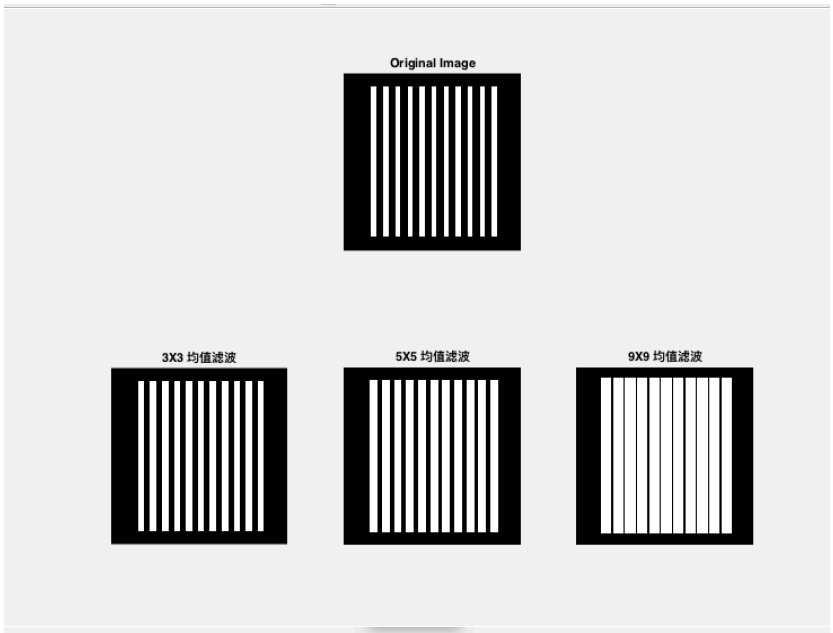
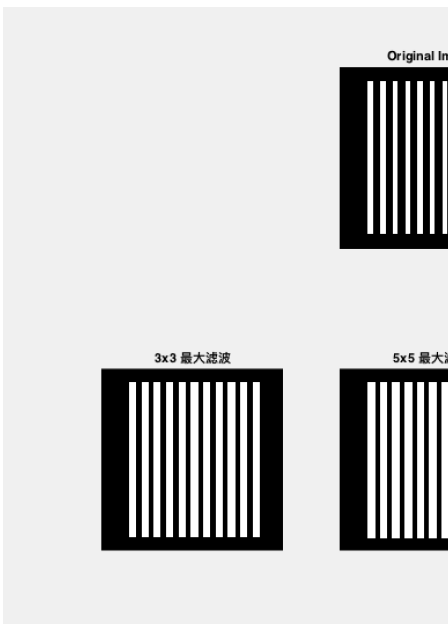
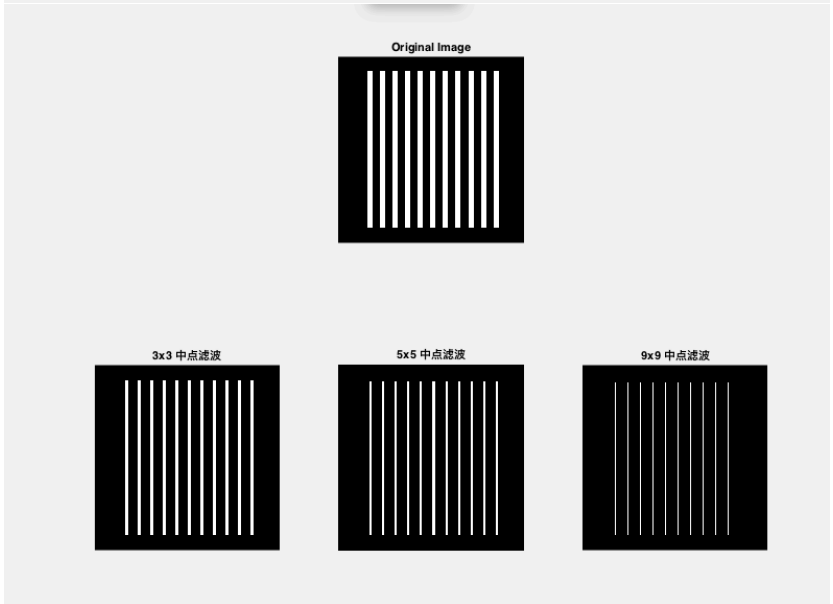
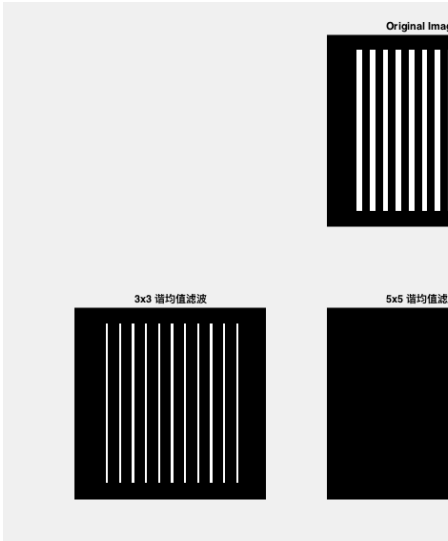
SeanChense

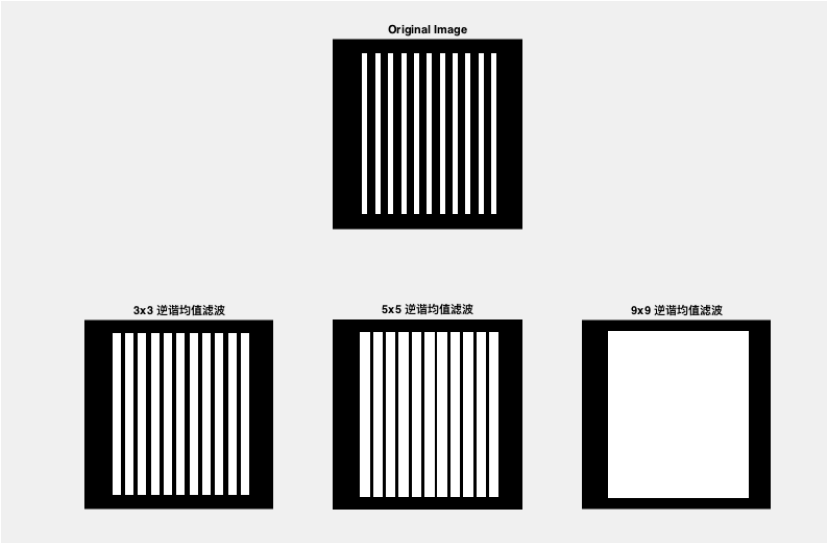
1


Original Image
3X3 均值滤波
5X5 均值滤波
9X9 均值滤波


3x3 最大滤波


Original Image
3x3 中点滤波
5x5 中点滤波
9x9 中点滤波


3x3 谐均值滤波



Main Script

```
periodPic = zeros(250, 250);
for i=-5:5
    periodPic(20:230, 125 + 17*i:125 + 17*i + 6) = 255;
end

threeFilter = [1, 1, 1;
               1, 1, 1;
               1, 1, 1]/9.0;

fiveFilter  = [1, 1, 1, 1, 1;
               1, 1, 1, 1, 1;
               1, 1, 1, 1, 1;
               1, 1, 1, 1, 1;
               1, 1, 1, 1, 1]/25.0;

nineFilter  = [1, 1, 1, 1, 1, 1, 1, 1, 1;
               1, 1, 1, 1, 1, 1, 1, 1, 1;
               1, 1, 1, 1, 1, 1, 1, 1, 1;
               1, 1, 1, 1, 1, 1, 1, 1, 1;
               1, 1, 1, 1, 1, 1, 1, 1, 1;
               1, 1, 1, 1, 1, 1, 1, 1, 1;
               1, 1, 1, 1, 1, 1, 1, 1, 1;
               1, 1, 1, 1, 1, 1, 1, 1, 1;
               1, 1, 1, 1, 1, 1, 1, 1, 1]/81.0;
```

```
% periodPic = imread('cameraman.tif');
subplot(2,3,2)
imshow(periodPic);
title('Original Image');
subplot(2,3,4)
imshow(SCFilter(periodPic, threeFilter));
title('3X3   ');
subplot(2,3,5)
imshow(SCFilter(periodPic, fiveFilter));
title('5X5   ');
subplot(2,3,6);
imshow(SCFilter(periodPic, nineFilter));
title('9X9   ');figure;


subplot(2,3,2)
imshow(periodPic);
title('Original Image');
subplot(2,3,4)
imshow(SCOrderFilter(periodPic, 1, 3));title('3x3   ');
subplot(2,3,5)
imshow(SCOrderFilter(periodPic, 1, 5));title('5x5   ');
subplot(2,3,6);
imshow(SCOrderFilter(periodPic, 1, 9));title('9x9   ');figure;

subplot(2,3,2)
imshow(periodPic);
title('Original Image');
subplot(2,3,4)
imshow(SCOrderFilter(periodPic, 0, 3));title('3x3   ');
subplot(2,3,5)
imshow(SCOrderFilter(periodPic, 0, 5));title('5x5   ');
subplot(2,3,6);
imshow(SCOrderFilter(periodPic, 0, 9));title('9x9   ');figure;

subplot(2,3,2)
imshow(periodPic);
title('Original Image');
subplot(2,3,4)
imshow(SCHarmonicMeanFilter(periodPic, 0, int32(3)));title('3x3    ');
subplot(2,3,5)
imshow(SCHarmonicMeanFilter(periodPic, 0, int32(5)));title('5x5    ');
subplot(2,3,6);
imshow(SCHarmonicMeanFilter(periodPic, 0, int32(9)));title('9x9    ');
figure;
```

```
subplot(2,3,2)
imshow(periodPic);
title('Original Image');
subplot(2,3,4)
imshow(SCHarmonicMeanFilter(periodPic, 1, int32(3)));title('3x3    ');
subplot(2,3,5)
imshow(SCHarmonicMeanFilter(periodPic, 1, int32(5)));title('5x5    ');
subplot(2,3,6);
imshow(SCHarmonicMeanFilter(periodPic, 1, int32(9)));title('9x9    ');
```

SCHarmonicMeanFilter.m

```
% SeanChense
% style: 0
%        1
function [ result ] = SCHarmonicMeanFilter(f, style, dim)
int_dim = int32(dim);
[m, n]= size(f);
result = zeros(m, n);
if style == 0
    for i = 1+int_dim/2:m-int_dim/2
        for j = 1+int_dim:n-int_dim/2
            con=0; s1=0;
            for k1 = i-int_dim/2:i+int_dim/2
                for p1 = j-int_dim/2:j+int_dim/2
                        con = con+1;
                        if f(k1,p1)==0
                            s1 = s1+0;
                        else
                            s1=s1+(1/f(k1,p1));
                        end
                end
            end
            temp = con/s1;
            if temp > 255
                temp = 0;
            end

            result(i,j)=temp;
        end
    end

else
    Q=-1.5;
```

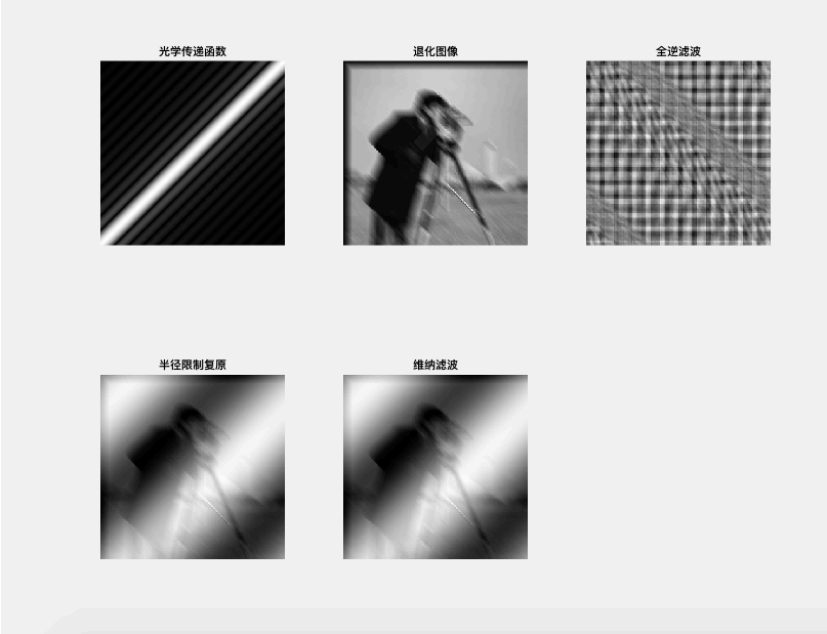
```
    for i = 1+int_dim/2:m-int_dim/2
        for j = 1+int_dim/2:n-int_dim/2
            con=0; s1=0; s2=0;
            for k1 = i-int_dim/2:i+int_dim/2
                for p1 = j-int_dim/2:j+int_dim/2
                        con = con+1;
                        if f(k1, p1) == 0
                            s1=s1+0;
                            s2=s2+(f(k1,p1)^(Q+1));
                        else
                            s1=s1+(f(k1,p1)^Q);
                            s2=s2+(f(k1,p1)^(Q+1));
                        end
                end
            end


            if s1 ~= 0 && s2 ~= 0
                temp = s2*1.0/(s1);
                if temp > 255
                    result(i,j)= 255;
                else
                    result(i,j)=temp;
                end

            else

            end
        end
    end
end% if
end
```

2



Main Script.m

```
%
% SeanChense
%
clear all; close all; clc;
image = imread('cameraman.tif');

[M,N] = size(image);
P = 2*M; Q = 2*N;      % zeros padding
[v,u] = meshgrid(1:Q,1:P);
u = u - floor(P/2);
v = v - floor(Q/2);

T = 1.0;              % The duration of the exposure
a = 0.025;
b = 0.025;
Duv = pi*( a.*u + b.*v + eps); %      2.2204e-016
Hmov = T./Duv .* sin(Duv) .* exp( -1j*Duv ); % Complex value

OTF = Hmov;           % Select the OTF (    )
subplot(231),imshow(log(1+abs(OTF)),[]),title('    ');
```

```
FI = fftshift(fft2(image,P,Q));
blurFT = FI.* OTF; % Filtering
blurIp = real(ifft2(ifftshift(blurFT)));
blurI =  blurIp(1:M,1:N);     % Crop the border
subplot(232)
g = im2uint8(mat2gray(blurI));
imshow(g),title('   ');

radius = 500;          %
%
fg = revfilter(g,OTF(1:M, 1:N),radius);
subplot(233)
imshow(fg,[]),title('   ');

radius = 0.001;
%    ,
fg = revfilter(g,OTF(1:M, 1:N),radius);
subplot(234)
imshow(fg,[]),title('    ');


%
wn = SCWinnerFilter(g, OTF(1:M, 1:N), 0.01);
subplot(235)
imshow(fg,[]),title('   ');
```

revfilter.m

```
function newim = revfilter(im,psf,radius)
%============================================================================
%
% newim = revfilter(ima,psf,radius)
% ima:
% PSF:
% radius:
% newim:
%============================================================================
%  rgb ,
if ndims(im)>=3
    im = rbg2gray(im);
end

im = im2double(im);%  /255
```

```
%    (FFT)
Fim = fftshift(fft2(im));
P = Fim;

[M,N] = size(P);
%
if radius > M/2         %
    P = P./(psf + eps); %
else
    %
    for i = 1:M
        for j = 1:N
            if sqrt((i-M/2).^2+(j-N/2).^2) < radius
                P(i,j) = P(i,j)./(psf(i,j)+eps);
            end
        end
    end
end
% IFFT
newim = real(ifft2(ifftshift(P)));
newim = (newim);
end
```

SCWinnerFilter.m

```
%
% SeanChense
%
function newim = SCWinnerFilter(im, psf, gamma)

im = im2double(im);%  /255

%    (FFT)
Fim = fftshift(fft2(im));
P = Fim;
[M,N] = size(P);

for i = 1:M
    for j = 1:N
        P(i,j) = conj(psf(i, j))./((psf(i, j).^2) + psf(i, j));
    end
end

% IFFT
newim = real(ifft2(ifftshift(P)));
```

```
newim = im2uint8(newim);
end
```