小作业 4

药 3 李宇晖 2013012470

实验目的

1） 运用 matlab 进行图片的模糊、加性高斯噪声处理

2） 了解图像恢复的基本原理并用 matlab 进行维纳滤波和逆滤波

实验过程

使用附件汽车图片作为例子。仿真两种运动模糊：1）汽车相对于相机向左运动导致的前景（即汽车）模糊；2）相机相对于背景向左运动导致的背景模糊。并施加不同程度（方差从小到大）的加性高斯噪声。采用维纳滤波和逆滤波进行图像恢复，比较它们的性能。

提示：

1） 可使用 MATLAB 工具箱。

2） 手动标注前景。

为了手动标注前景，首先写出 Get_FrontGround()函数，利用 ginput 和 roipoly()函数手工标注前景，从而可以得到前景的二值图。代码如下：

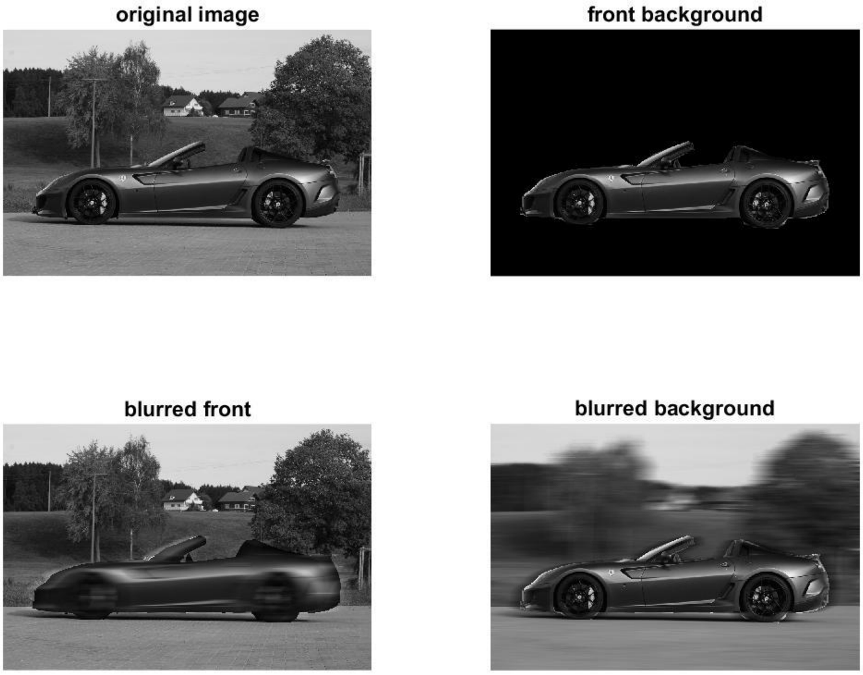
```
function [ binary_image ] = Get_FrontGround( image )
    close all;
    imshow(image);
    [x,y] = ginput;
    binary_image = roipoly(image,x,y);
    imwrite(binary_image,'binary.bmp');
    binary_image = 1 - binary_image;
    imwrite(binary_image,'without_front.bmp');
end
```

而后，采用 fspecial()函数和 imfilter()函数，可以达到运动造成的模糊。代码如下：

```
I = imread('car.bmp');
[m,n,c] = size(I);
if c==3,
    I = rgb2gray(I); %promise changing to gray
end
binary_image = Get_FrontGround(I); %get binary front image
%binary_image = imread('without_front.bmp')/255;
front_image = uint8(I) .* uint8(1-binary_image);
background_image = uint8(I).* uint8(binary_image);
H_front = fspecial('motion',70,180);    %blur
MotionBlur_front = imfilter(front_image,H_front,'replicate');    %filtering
MotionBlur_front = MotionBlur_front + background_image;
figure(1),clf;
subplot(2,2,1),imshow(I),title('original image');
subplot(2,2,2),imshow(front_image),title('front background');
subplot(2,2,3),imshow(MotionBlur_front,[]),title('blurred front');
H_back = fspecial('motion',70,0); %blur, the opposite orientation
```

```
MotionBlur_back = imfilter(background_image, H_back, 'replicate');
MotionBlur_back = MotionBlur_back + front_image;
subplot(2,2,4), imshow(MotionBlur_back,[]),title('blurred background');
saveas(figure(1),'blur.jpg');
close all;
```

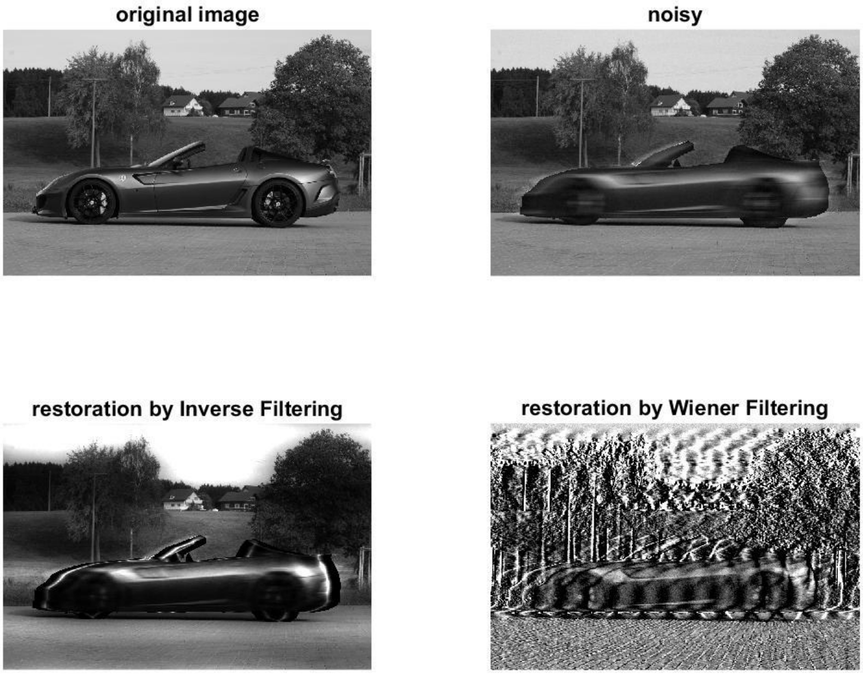
结果如下：



之后使用 matlab 自带的 imnoise()函数加入均值为 0，反差分别为 0.0001,0.001,0.01,0.1 的高斯噪声，使用盲目去卷积函数 deconvblind()进行逆滤波，并使用 deconvwnr()函数进行维纳滤波。代码如下：

```
noise_mean = 0;
noise_var_list = [0.0001;0.001;0.01;0.1];
I = im2double(I);
for k = 1:4,
    noise_var = noise_var_list(k);
    blurred_noisy = imnoise(MotionBlur_front, 'gaussian',noise_mean, noise_var);%gaussian noise blur
    noisy = blurred_noisy - MotionBlur_front;%get pure noise
    deblur = deconvblind(blurred_noisy,noisy);%inverse filtering
    front_image = uint8(deblur) .* uint8(1-binary_image);
    background_image = uint8(deblur).* uint8(binary_image);
    front = deconvblind(front_image,H_front);%inverse filtering
    final_image = front + background_image;%final image
    figure(1),clf;
    subplot(2,2,1),imshow(I),title('original image');
    subplot(2,2,2),imshow(blurred_noisy),title('noisy');
    subplot(2,2,3),imshow(final_image),title('restoration by Inverse Filtering');
    estimated_nsr= noise_var/ var(I(:));
    front = deconvwnr(blurred_noisy,H_front, estimated_nsr);%winer filtering
    subplot(2,2,4),imshow(front),title('restoration by Wiener Filtering');
    saveas(figure(1),[num2str(k) 'front.jpg']);
    close all;
    figure(1),clf;
    blurred_noisy = imnoise(MotionBlur_back, 'gaussian',noise_mean, noise_var); %gaussian noise blur
    noisy = blurred_noisy - MotionBlur_back; %get pure noise
    deblur = deconvblind(blurred_noisy,noisy);%inverse filtering
    front_image = uint8(deblur) .* uint8(1-binary_image);
    background_image = uint8(deblur).* uint8(binary_image);
    back = deconvblind(background_image,H_back);%inverse filtering
    final_image = back + front_image;%final image
    figure(1),clf;
    subplot(2,2,1),imshow(I),title('original image');
    subplot(2,2,2),imshow(blurred_noisy),title('noisy');
    subplot(2,2,3),imshow(final_image),title('restoration by Inverse Filtering');
    front = deconvwnr(blurred_noisy,H_back, estimated_nsr); %winer filtering
    subplot(2,2,4),imshow(front),title('restoration by Wiener Filtering');
    saveas(figure(1),[num2str(k) 'back.jpg']);
    close all;
end
```

结果如下：

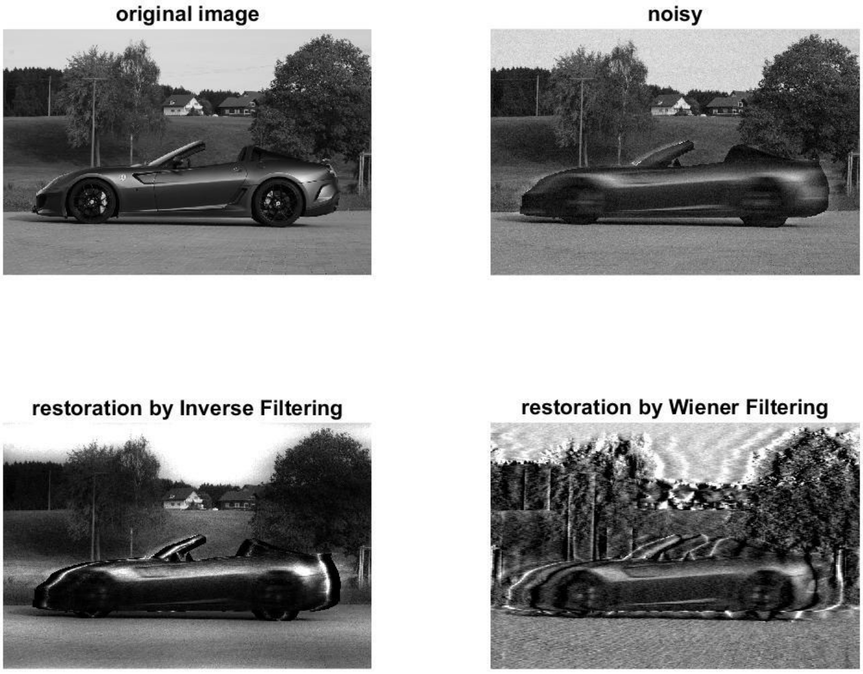


original image

noisy

restoration by Inverse Filtering

restoration by Wiener Filtering

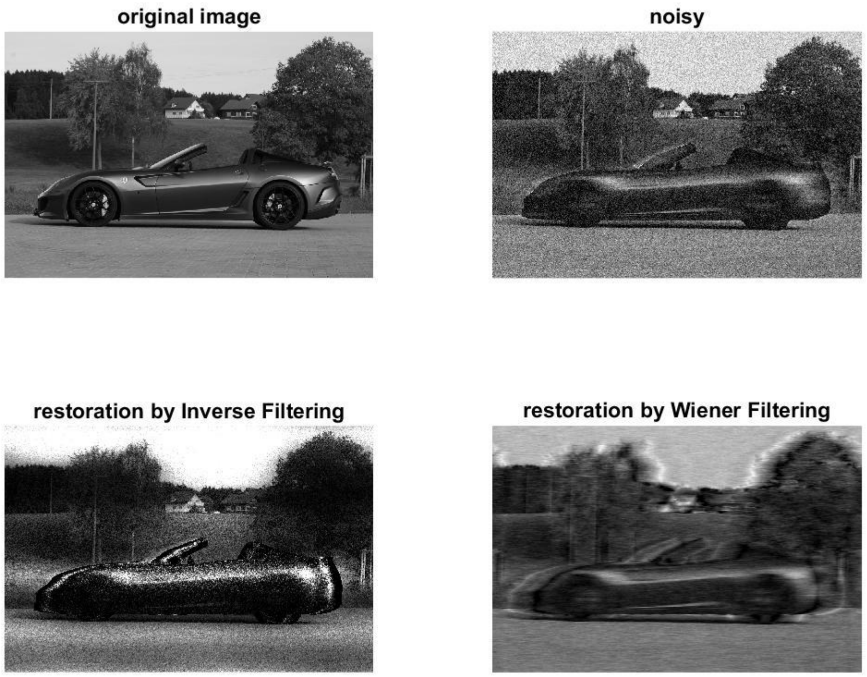


original image

noisy

restoration by Inverse Filtering

restoration by Wiener Filtering

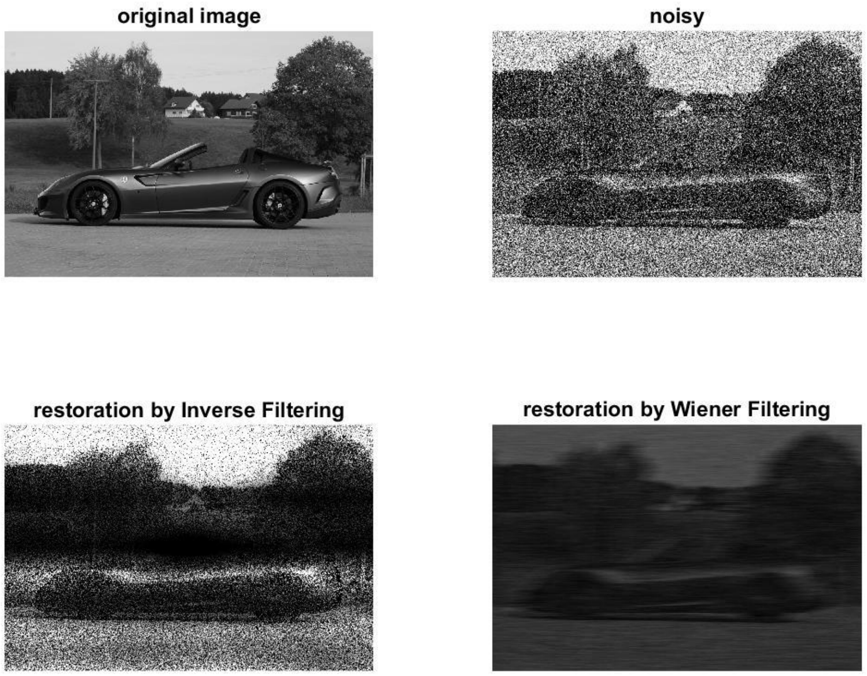


original image

noisy

restoration by Inverse Filtering

restoration by Wiener Filtering

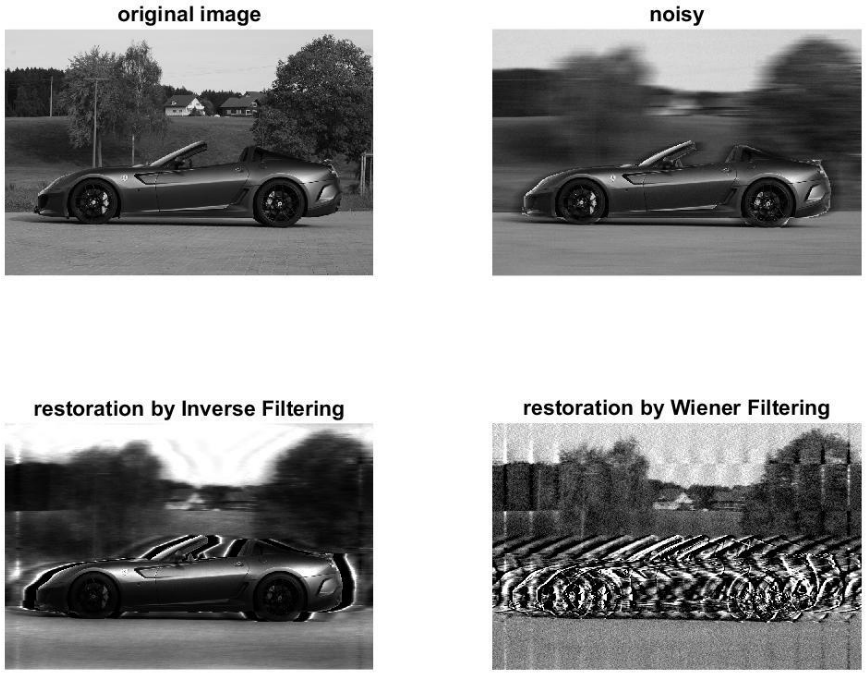


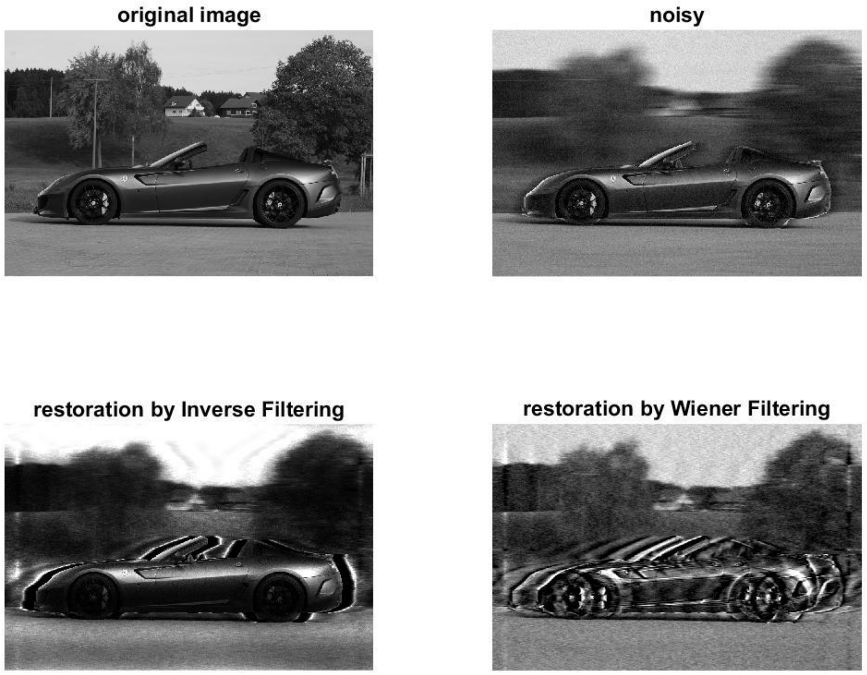
original image

noisy

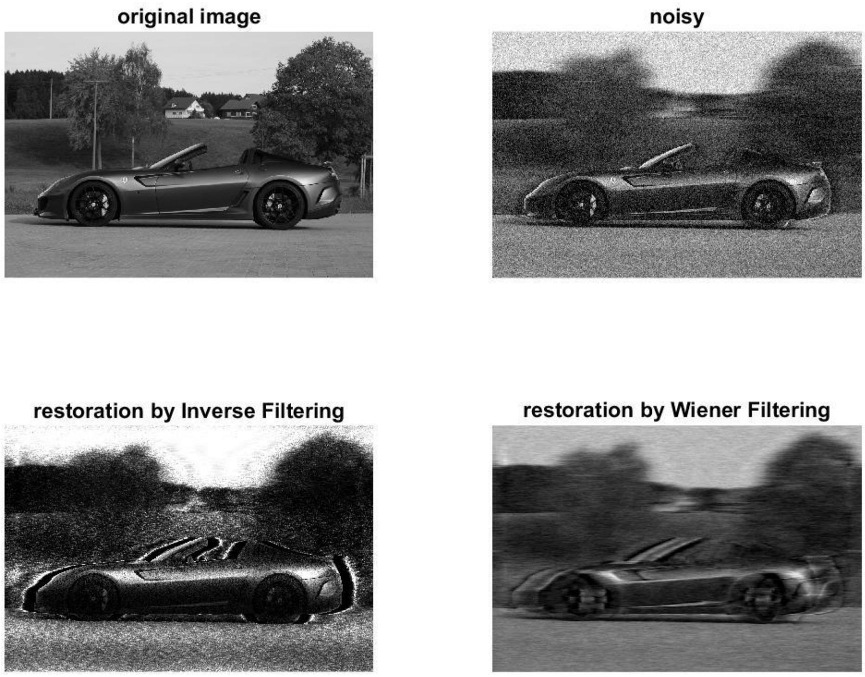
restoration by Inverse Filtering

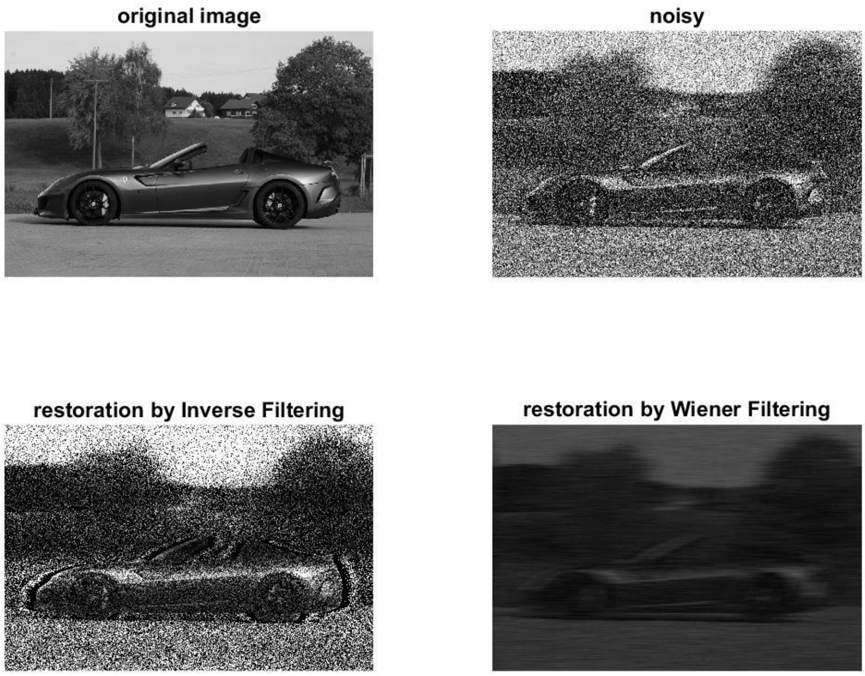
restoration by Wiener Filtering


original image
noisy
restoration by Inverse Filtering
restoration by Wiener Filtering


original image
noisy
restoration by Inverse Filtering
restoration by Wiener Filtering

可以看出，当加性高斯噪声方差较小时，使用逆滤波就可以达到图像复原的效果。而当加性高斯噪声方差较大时，逆滤波必须循环多次，复原的效果也并不好。而维纳滤波此时可以达


original image
noisy
restoration by Inverse Filtering
restoration by Wiener Filtering


original image
noisy
restoration by Inverse Filtering
restoration by Wiener Filtering

到更好的效果。