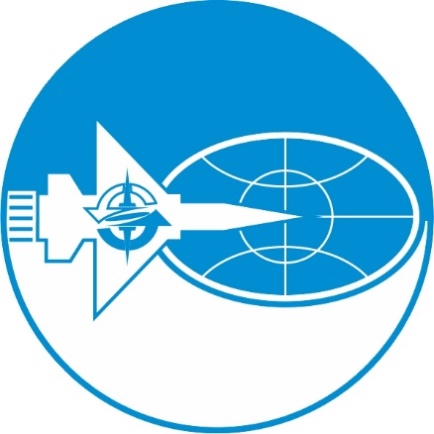
МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ

(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

КАФЕДРА 305

«ПИЛОТАЖНО-НАВИГАЦИОННЫЕ И ИНФОРМАЦИОННО-ИЗМЕРИТЕЛЬНЫЕ КОМПЛЕКСЫ»



Дисциплина «Специальные технологии 1 (обработка изображений)»

Отчет по лабораторной работе № 4

Вариант №6

Выполнил: студент группы М3О-406С-20

Орлов П.А.

Принял: доцент кафедры 305

к.т.н., Нгуен Н.М.

Цель работы:

Разработка программ для реализации методов фильтрации изображения на основе его производных.

# Теоретическая часть

Рассмотрим свойства первой и второй производных одномерного сигнала , где аргумент принимает дискретные значения с шагом 1. Первая производная дискретной функции вычисляется по формуле:

(1)

Первая производная имеет следующие свойства:

* она равна 0 на плоских участках;
* она ненулевая в начале и конце ступеньки или склона яркости;
* она ненулевая на склонах яркости.

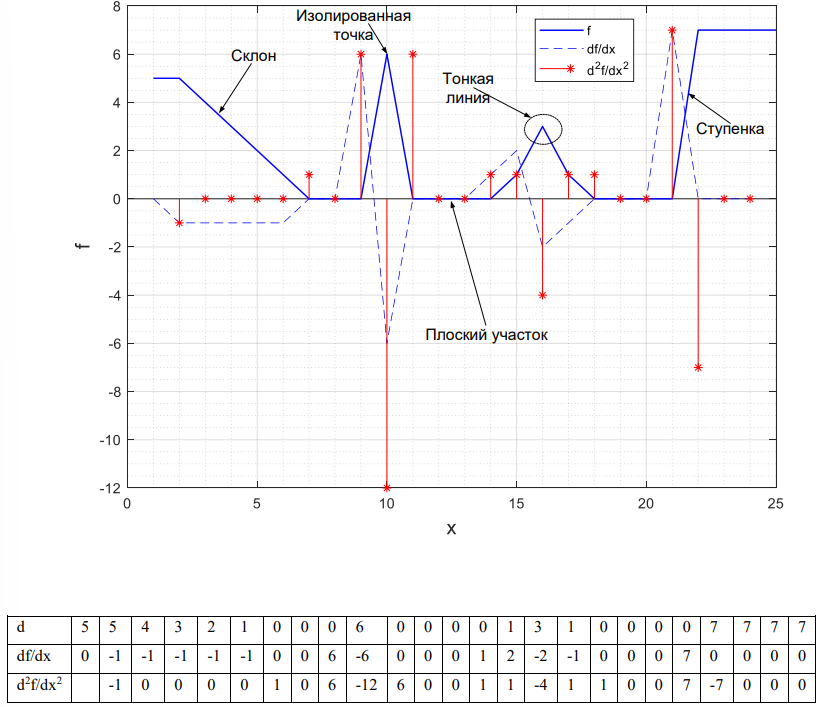
Вторая производная функции определяется по формуле:

(2)

Вторая производная имеет следующие свойства:

* она равна 0 на плоских участках;
* она ненулевая в начале и конце ступеньки или склона яркости;
* она равна 0 на склонах постоянной крутизны.

Пример производных дискретной функции показан на рисунке 1.

  
Рисунок 1 – Производные дискретной функции одной переменной

Как видно из рисунка 1, первая производная дает в результате более толстые контуры, а вторая производная больше реагирует на мелкие детали (изолированная точка и тонкая линия). Исходя из этого, на практике предпочтительно используют вторую производную для улучшения изображения, в то же время первую производную в основном используют для выделения контуров на изображении. Тем не менее, первая производная все же находит применение и в задачах улучшения изображения.

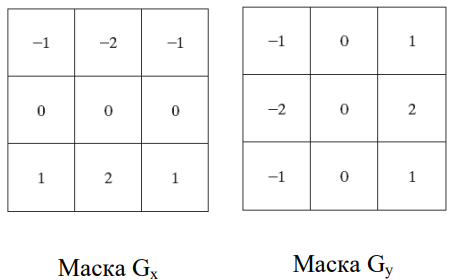
В прикладных приложениях по обработке изображений значение градиента изображения в точке приблизительно находится по модулям частных производных по строкам и столбцам:

, (3)

где ;

.

Для получения большей гладкости градиента-изображения применяют маски Собела, представленные на рисунке 2. Особенность масок Собела и, в частности, масок вычисления производных изображения в целом, заключается в том, что сумма всех элементов маски равна нулю

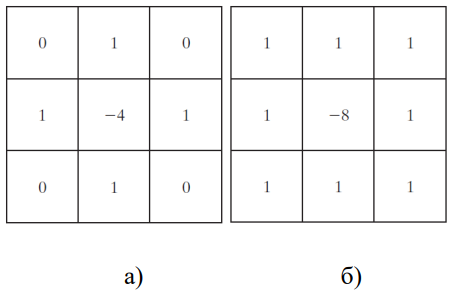
  
Рисунок 2 – Маски Собела

Двумерный лапласиан определяется так:

Для цифрового изображения лапласиан можно получить по формуле:

(4)

Правая часть уравнения (формула 4) является линейной комбинацией пикселя и его соседних пикселей, которая может быть реализована с помощью маски, представленной на рисунке 3а.

  
Рисунок 3 – Маски лапласиана: а – без учёта диагональных направлений, б – с учётом диагональных направлений

Чтобы получить более резкое изображение, в формуле лапласиана учитывают еще диагональные направления (т.е. вычисляются и частные производные изображения по диагональным направлениям). В этому случае применяется маска, показанная на рисунке 3б.

Границы объектов на изображении характеризуются высокочастотными элементами изображения. С целью улучшения изображения иногда необходимо подчеркивать эти детали на изображении. Для этого можно использовать фильтр с подъемом высоких частот, задаваемый формулой:

, (5)

где ;

‒ копия исходного изображения, которая получается путем применения либо оператора градиента, либо оператора лапласиана:

или

## Практическая часть

Листинг файла derivative.m:

%функция расчёта производной 1 порядка

function [der\_1] = derivative(v)

    v=double(v);

    [m,n]=size(v);

    der\_1=zeros(m,n);

    for i = 1:m-1

        der\_1(i,1)=v(i+1,1)-v(i,1);

    end

end

Листинг файла second\_derivative.m

%функция расчёта производной 2 порядка

function [der\_2] = second\_derivative(v)

    v=double(v);

    [m,n]=size(v);

    der\_2=zeros(m,n);

    for i = 2:m-1

        der\_2(i,1)=v(i+1,1)+v(i-1,1)-2\*v(i,1);

    end

end

Листинг файла lab\_4\_1\_demo.m:

clear all;

k=6;

addpath('images','lab\_1','lab\_2');

image=imread('abstraction.png');

[image,m,n]=is\_gray(image);

imshow(image);

figure;

V=(image(k,:))';

plot(V);

title(sprintf('Распределение яркости пикселей данного профиля по %i-ой строке',k));

der\_1\_V=derivative(V);

hold on;

plot(der\_1\_V);

der\_2\_V=second\_derivative(V);

plot(der\_2\_V);

legend(sprintf('Распределение яркости пикселей данного профиля'), ...

    'Производная первого порядка', ...

    'Производная второго порядка','Location','best');

figure;

imshow(V');

title('Обрабатываемая строка');

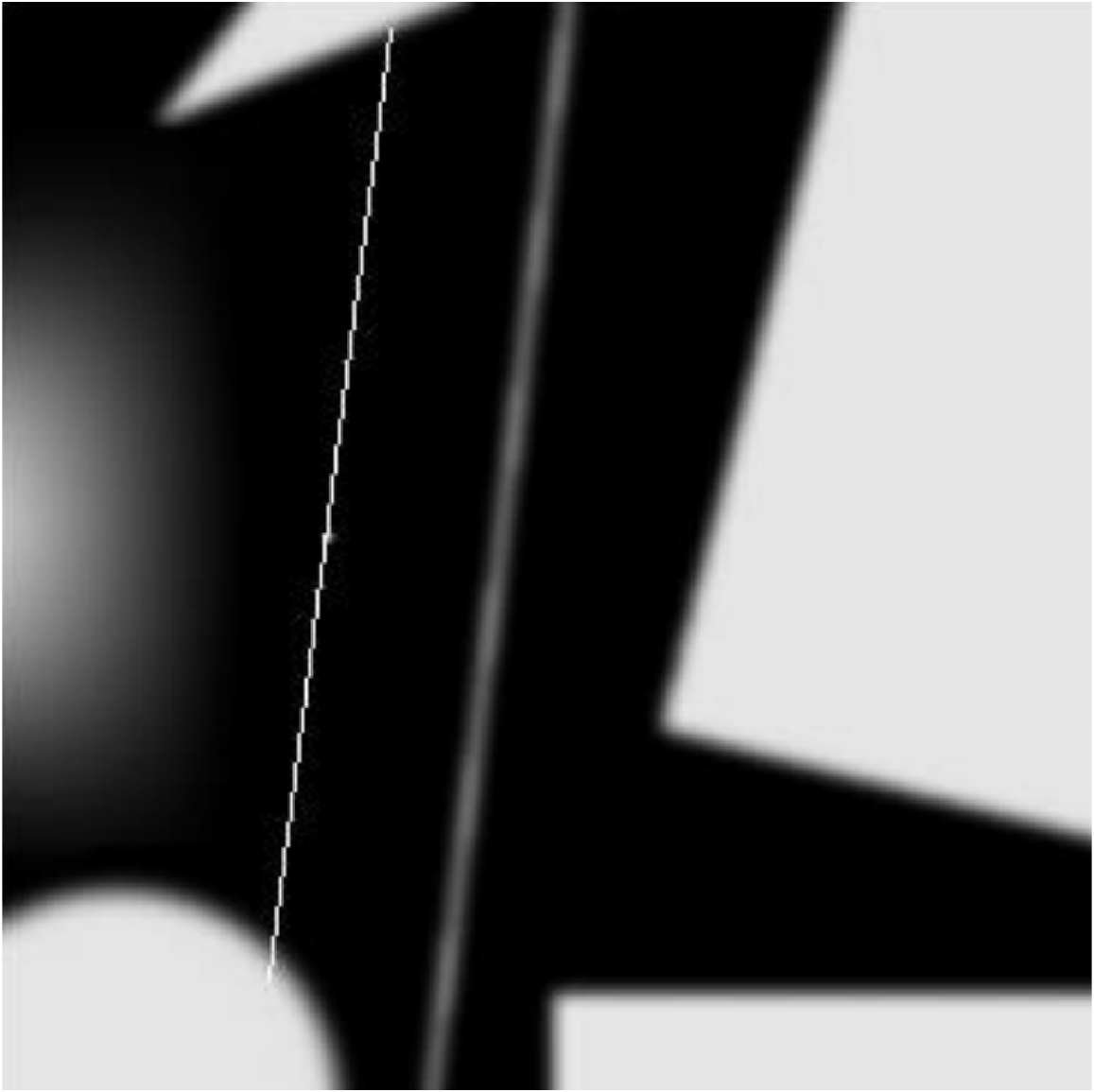
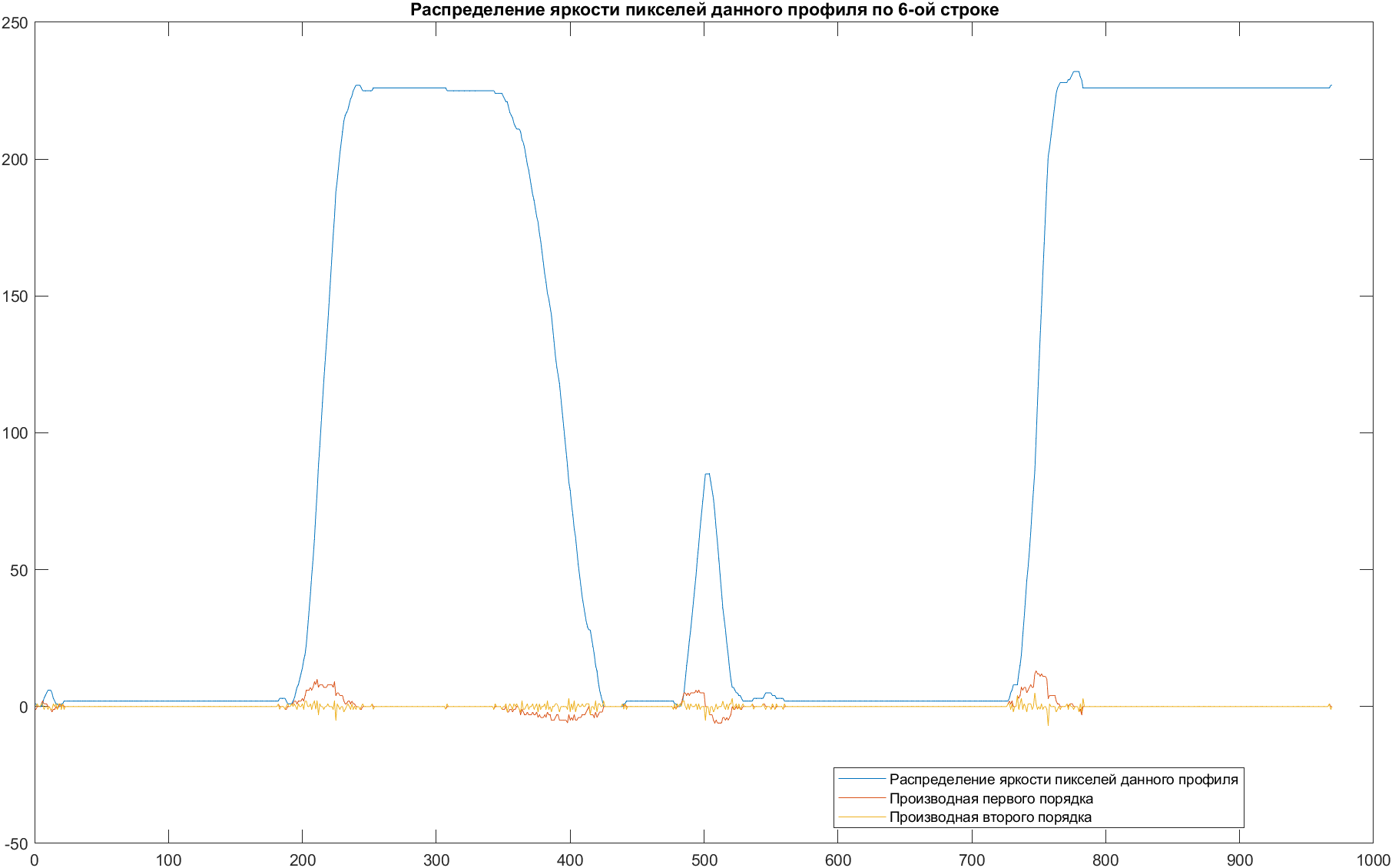
  
Рисунок 4 – Исходное изображение

  
Рисунок 5 – Обрабатываемая строка

  
Рисунок 6 – Результат работы программы

Листинг файла LINEARFILTER\_CASTOM\_MASK:

%функция пространственной фильтрации изображения с заданной маской

function [filtered\_image] = LINEARFILTER\_CASTOM\_MASK(image,mask)

    %добавление папки в path

    addpath('lab\_1')

    [m,n]=size(mask);

    a=(m-1)/2;

    b=(n-1)/2;

    %проверка изображения на ЧБ

    [image,M,N]=is\_gray(image);

    %увеличение размеров изображения для корректной работы фильтра

    M\_1=M+2\*a;

    N\_1=N+2\*b;

    f\_1=zeros(M\_1,N\_1);

    f\_1(1+a:M+a, 1+b:N+b)=image;

    filtered\_image=zeros(M,N);

    %фильтрация изображения

    for i = (1+a):(M+a)

        for j = (1+b):(N+b)

            R=0;

            for s=1:m

                for t=1:n

                    R=R+mask(s,t)\*f\_1(i-a-1+s,j-b-1+t);

                end

            end

            filtered\_image(i-a,j-b)=R;

        end

    end

end

Листинг файла lab\_4\_2\_demo.m:

clear all;

addpath('images','lab\_1');

f\_image=imread('terminal\_dark.png');

f=im2double(f\_image);

%маска Собела G\_x

w\_G\_x=[-1 -2 -1;

    0 0 0;

    1 2 1];

%маска Собела G\_y

w\_G\_y=[-1 0 2;

    -2 0 2;

    -1 0 1];

%маска лапласиана без учёта диагональных направлений

w\_L=[0 1 0;

    1 -4 1;

    0 1 0];

%маска лапласиана с учётом диагональных направлений

w\_L\_diag=[1 1 1;

            1 -8 1;

            1 1 1];

g\_G\_x=LINEARFILTER\_CASTOM\_MASK(f,w\_G\_x);

g\_G\_y=LINEARFILTER\_CASTOM\_MASK(f,w\_G\_y);

g\_L=LINEARFILTER\_CASTOM\_MASK(f,w\_L);

g\_L\_diag=LINEARFILTER\_CASTOM\_MASK(f,w\_L\_diag);

figure;

imshow(f\_image);

title('Исходное изображение');

figure;

imshow(mat2gray(g\_G\_x));

title('Изображение, обработанное маской Собела G\_x');

figure;

imshow(mat2gray(g\_G\_y));

title('Изображение, обработанное маской Собела G\_y');

figure;

imshow(mat2gray(abs(g\_G\_x)+abs(g\_G\_y)));

title('Изображение, обработанное фильтром на основе первой производной');

figure;

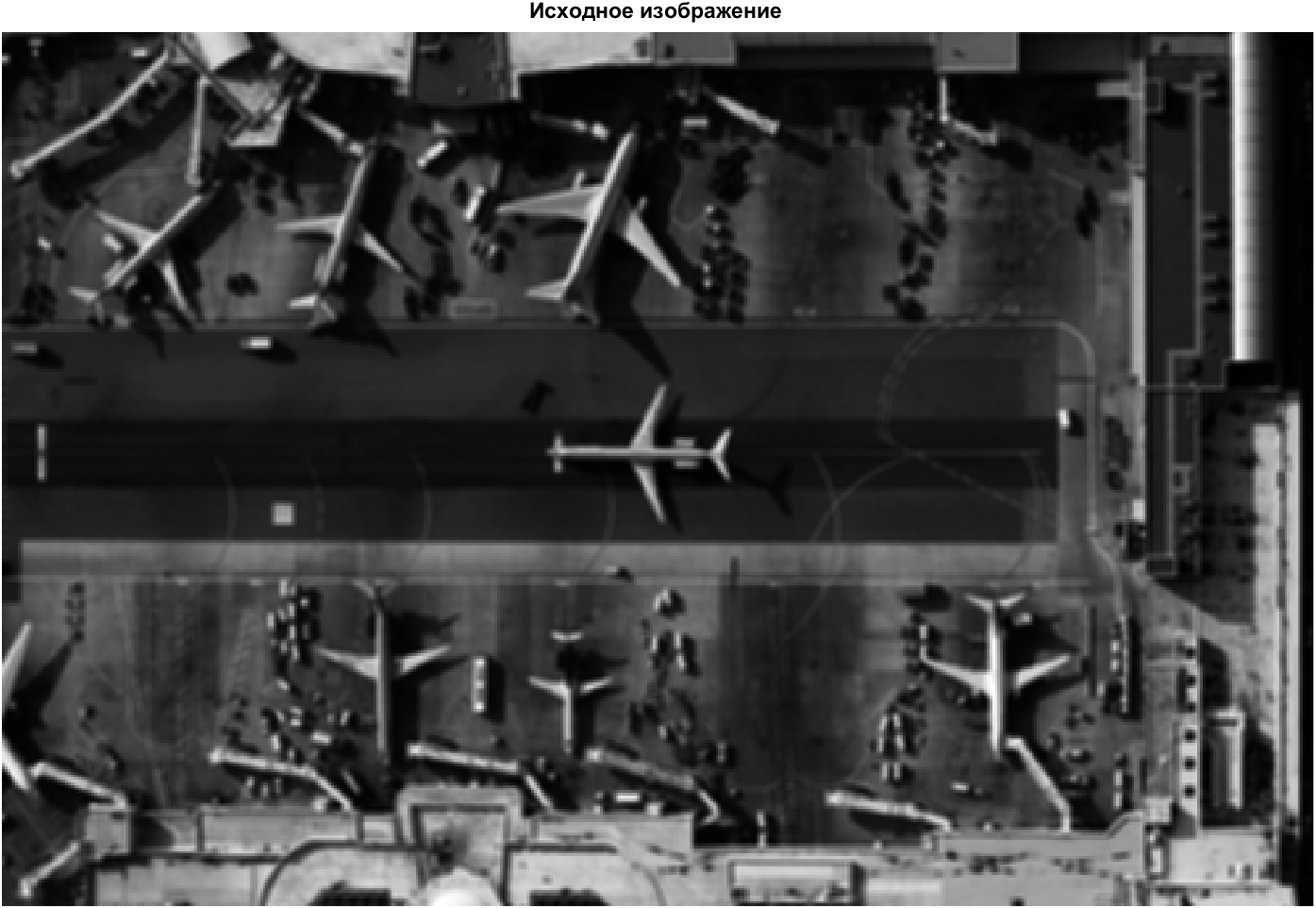
imshow(mat2gray(g\_L));

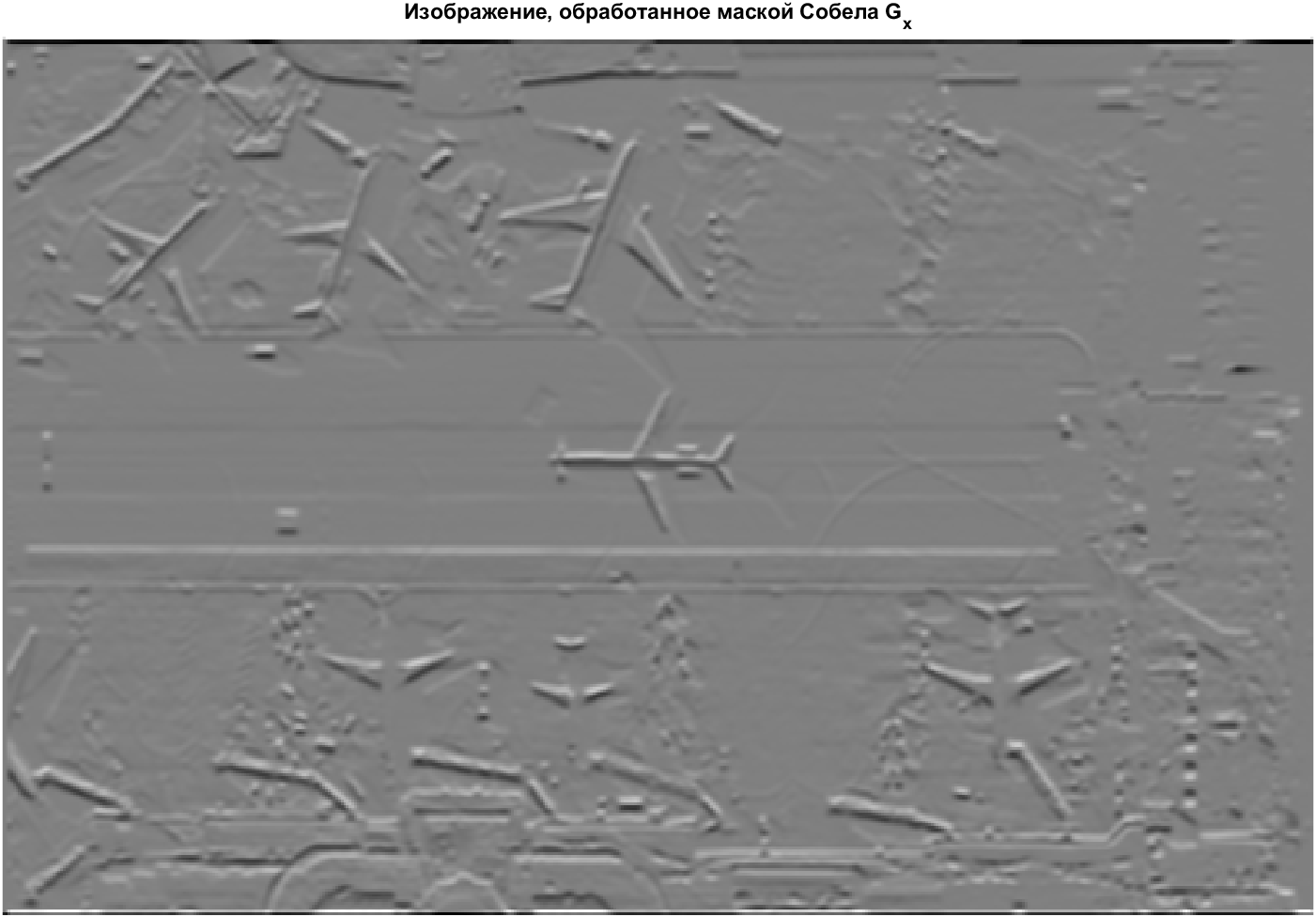
title({'Изображение, обработанное маской лапласиана';'без учёта диагональных направлений'});

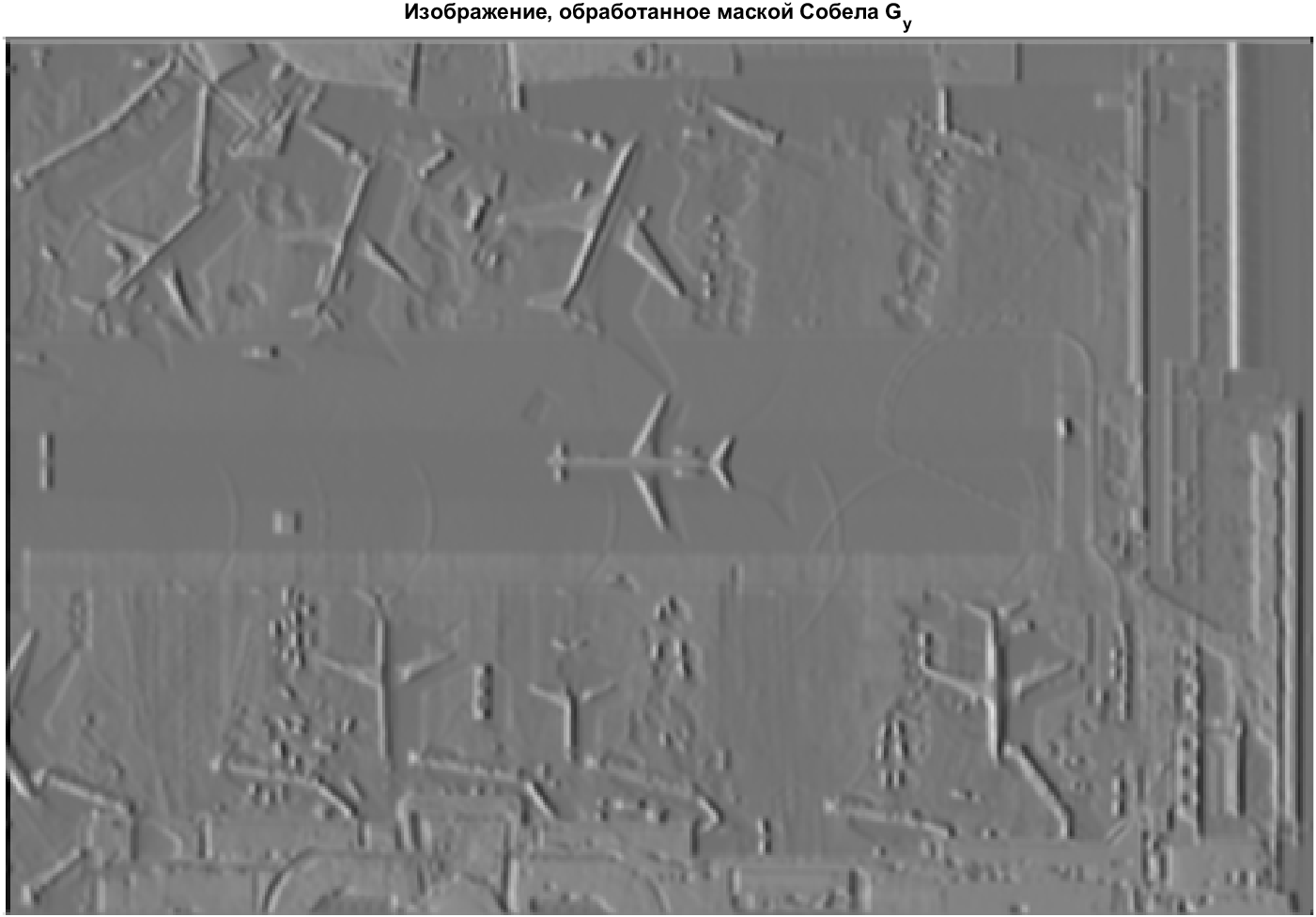
figure;

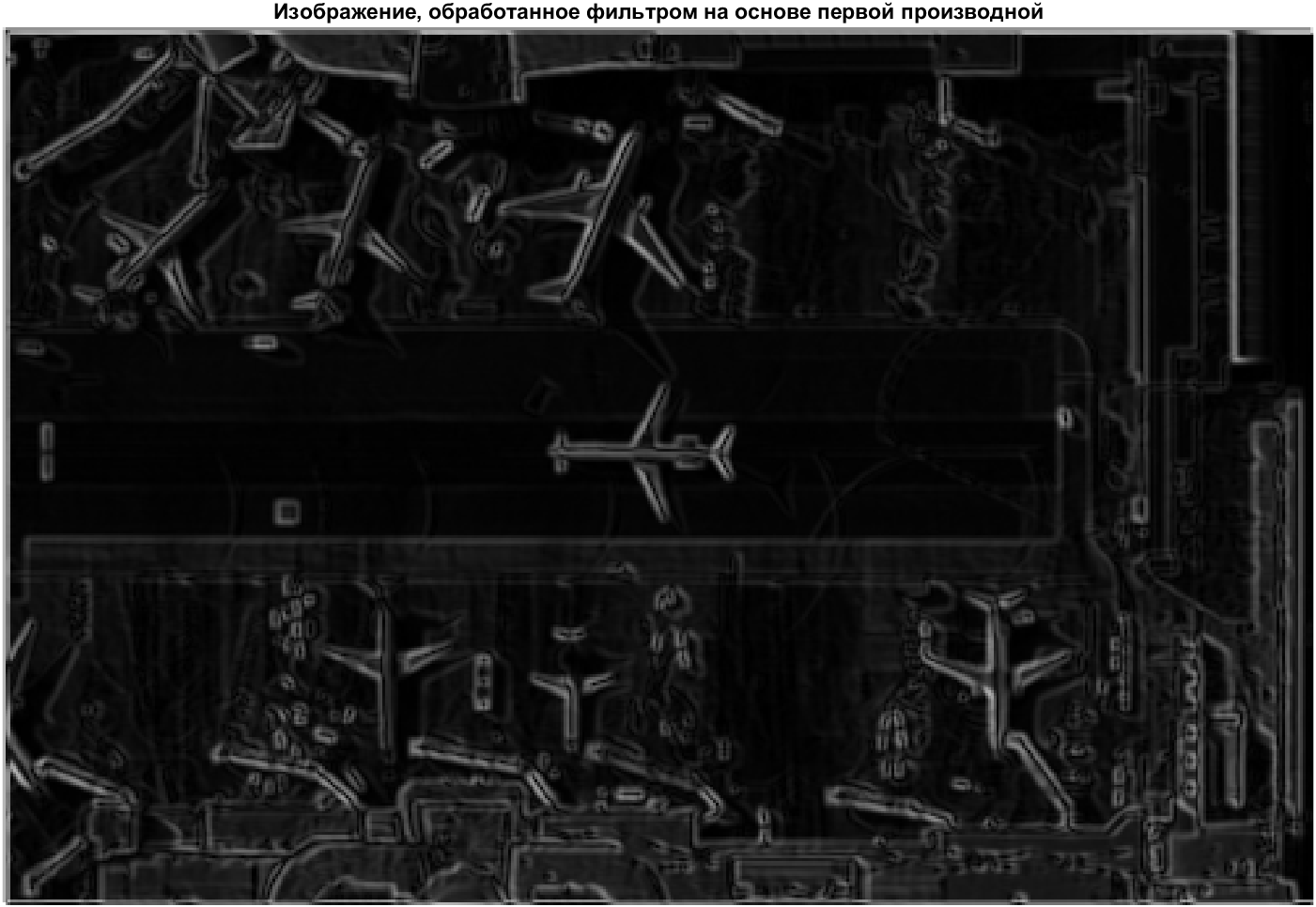
imshow(mat2gray(g\_L\_diag));

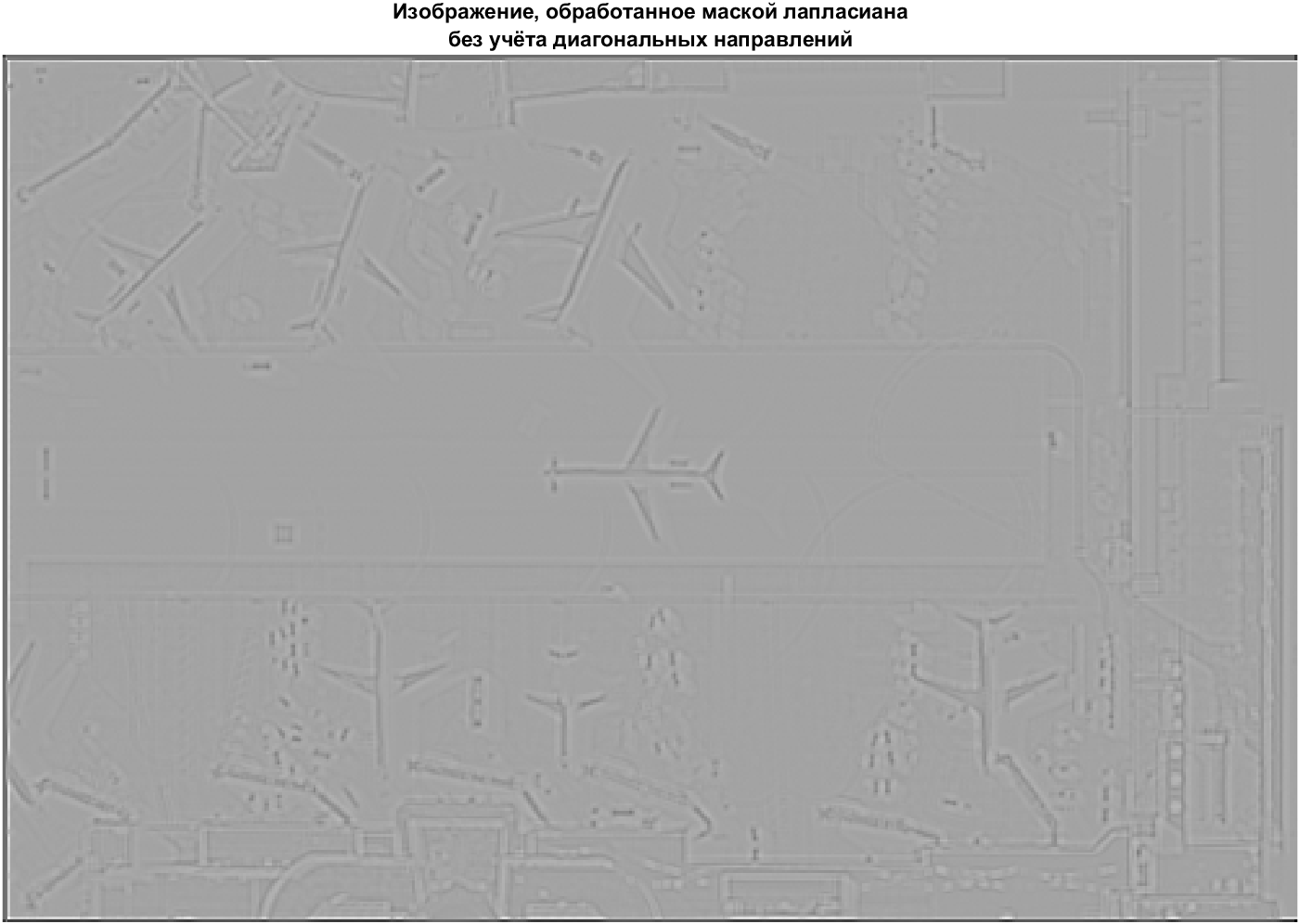
title({'Изображение, обработанное маской лапласиана';'с учётом диагональных направлений'});

  
Рисунок 7 – Исходное изображение









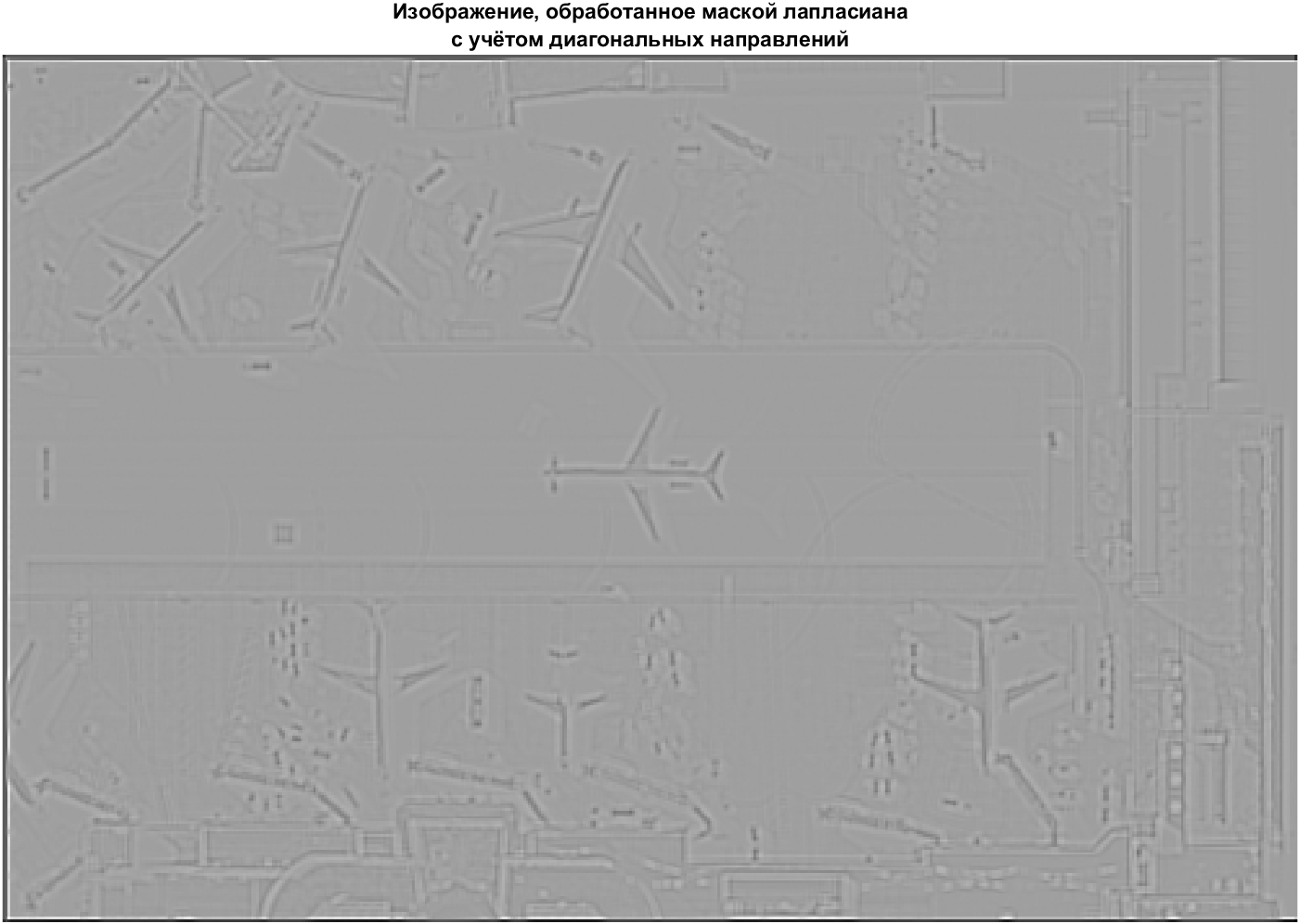


Рисунок 8 – Результат работы программы

Листинг файла lab\_4\_3\_demo.m:

clear all;

f\_image=imread('terminal\_dark.png');

%f2=rgb2gray(f1);

f=im2double(f\_image);

%маска Собела G\_x

w\_G\_x=[-1 -2 -1;

    0 0 0;

    1 2 1];

%маска Собела G\_y

w\_G\_y=[-1 0 2;

    -2 0 2;

    -1 0 1];

%маска лапласиана без учёта диагональных направлений

w\_L=[0 1 0;

    1 -4 1;

    0 1 0];

%маска лапласиана с учётом диагональных направлений

w\_L\_diag=[1 1 1;

            1 -8 1;

            1 1 1];

A=1.0;

g\_G\_x=LINEARFILTER\_CASTOM\_MASK(f,w\_G\_x);

g\_G\_y=LINEARFILTER\_CASTOM\_MASK(f,w\_G\_y);

g\_L=LINEARFILTER\_CASTOM\_MASK(f,w\_L);

g\_L\_diag=LINEARFILTER\_CASTOM\_MASK(f,w\_L\_diag);

g\_G=abs(g\_G\_x)+abs(g\_G\_x);

figure;

imshow(f);

title('Исходное изображение');

figure;

imshow(mat2gray(g\_G));

title('$|G\_{x}|+|G\_{y}|$','interpreter','latex');

for i=A:0.5:2.0

    figure;

    imshow(mat2gray(i\*im2double(f)-g\_G));

    title(sprintf('Преобразованное оператором градиента изображение, A=%d',i));

end

figure;

imshow(mat2gray(g\_L));

title('Преобразованное маской лапласиана без учета диагональных направлений изображение')

for i=A:0.5:2

    figure;

    imshow(mat2gray(i\*f+g\_L));

    title(sprintf('Преобразованное оператором лапласиана без учета диагональных направлений изображение, A=%d',i));

 end

figure;

imshow(mat2gray(g\_L\_diag));

title('Преобразованное маской лапласиана с учетом диагональных направлений изображение')

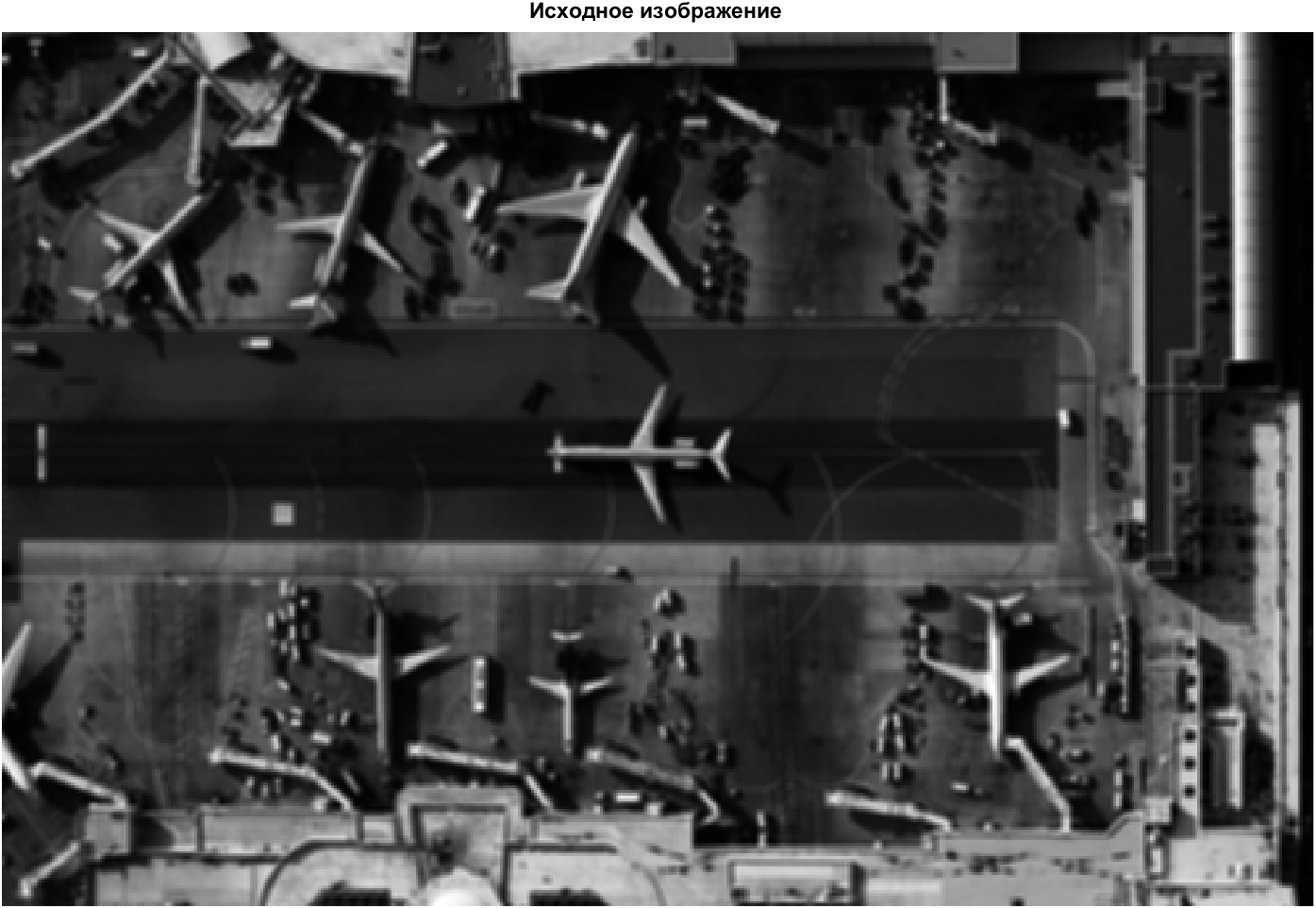
for i=A:0.5:2

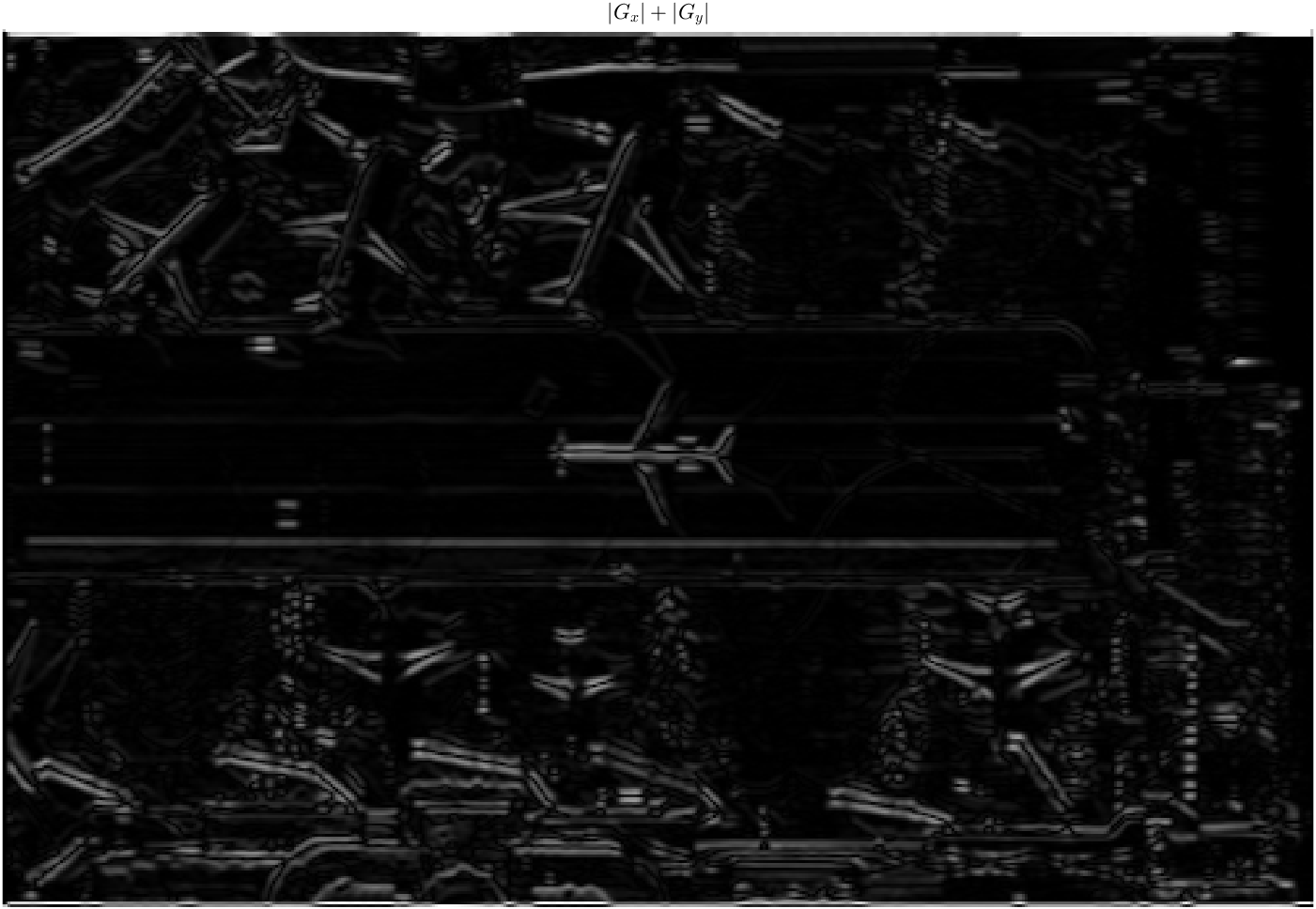
    figure;

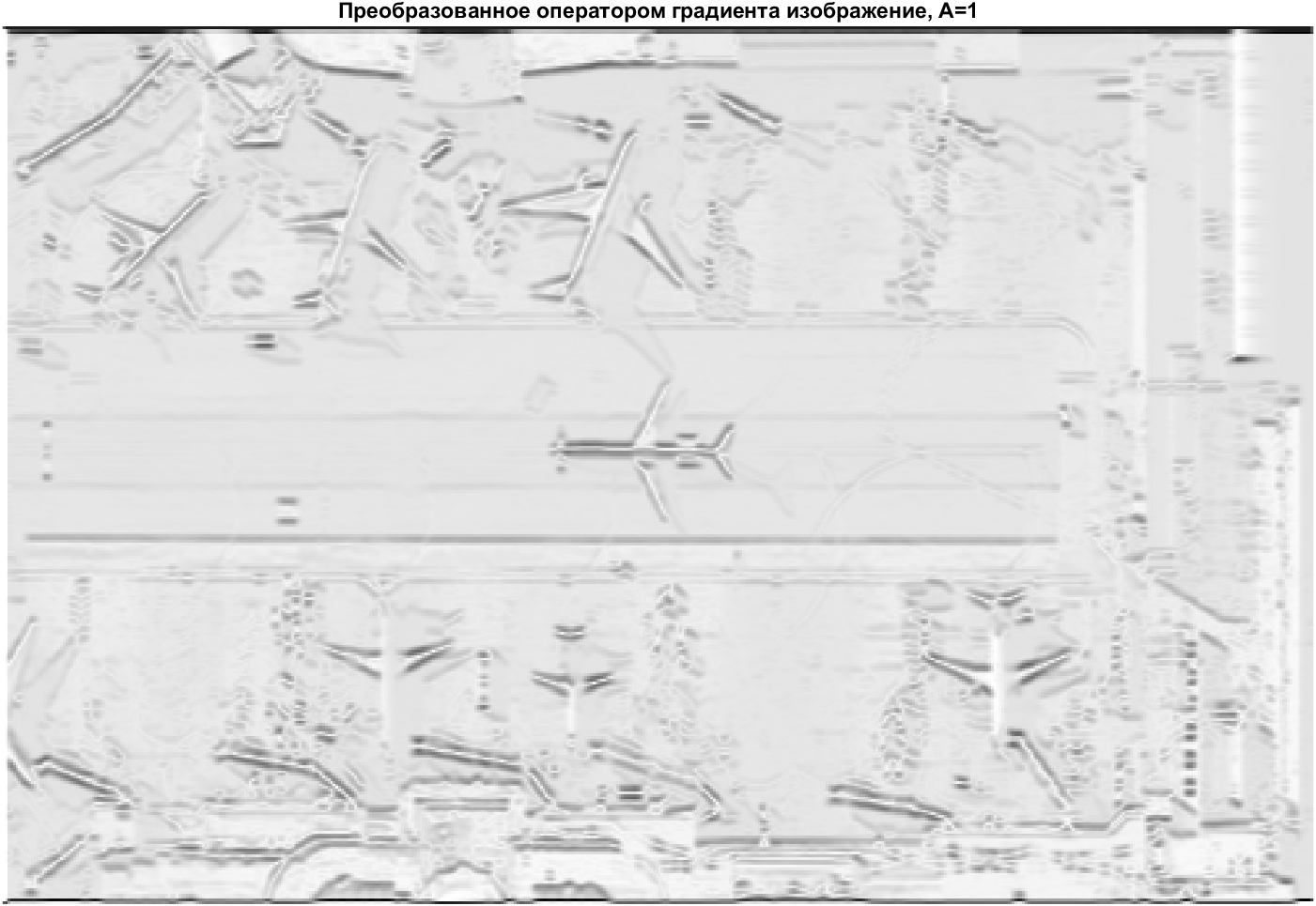
    imshow(mat2gray(i\*im2double(f)-g\_L\_diag));

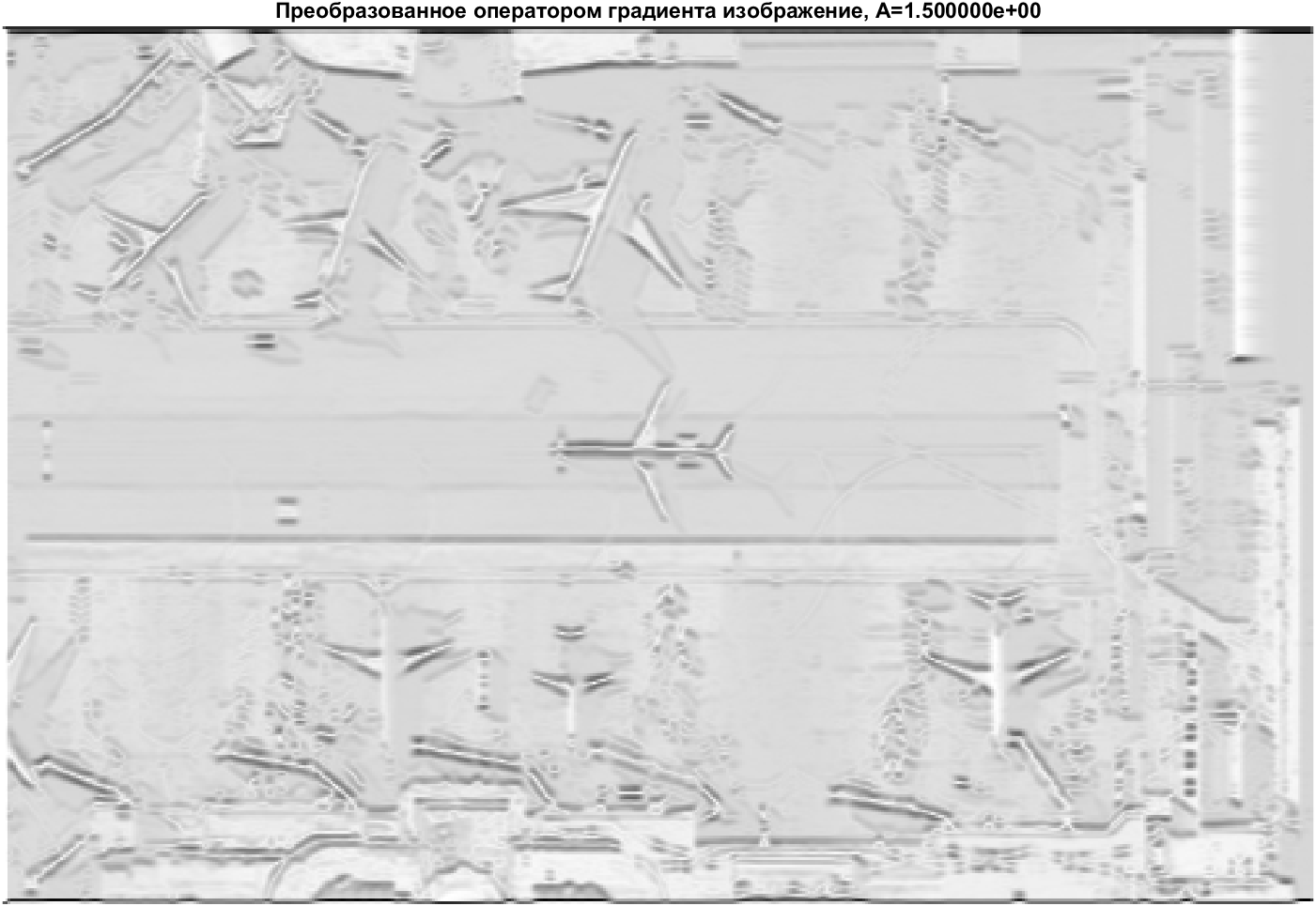
    title(sprintf('Преобразованное оператором лапласиана с учетом диагональных направлений изображение, A=%d',i));

end

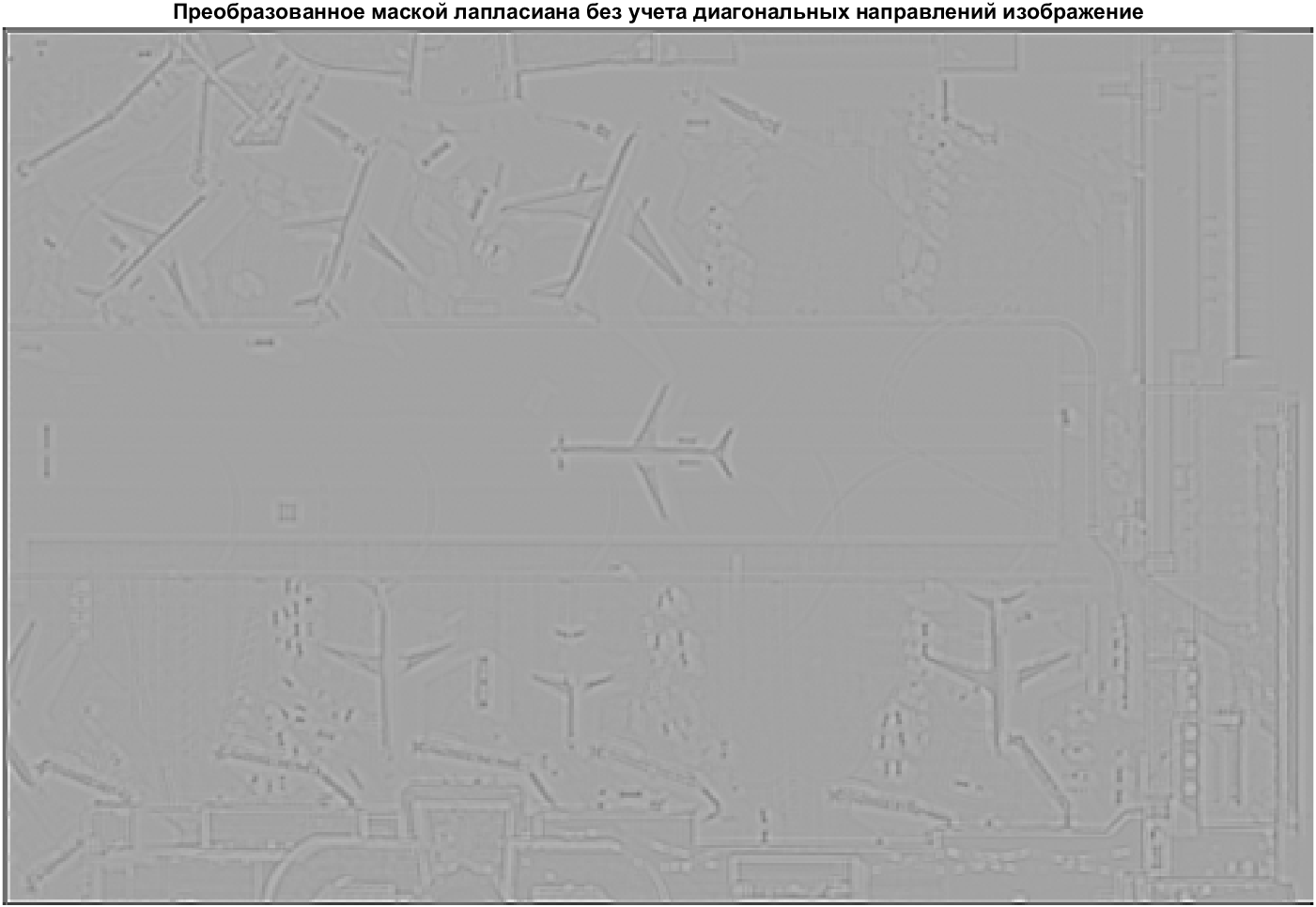








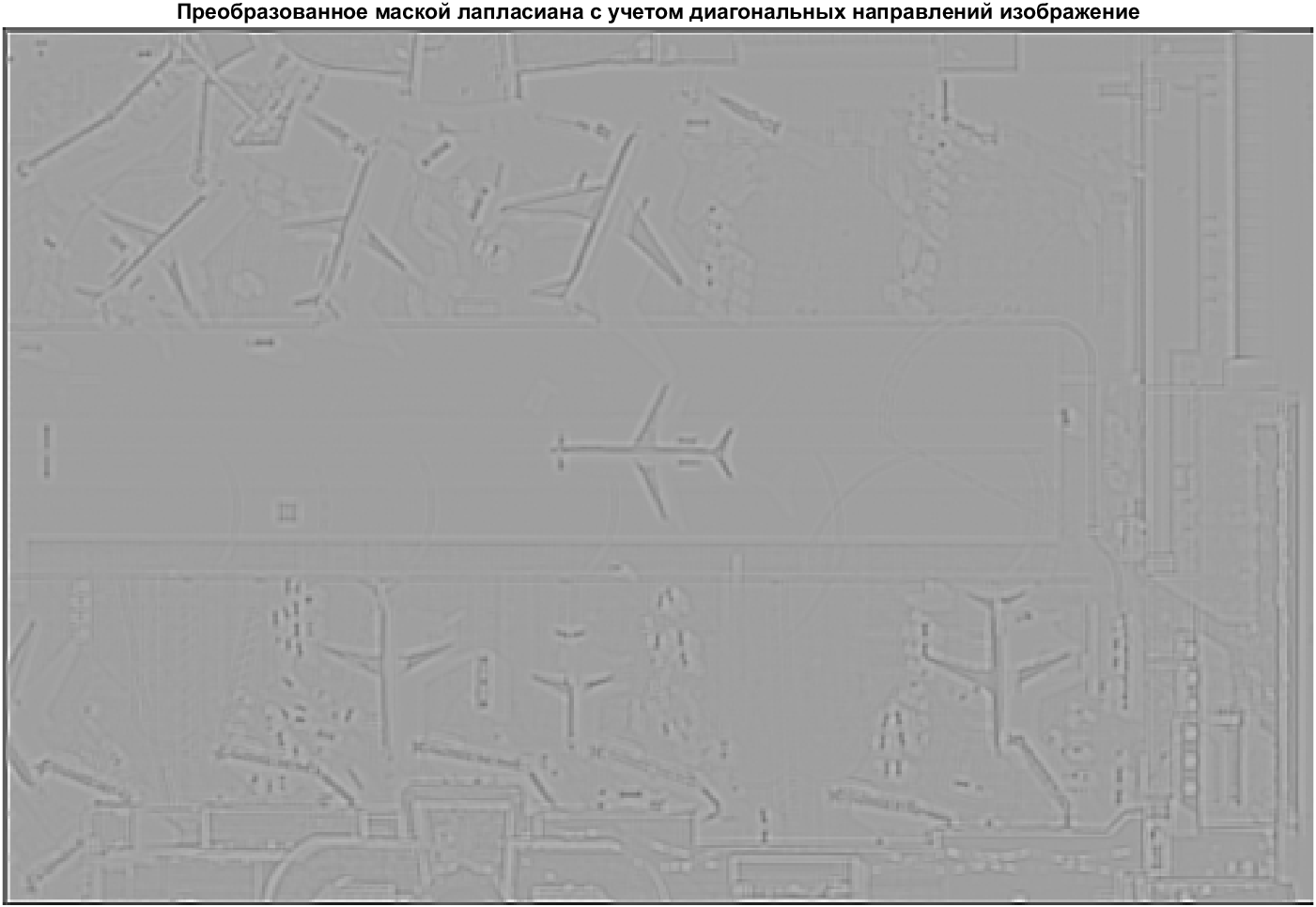


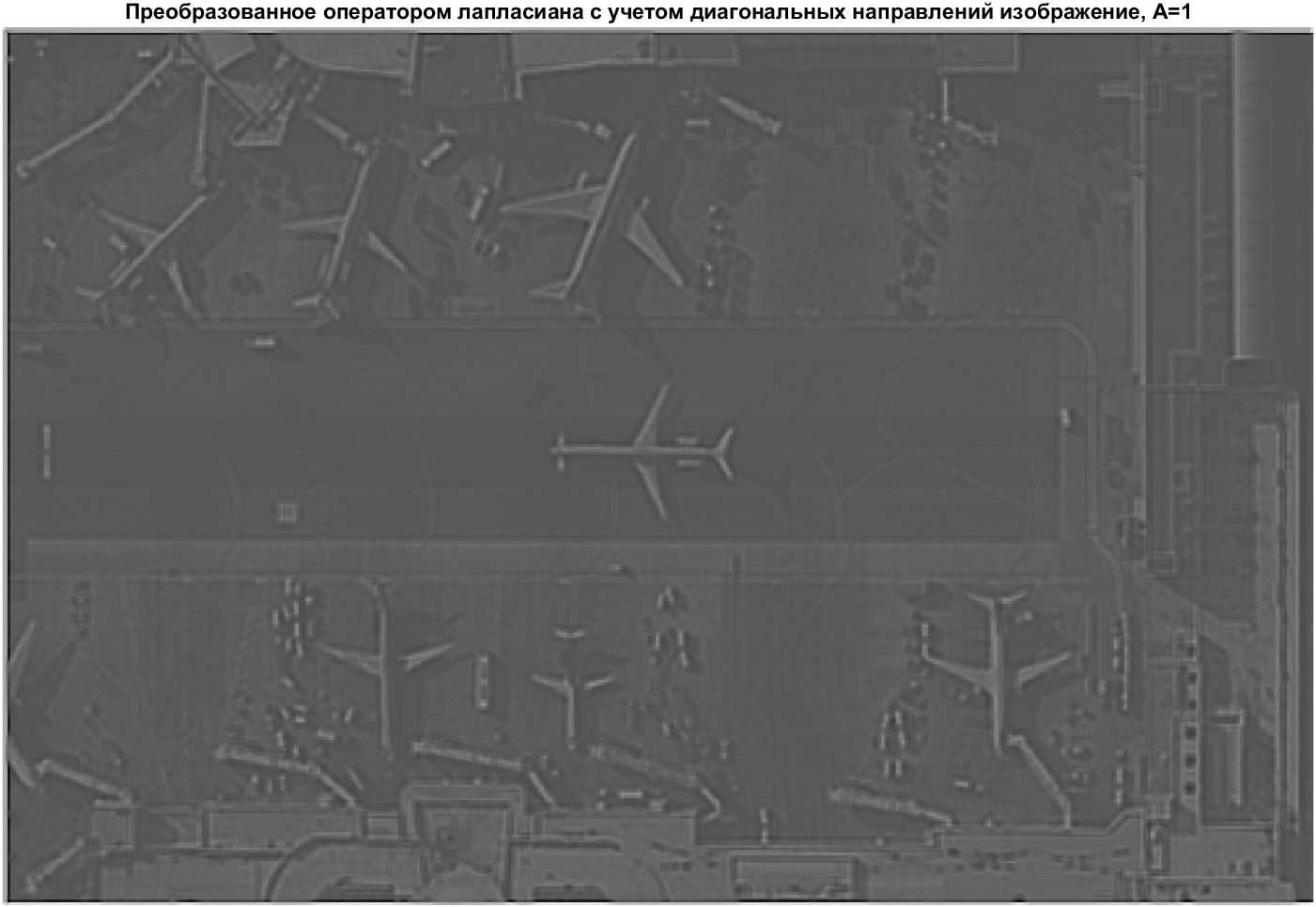


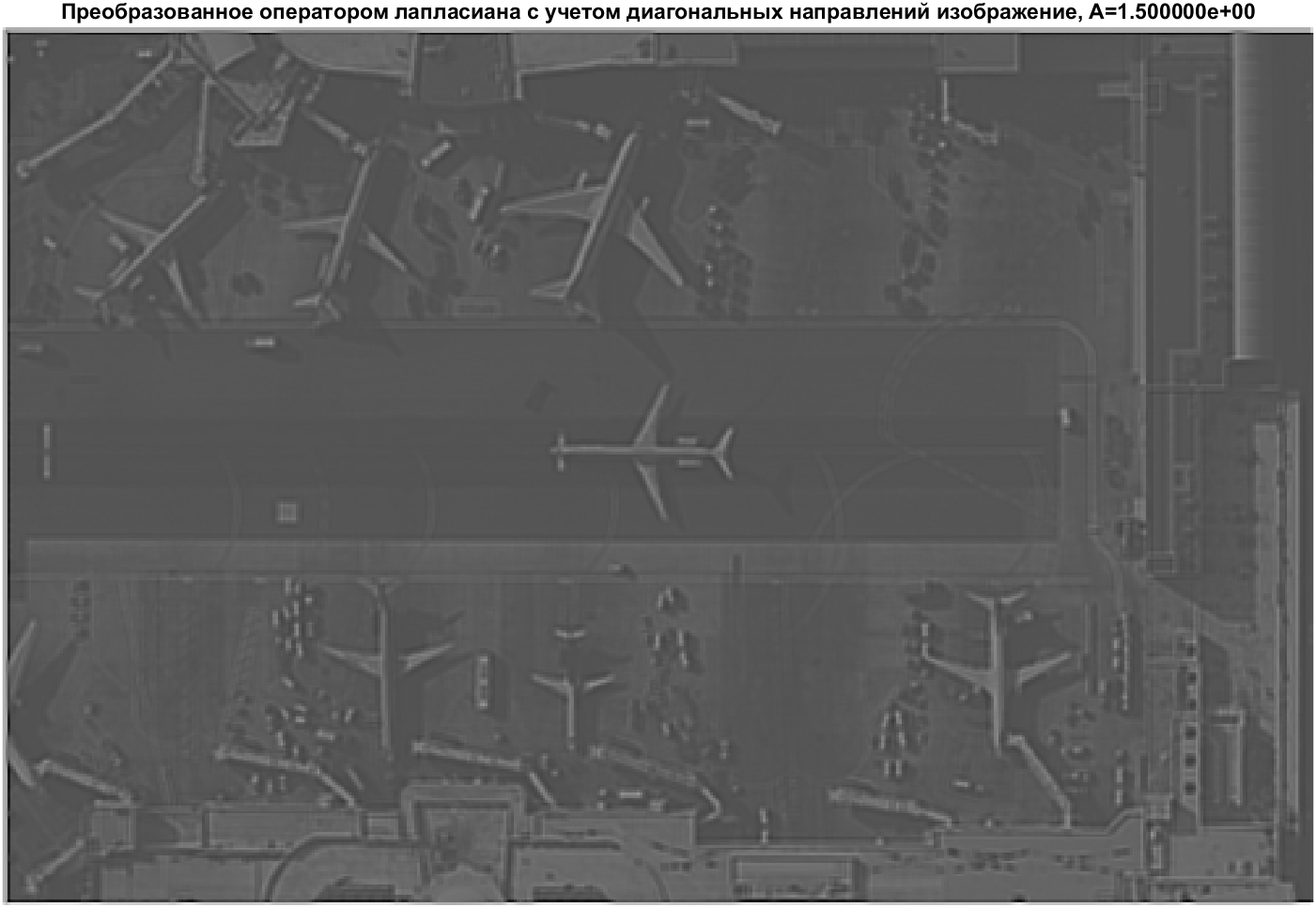












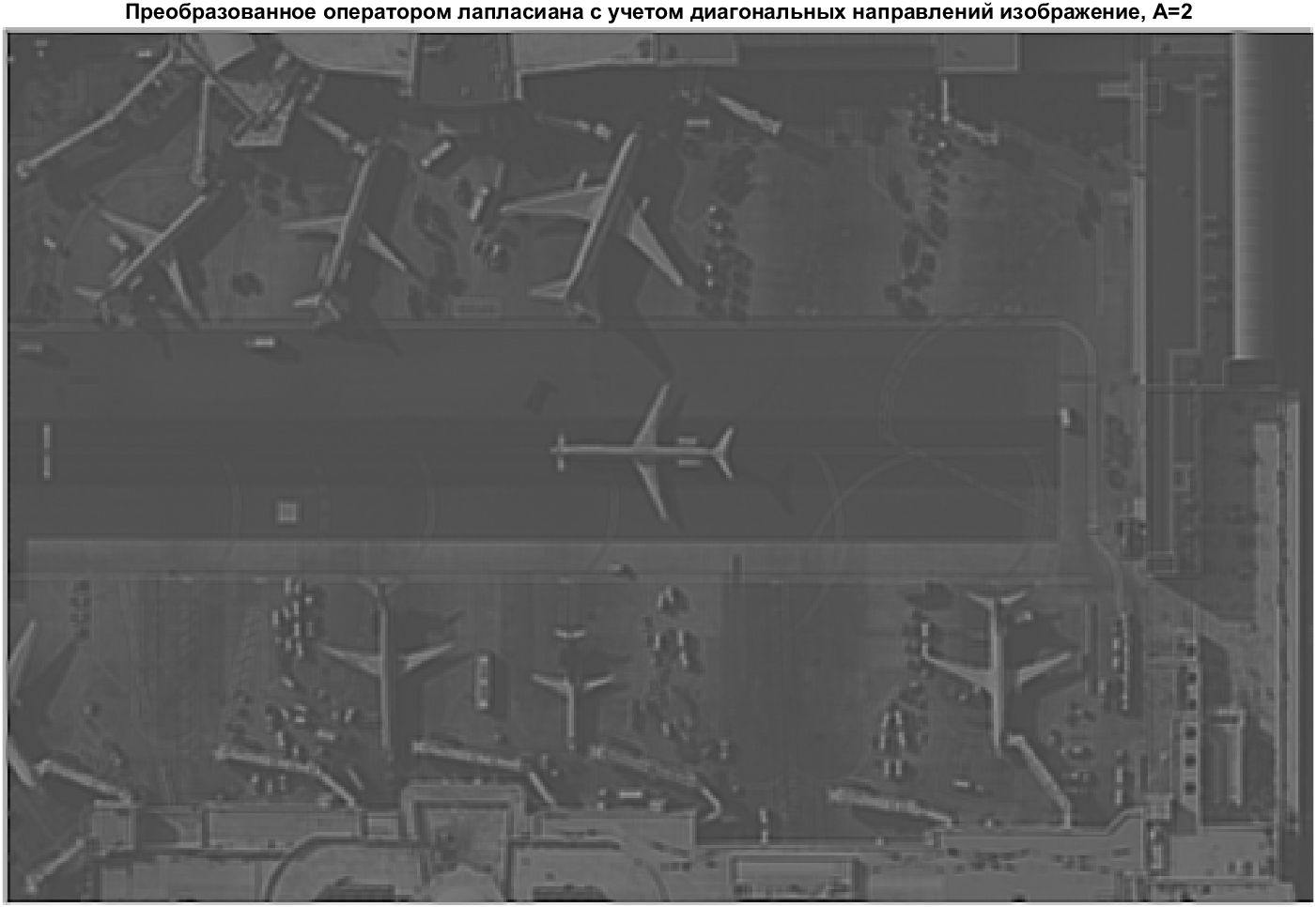


Рисунок 9 – Результат работы программы

Вывод:

В ходе выполнения лабораторной работы была разработана программа для реализации методов фильтрации изображения на основе его производных.