

## 建模估计——由运动引起的图像模糊

在实际生活中，我们常常会遇到当被摄物体与相机之间存在相对运动的情形，由于相机拍摄存在曝光时间，所以拍摄出来的图像往往会产生一些模糊。这里我们尝试模拟这种模糊。

在实际情况中，这种模糊可以看作是一种积分运算。假设图像 $f(x, y)$ 进行平面运动， $x_0(t)$ 和 $y_0(t)$

分别是在 $x$ 和 $y$ 方向上随时间变化的分量。成像 $g(x, y)$ 则可以由以下式子得到：

$$g(x, y) = \int_0^t f[x - x_0(t), y - y_0(t)] dt$$

如果在频域中处理，则为上式的傅里叶变换，

$$G(u, v) = \int_0^T F(u, v) e^{-j2\pi[ux_0(t) + vy_0(t)]} dt$$

那么，退化函数变为

$$H(u, v) = \int_0^t e^{-j2\pi[ux_0(t) + vy_0(t)]} dt$$

将运动分解成 $x, y$ 方向， $x_0(t) = \frac{at}{T}$ 以及 $y_0(t) = \frac{bt}{T}$

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

我们的方案是：先进行傅里叶变换，在乘以 $H(u, v)$ 函数，最后对结果进行反变换得到变换后的图像。

```
1 for u=1:h
2     for v =1:w
3         H(u,v) = (T./(pi*(u*a+v*b))).*sin(pi*(u*a+v*b)).*exp(-1i*pi*
(u*a+v*b));
4         Gi(u,v)=H(u,v).*Fi(u,v);
5     end
6 end
```

## 逆滤波

原始图像的傅里叶变换可由下式得到： $\hat{F}(u, v) = \frac{G(u, v)}{H(u, v)}$

将 $G(u, v) = F(u, v)H(u, v) + N(u, v)$ 代入

$$\hat{F}(u, v) = F(u, v) + \frac{N(u, v)}{H(u, v)}$$

实际上我们不可能知道 $N(u, v)$

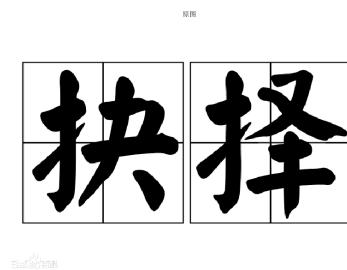
所以在当我们不知道噪声信息时，如果进行全域滤波，则效果很差。

```

1  a = 0.1; %x 方向上的秒速度
2  b = 0.1; %y方向上的秒速度
3  T = 1; % 快门时间
4
5  for u=1:h
6      for v =1:w
7          A =pi.*(u.*a+v.*b);
8          H(u,v) = (T./A).*sin(A).*exp(-1i.*A);%退化函数
9      end
10 end
11 %频域退化函数

```

运动模糊处理图像如下



```

1  noise = imnoise(zeros(h,w),'gaussian',0,0.00005);%gaussian noise
2  Fnoise = fft2(noise);%加性噪声谱
3  Gnoise = Gi+Fnoise;%噪声图像谱

```

高斯噪声图像:

gaussian



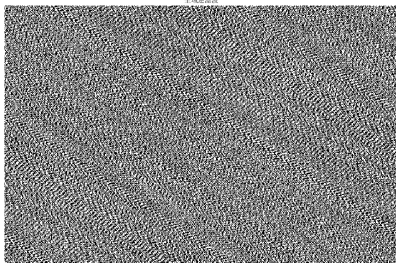
逆滤波图像：左图1 是进行全域逆滤波，可见此种方法对有噪声的图像并没有办法很好的进行重建。  
右图是减去了噪声频谱进行的滤波。

```

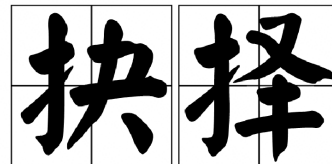
1  Fhat = Gnoise ./H; %全域逆滤波
2  fhat = ifft2(Fhat);
3  figure,subplot(1,2,1),imshow(uint8(fhat)),title("全域逆滤波 ");
4
5  Fhat2 =(Gnoise -Fnoise)./H; %噪声图像谱-噪声谱
6  fhat2 = ifft2(Fhat2);
7  subplot(1,2,2),imshow(uint8(fhat2)),title("在已知噪声谱的情况下逆滤波");

```

全域逆滤波



在已知噪声谱的情况下逆滤波



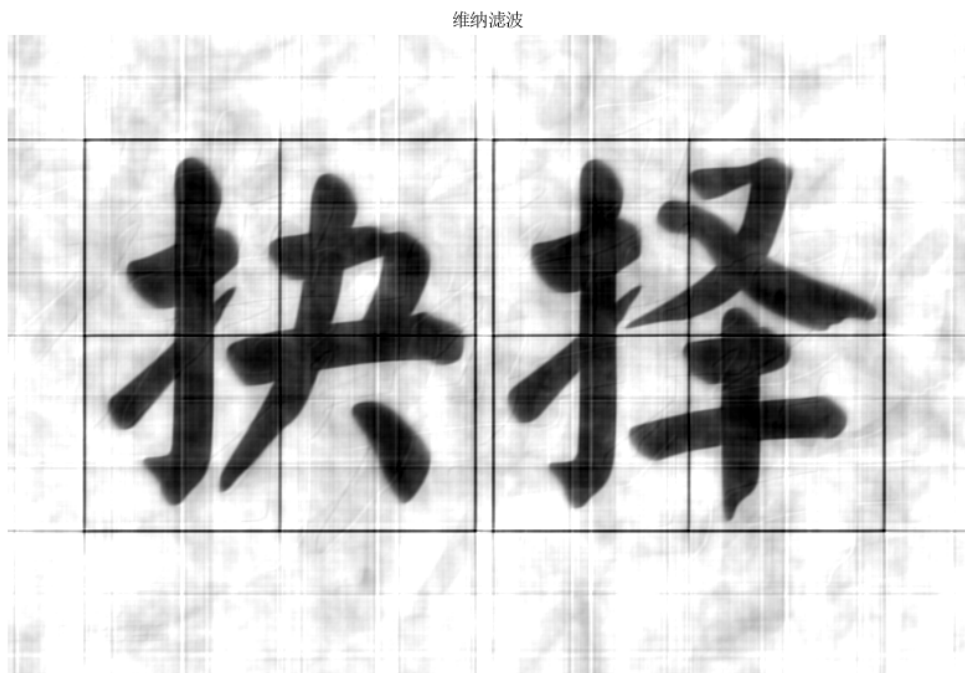
维纳滤波

维纳滤波

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

k 是一个估计值,  $k = \frac{|N(u, v)|^2}{|F(u, v)|^2}$  由于噪声的功率谱和原图的功率谱并不清楚, 所以k只能根据经验选择一些经验值。

```
1 %维纳滤波
2 k=(Fnoise.*conj(Fnoise))./(Gi.*conj(Gi)); %我们已知噪声和原图像
3 w = (1./H) .* (H.*conj(H))./(H.*conj(H)+k);
4 Fhat3 = Gnoise .*w; %维纳逆滤波
5 fhat3 = ifft2(Fhat3);
6 figure,imshow(uint8(fhat3)),title("维纳滤波 ");
```



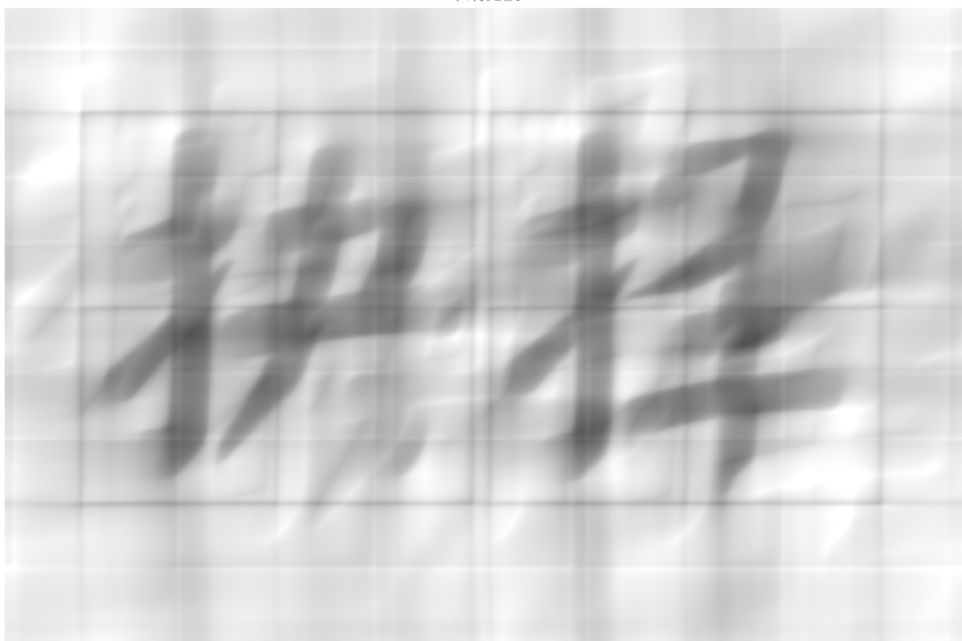
当k是一个估计值 0.05时, 复原图像如下:

维纳滤波



当 $k$ 是一个估计值 0.005时，复原图像如下：

维纳滤波

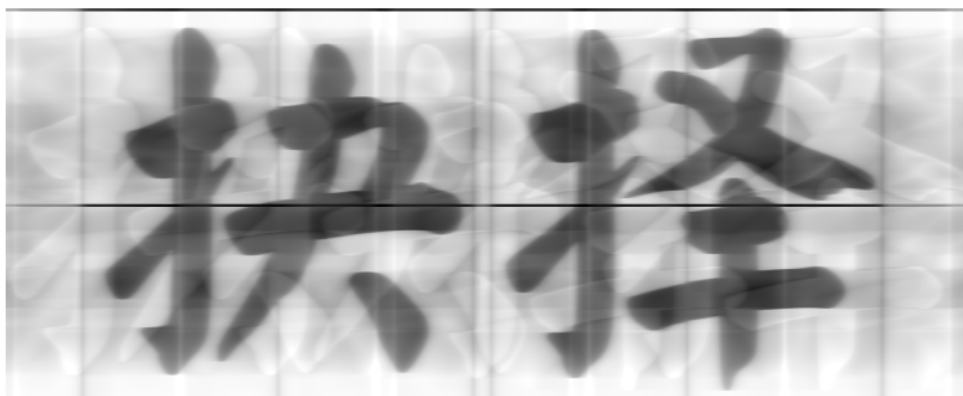


以上均为噪声方差为0.00005时的情况。

在噪声方差提高至0.005之后，

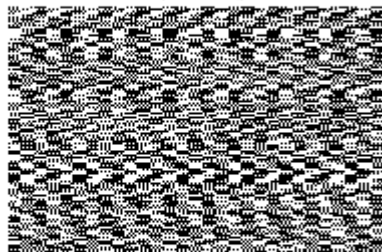
维纳滤波如下：

维纳滤波

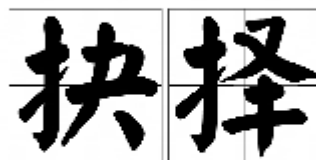


全域滤波依旧难以解决噪声问题。

全域逆滤波



在已知噪声谱的情况下逆滤波



4.0.0.0