

## 数字图像处理 作业4 图像的复原和重建

徐达烽

16340260

1. Implement a blurring filter using the equation (5.6-11, 数字图像处理 (第三版) ) in textbook, and blur the test image 'book\_cover.jpg' using parameters  $a=b=0.1$  and  $T=1$ .

算法描述:

运动模糊的退化函数是:  $H(u,v) = T \text{sinc}(ua + vb) e^{-j\pi(ua+vb)}$ , 其中  $\text{sinc}(x) = \frac{\sin x}{\pi x}$

先将图像转换成[0,1]浮点图像 (方便进行滤波操作), 然后做傅里叶变换, 乘上退化函数, 再反变换即可得到模糊后的图像。

源代码:

```
im=im2double(imread('book_cover.jpg'));
im = im(:,:,1);
[m,n]=size(im);
%% degradation
fm = fftshift(fft2(im));
H = motionDegradation(m, n, 0.1, 0.1, 1);
g = real(ifft2(ifftshift(H .* fm)));
```

其中motionDegradation定义如下:

```
function H = motionDegradation(m, n, a, b, T)
    [u, v] = meshgrid((1:m)-m/2, (1:n)-n/2);
    H = T .* sinc(u*a+v*b) .* exp(-1i*pi*(u*a+v*b));
    H(isinf(H)) = 1;
    H(isnan(H)) = 1;
end
```

2. Add Gaussian noise of 0 mean and variance of 500 to the blurred image.

算法描述:

使用randn(m,n)函数产生标准正态分布的随机数, 然后放大 $\sigma = \sqrt{500}$  倍即可。由于步

骤1中, 我们对图像进行了归一化操作, 所以 $\sigma$ 也需要归一化, 应等于 $\frac{\sqrt{500}}{255}$

源代码:

```
sigma = sqrt(500/(255^2));
noise = sigma*randn(m,n);
```

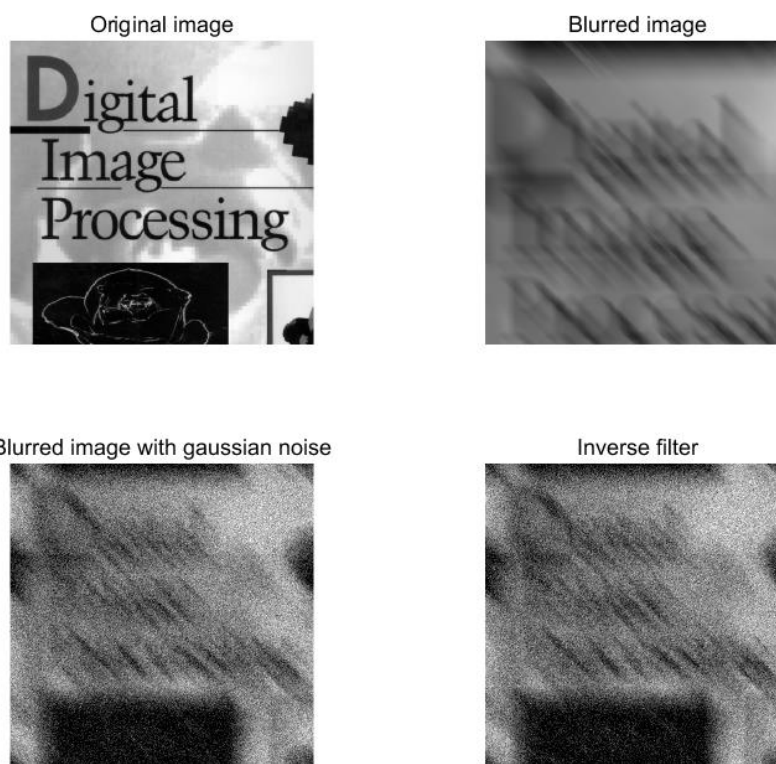
```
g_gau = g + noise;
g_gau(g_gau>1) = 1;
g_gau(g_gau<0) = 0;
```

3. Restore the blurred image and the blurred noisy image using the inverse filter.

算法描述：对退化后的图像进行傅里叶逆变换，然后除以步骤1中的退化函数即可。注意，当H取值很小时，需要做截止的处理，避免病态问题。

源代码：

步骤1-3实验结果图像：



可见，直接逆变换的效果非常差。

4. Restore the blurred noisy image using the parametric Wiener filter with at least 3 different parameters, and compare and analyse results with that of 3.

算法描述：

$$\hat{F}(u, v) = \left[ \frac{1}{H(u, v)} \frac{|H(u, v)|^2}{|H(u, v)|^2 + k} \right] G(u, v)$$

维纳滤波的定义是

根据定义，对不同的参数 k，生成不同的滤波器，并依次实验即可。

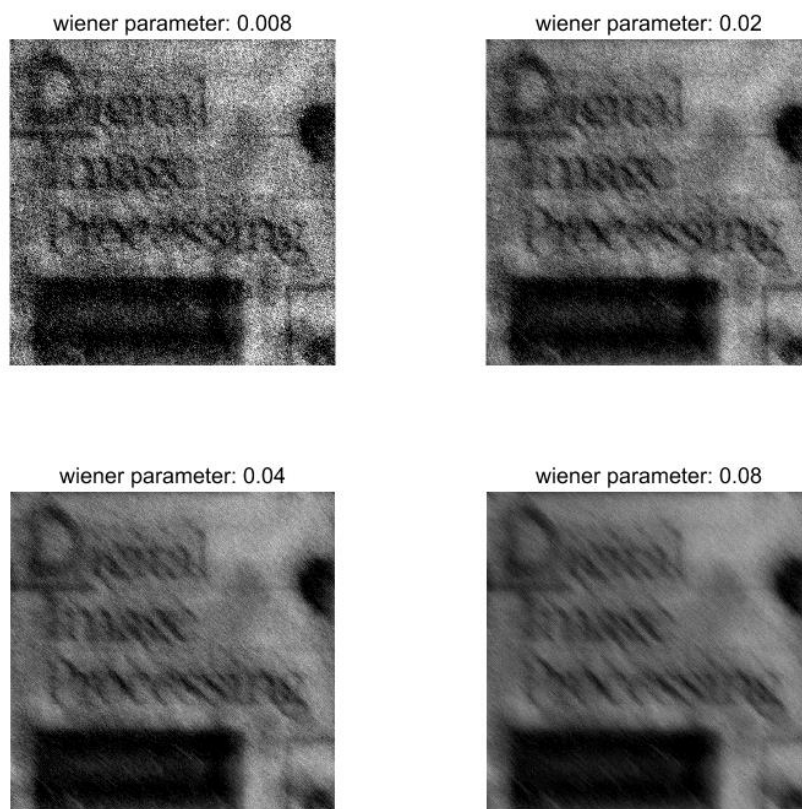
源代码:

```
wiener_parameters = [0.008, 0.02, 0.04, 0.08];  
i=1;  
figure(3);  
for para = wiener_parameters  
    rst = wienerFilter(H, para);  
    im_res = real(ifft2(ifftshift(f_gau .* rst)));  
    subplot(2,2,i); imshow(im_res);  
    title(['wiener parameter: ', num2str(para)]);  
    i = i+1;  
end
```

其中, 函数 wienerFilter 定义如下:

```
function [rst] = wienerFilter(H, k)  
    rst = H .^ (-1) .* (abs(H) .^2 ./ (abs(H) .^2 +  
k));  
end
```

实验结果图像:



由图可知, 参数  $k$  取 0.02 附近时效果最佳。