МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ
Национальный аэрокосмический университет им. Н.Е. Жуковского
«Харьковский авиационный институт»

Кафедра 502

ДОМАШНЕЕ ЗАДАНИЕ №4
по дисциплине «Программирование прикладных ГИС-задач »
на тему: «ПРОСТРАНСТВЕННАЯ ФИЛЬТРАЦИЯ, ОБРАБОТКА В ЧАСТОТНОЙ ОБЛАСТИ И ВОССТАНОВЛЕНИЕ ИЗОБРАЖНИЯ»

Выполнила:

Студентка 523 группы
Халаимова А.Н.

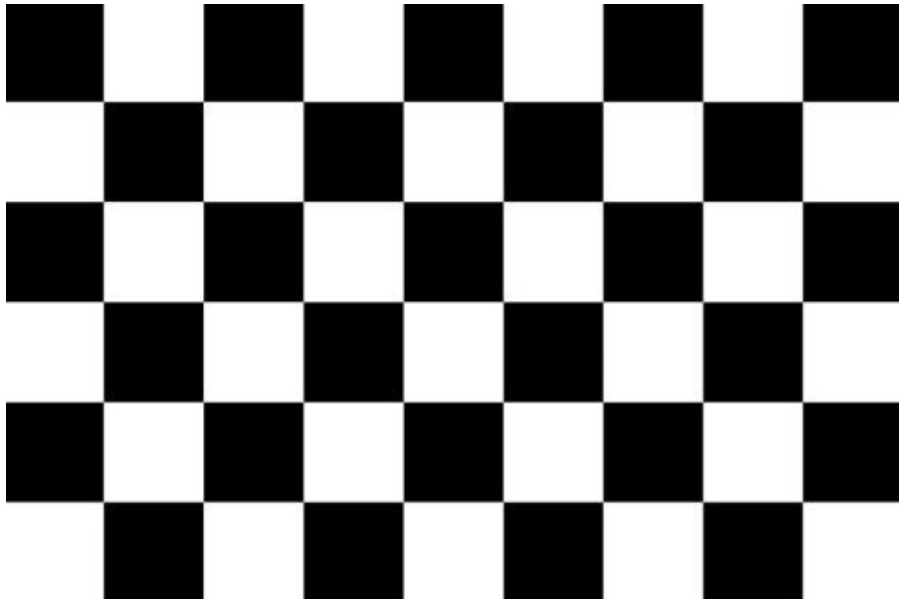
Харьков
2015

Задание:

- Выполнить над заданным (по вариантам) изображением все команды и примеры из приведенных в данном описании. Программные скрипты (из окна Command Window), M-файлы и результаты по обработке изображений (из окон Figure) поместить в отчёте, сформированном в MS Word (имя файла отчёта должно иметь вид: Фамилия-шифр группы-ЦОИ.doc).

Вычислительный сценарий:

- Для выполнения заданий мною было взято изображение с именем mlflagga_3-2.jpg. При фильтрации такого изображения будут хорошо видны изменения.



1. Линейная пространственная фильтрация

```
>>  f=imread('D:\mlflagga_3-2.jpg');
>> imshow(f)
```

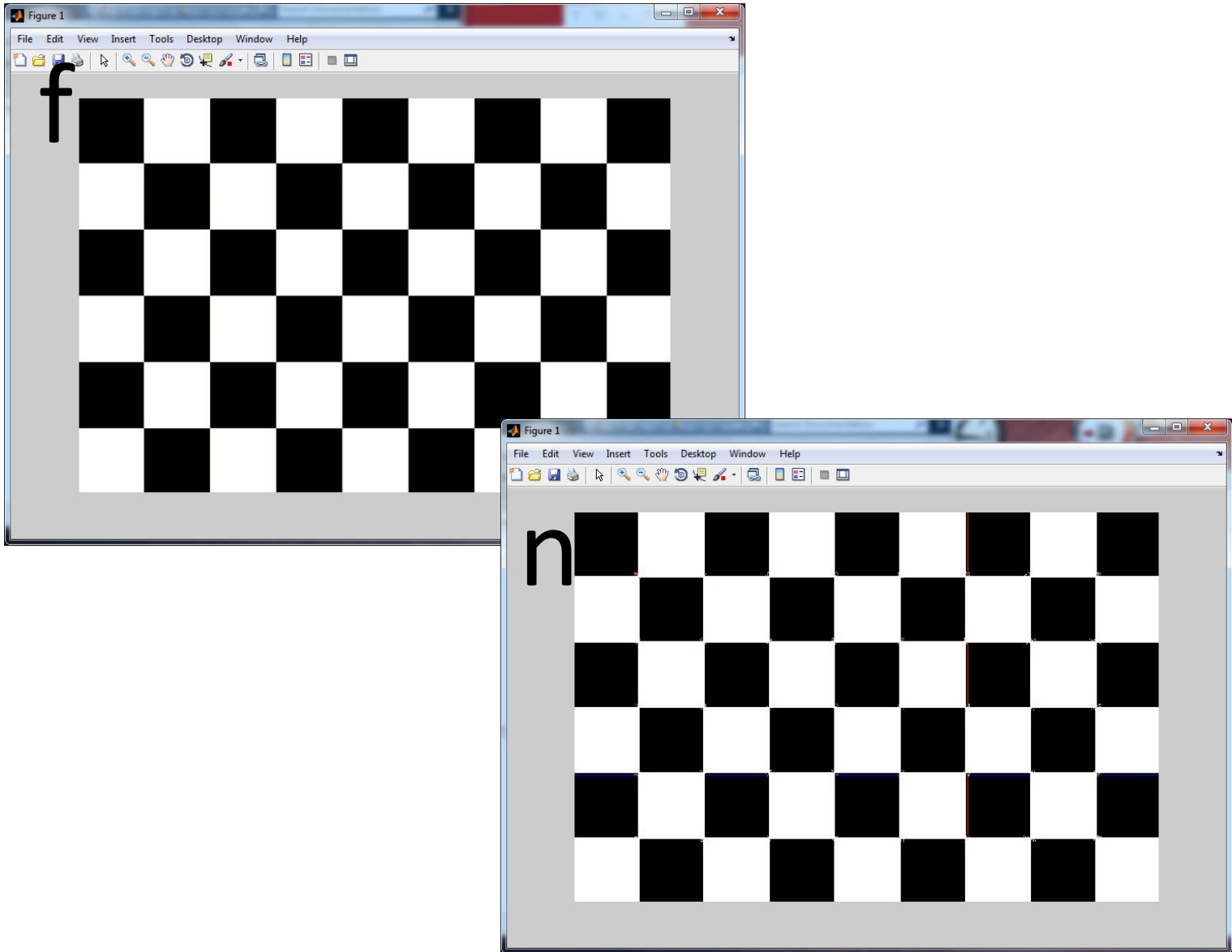
Определяем класс изображения f :

```
>>  class(f)
ans =
uint8
```

Изменяем класс изображения f на класс double:

```
>>n=double(f);
>>class(n)
ans =
Double
```

- Фильтрующая маска:
- w=ones(5,5)/25;



Применение imfilter с нулевым проложением:

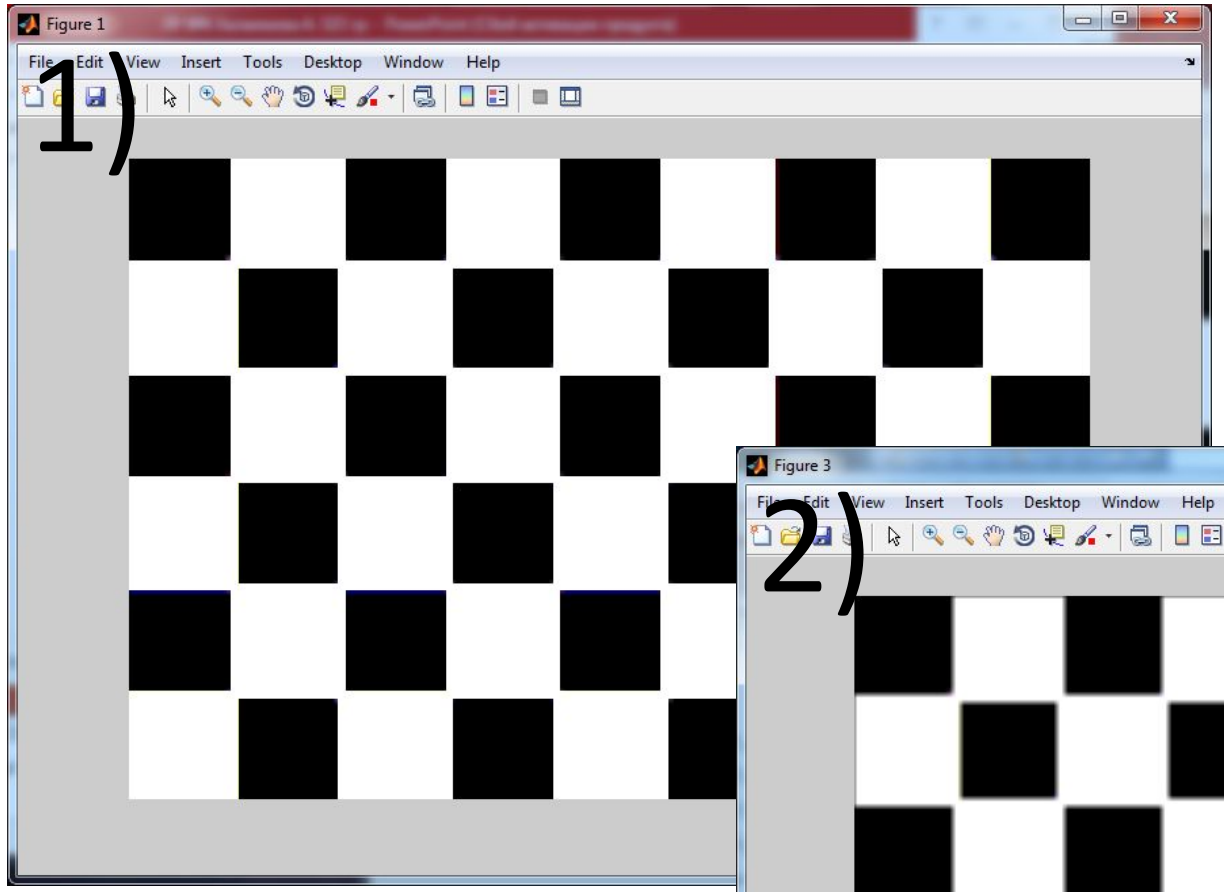
1)gd=imfilter(n, w);

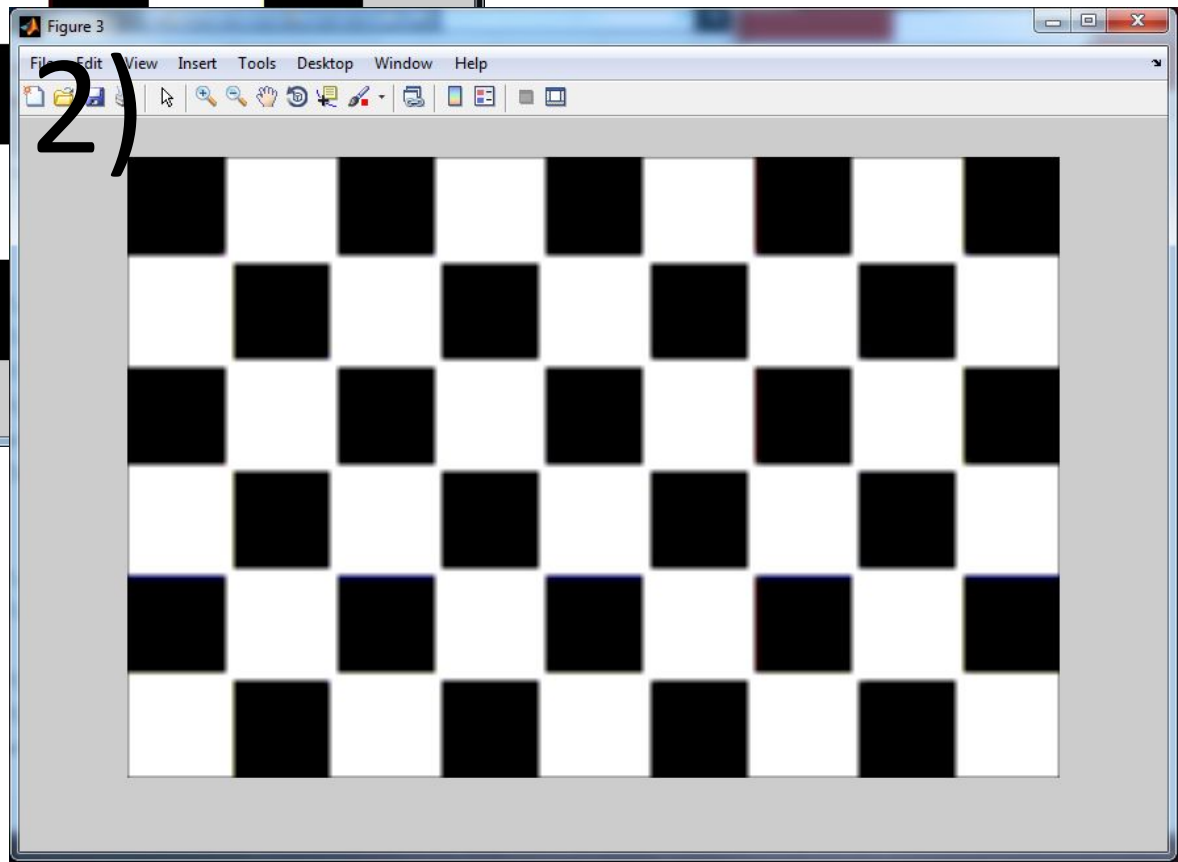
figure, imshow(gd,[])

2)gd8=im2uint8(gd);

g8d=imfilter(gd8,w);

figure,imshow(g8d,[])

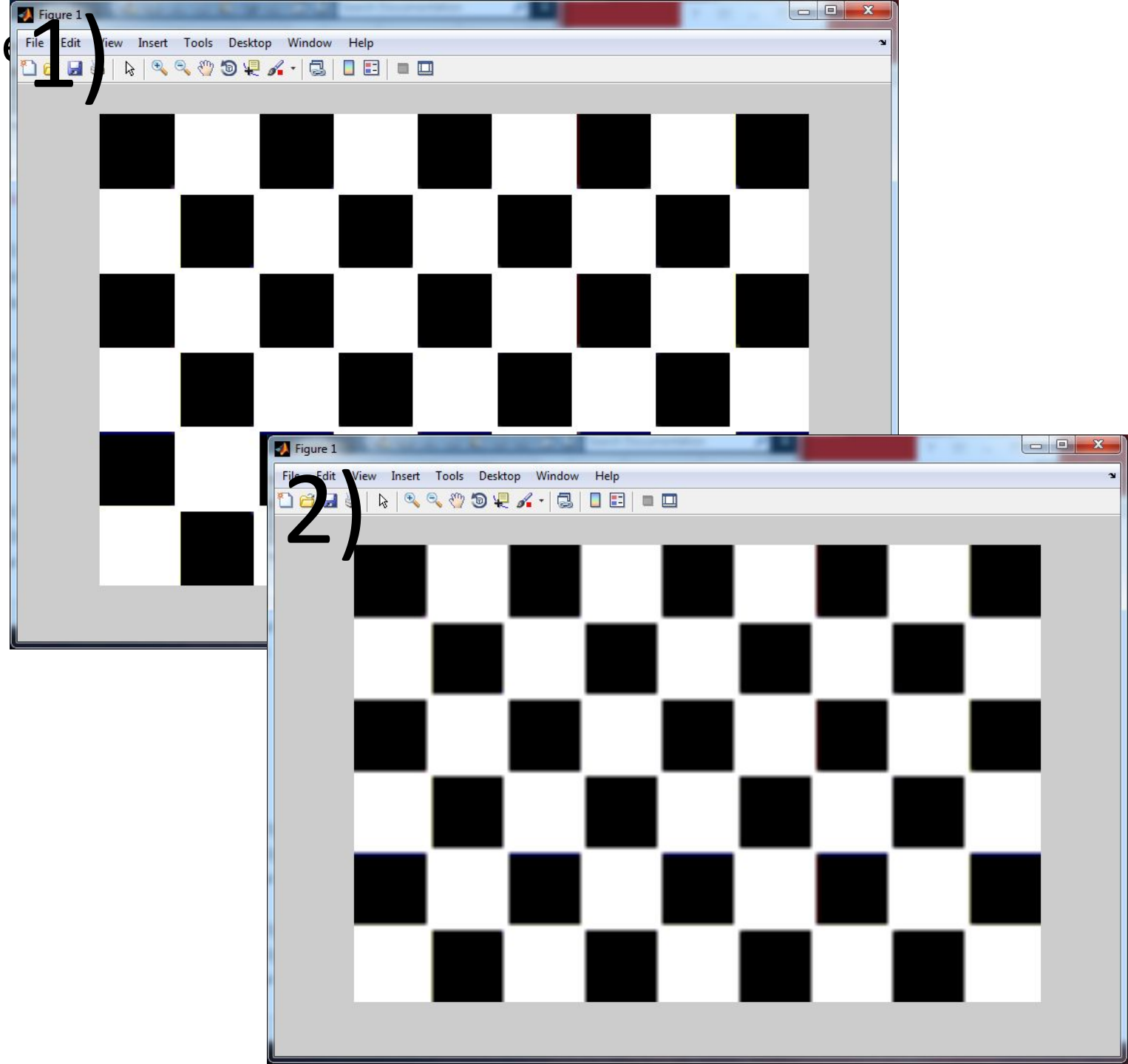




Использование опции 'replicate

1) gr=imfilter(n, w,'replicate');
figure,imshow(gr,[])

2) gr8=im2uint8(gr);
g8r=imfilter(gr8,w,'replicate');
figure,imshow(g8r,[])



Использование опции 'symmetric'

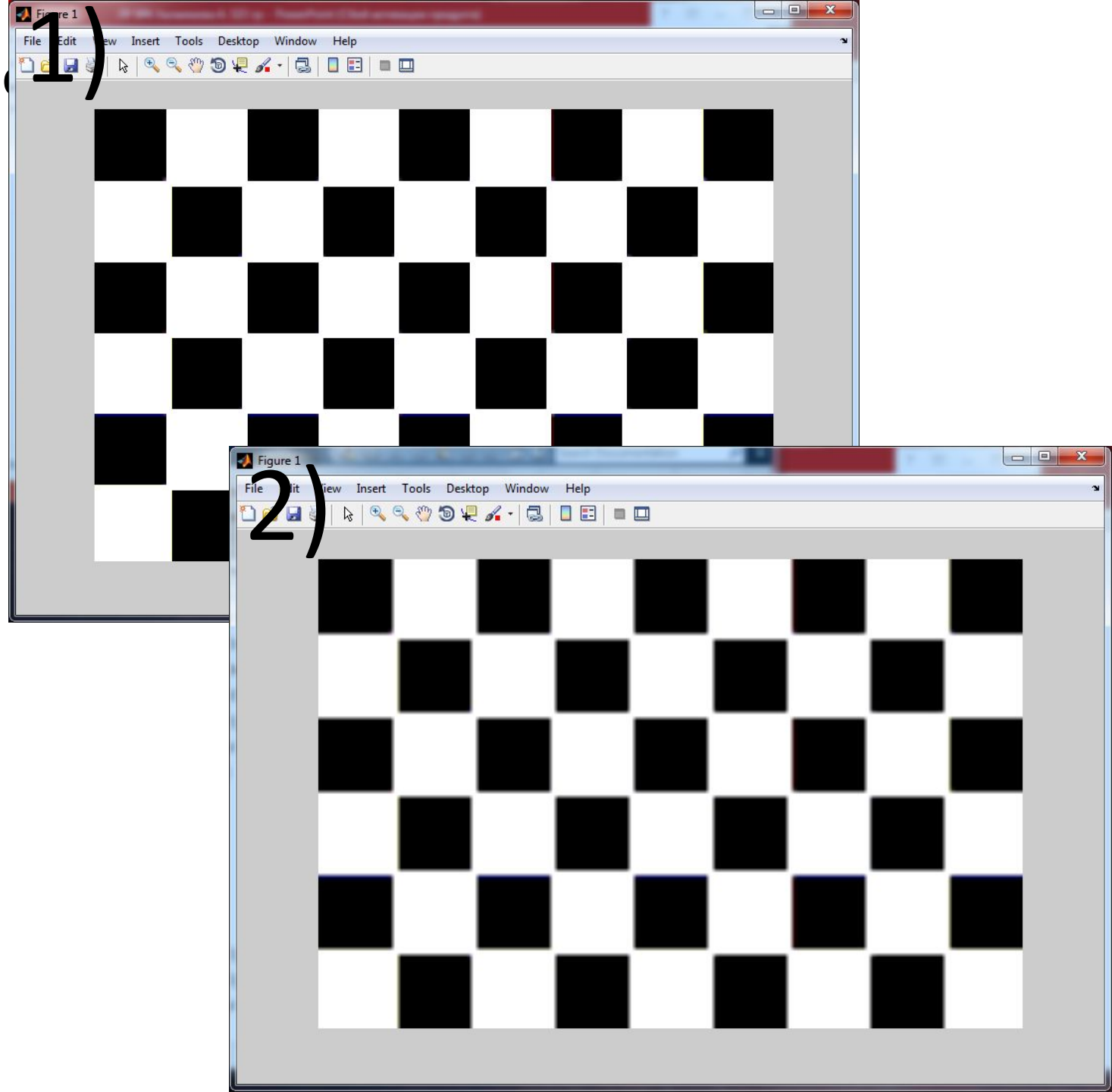
1) gs= imfilter(f,w,'symmetric');

figure,imshow(gs,[])

2) gs8=im2uint8(gs);

g8s=imfilter(gs8,w,'symmetric');

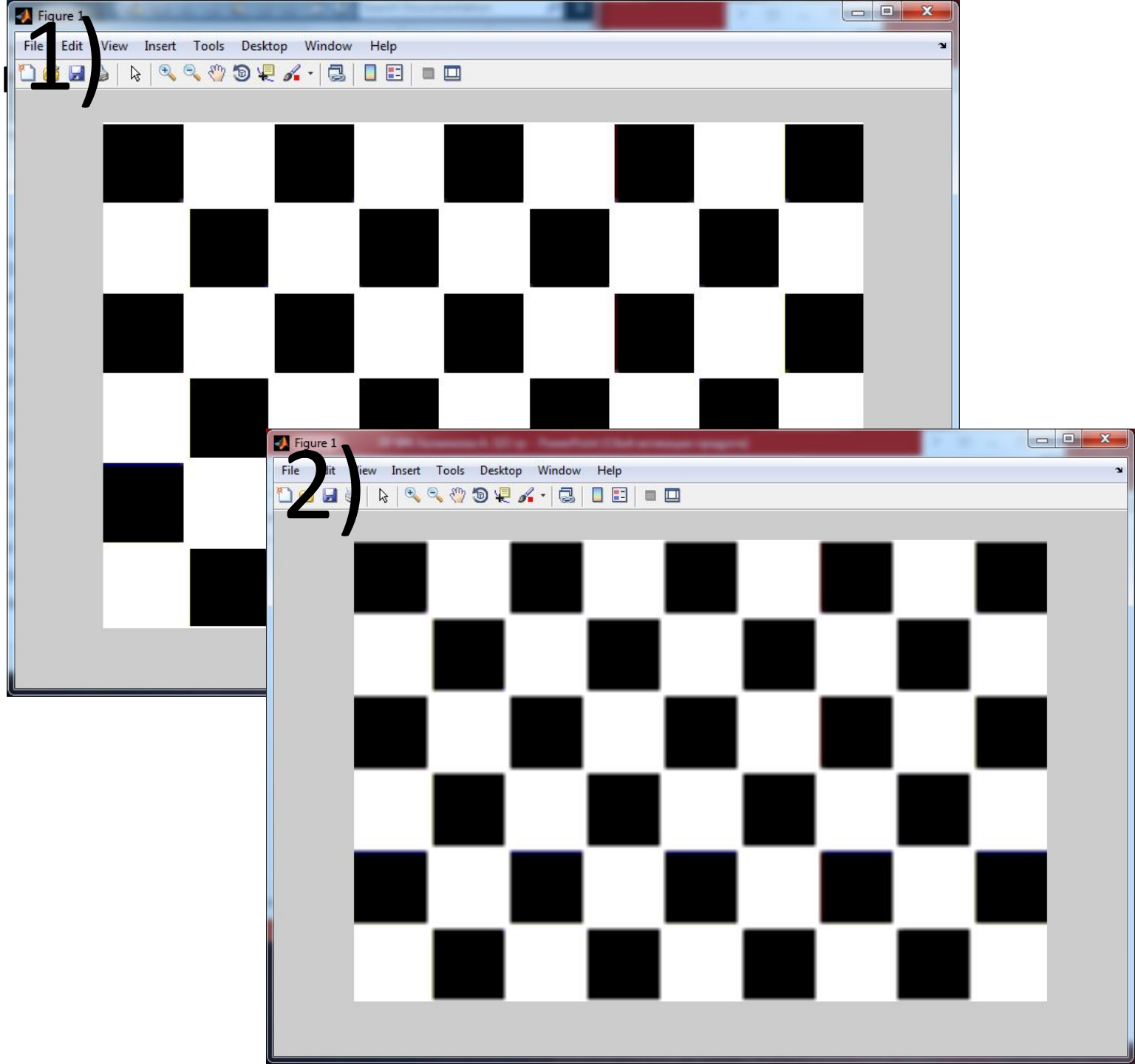
figure,imshow(g8s,[])



Использование опции 'circular'

1) gc= imfilter(n,w,'circular');
figure,imshow(gc,[])

2) gc8=im2uint8(gc);
g8c=imfilter(gc8,w,'circular');
figure,imshow(g8c,[])



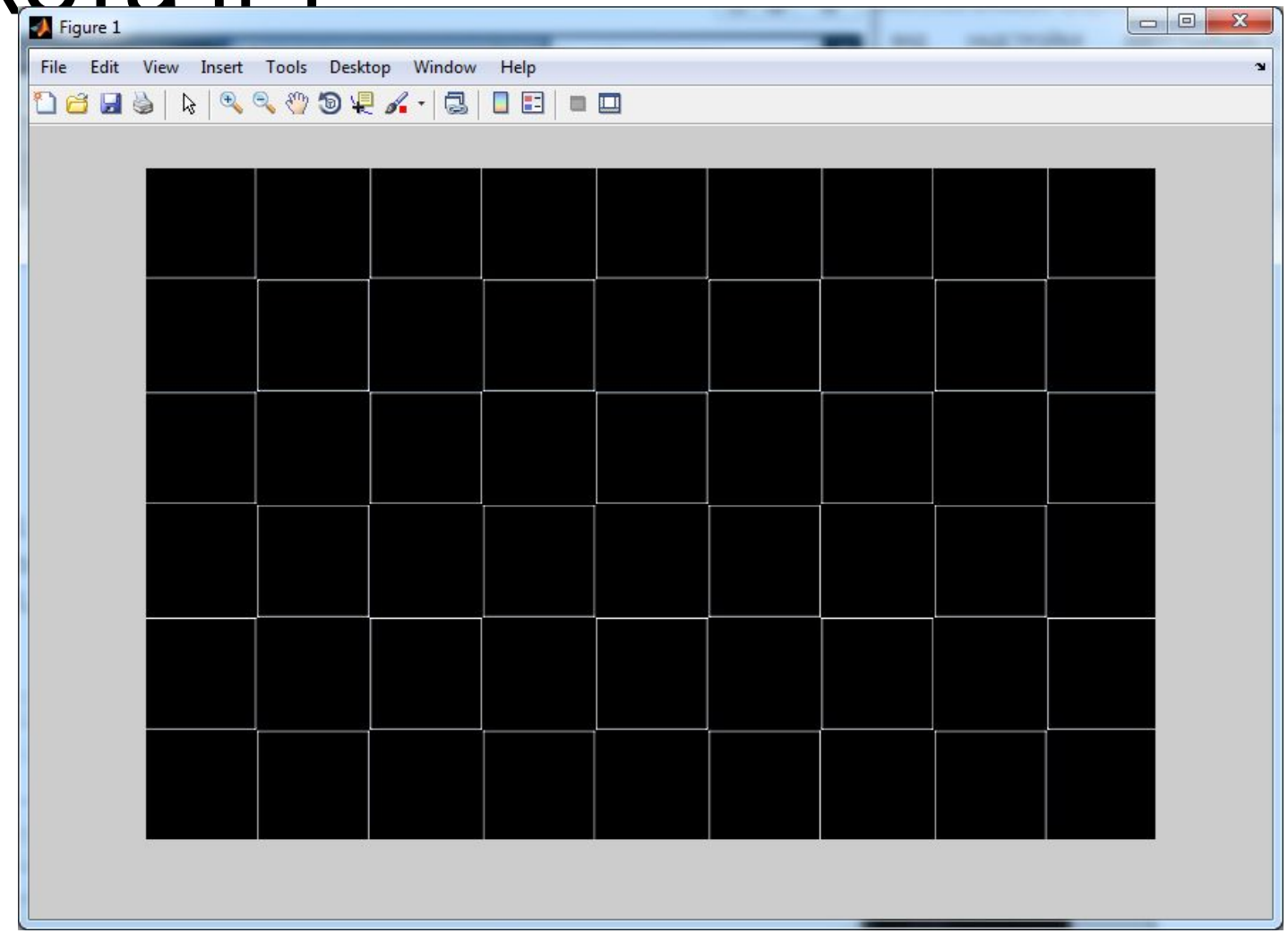
2. Стандартные пространственные фильтры из пакета IPT

1)Фильтр Лапласа:

- >> w=fspecial('laplacian',0)

w =

- 0 1 0
- 1 -4 1
- 0 1 0
- >> w=[0 1 0;1 -4 1;0 1 0];
- g1=imfilter(f,w,'replicate');
- imshow(g1,[])



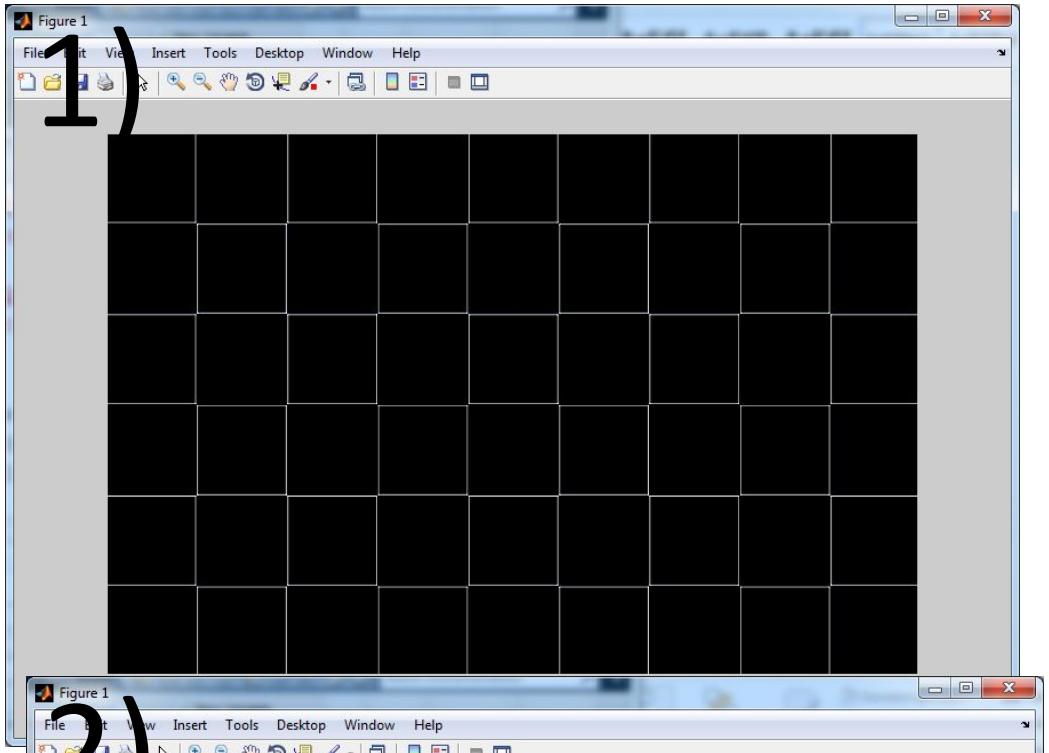
1) f2=im2double(f);

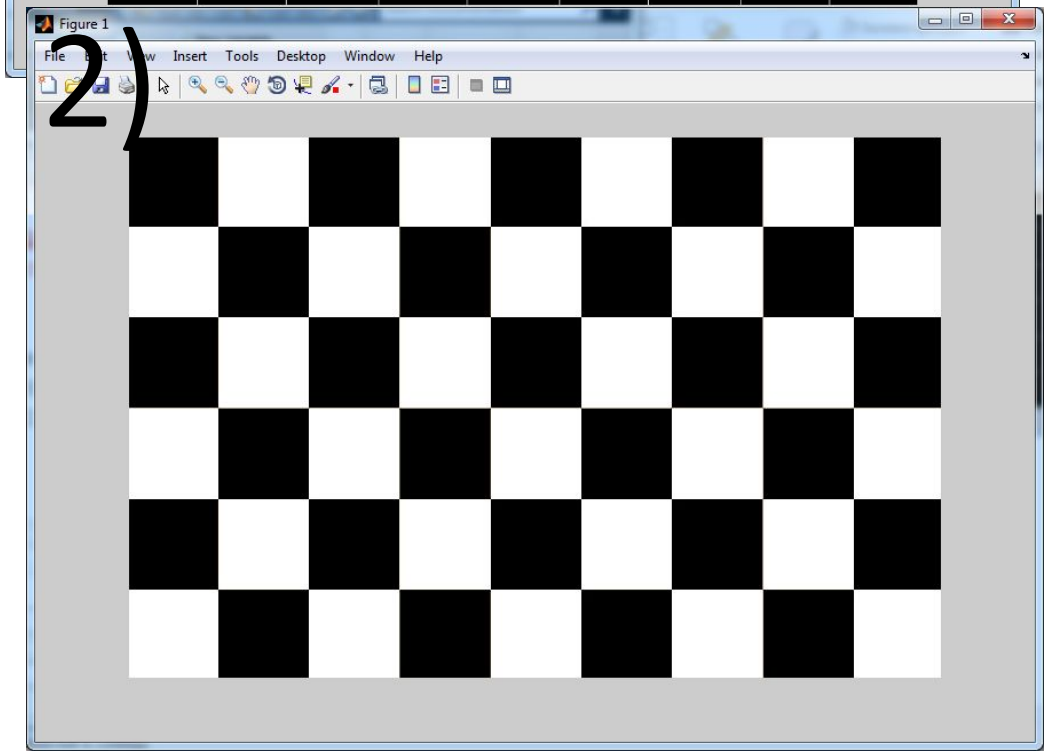
- >> g2=imfilter(f2,w,'replicate');
- >> imshow(g2,[])

2) f2=im2double(f);

>> g2=imfilter(f2,w,'replicate');

>> imshow(g2,[])

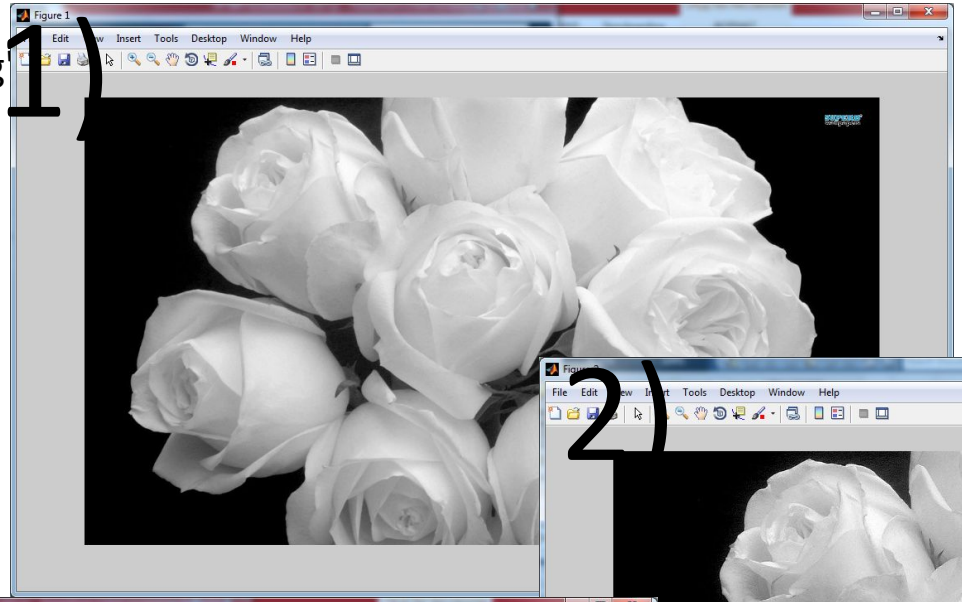




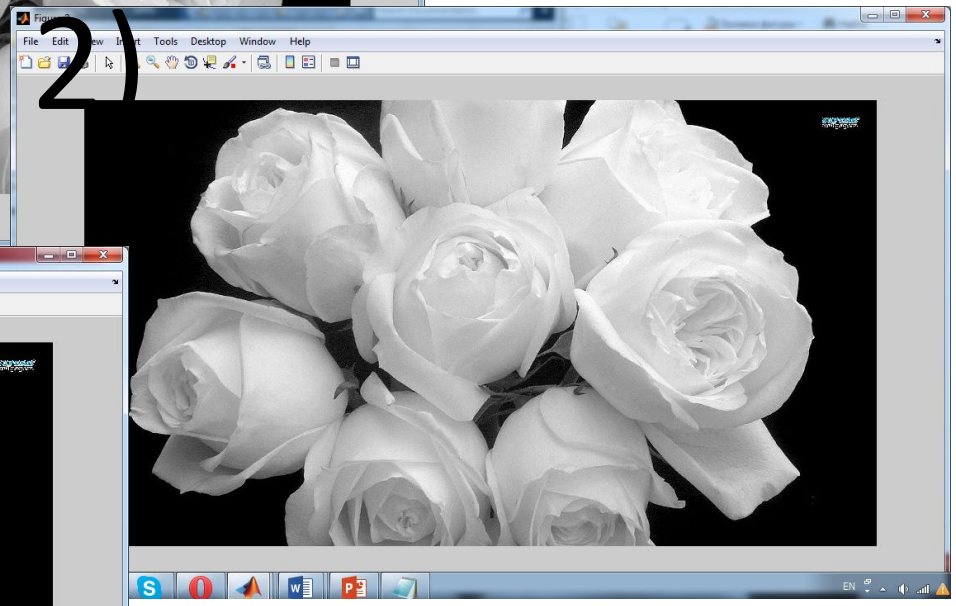
Подбор параметров фильтров и сравнение разных техник улучшения изображения:

- >> f=imread('D:\rose-white-bouquet-flower-227196.jpg');
- >> w4=fspecial('laplacian',0);
- >> w8=[1 1 1 ; 1 -8 1;1 1 1];
- >> f= im2double(f);
- >> g4=f-imfilter(f,w4, 'replicate');
- >> g8=f-imfilter(f,w8, 'replicate');
- >> imshow(f)
- >> figure;imshow(g4)
- >> figure;imshow(g8)

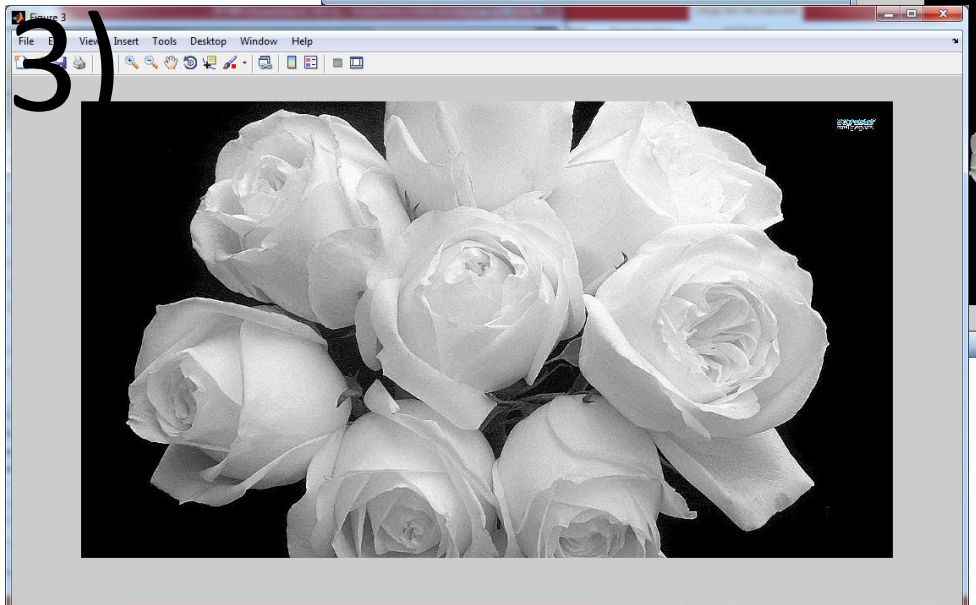
1)



2)



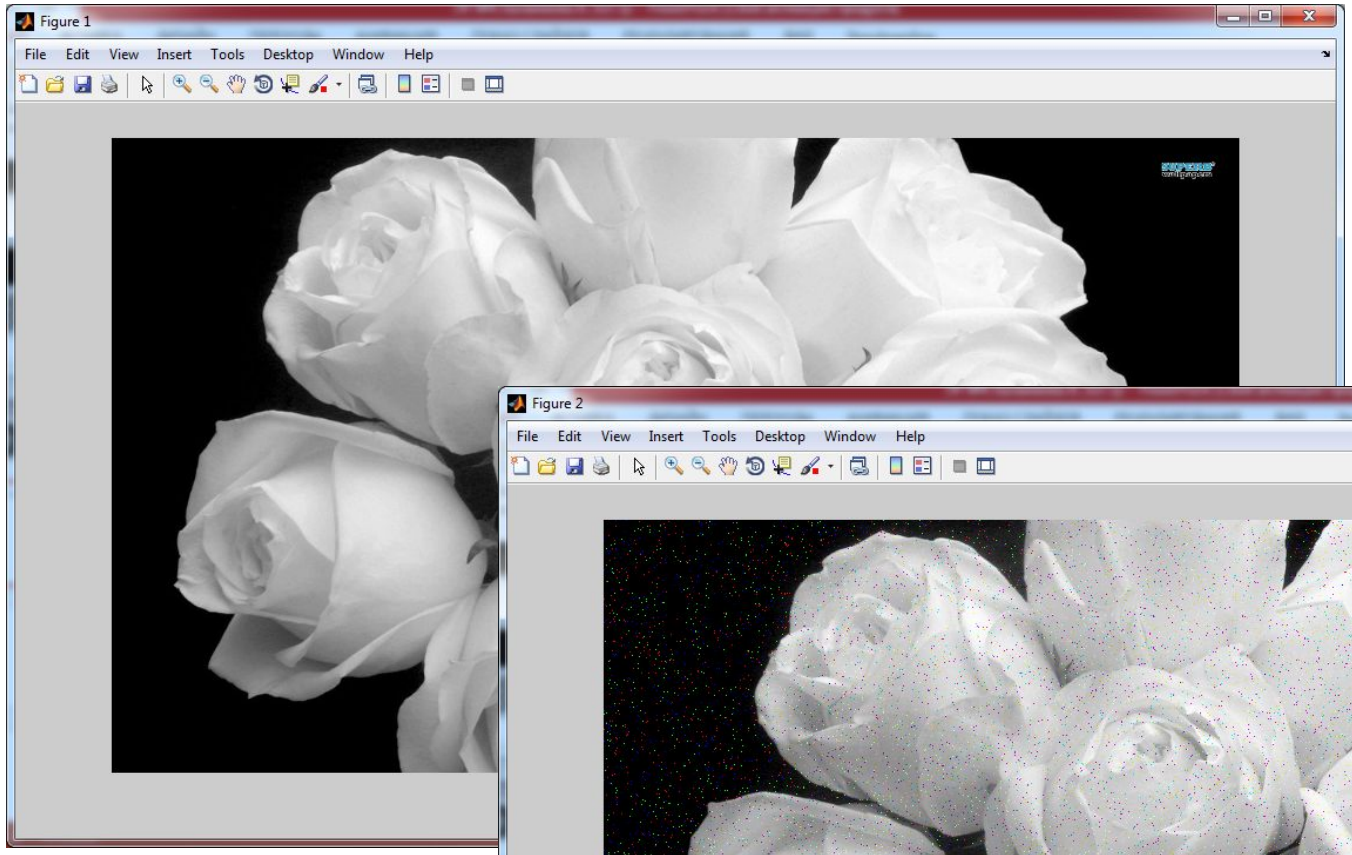
3)

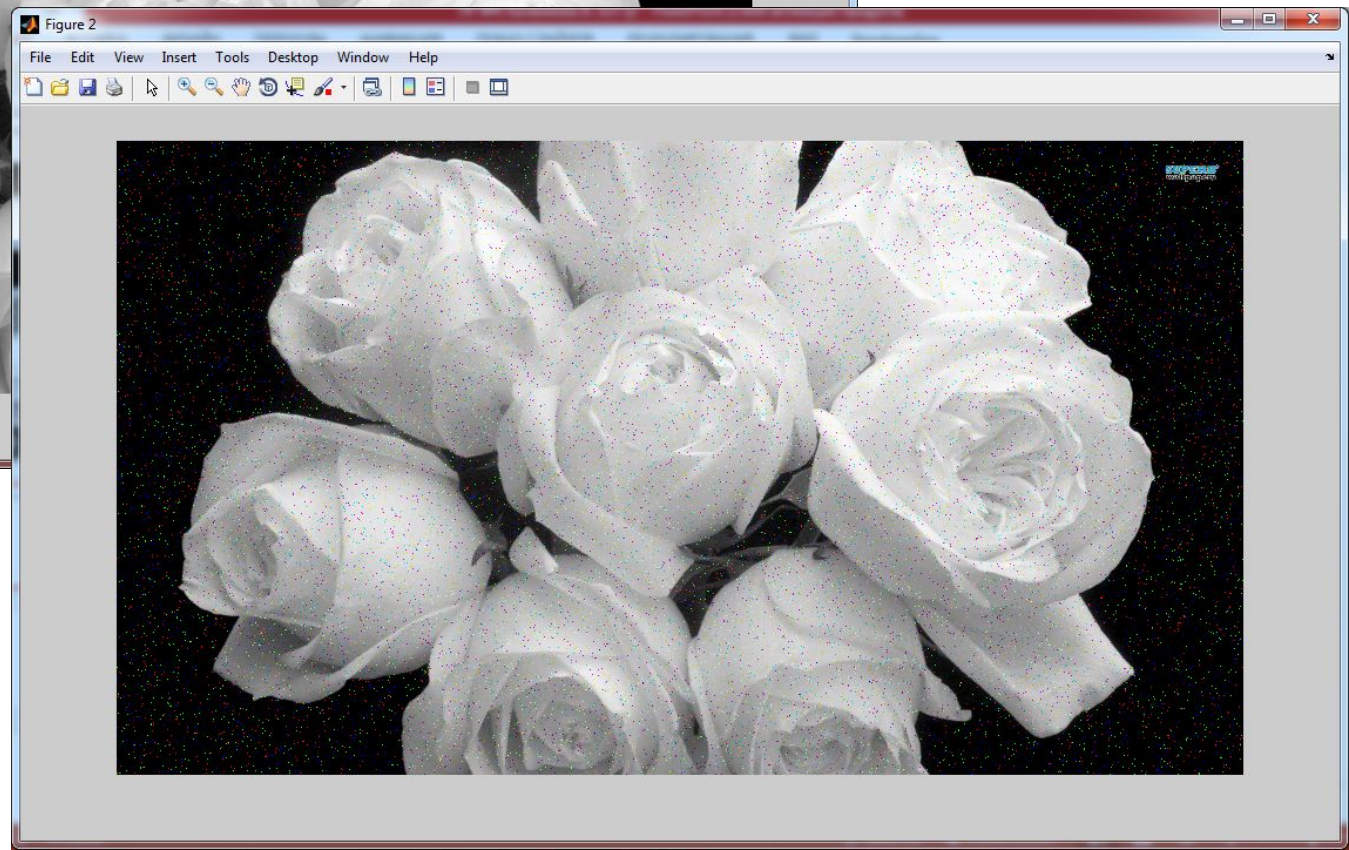


3. Нелинейный пространственный фильтр

Шум типа «соль и перец»:

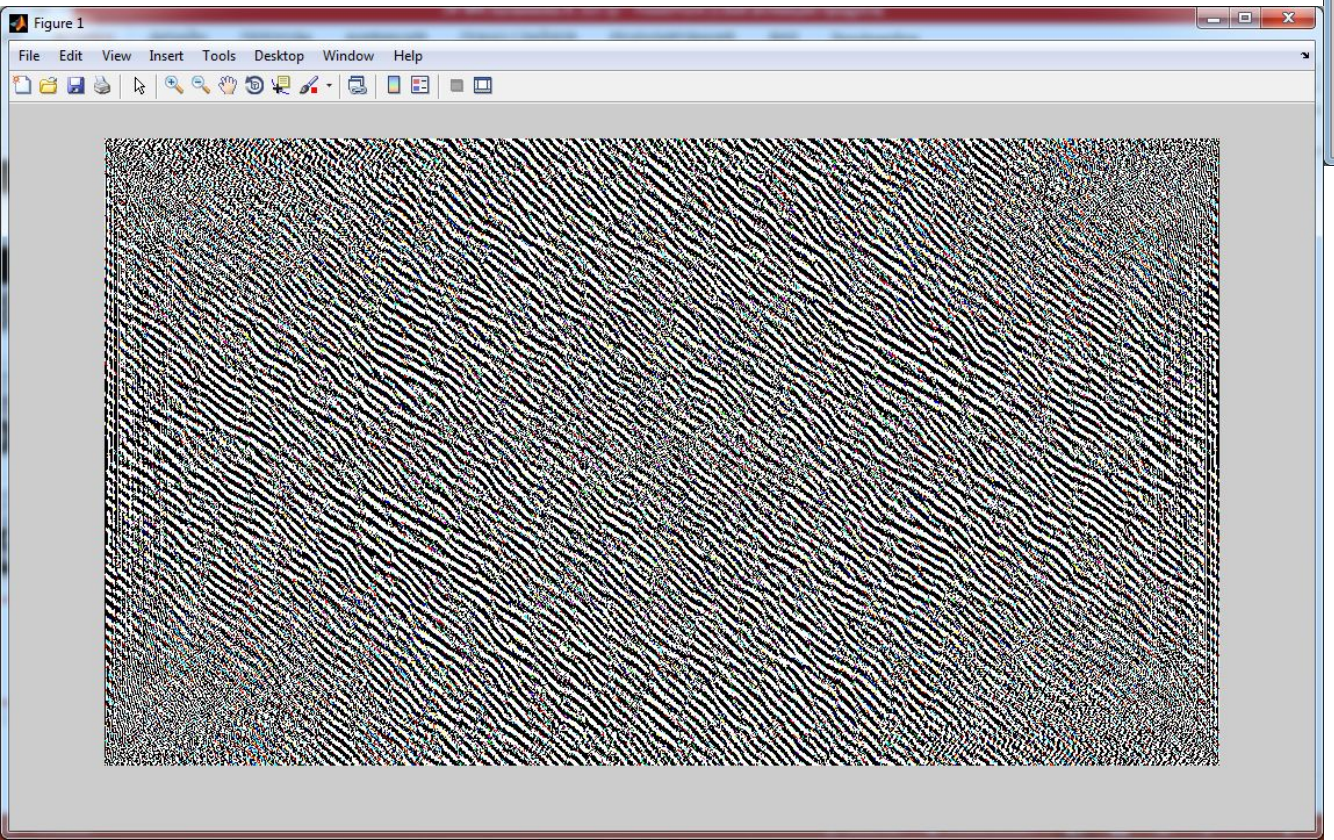
```
f1=imnoise(f,'salt & pepper',0.02);
imshow(f)
figure,imshow(f1)
```

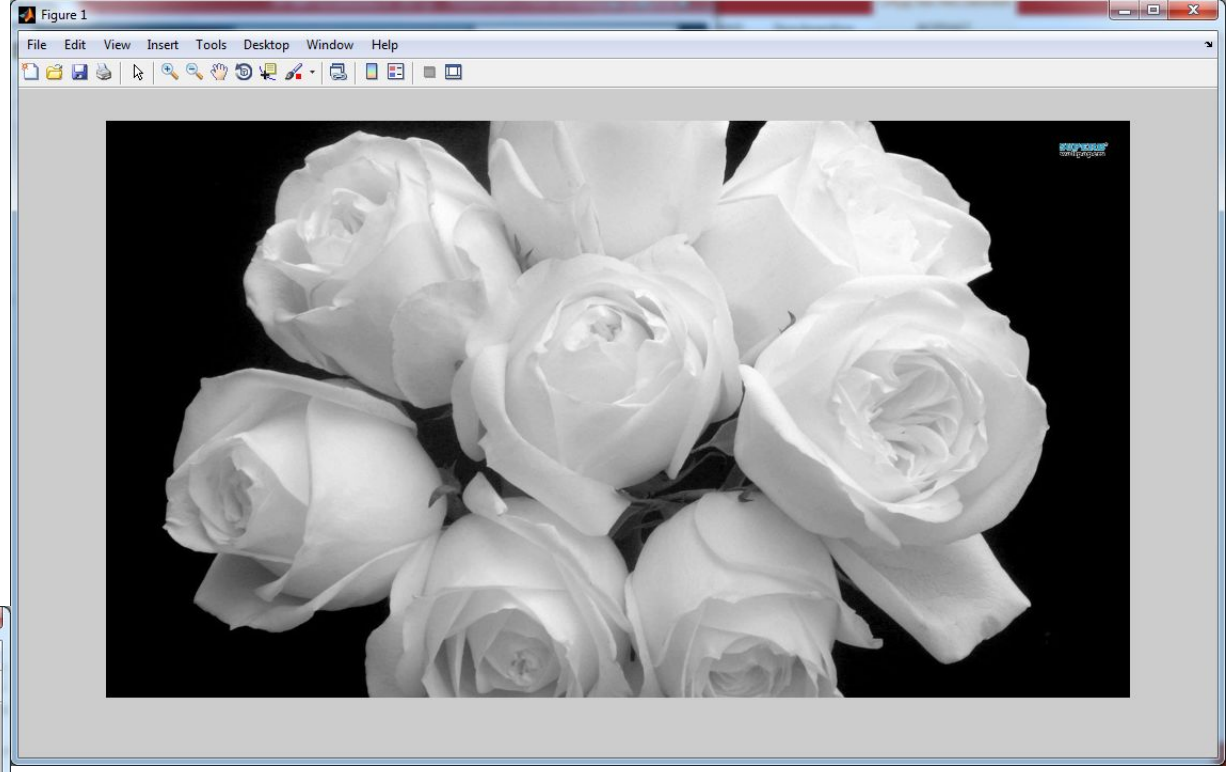




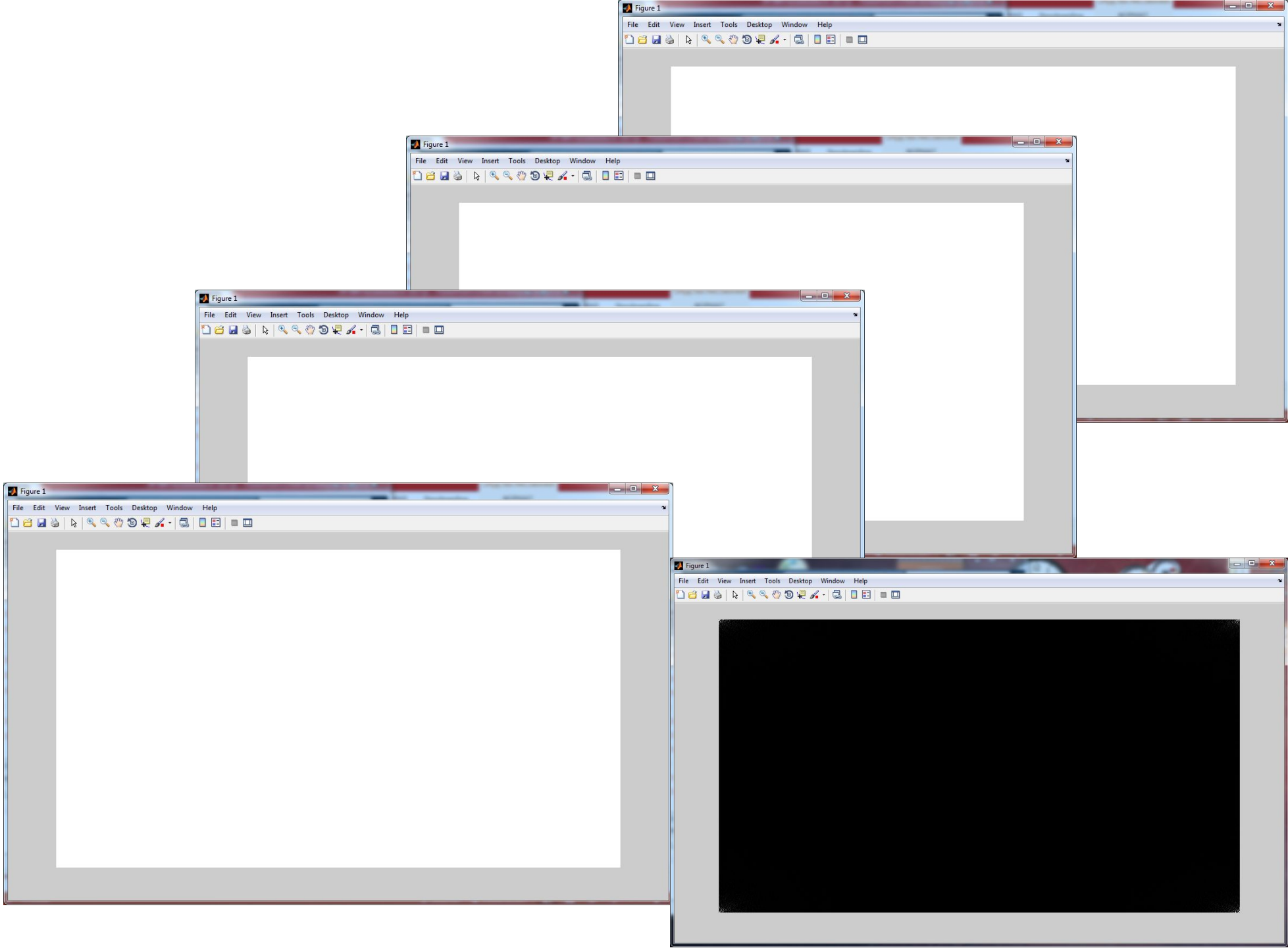
4.Фильтр Фурье

- >> f=imread('D:\rose-white-bouquet-flower-227196.jpg');
- >> F=fft2(f);
- >> imshow(F)





- >> S=abs(F);
- >> imshow(S,[]);
- >> Fc=fftshift(F);
- >> imshow(abs(Fc),[]);
- >> V=abs(Fc);
- >> imshow(V,[]);
- >> S2=log(1+abs(Fc));
- >> imshow(S2,[]);
- >> F=ifftshift(Fc);
- >> f=ifft2(F);
- >> f=real(ifft2(f));
- >> imshow (f)



5.Фильтрация в частотной области

Создаем M-функцию с именем paddedsize :

- function PQ = paddedsize(AB, CD, PARAM)
- if nargin==1
- PQ=2*AB;
- elseif nargin == 2&~ischar(CD)
- PQ = AB+CD ;
- PQ= 2 *ceil (PQ/2);
- elseif nargin == 2
- m= max (AB);
- P=2^nextpow2(2*m);
- PQ=[P, P];

Создаем M-функцию с именем dftfilt :

- function g =dftfilt(f, H)
- F=fft2(f, size(H,1), size(H,2));
- Gi=H.*F;
- g=real(ifft2(Gi));
- g=g(1:size(f,1),1:size(f,2));
- end

Создаем M-функцию с именем gscale :

```
function g = gscale( f, varargin )
if length(varargin)==0
    method='full8';
else
    method = varargin{1};
end
if strcmp(class(f), 'double')&(max(f(:))>1| min (f(:))<0)
    f=mat2gray(f);
end
switch method
    case 'full8'
      g=im2uint8(mat2gray(double (f)));
    case 'full16'
      g=im2uint16(mat2gray(double (f)));
    case 'minmax'
      low=varargin{2}; high=varargin{3};
      if low>1 | low<0 | high>1 | high<0
         error('Параметры low и high должны быть изменены')
```

Создаем M-функцию с именем dftuv :

- function [U , V]= dftuv(M, N)
- u=0:(M);
- v=0:(N);
- idx =find(u>M/2);
- u(idx) = u(idx);
- idy=find(v>N/2);
- v(idy) = v(idy) ;
- [V, U]=meshgrid(v, u);
- end

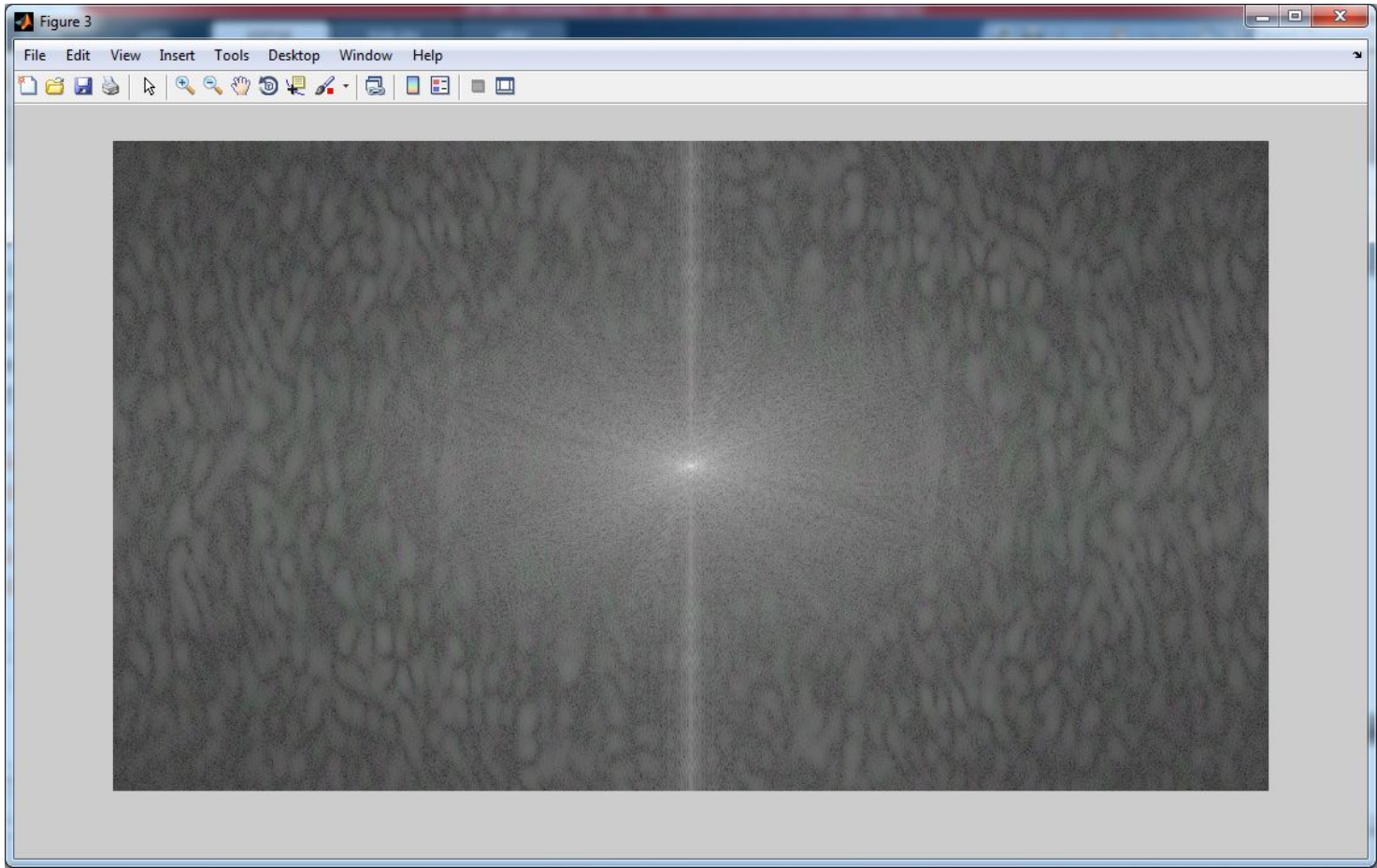
Создаем M-функцию с именем lpfilter :

```
function [H, D] = lpfilter( type, M, N, D0, n )
[U, V]=dftuv(M, N);
D=sqrt(U.^2+V.^2);
switch type
    case 'ideal'
        H=double (D<=D0);
    case 'btw'
        if nargin==4
            n=1;
        end
        H=1./(1+(D/D0).^(2*n));
    case 'gaussian'
        H= exp ((D.^2)./(2*(D0^2)));
    otherwise
        error ('Неизвестный тип фильтра');
    end
end
```

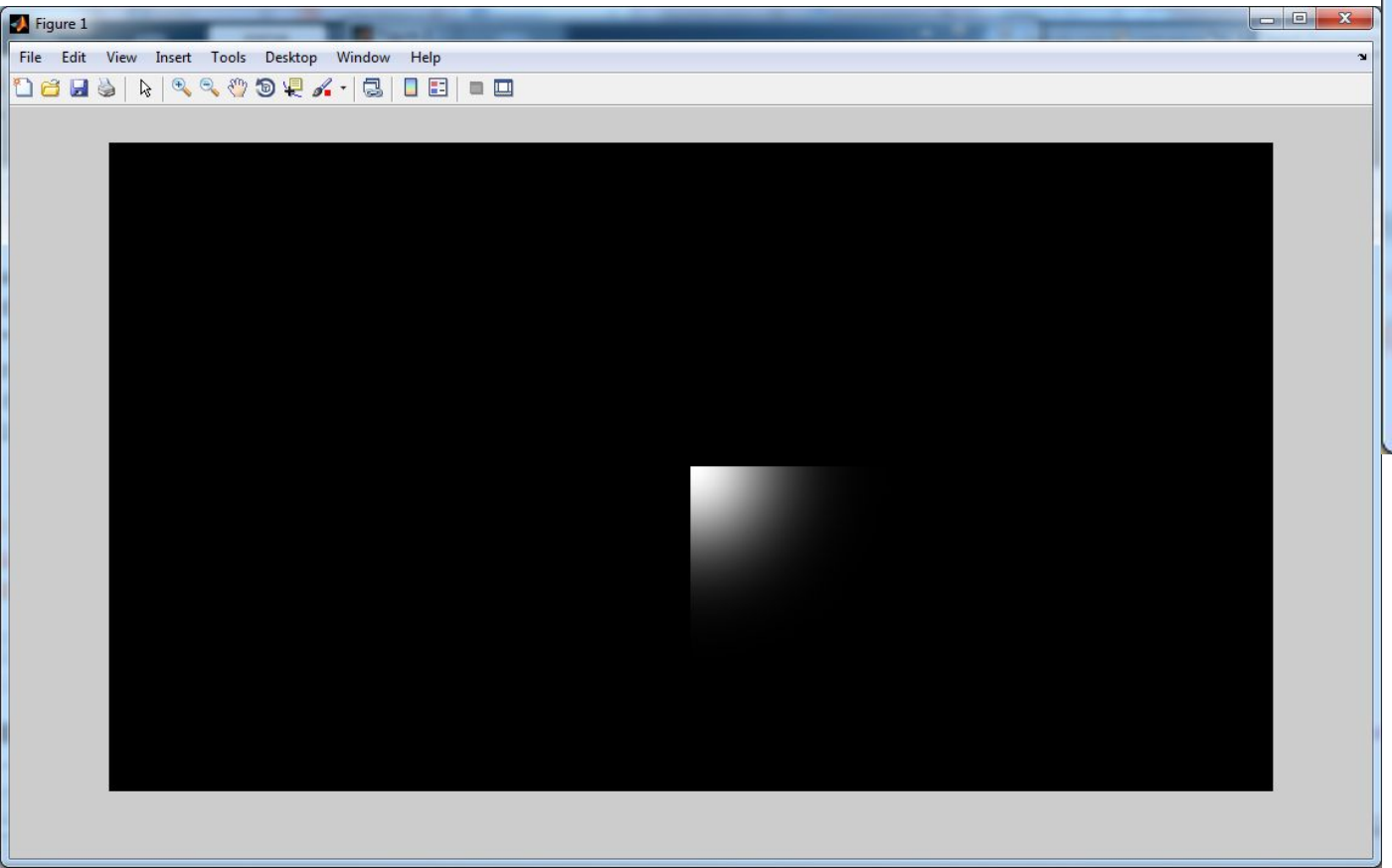
Создаем M-функцию с именем hpfilter :

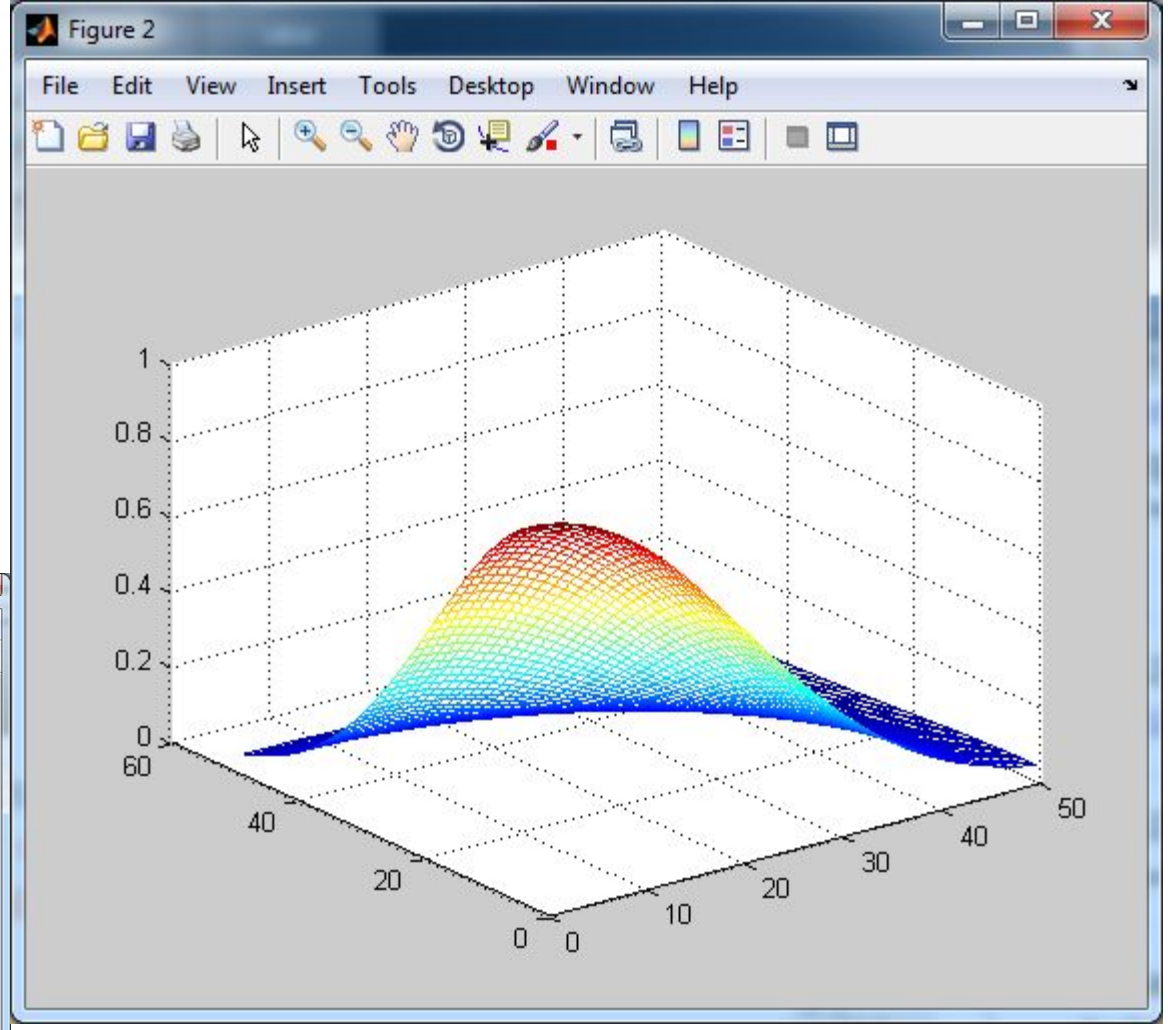
```
function H = hpfilter( type, M, N, D0, n )
   [U, V]=dftuv(M, N);
D=sqrt(U.^2+V.^2);
switch type
    case 'ideal'
        H=double (D<=D0);
    case 'btw'
        if nargin==4
            n=1;
        end
        H=1-(1./(1+(D/D0).^(2*n)));
    case 'gaussian'
        H=1- (exp ((D.^2)./(2*(D0^2))));
    otherwise
        error ('Неизвестный тип фильтра');
    end
end
```

- f=imread('D:\rose-white-bouquet-flower-227196.jpg');
- red = f(:,:, 1);
- F=fft2(f);
- S=fftshift(log(1+abs(F)));
- S=gscale(S);
- imshow(f), figure, imshow (S);

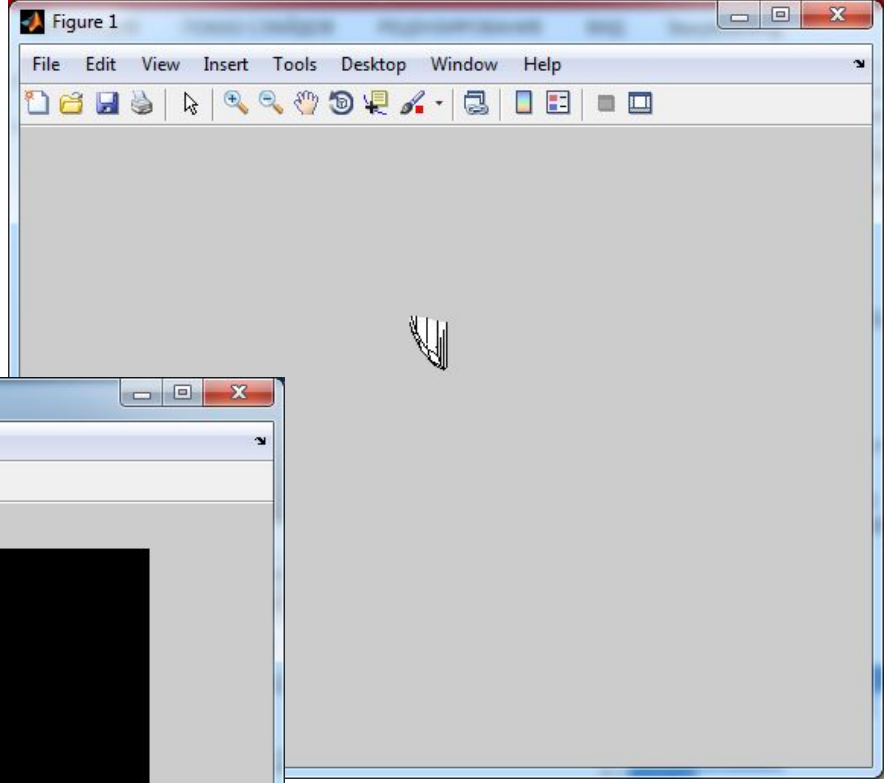


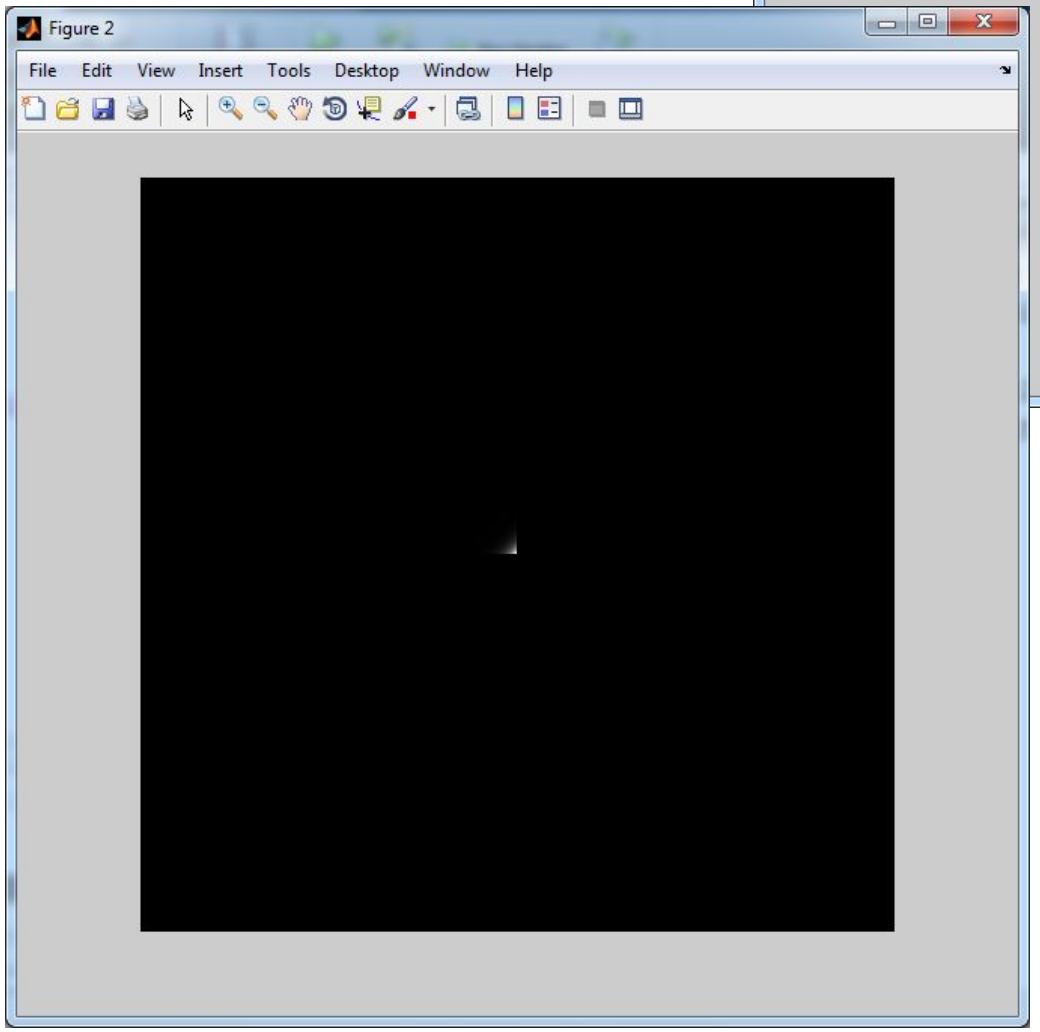
- PQ=paddedsize(size(red));
- [U, V]=dftuv(PQ(1), PQ(2));
- D0=0.05*PQ(2);
- F=fft2(red, PQ(1), PQ(2));
- H=exp(-(U.^2+V.^2)/(2*(D0^2)));
- g=dftfilt(red, H);
- figure, imshow(fftshift(H), []);
- figure, mesh(H(1:10:500, 1:10:500))



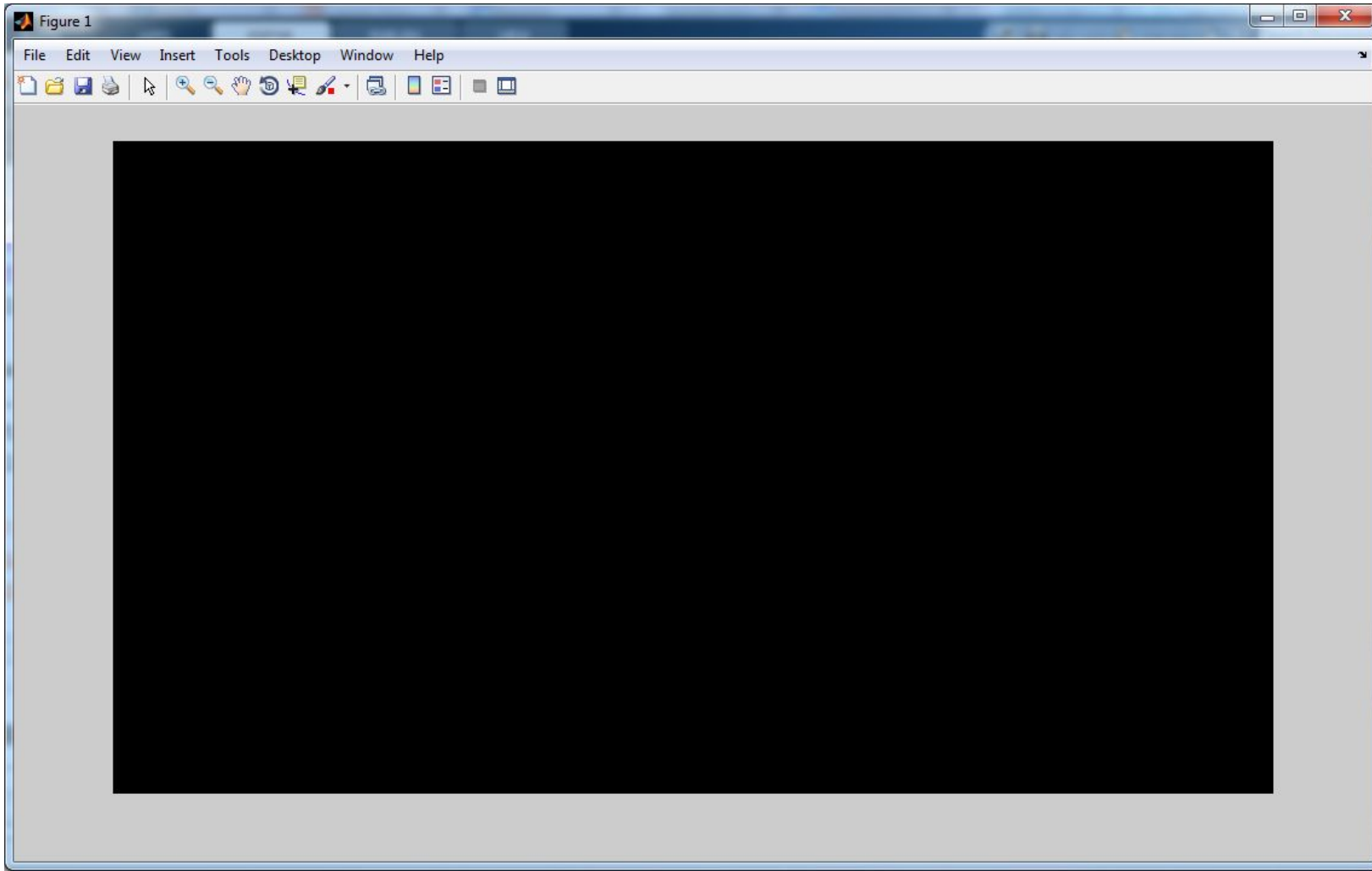


- axis([0 50 0 50 0 1])
- colormap(gray)
- grid off
- axis off
- view (-25, 0)
- H=fftshift (lpfilter ('gaussian', 500, 500, 50));
- mesh(H(1:10:500, 1:10:500))
- axis([0 50 0 50 0 1])
- colormap([0 0 0])
- grid off
- axis off
- view (-163, 64)
- figure, imshow (H, []);

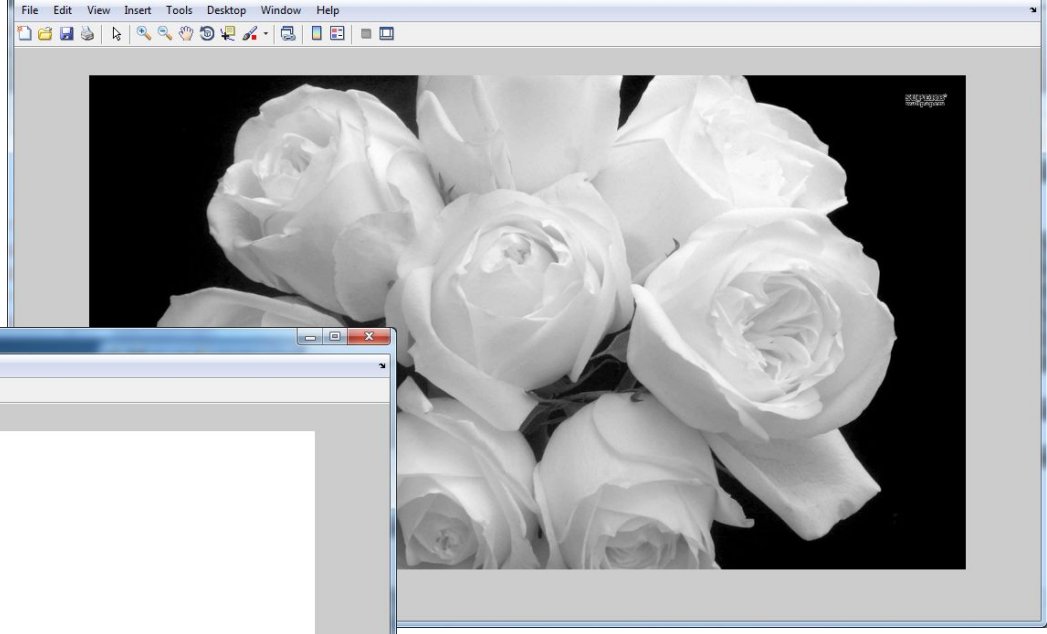


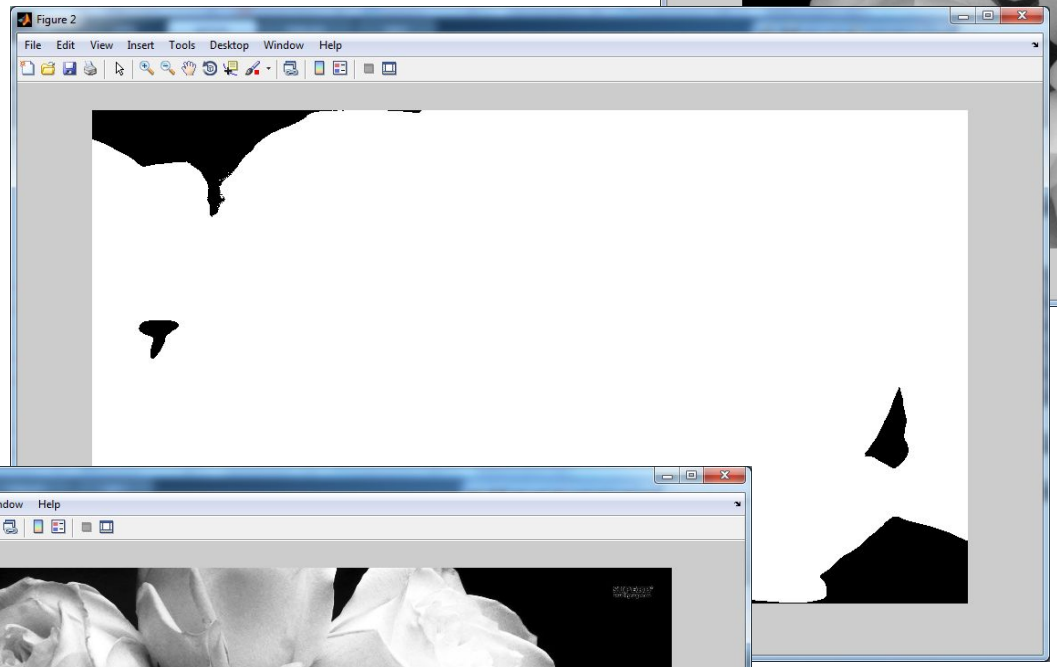


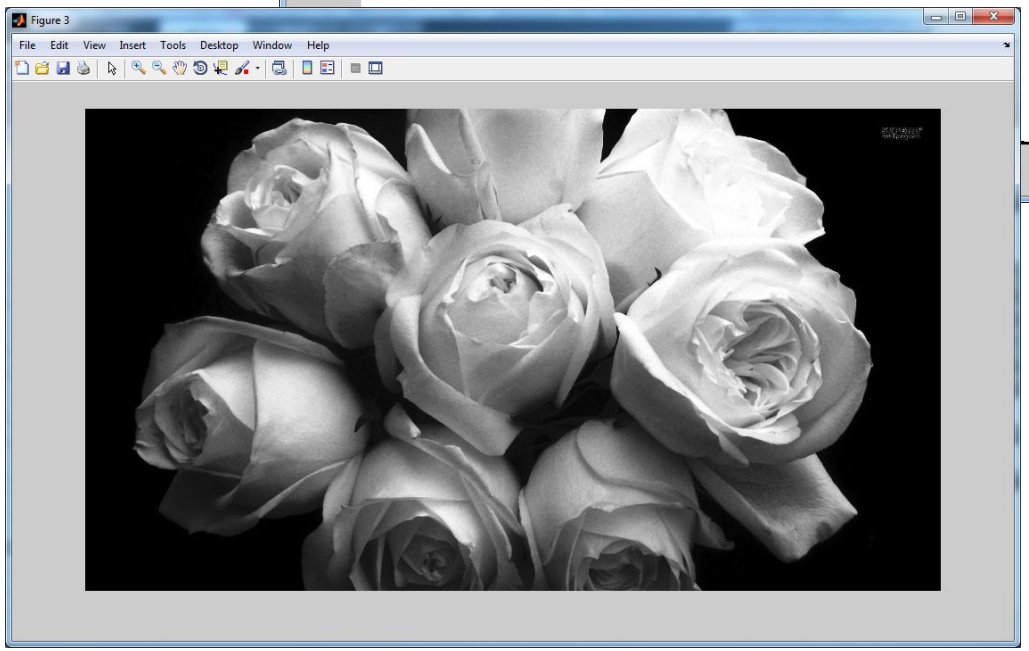
- PQ=paddedsize (size(red));
- D0=0.05.*PQ(1);
- H=lpfilter('gaussian', PQ(1), PQ(2), D0);
- g=dftfilt(red, H);
- figure, imshow (g, [0 255]);



- PQ=paddedsize (size(red));
- D0=0.05.*PQ(1);
- HBW=lpfilter('btw', PQ(1), PQ(2), D0,2);
- H=0.5+2*HBW;
- gbw=dftfilt(red, HBW);
- gbw=gscale(gbw);
- ghf=dftfilt(red, H);
- gbf=gscale(ghf);
- f1=histeq(red, 256);
- ghe=histeq(ghf, 256);
- figure, imshow (red);
- figure, imshow (ghe, []);
- figure, imshow (f1, []);



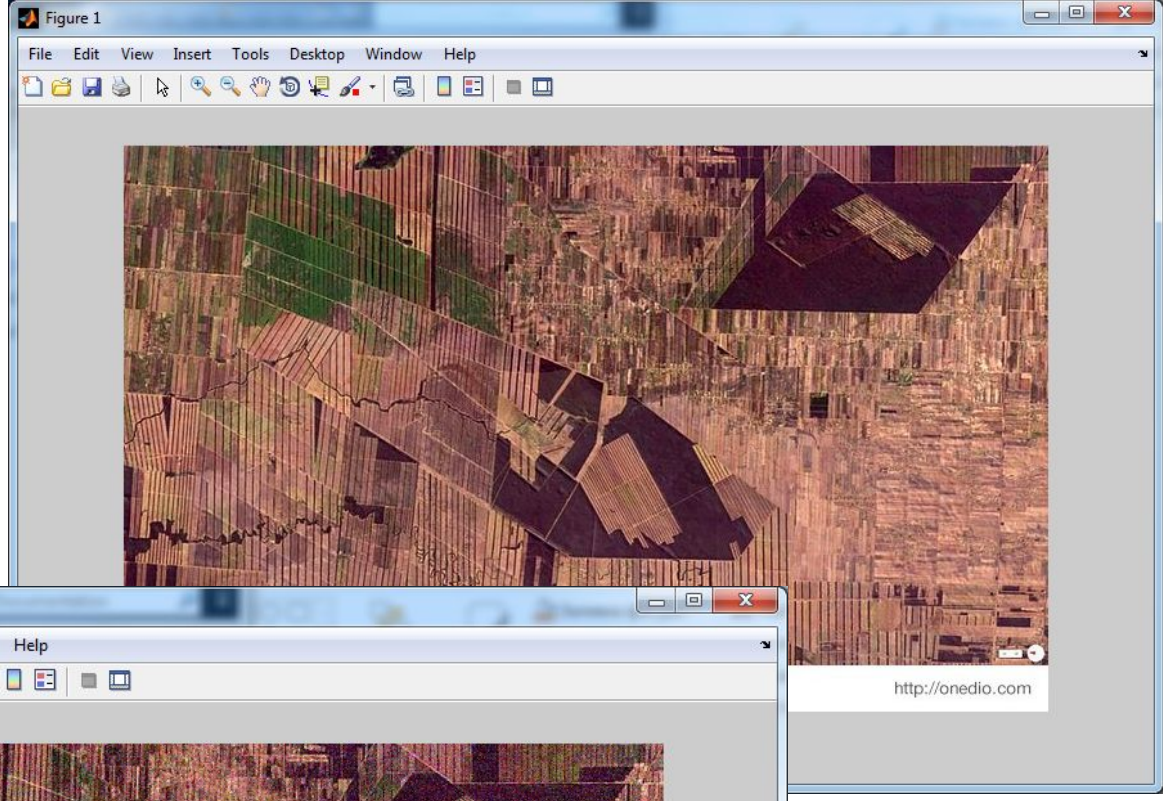


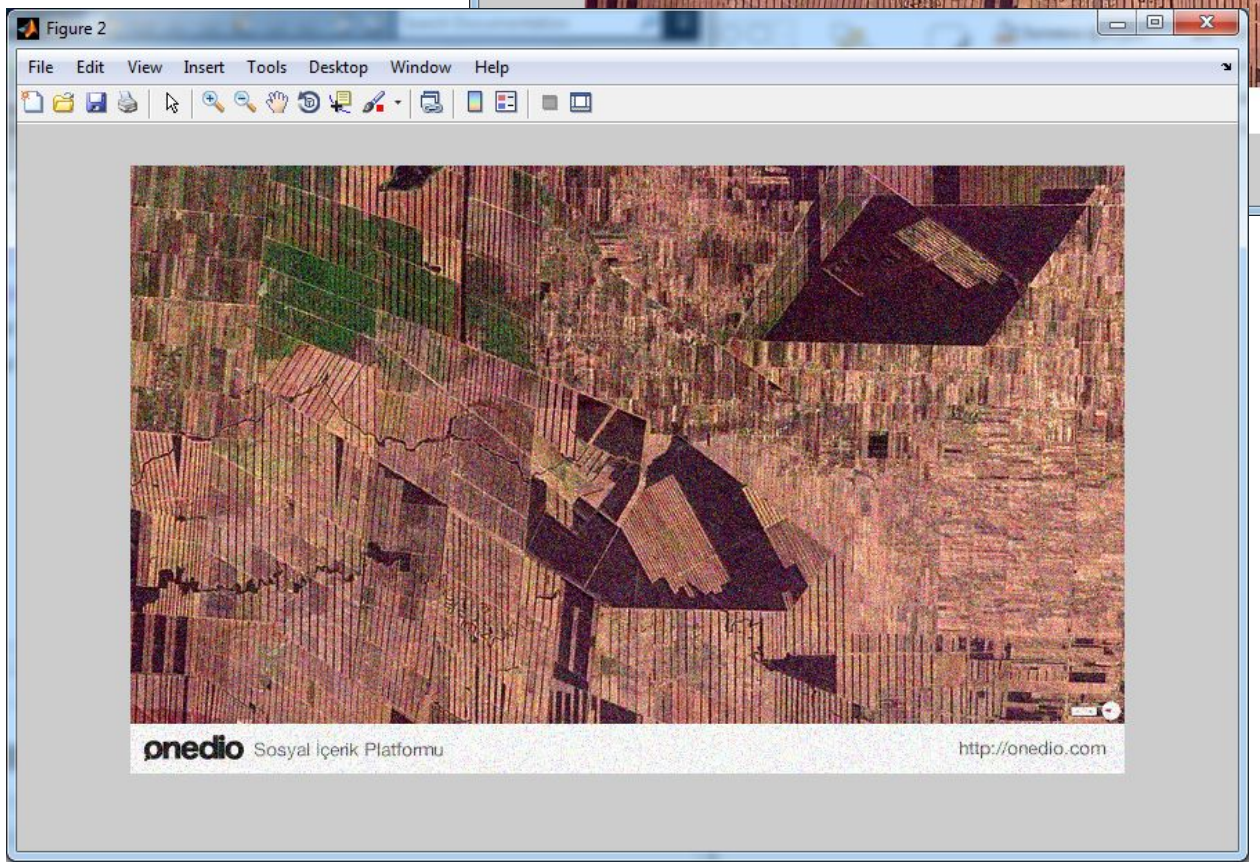


6.Модели шума

- Гауссов шум:

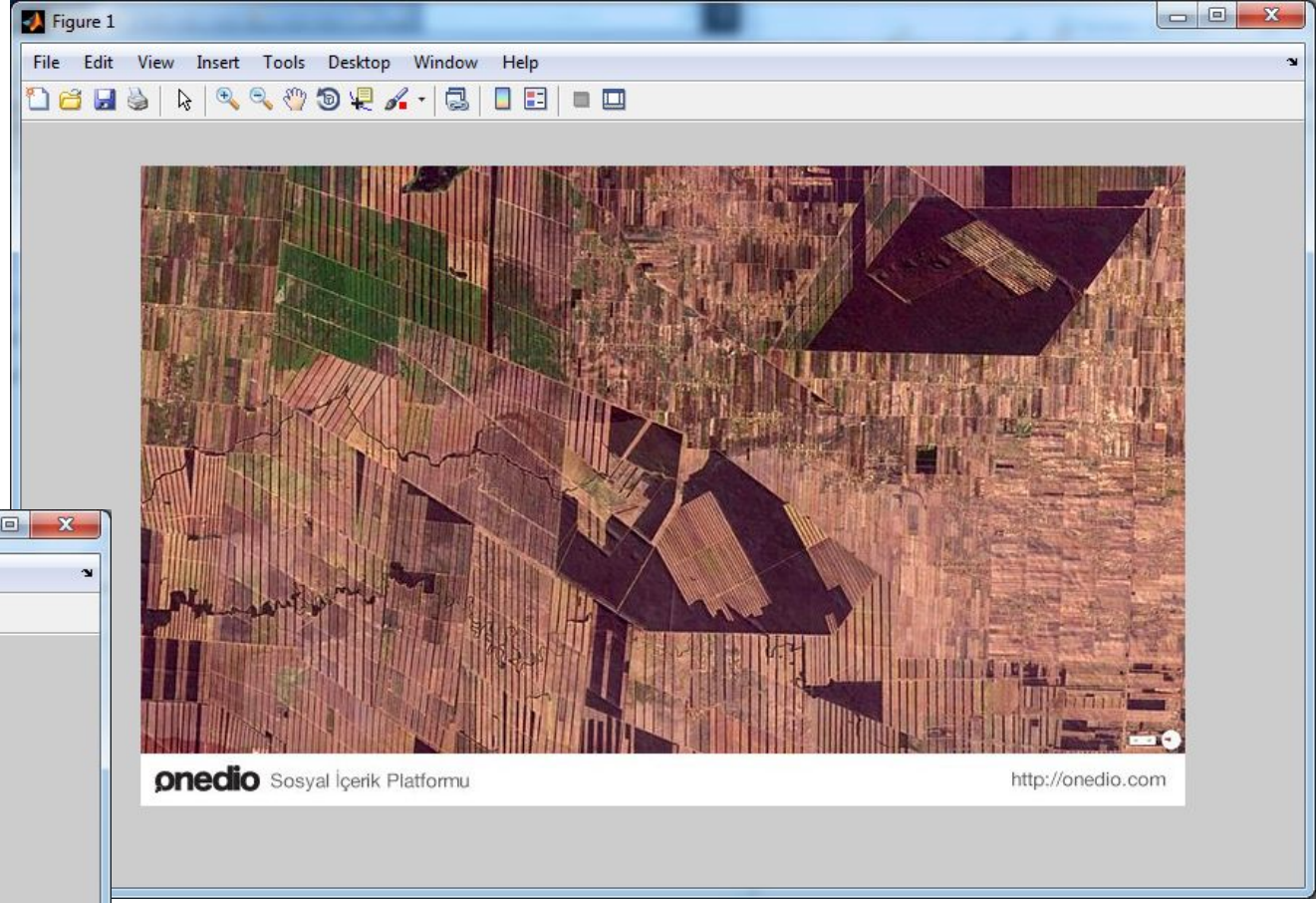
- f=imread('D:\539ee50c8597c8387f47b87b.jpg');
- >> g=imnoise(f,'gaussian',0,0.01);
- >> figure, imshow(f);
- >> figure, imshow(g);

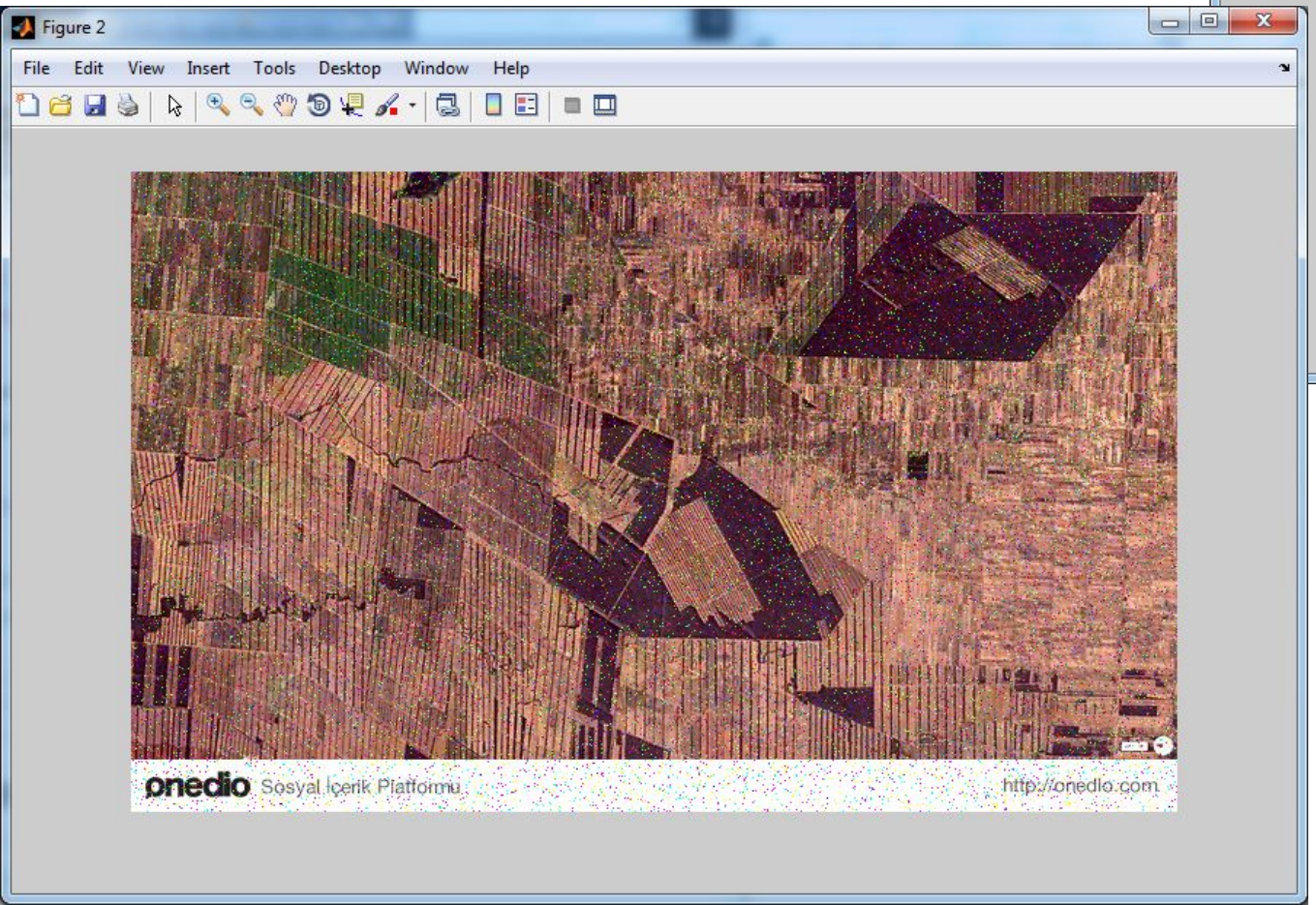




- Шум типа «соль и перец»:

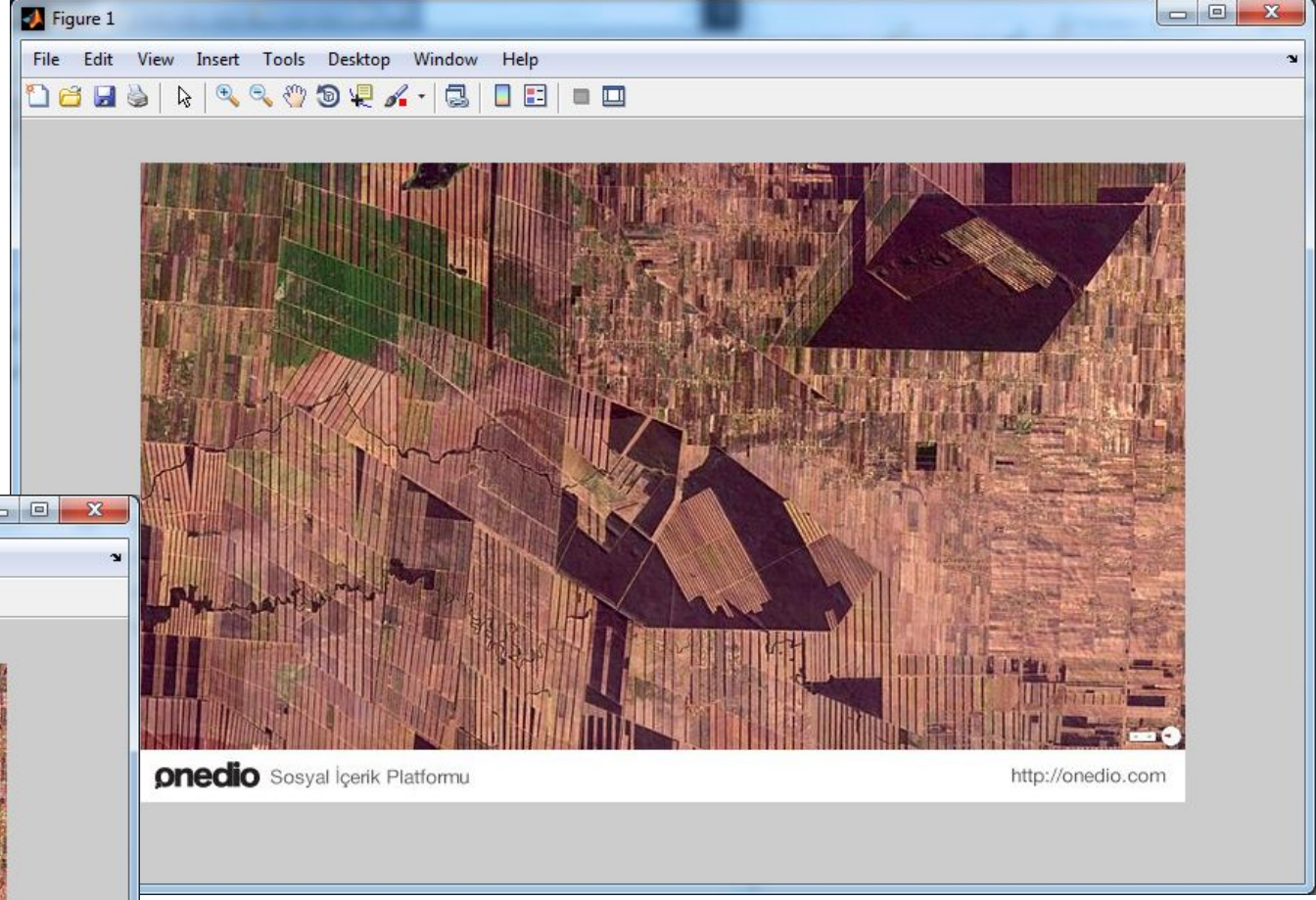
- f=imread('D:\539ee50c8597c8387f47b87b.jpg');
- >> g=imnoise(f,'salt & pepper',0.05);
- >> figure, imshow(f);
- >> figure, imshow(g);

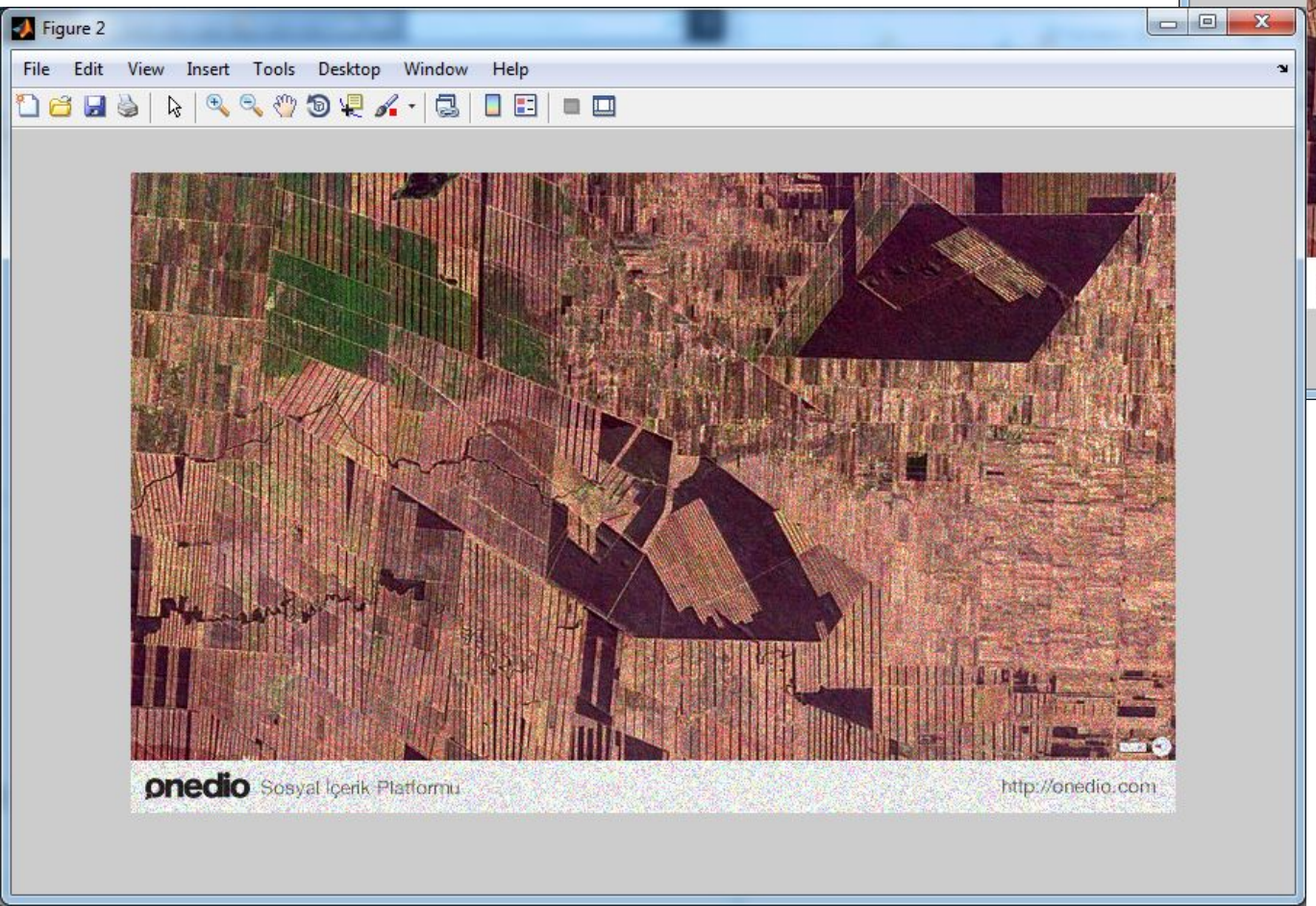




- Мультипликатный шум:

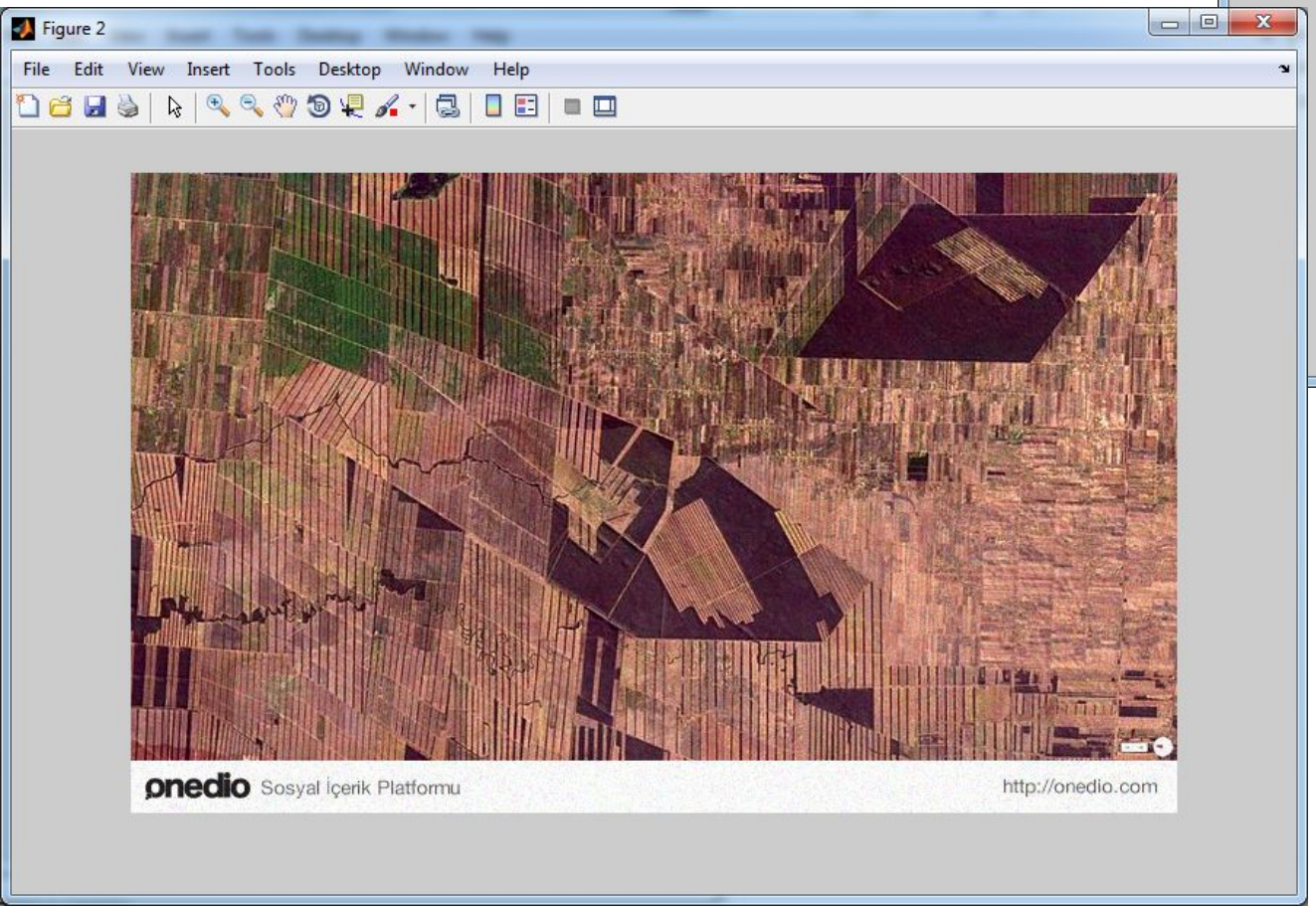
- f=imread('D:\539ee50c8597c8387f47b87b.jpg');
- >> g=imnoise(f,'speckle',0.04);
- >> figure, imshow(f);
- >> figure, imshow(g);

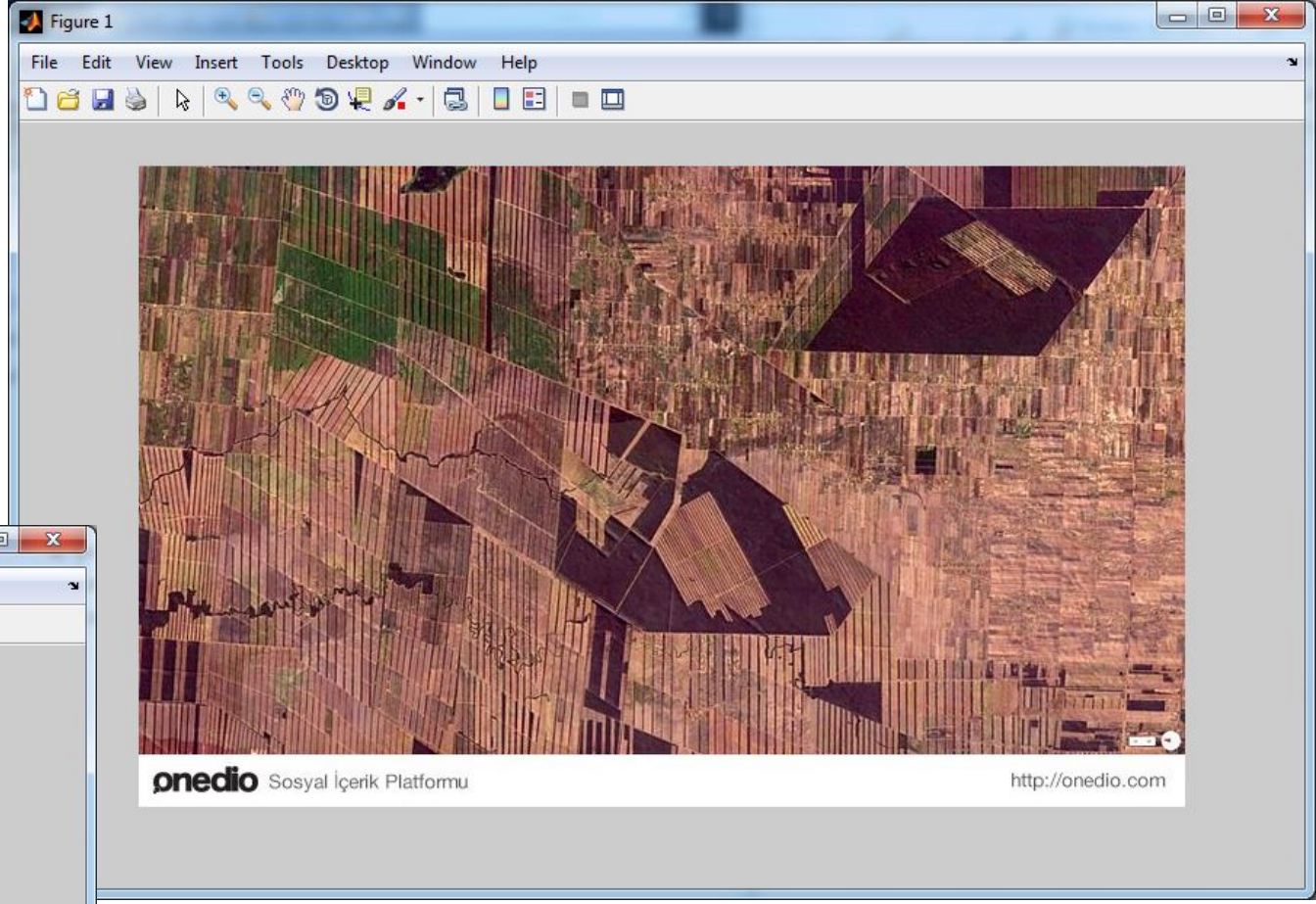




- Шум Пуассона:

- f=imread('D:\539ee50c8597c8387f47b87b.jpg');
- >> g=imnoise(f,'poisson');
- >> figure, imshow(f);
- >> figure, imshow(g);



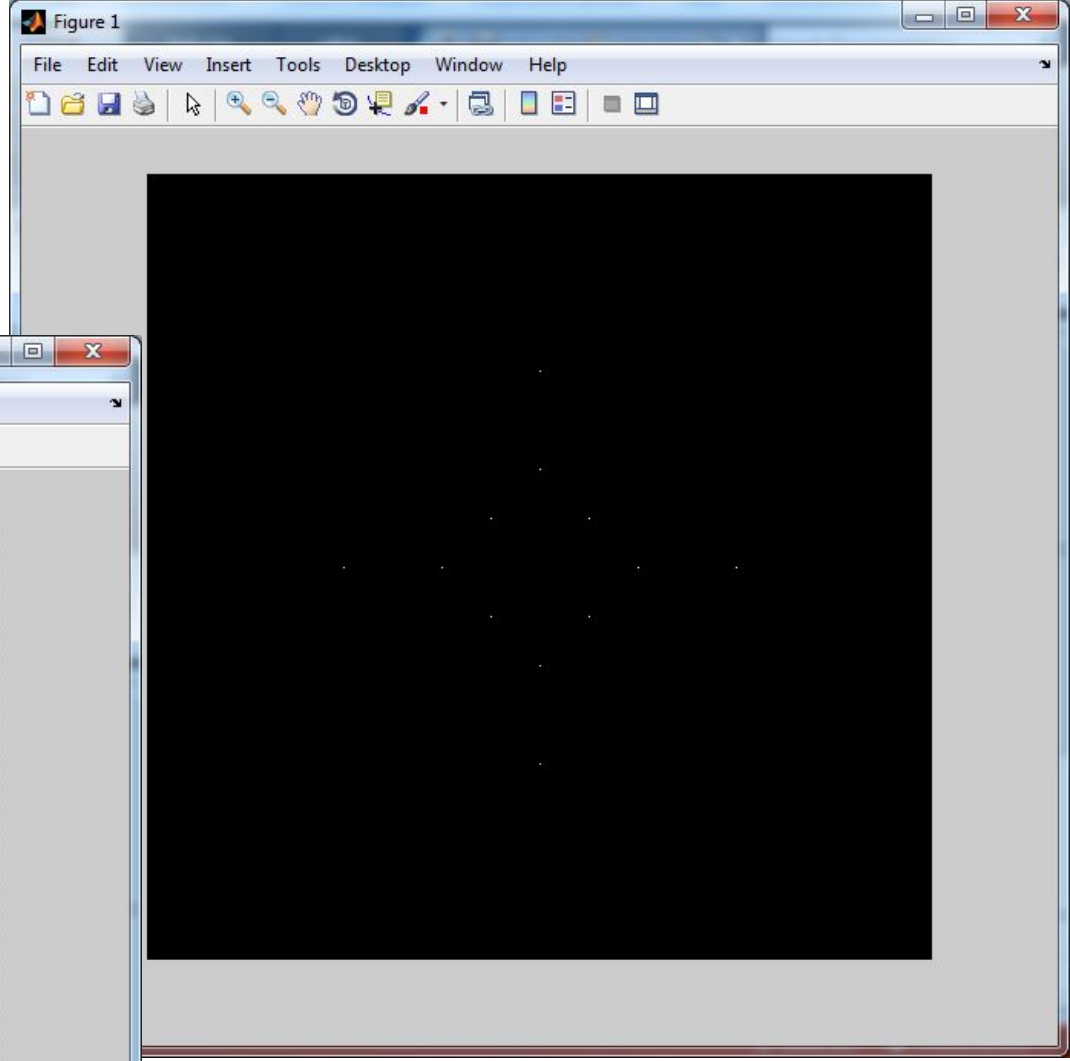


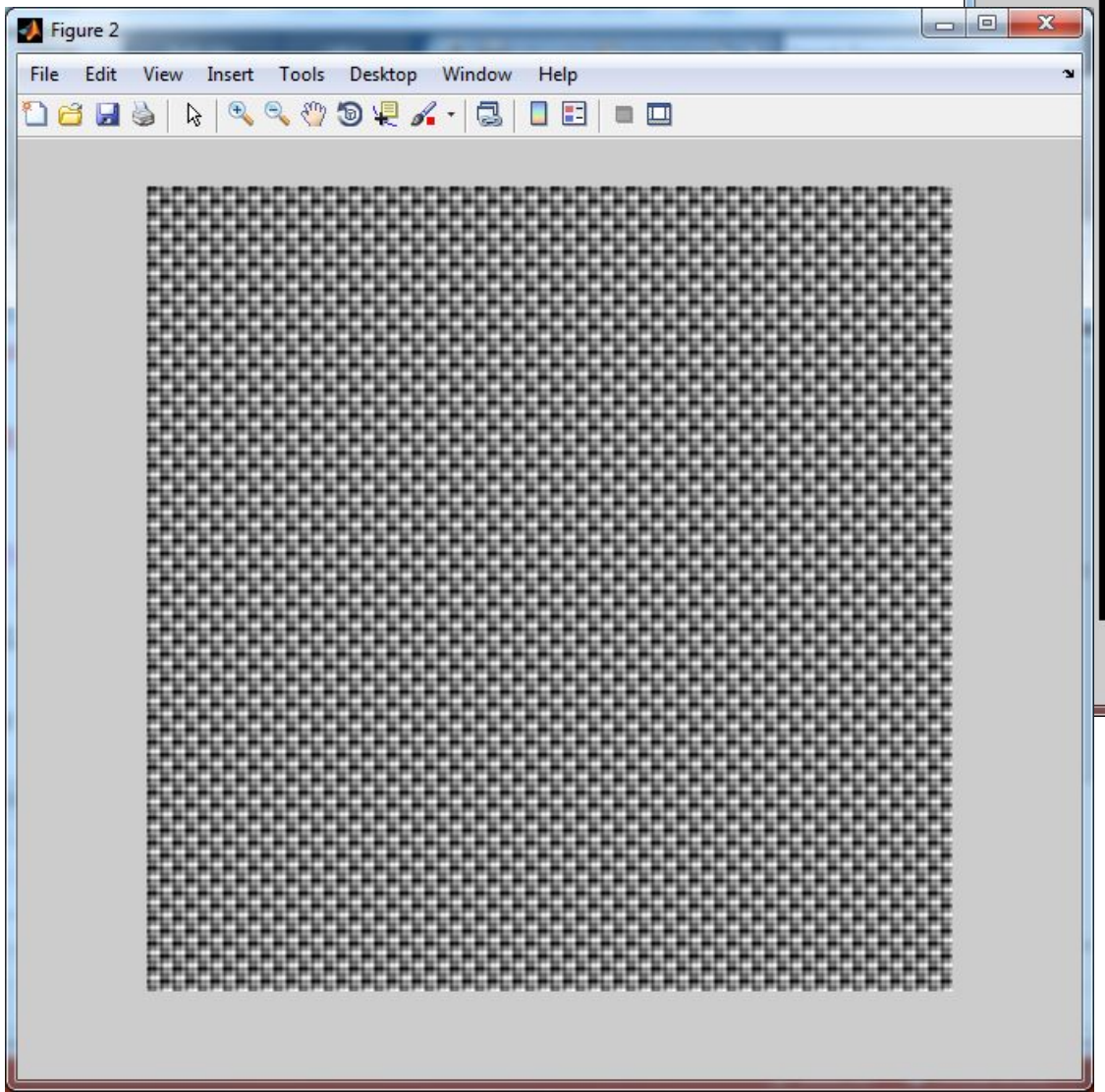
Периодический шум

Создаем функцию с именем imnoise3:

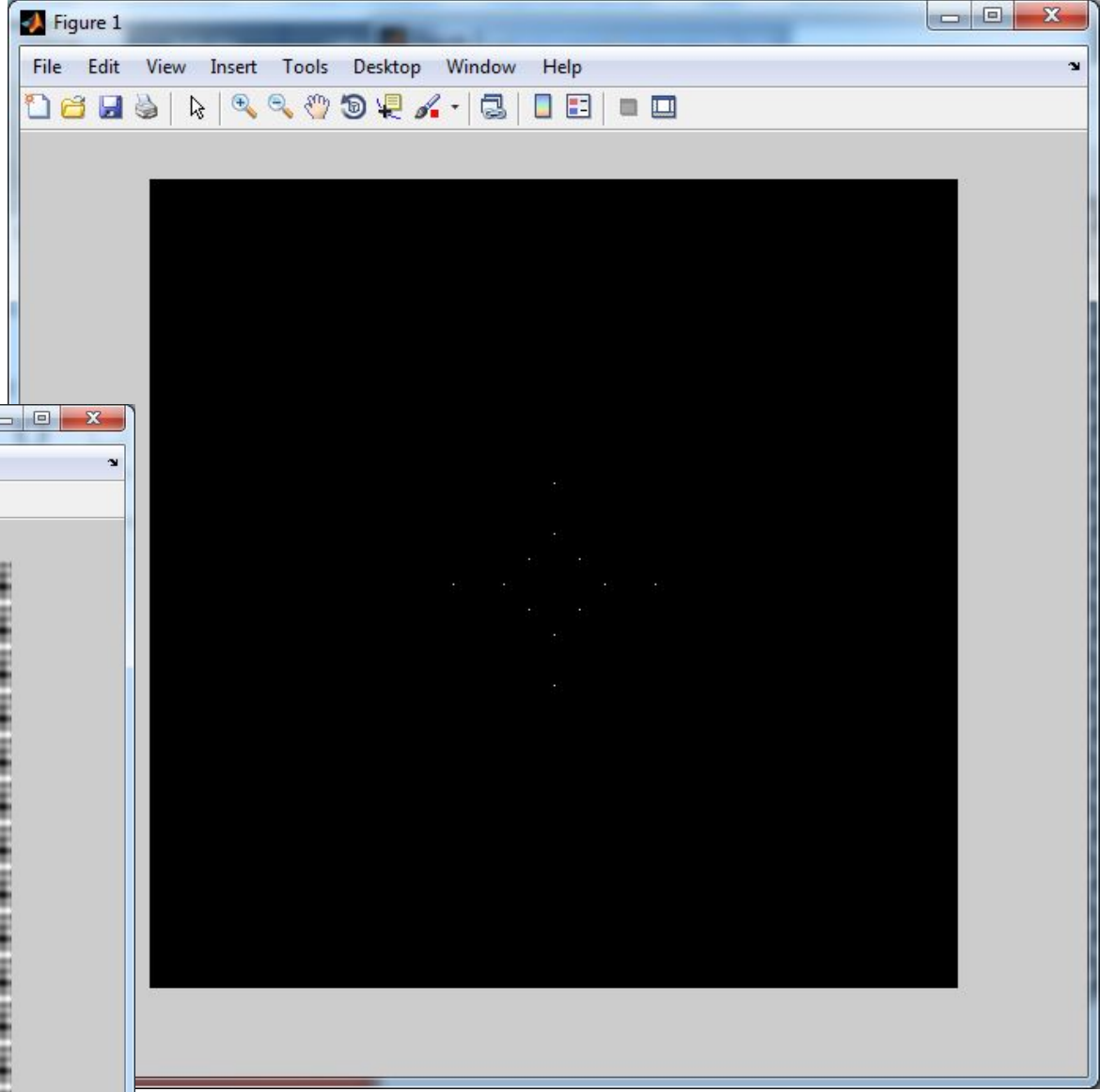
```
function [ r,R,S ] = imnoise3(M,N,C,A,B)
[K,n]=size(C);
if nargin==3
    A(1:K)=1.0;
    B(1:K,1:2)=0;
end
R=zeros(M,N);
for j=1:K
    u1=M/2+1+C(j,1); v1=N/2+1+C(j,2);
    R(u1,v1)=i*(A(j)/2)*exp(i*2*pi*C(j,1)*B(j,1)/M);
    u2=M/2+1-C(j,1); v2=N/2+1-C(j,2);
    R(u2,v2)=-i*(A(j)/2)*exp(i*2*pi*C(j,2)*B(j,2)/M);
end
S=abs(R);
r=real(ifft2(ifftshift(R)));
end
```

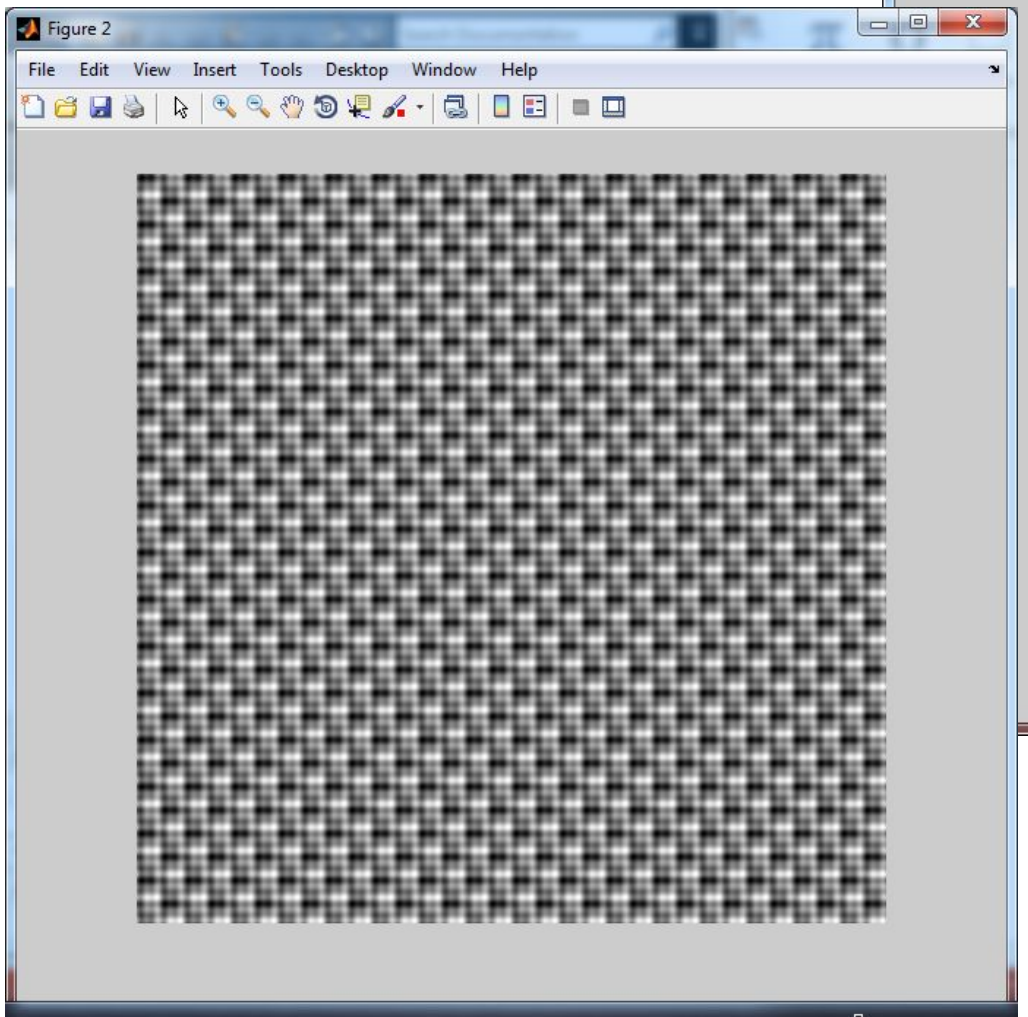
- C=[0 64; 0 128; 32 32; 64 0; 128 0;-32 32];
- [r,R,S]=imnoise3(512,512,C);
- >> imshow(S,[])
- >> figure,imshow(r,[]);



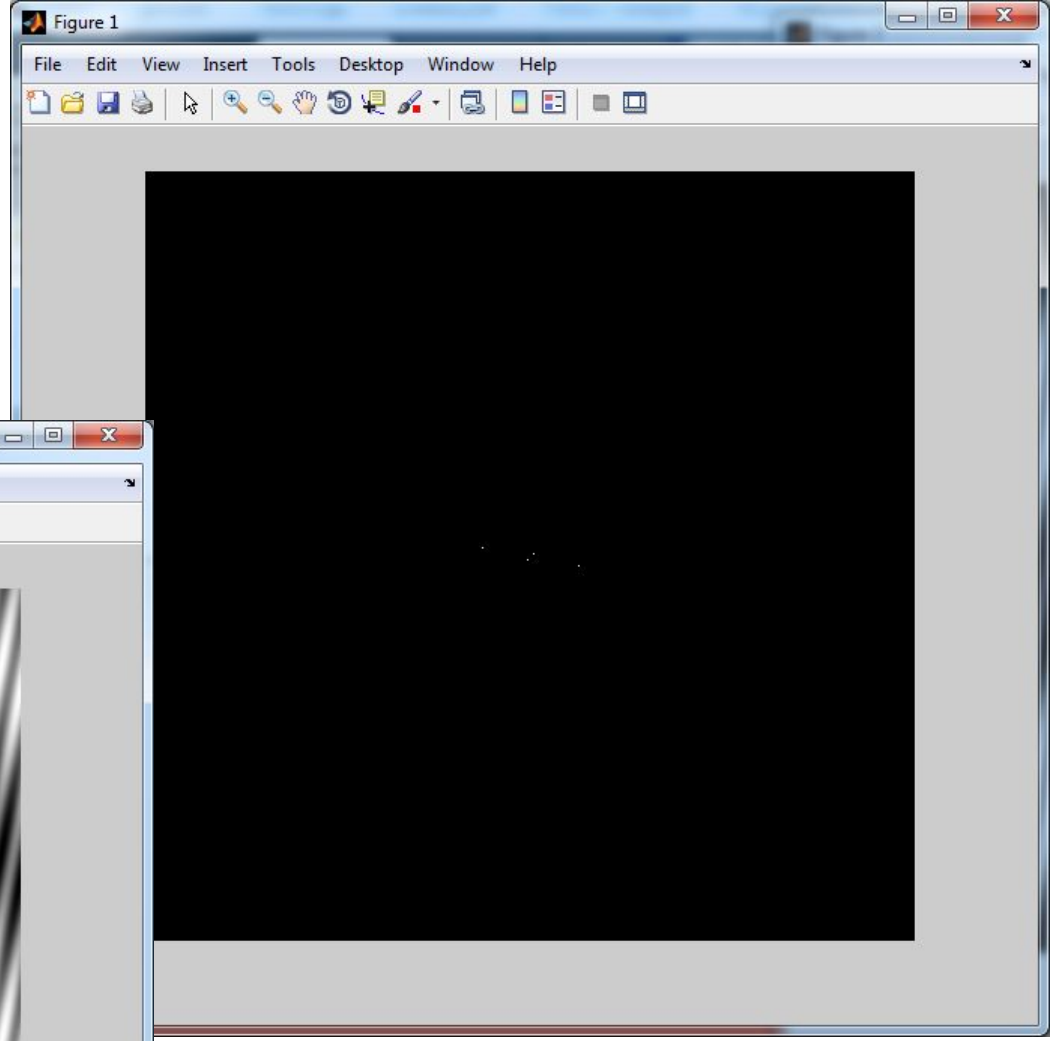


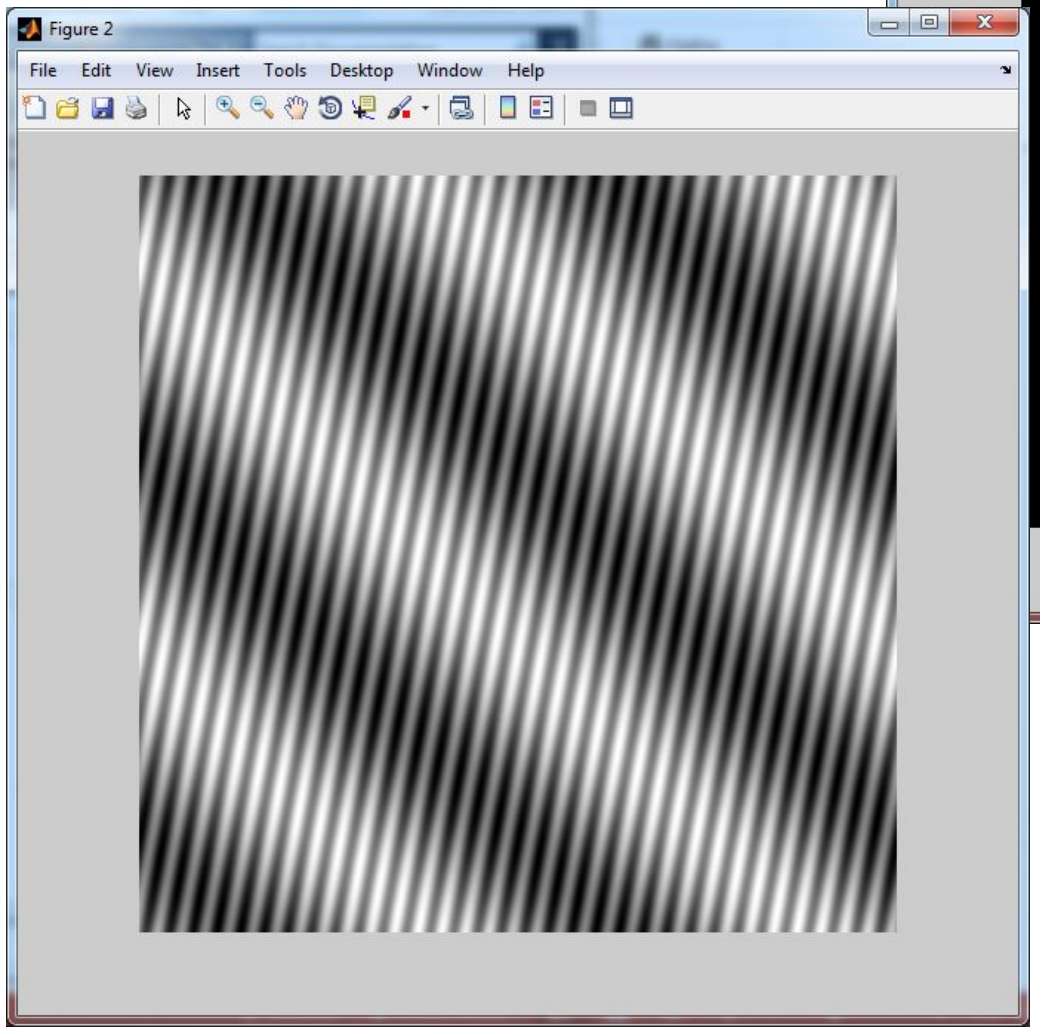
- C=[0 32; 0 64; 16 16;32 0;64 0;-16 16];
- >> [r,R,S]=imnoise3(512,512,C);
- imshow(S,[])
- figure,imshow(r,[]);





- C=[6 32; -2 2];
- >> [r,R,S]=imnoise3(512,512,C);
- imshow(S,[])
- figure,imshow(r,[]);

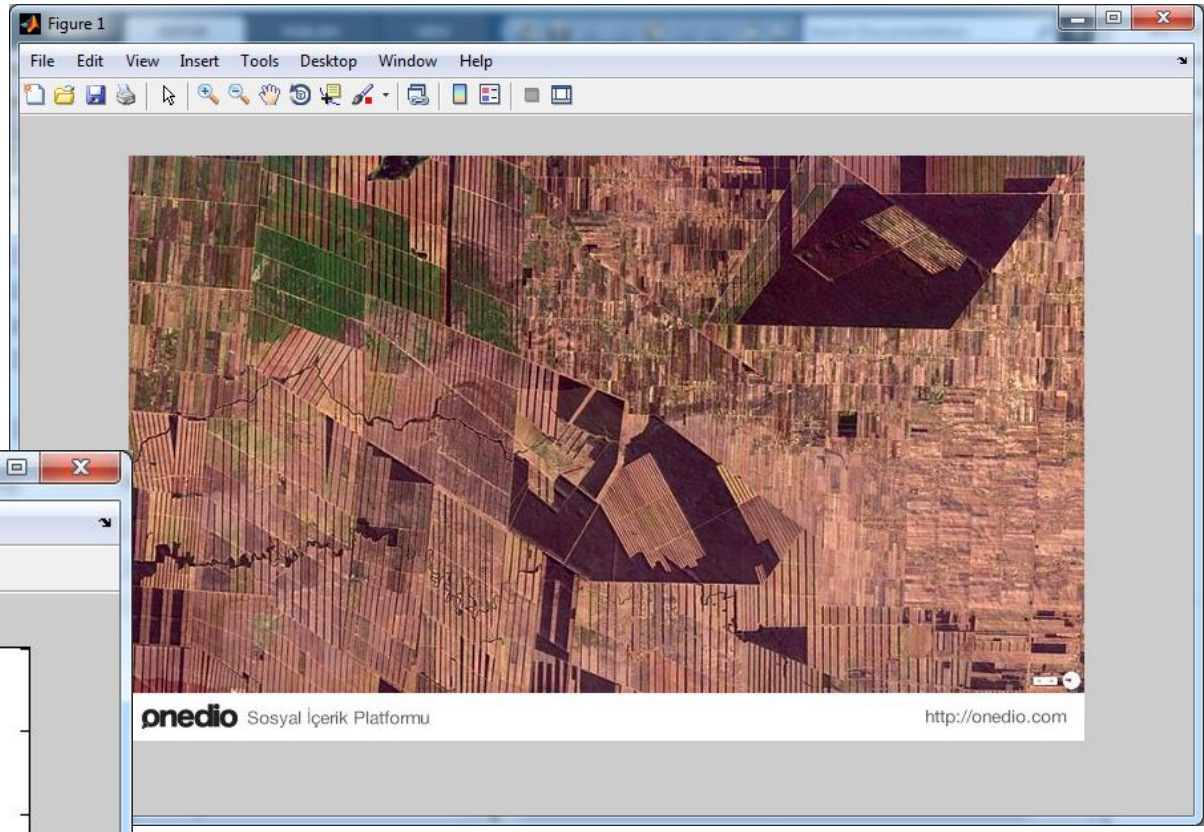


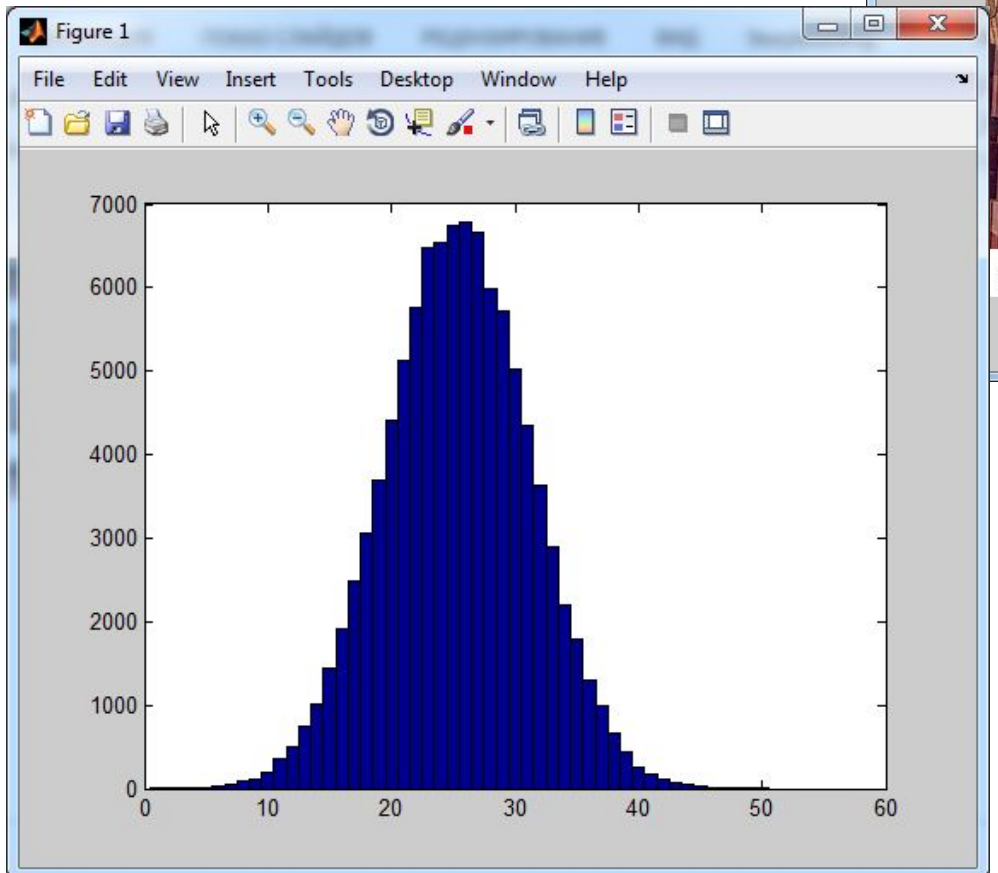


Оценивание параметров шума

- Создаем M-функцию с именем histroi:
- function [p,npix]=histroi(f,c,r)
- B=roipoly(f,c,r);
- p=imhist(f(B));
- if nargout>2
- npix=sum(B(:));
- end
- end

- f=imread('D:\539ee50c8597c8387f47b87b.jpg');
- >> [B,c,r]=roipoly(f);
- >> [p,npix]=histroi(f,c,r);
- >> figure,bar(p,1);





- >> f=imread('D:\539ee50c8597c8387f47b87b.jpg');

- B=roipoly(f,r,c);
- p=imhist(f(B));
- if nargout >2
- npix = sum(B(:));

- end
- [B,c,r] = roipoly(f);
- [p,npix]=histroi(f,c,r);
- figure,bar(p,1);

Создаем M-функцию с именем spfilt:

```
function  f= spfilt(g,type,m,n,parameter)
if nargin == 2
    m=3;n=3;Q=1.5;d=2;
elseif nargin==5
    Q=parameter;d=parameter;
elseif nargin==4
    Q=1.5;d=2;
else
    error('Wrong number of inputs.');
end
switch type
    case 'amean'
        w=fspecial('average',[m n]);
        f=imfilter(g,w,'replicate');
    case 'gmean'
        f=gmean(g,m,n);
    case 'hmean'
        f=harmean(g,n,m);
    case 'chmean'
        f=charmean(g,m,n,Q);
    case 'median'
```

Создаем M-функцию с именем gmean:

- function f=gmean(g,m,n)
- inclass=class(g);
- g=im2double(g);
- warning off;
- f=exp(imfilter(log(g),ones(m,n),'replicate')).^(1/m/n);
- warning on;
- f=changeclass(inclass,f);
- end

Создаем M-функцию с именем harmean :

- function f= harmean(g,m,n)
- inclass=class(g);
- g=im2double(g);
- f=m*n./imfilter(1./(g+eps),ones(m,n), 'replicate');
- f=changeclass(inclass,f);
- end

Создаем M-функцию с именем charmean:

- function f=charmean(g,m,n,q)
- inclass=class(g);
- g=im2double(g);
- f=imfilter(g.^(q+1),ones(m,n),'replicate');
- f=f./(imfilter(g.^(q+1),ones(m,n),'replicate')+eps);
- f=changeclass(inclass,f);
- end

- >> f=imread('D:\539ee50c8597c8387f47b87b.jpg');
- g=imnoise(f,'salt & pepper',0.05);
- figure, imshow(f);
- figure, imshow(g);
- >> fg=spfilt(g,'chmean',3,3,1.5)
- Изображение было подпорчено шумом типа соль. Затем с помощью вызова M-функции с именем spfilt, мы избавляемся от данного шума.

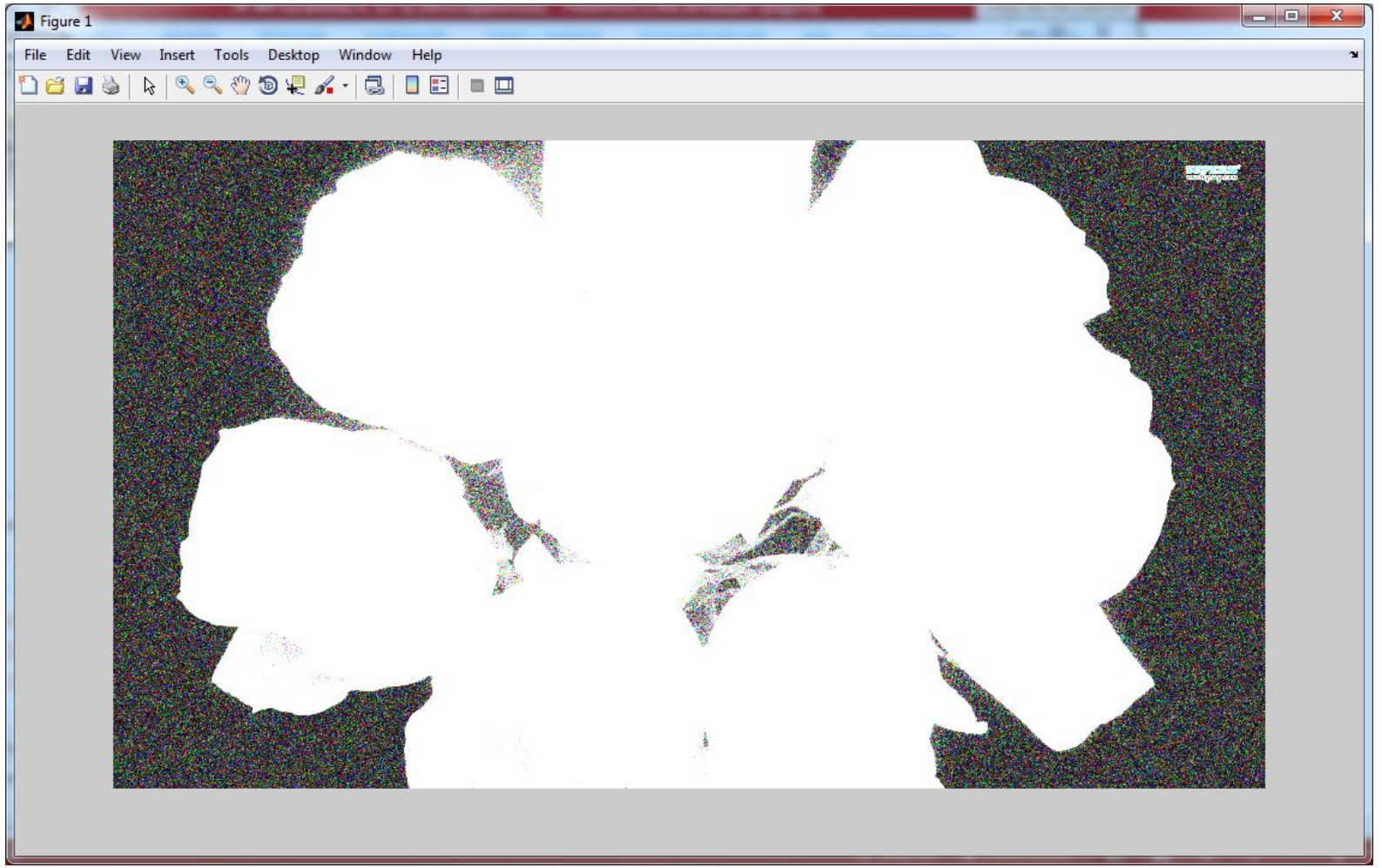
Моделирование размытого зашумленного изображения

- >> f=imread('D:\rose-white-bouquet-flower-227196.jpg');
- >> PSF=fspecial('motion',7,45);
- >> gd=imfilter(f,PSF,'circular');
- >> PSF

PSF =

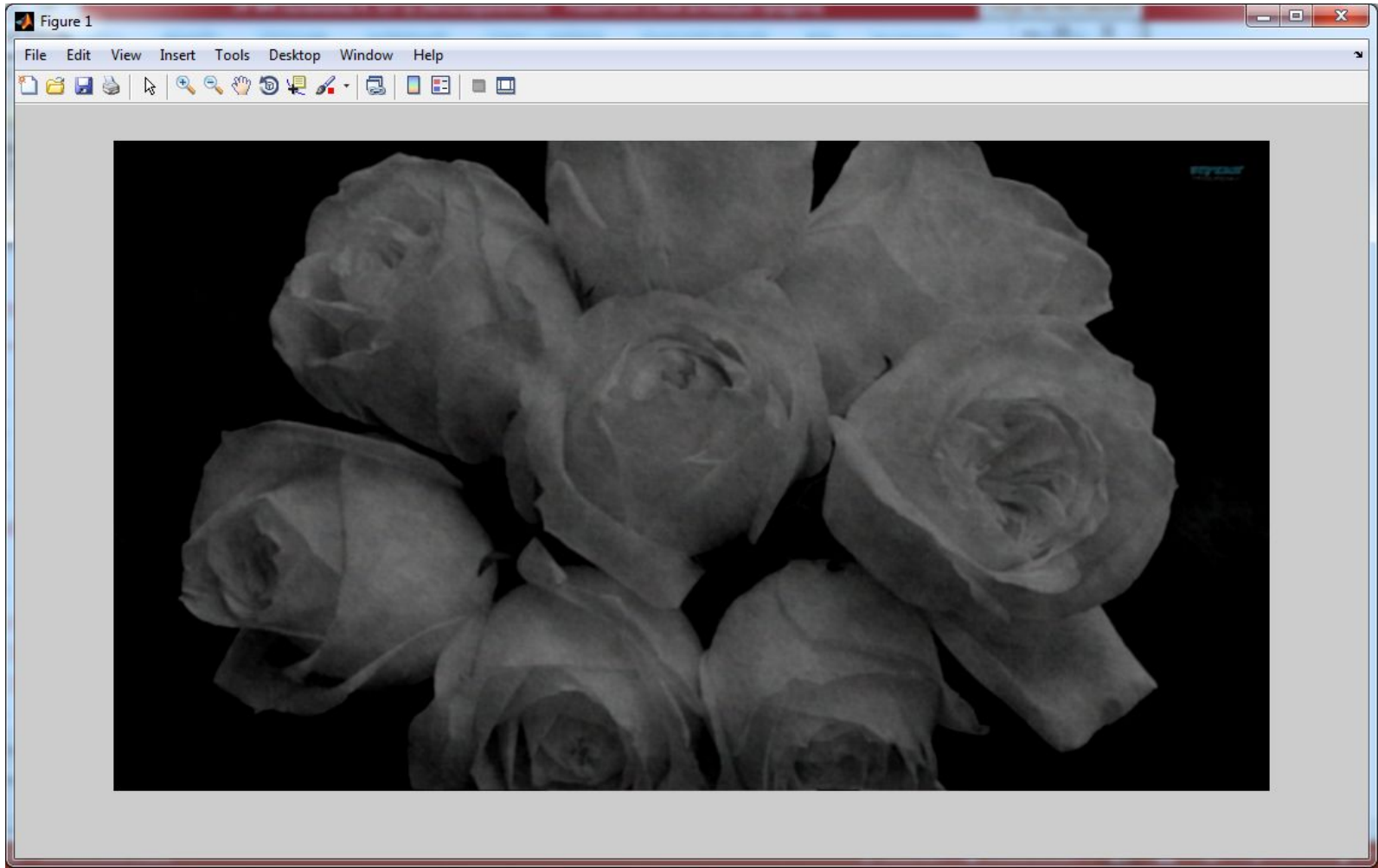
```
         0         0         0         0         0    0.0145         0
         0         0         0         0    0.0376    0.1283    0.0145
         0         0         0    0.0376    0.1283    0.0376         0
         0         0    0.0376    0.1283    0.0376         0         0
         0    0.0376    0.1283    0.0376         0         0         0
    0.0145    0.1283    0.0376         0         0         0         0
         0    0.0145         0         0         0         0         0
```

- >> noise=imnoise(gb,'gaussian',0,0.001);
- gb=g+noise;
- >> imshow(gb);



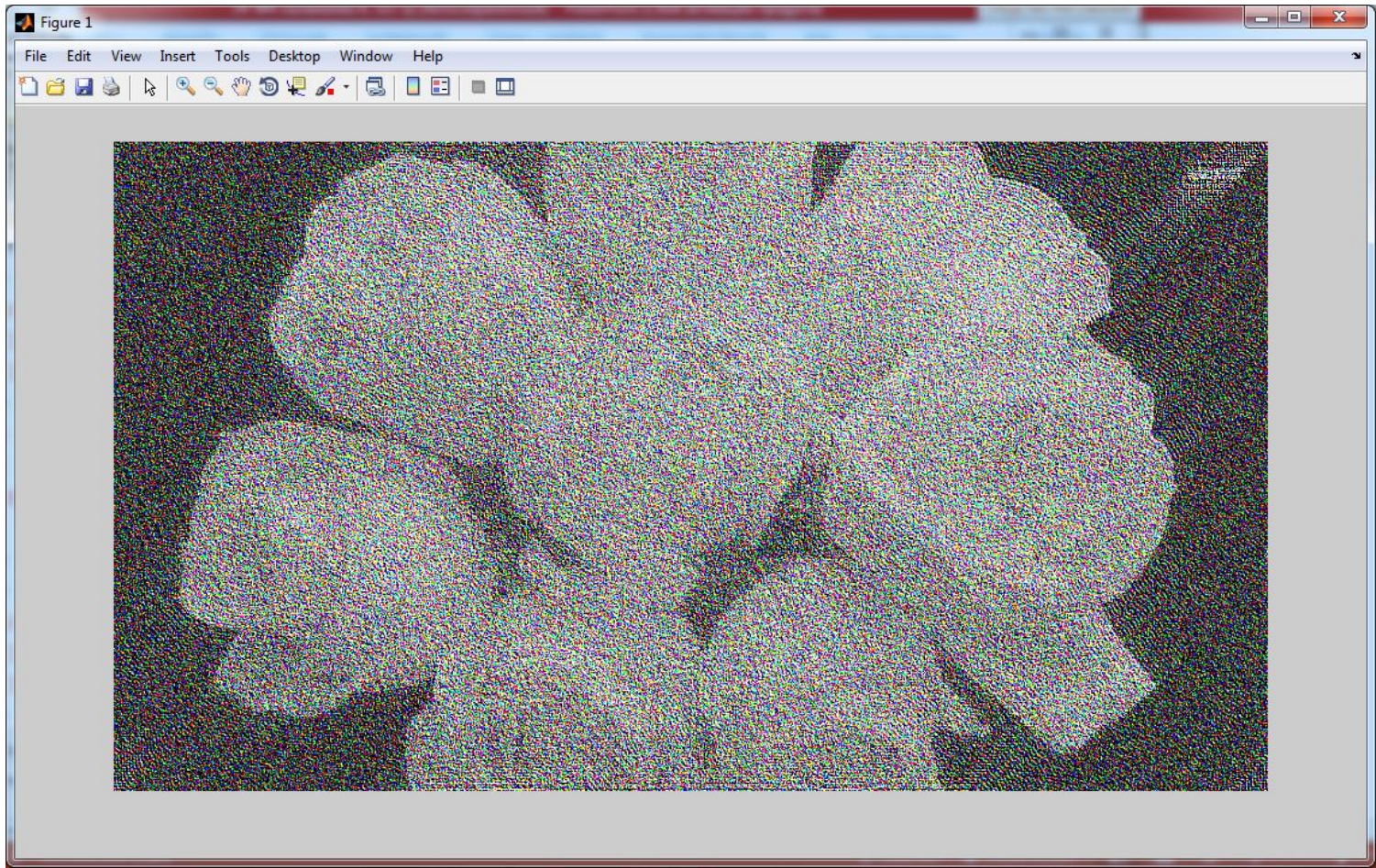
Использование автокорреляционной функции при восстановлении изображения

- >> f=imread('D:\rose-white-bouquet-flower-227196.jpg');
- PSF=fspecial('motion',7,45);
- gd=imfilter(f,PSF,'circular');
- >> NCORR=fftshift(real(ifft2(Sn)));
- >> ICORR=fftshift(real(ifft2(Sf)));
- >> fr3=deconvwnr(g,PSF,NCORR,ICORR);
- >> imshow(fr3);



Использование функции deconvreg при восстановлении смазанного зашумленного изображения

- >> f=imread('D:\rose-white-bouquet-flower-227196.jpg');
- PSF=fspecial('motion',7,45);
- gd=imfilter(f,PSF,'circular');
- fr=deconvreg(g,PSF,4);
- >> imshow(fr)



- >> f=imread('D:\rose-white-bouquet-flower-227196.jpg');
- PSF=fspecial('motion',7,45);
- gd=imfilter(f,PSF,'circular');
- fr=deconvreg(g,PSF,0.4,[1e-7 1e-7]);
- >> imshow(fr)