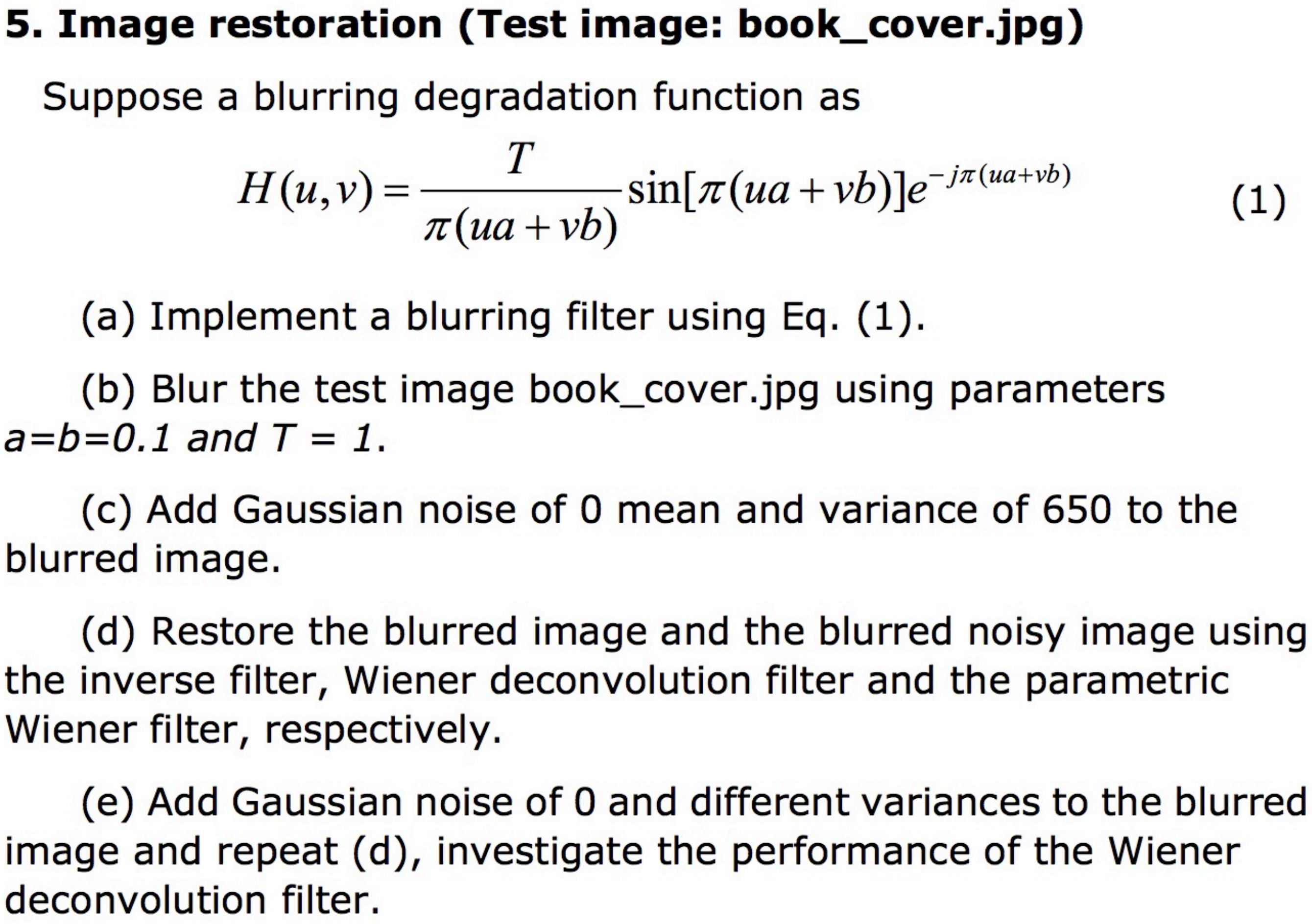
Digital Image Processing

BB1603391 116033910045 修宇亮

Problem 5 Requirement



5. Image restoration (Test image: book_cover.jpg)

Suppose a blurring degradation function as

$$H(u,v)=\frac{T}{\pi(ua+vb)}\sin[\pi(ua+vb)]e^{-j\pi(ua+vb)} \quad (1)$$

(a) Implement a blurring filter using Eq. (1).

(b) Blur the test image book_cover.jpg using parameters *a=b=0.1 and T = 1*.

(c) Add Gaussian noise of 0 mean and variance of 650 to the blurred image.

(d) Restore the blurred image and the blurred noisy image using the inverse filter, Wiener deconvolution filter and the parametric Wiener filter, respectively.

(e) Add Gaussian noise of 0 and different variances to the blurred image and repeat (d), investigate the performance of the Wiener deconvolution filter.

Problem 5 solution

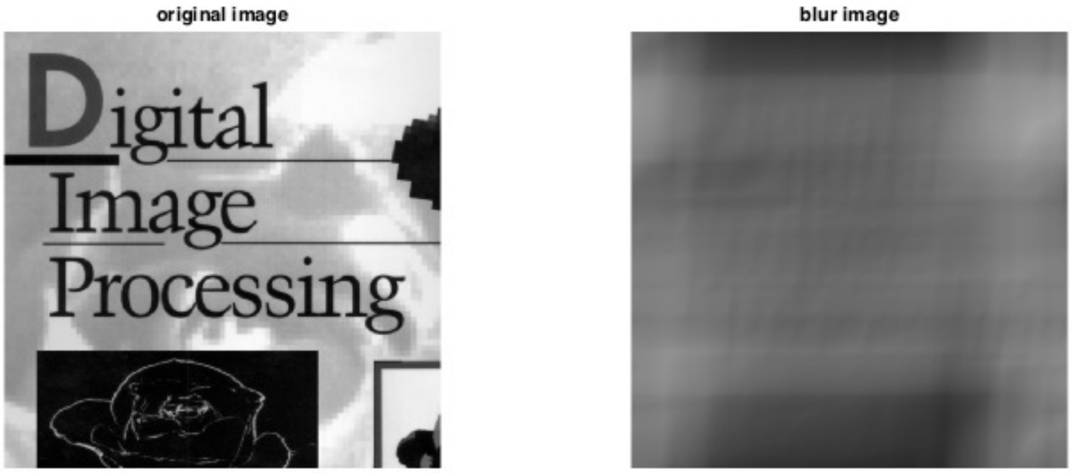
Blurring Filter Code

MATLAB

```
orig_img = imread('book_cover.jpg');
set(gcf,'position',[0,0,800,300]);
subplot(1,2,1),imshow(orig_img);title('original image');
F = fft2(double(orig_img));
H = blur_filter(F,0.1,0.1,1);
G = F.*H;
result = ifft2(G);
blur_img = uint8(real(ifft2(G)));
subplot(1,2,2),imshow(blur_img);title('blur image')

function H = blur_filter(F,a,b,T)
    [M,N] = size(F);
    H = F;
    for U=1:M
        for V=1:N
            puab = pi*(U*a+V*b);
            H(U,V) = T*sin(puab).*exp(-j*puab)./puab;
        end
    end
end
```

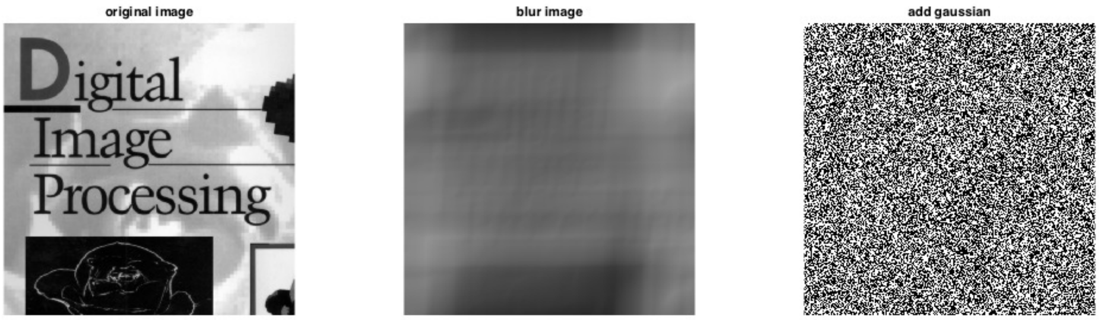
Blurring Result



Add Gaussian Noise Code

```MATLAB
orig_img = imread('book_cover.jpg');
set(gcf,'position',[0,0,1200,300]);
subplot(1,3,1),imshow(orig_img);title('original image');
[M,N] = size(orig_img);
F = fft2(double(orig_img));
H = blur_filter(F,0.1,0.1,1);
G = F.*H;
result = ifft2(G);
blur_img = uint8(real(ifft2(G)));
subplot(1,3,2),imshow(blur_img);title('blur image');
gau_noise = imnoise2('gaussian',M,N,0,650);
blur_gau_img = im2uint8(im2double(blur_img)+gau_noise);
subplot(1,3,3),imshow(blur_gau_img);title('add gaussian');
```

Add Gaussian Noise Result



Restore Blurred Image and Blurred Gaussian Image

Inverse Filter

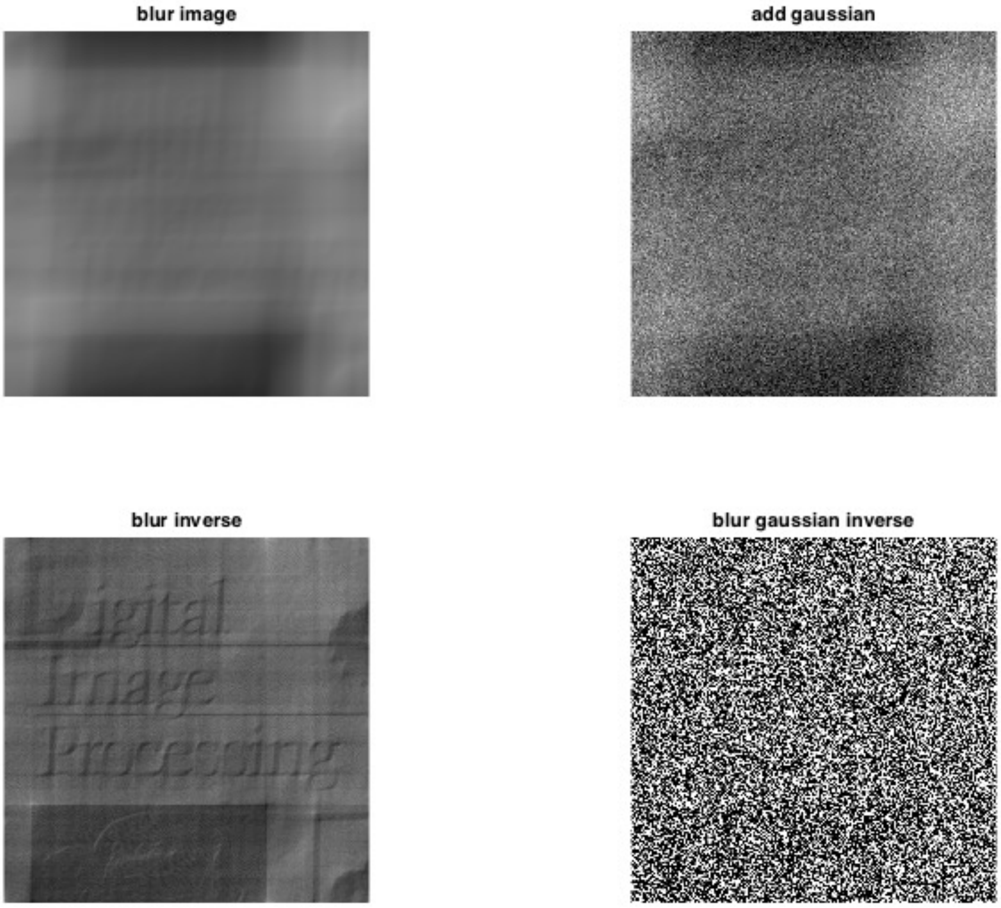
MATLAB

```matlab
orig_img = imread('book_cover.jpg');
[M,N] = size(orig_img);
% produce noise image
F = fft2(double(orig_img));
H = blur_filter(F,0.1,0.1,1);
G = F.*H;
result = ifft2(G);
blur_img = uint8(real(ifft2(G)));
gau_noise = imnoise2('gaussian',M,N,0,0.1);
blur_gau_img = im2uint8(im2double(blur_img)+gau_noise);

%inverse filter result

G1 = fft2(double(blur_img));
G2 = fft2(double(blur_gau_img));
F1 = G1./H;
F2 = G2./H;
blur_inv_img = uint8(real(ifft2((abs(H)>0.01).*F1)));
blur_gau_inv_img = uint8(real(ifft2((abs(H)>0.01).*F2)));

% plot
set(gcf,'position',[0,0,800,600]);
subplot(2,2,1),imshow(blur_img);title('blur image');
subplot(2,2,2),imshow(blur_gau_img);title('add gaussian');
subplot(2,2,3),imshow(blur_inv_img);title('blur inverse');
subplot(2,2,4),imshow(blur_gau_inv_img);title('blur gaussian inverse');
```



Wiener Deconvolution Filter

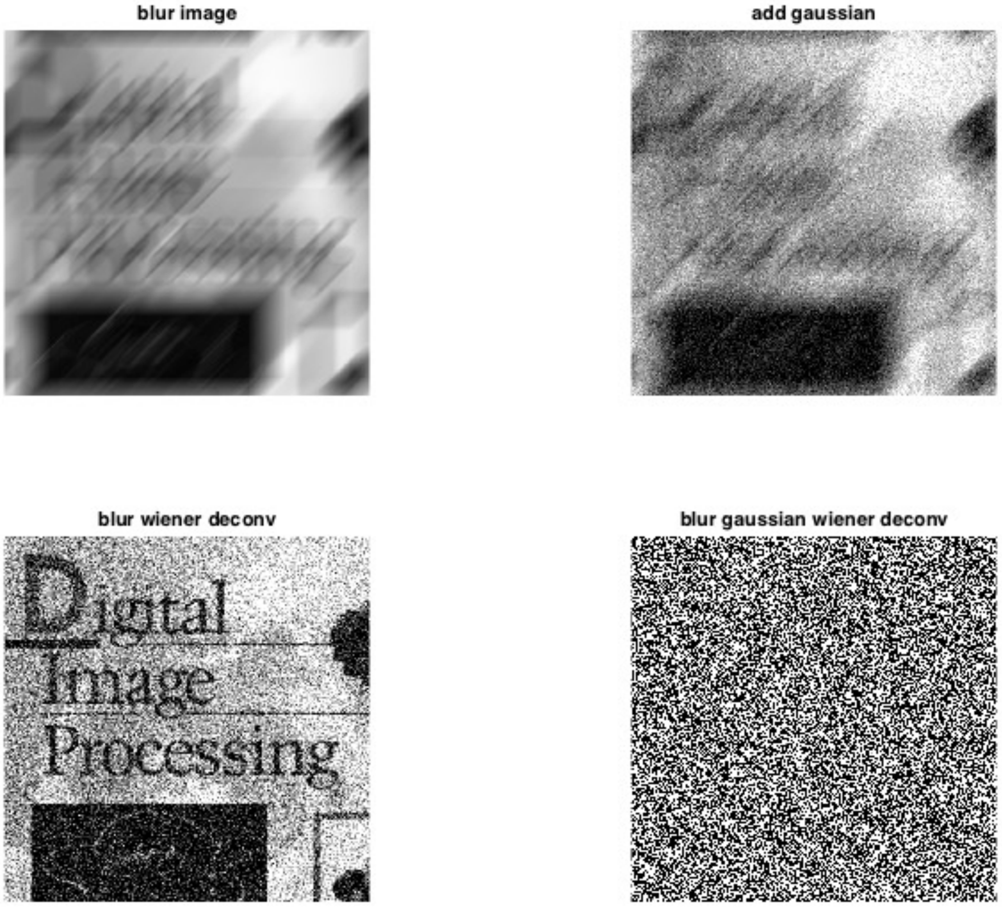
MATLAB

```matlab
orig_img = imread('book_cover.jpg');
[M,N] = size(orig_img);
% produce noise image
PSF = fspecial('motion',100,45);
blur_img = imfilter(orig_img,PSF,'circular');
gau_noise = imnoise2('gaussian',M,N,0,0.1);
blur_gau_img = im2uint8(im2double(blur_img)+gau_noise);

%wiener deconvolution filter result
sn = abs(fft2(gau_noise)).^2;
na = sum(sn(:))/numel(gau_noise);
sf = abs(fft2(orig_img)).^2;
fa = sum(sf(:))/numel(orig_img);
R = na/fa;

blur_wd_img = deconvwnr(blur_img,PSF,R);
blur_gau_wd_img = deconvwnr(blur_gau_img,PSF,R);

% plot
set(gcf,'position',[0,0,800,600]);
subplot(2,2,1),imshow(blur_img);title('blur image');
subplot(2,2,2),imshow(blur_gau_img);title('add gaussian');
subplot(2,2,3),imshow(blur_wd_img);title('blur wiener deconv');
subplot(2,2,4),imshow(blur_gau_wd_img);title('blur gaussian wiener deconv');
```



blur image

add gaussian

blur wiener deconv

blur gaussian wiener deconv

Wiener Parametric Filter

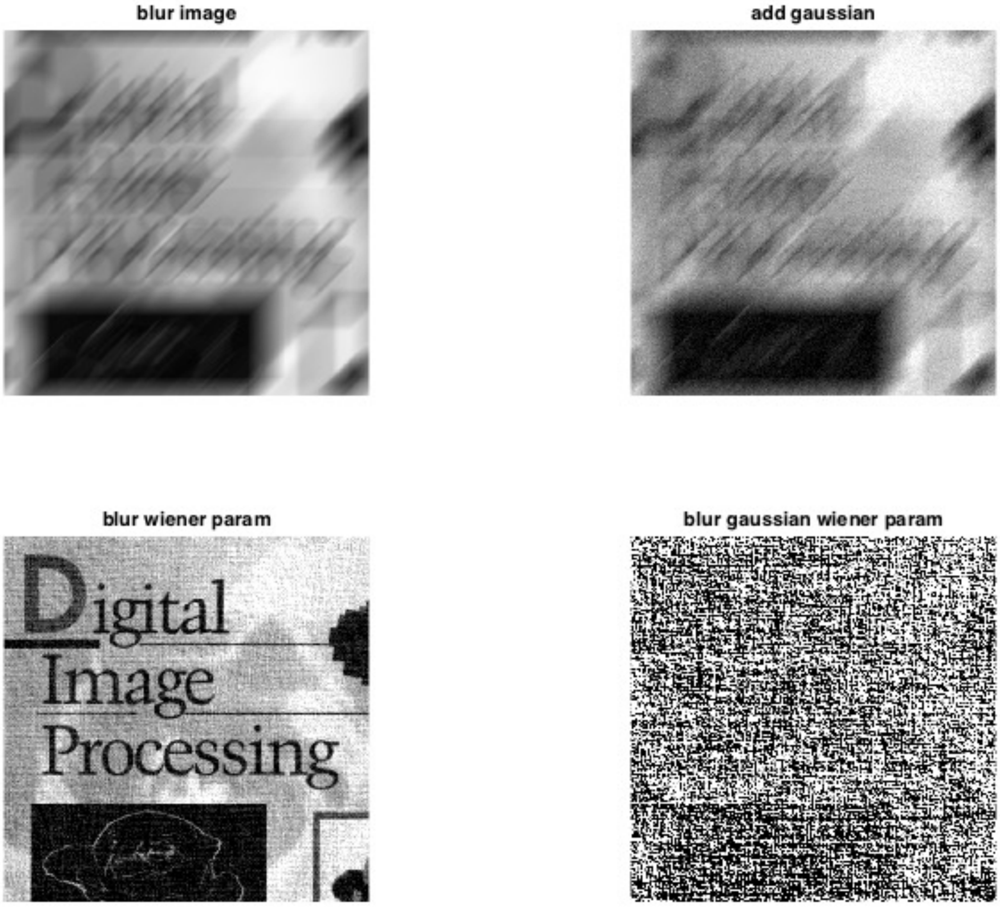
MATLAB

```
orig_img = imread('book_cover.jpg');
[M,N] = size(orig_img);
% produce noise image
PSF = fspecial('motion',100,45);
blur_img = imfilter(orig_img,PSF,'circular');
gau_noise = imnoise2('gaussian',M,N,0,sqrt(0.001));
blur_gau_img = im2uint8(im2double(blur_img) + gau_noise);

%wiener deconvolution filter result
sn = abs(fft2(gau_noise)).^2;
na = sum(sn(:))/numel(gau_noise);
sf = abs(fft2(orig_img)).^2;
fa = sum(sf(:))/numel(orig_img);
R = na/fa;

%wiener parametric filter result
ncorr = fftshift(real(ifft2(sn)));
icorr = fftshift(real(ifft2(sf)));
blur_wd_img = deconvwnr(blur_img,PSF,ncorr,icorr);
blur_gau_wd_img = deconvwnr(blur_gau_img,PSF,ncorr,icorr);

% plot
set(gcf,'position',[0,0,800,600]);
subplot(2,2,1),imshow(blur_img);title('blur image');
subplot(2,2,2),imshow(blur_gau_img);title('add gaussian');
subplot(2,2,3),imshow(blur_wd_img);title('blur wiener param');
subplot(2,2,4),imshow(blur_gau_wd_img);title('blur gaussian wiener param');
```



blur image

add gaussian

blur wiener param

blur gaussian wiener param