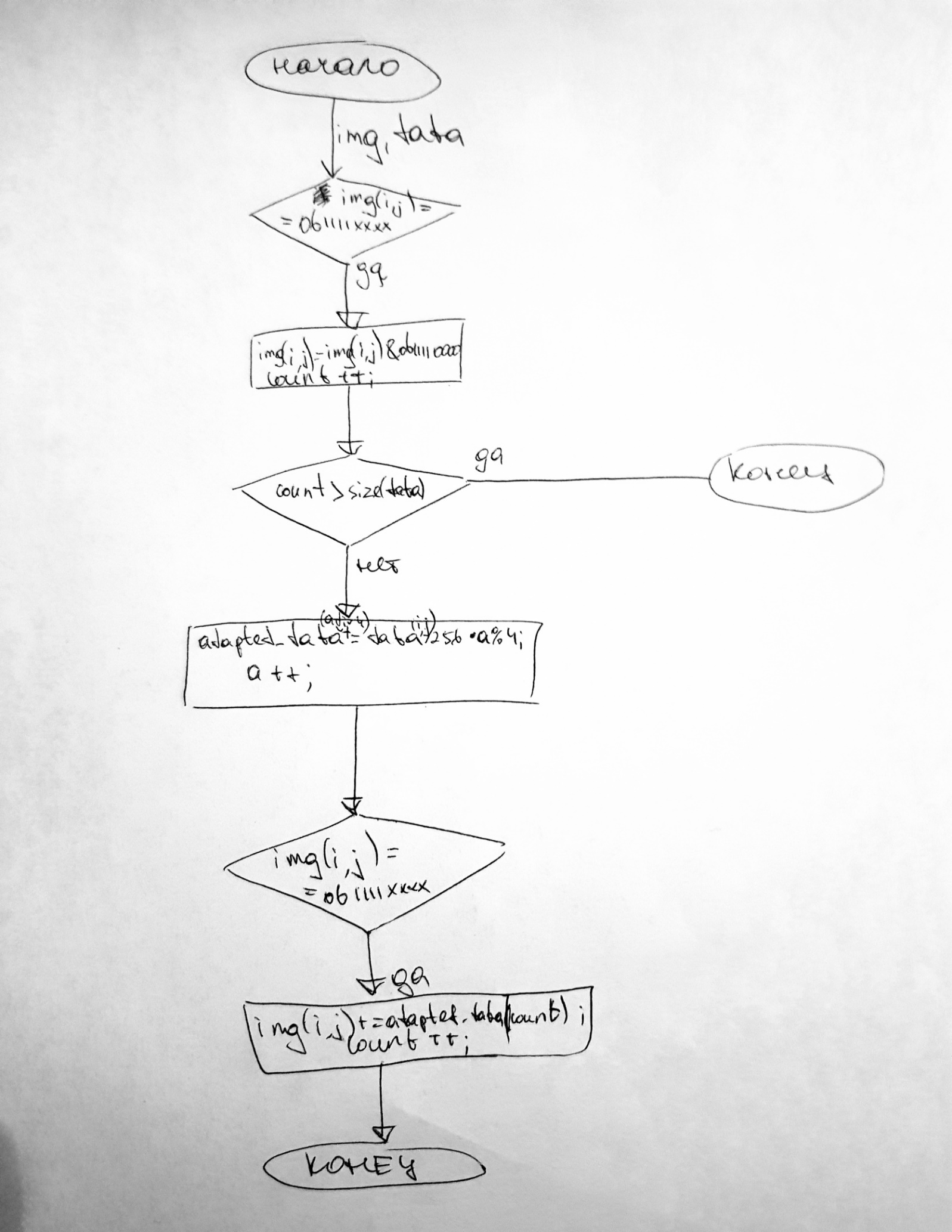
Изображение 5 Изображение со скрытым сообщением



Изображение 6 Извлечённое изображение



Изображение 7 Гистограммы яркости изображения до и после кодирования скрытого сообщения



Изображение 8 Блок-схема алгоритма сокрытия изображения в псевдо-чёрных и -белых пикселах

# Выводы

В ходе выполнения данной лабораторной работы были изучены методы стеганографии, такие как сокрытие данных в младшем разрядном срезе и сокрытие данных в псевдо-белых и псевдо-черных пикселах изображения.