

厦大数字图像处理期末复习中

原创

Karon_NeverAlone 于 2022-01-10 20:34:09 发布

版权

阅读量1.8k 收藏 6 点赞数 1

分类专栏: 数字图像处理 文章标签: 图像处理



数字图像处理 专栏收录该内容

6 订阅

10 篇文章

订阅专栏

内容概括

5.图像复原

基本模型

比较认真的讲了噪声的模型，前几个单元都只是说了 滤波器

全逆滤波

维纳滤波

5.图像复原

图像退化/恢复过程的模型•噪声模型•仅存在噪声的恢复-空间滤波•通过频域滤波周期性降噪•线性位置不变退化•估计退化 函数 •逆滤波•最小均方误差(Wiener)滤波•约束最小二乘滤波

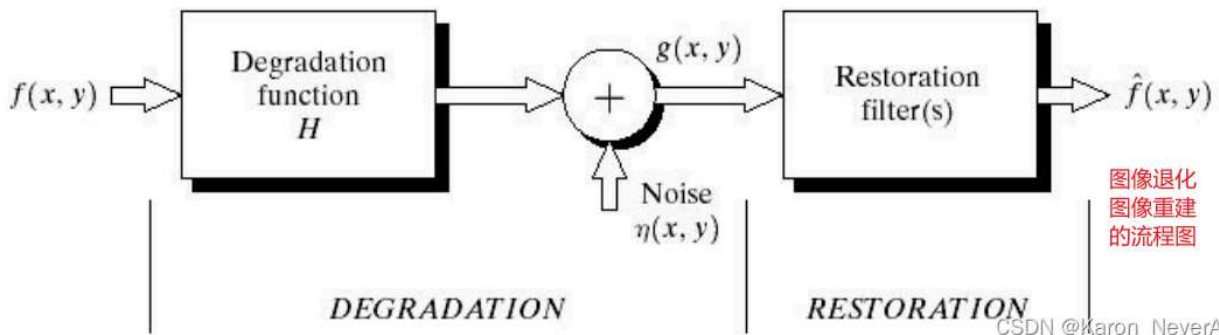
恢复技术主要是对退化过程进行建模，并应用逆过程恢复原始图像，试图利用退化现象的先验知识来重建或恢复已退化的图像

或者你可以理解为一方面是对传输过程中失真的照片的恢复，或者是对主观图像增强后的照片的复原

--基本退化模型

是什么让图像退化？噪声和退化函数

噪声的来源？图像采集 的过程中和图像传输的过程中产生



空域上的退化模型

$$g(x, y) = h(x, y) * f(x, y) + \eta(x, y)$$

可以视为是一个卷积操作

The model in frequency domain

频域上的退化模型

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

CSDN @Karon_NeverAlone

---噪声模型

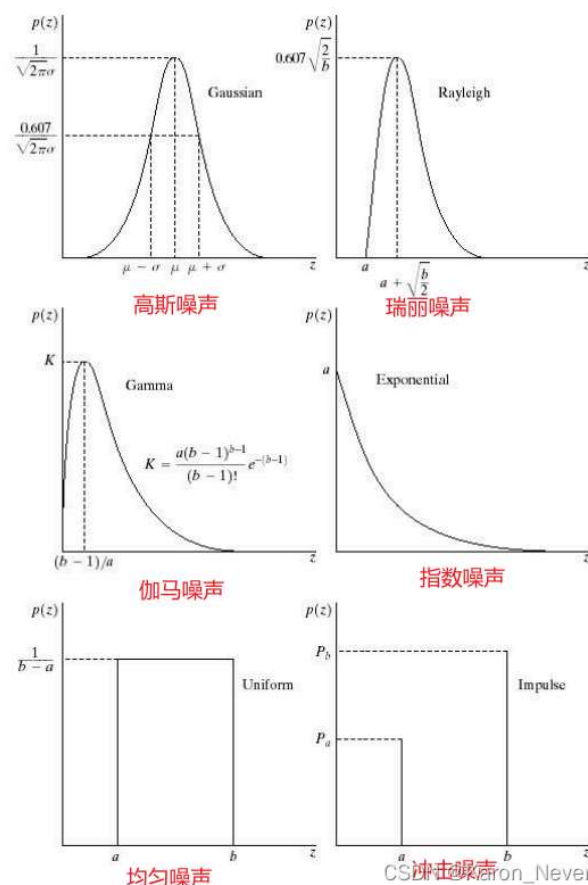
白噪声： 傅里叶变换后频谱是常数

空间周期噪声： 有周期性的噪声，来自电气或机电干扰

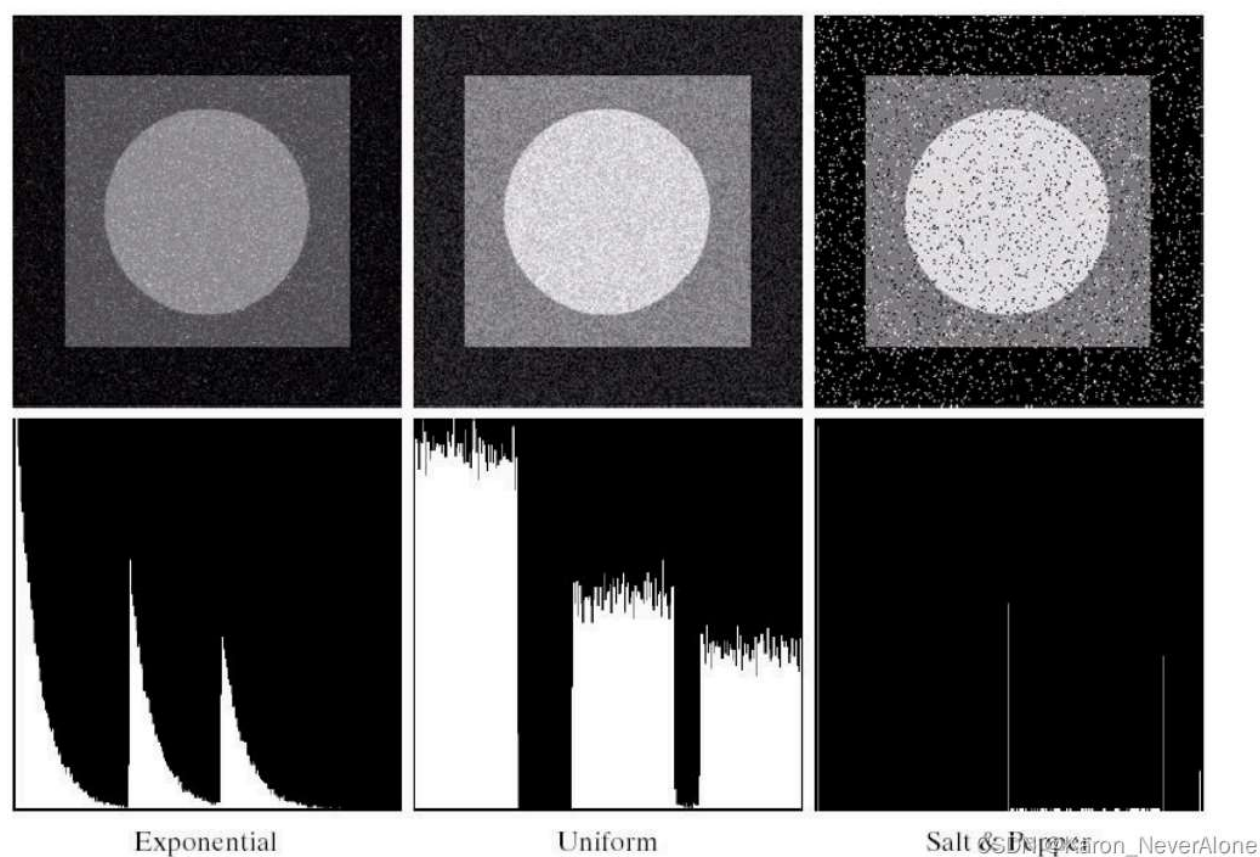
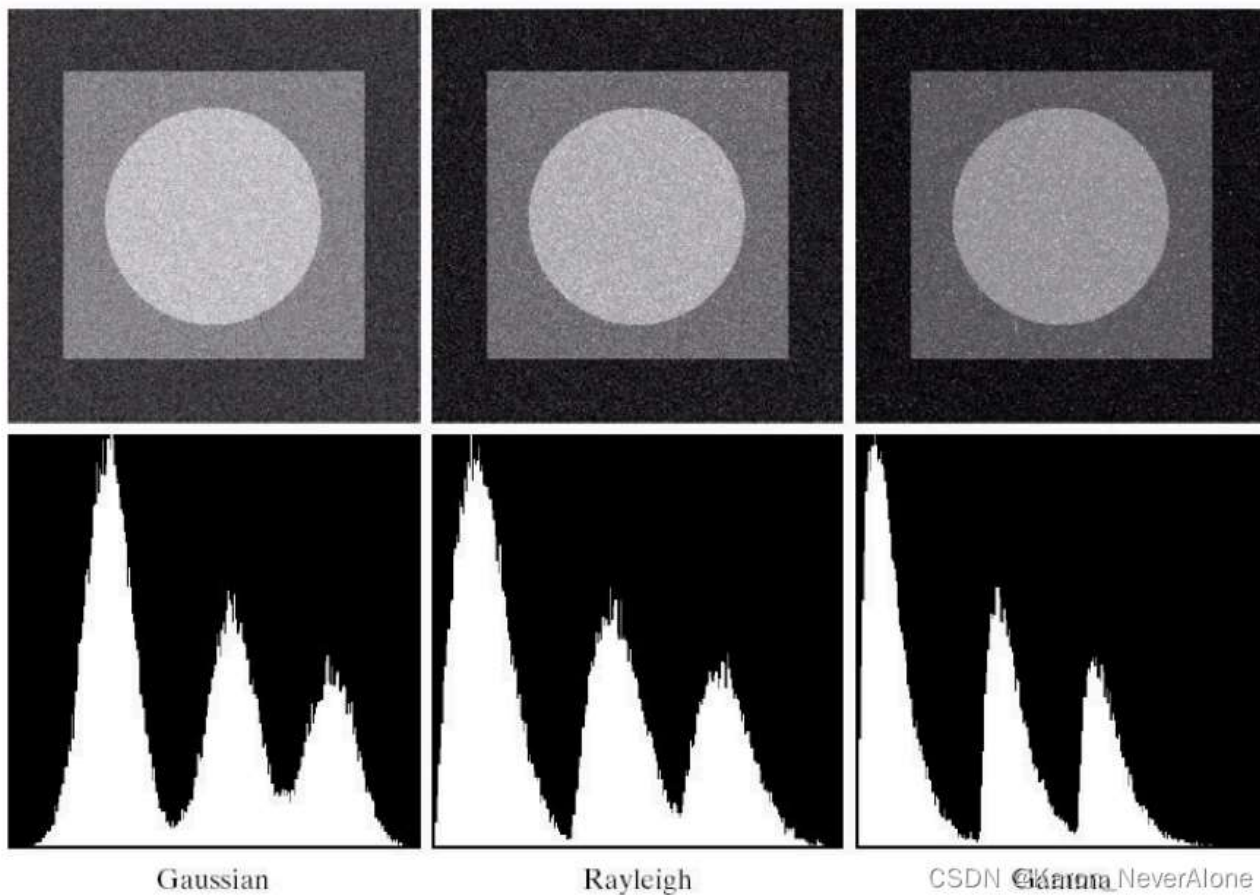
其他常见噪声：

Some important noise PDFs

- Gaussian noise
- Rayleigh noise
- Erlang (Gamma) noise
- Exponential noise
- Uniform noise
- Impulse (salt-and-pepper) noise



要能看出来是什么噪声



---只存在噪声的退化模型

当退化图像只是因为噪声而和原图有了差别，那就可以想办法去掉噪声，滤波是首选，问题转化为图像去噪！

•平均滤波器-算术平均滤波器-几何平均滤波器（适合随机噪声）-谐波平均滤波器-反谐波平均滤波器（适合椒盐噪声）

算术均值滤波器 Arithmetic Mean Filter 这是线性的

这是最简单的均值滤波器，可以去掉均匀噪声和高斯噪声，但对图像造成一定程度的模糊。

令 $S_x y$ 表示中心点在 (x, y) 处，大小为 $m \times n$ 的滤波器窗口。算术均值滤波器就是简单的计算窗口区域的像素均值，然后将均值赋值给窗口中心点处的像素：

$$f(x, y) = \frac{1}{mn} \sum_{(s, t) \in S_{xy}} g(s, t)$$

其中， $g(s, t)$ 表示原始图像， $f(x, y)$ 表示均值滤波后得到的图像。

基于上述公式，可以很容易的得到算术均值滤波器的窗口模板，下面以 3×3 为例

$$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

CSDN @Karon_NeverAlone

其他的一些非线性均值滤波器

除了上述算术均值滤波器，根据计算均值方法的不同还有几种均值滤波。

几何均值滤波器 Geometric Mean Filter 其公式如下：

$$f(x, y) = \left[\prod_{(s, t) \in S_{xy}} g(s, t) \right]^{\frac{1}{mn}}$$

滤波后图像的像素由模板窗口内像素的乘积的 $\frac{1}{mn}$ 幂给出。和算术均值滤波器相比，几何均值滤波器能够更好的取出高斯噪声，并且能够更多的保留图像的边缘信息。但，其对0值是非常敏感的，在滤波器的窗口内只要有一个像素的灰度值为0，就会造成滤波器的输出结果为0。

谐波均值滤波器 Harmonic Mean Filter 其公式如下：

$$f(x, y) = \frac{mn}{\sum_{(s, t) \in S_{xy}} \frac{1}{g(s, t)}}$$

谐波均值滤波器对盐粒噪声（白噪声）效果较好，不适用于胡椒噪声；比较适合处理高斯噪声。

逆谐波均值滤波器 Contra-Harmonic Mean Filter 其公式如下：

$$f(x, y) = \frac{\sum_{(s, t) \in S_{xy}} g(s, t)^{Q+1}}{\sum_{(s, t) \in S_{xy}} g(s, t)^Q}$$

其中Q称为滤波器的阶数，该滤波器可以用来消除椒盐噪声。但是需要不同同时处理盐粒噪声和胡椒噪声，当Q为正时，可以消除胡椒噪声；当Q为负时，消除盐粒噪声。当Q=0时，该滤波器退化为算术均值滤波器；Q=-1时，退化为谐波均值滤波器。

CSDN @Karon_NeverAlone

•顺序统计滤波器-中值滤波器-最大值和最小滤波器

这个也是非线性的，比较好理解

中值可以比较好的去除椒盐噪声

最大值可以去掉黑点——即pepper噪声，最小值可以去掉白点即——salt噪声

•自适应滤波器-自适应局部（均值）滤波器-自适应中值滤波器

滤波器在滤波的过程中根据实际需要改变策略，滤波性能 更好但是计算代价更高

•自适应局部滤波器

处理(x,y)的时候每次选取一个小区域

$$\hat{f}(x, y) = g(x, y) - \frac{\sigma_{\eta}^2}{\sigma_L^2} [g(x, y) - m_L]$$

噪声的方差，这个需要估计

小区域的方差

小区域的均值

CSDN @Karon_NeverAlone

那么根据局部像素块的不同，就有三个结果：

01 噪声的方差为0，根本没有噪声，那么这个式子也就返回 $g(x,y)$

02 区域的方差和噪声的方差相同，说明区域的方差是由图像本身的**细节**造成的，那么做均值滤波就好了，也就是返回这个小区域的均值**ML**

03 其他情况，说明区域的方差是由噪声引起的，则根据公式返回一个 $g(x,y)$ 的近似值，（一般认为噪声的方差是比区域方差小的）

优势：去噪干净而且细节保留完好，因为细节处的区域方差会很大

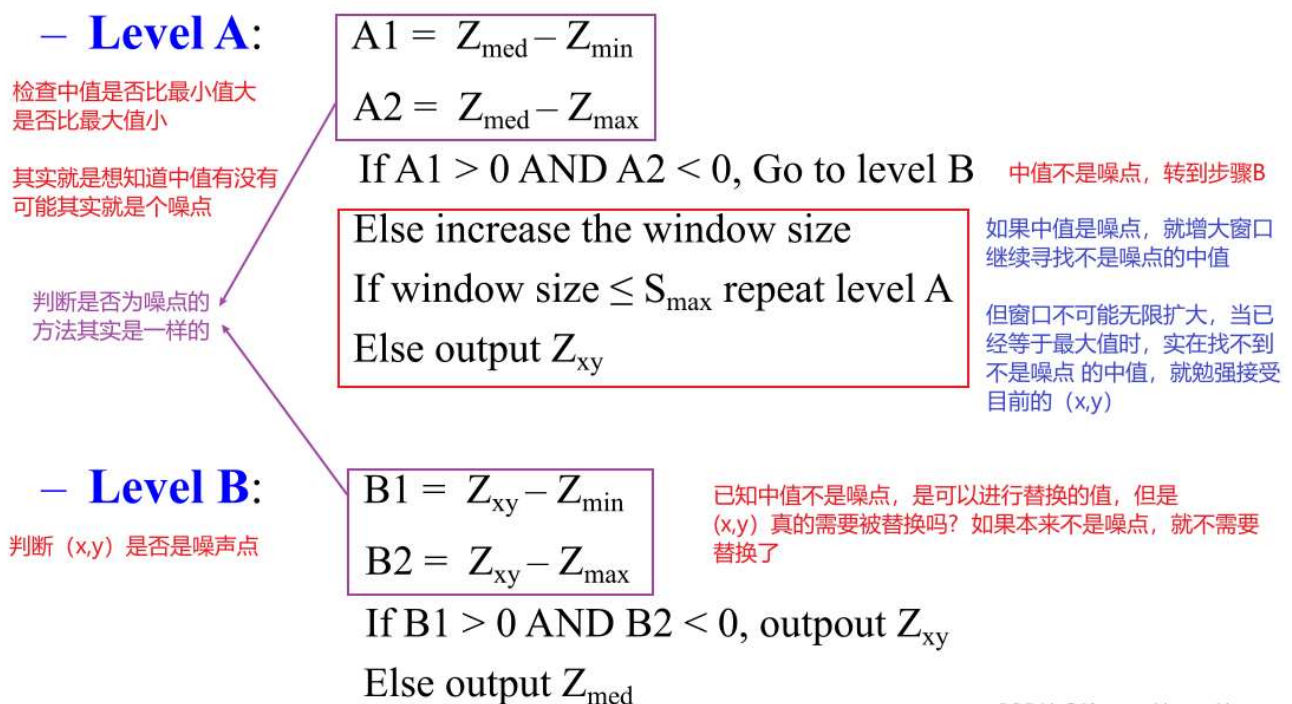
·自适应中值滤波器——此处有实验哦！！

自适应中值和中值滤波的区别：中值滤波并没有考虑当前像素是不是噪声，就直接换成了滤波器的中值（而且噪声密度很大的时候，你也不清楚中值是不是噪点），万一原来并非噪声点，这样操作反而是画蛇添足，自适应就是加入了**判断当前像素是否是噪声点的算法**

一些参数说明

- Z_{\min} = minimum gray level value in S_{xy} Sxy区域内的最小值
- Z_{\max} = maximum gray level value in S_{xy} Sxy区域内的最大值
- Z_{med} = median of gray level in S_{xy} Sxy区域内的中值
- Z_{xy} = gray level at coordinates (x,y) 坐标(x,y)的灰度值
- S_{\max} = maximum allowed size of S_{xy} 最大允许尺度

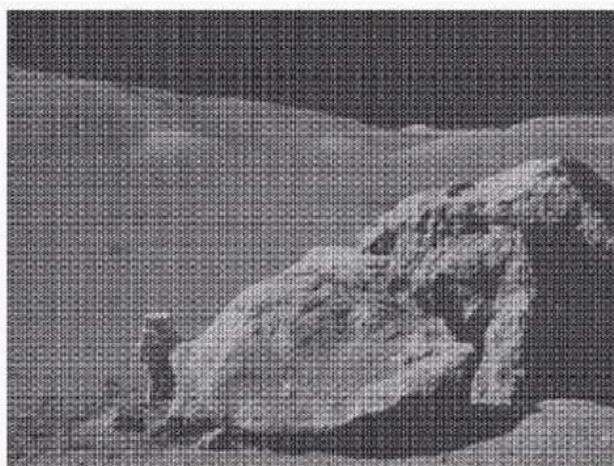
CSDN @Karon_NeverAlone



CSDN @Karon_NeverAlone

--处理空间周期噪声

空间周期性噪声傅里叶之后，是右图



如果先去除该噪声，应该让这个圈上的分量被阻，其他部分分量通过

即**带通、带阻、陷波滤波器**

带阻滤波器：那三个经典的滤波器的带阻滤波器

• Bandreject filters W指的是带宽

Ideal

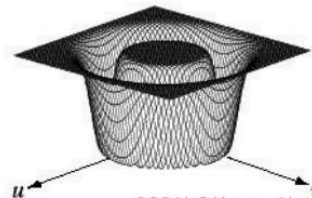
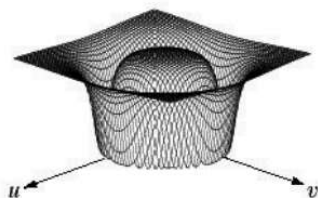
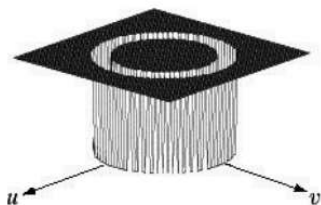
$$H(u, v) = \begin{cases} 1, & \text{if } D(u, v) < D_0 - \frac{W}{2} \\ 0, & \text{if } D_0 - \frac{W}{2} \leq D(u, v) \leq D_0 + \frac{W}{2} \\ 1, & \text{if } D(u, v) > D_0 + \frac{W}{2} \end{cases}$$

Butterworth

$$H(u, v) = \frac{1}{1 + \left[\frac{D(u, v)W}{D_0^2 - D(u, v)^2} \right]^{2n}}$$

Gaussian

$$H(u, v) = 1 - e^{-\frac{1}{2} \left[\frac{D^2(u, v) - D_0^2}{D(u, v)W} \right]^2}$$



CSDN @Karon_NeverAlone

带通滤波器：用1-带阻滤波器就好啦

陷波滤波器

就是挖洞的滤波器，主要是用在不对称（不能形成圆形这种规则图形）的时候

Ideal notch filter
$$H(u, v) = \begin{cases} 0, & \text{if } D_1(u, v) \leq D_0 \text{ or } D_2(u, v) \leq D_0 \\ 1, & \text{otherwise} \end{cases} \quad (189)$$

Where

$$D_1(u, v) = [(u - M/2 - u_0)^2 + (v - N/2 - v_0)^2]^{1/2} \quad (190)$$

and

$$D_2(u, v) = [(u - M/2 + u_0)^2 + (v - N/2 + v_0)^2]^{1/2} \quad (191)$$

Butterworth notch filter
$$H(u, v) = \frac{1}{1 + [\frac{D_0^2}{D_1(u, v)D_2(u, v)}]^n}$$

Gaussian notch filter
$$H(u, v) = 1 - e^{-\frac{1}{2}[\frac{D_1(u, v)D_2(u, v)}{D_0^2}]}$$

CSDN @Karon_NeverAlone

--基本退化模型的图像复原

主要是要得出 $H(u, v)$: 三个方法 观察、实验、数学建模

空域上的退化模型

$$g(x, y) = h(x, y) * f(x, y) + \eta(x, y)$$

可以视为是一个卷积操作

The model in frequency domain

频域上的退化模型

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

CSDN @Karon_NeverAlone

(1) 观察法

因为局部的 H 和全局的 H 应该是一样的，这样就能知道全局的 H

$$H_s(u, v) = \frac{G_s(u, v)}{\bar{F}_s(u, v)}$$

取一小块区域

预估出来的退化函数

手动构建一个相同大小和特征的未模糊的图像

CSDN @Karon_NeverAlone

(2) 实验法

通过使用相同的系统设置成像一个脉冲(由一个明亮的光点模拟)来获得退化的脉冲响应

预测退化函数

$$H(u, v) = \frac{G(u, v)}{A}$$

傅里叶变化后的图像

脉冲强度

CSDN @Karon_NeverAlone

(3) 建模

根据大气湍流的物理特征

$$H(u, v) = e^{-k(u^2 + v^2)^{5/6}}$$

CSDN @Karon_NeverAlone

根据平面运动 (运动模糊恢复)

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

$x_0(t)$ 是x上的一小段移动

CSDN @Karon_NeverAlone

---两个滤波 (这里都是假设H已知了)

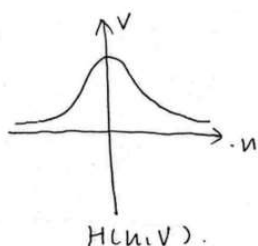
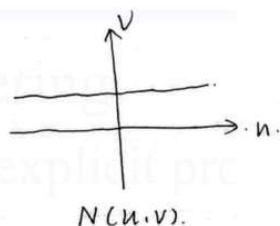
全逆滤波

全逆滤波 (两边同除 $H(u, v)$)

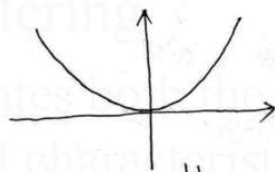
$$\hat{F}(u, v) = \frac{H(u, v)F(u, v)}{H(u, v)} + \frac{N(u, v)}{H(u, v)}$$

① 如果没有 $N(u, v)$ ，可以这样做。

② 有 $N(u, v)$ ，不可以全逆，why?



相除: $\frac{N(u, v)}{H(u, v)}$ 有在两侧会非常大。



有一个解决方法，只存在低频率的部分除高频部分仍保持 $H(u, v)$ 。

对 $\frac{N(u, v)}{H(u, v)}$ 低通滤波 比如乘一个巴特沃斯

CSDN @Karon_NeverAlone

但是问题在于，我们只是假想这个噪声是白噪声，但具体噪声的类型不清楚，所以低通滤波的截止频率需要一点点尝试，这个问题在维纳滤波可以解决！

维纳滤波

要用到噪声的统计特性

维纳滤波 退化后的图像

$$\hat{F}(u, v) = \frac{G(u, v)}{H(u, v)} \cdot \frac{|H(u, v)|^2}{|H(u, v)|^2 + \frac{S_y(u, v)}{S_f(u, v)}}$$

① 刚刚会连滤波 $\hat{F}(u, v) = \frac{G(u, v)}{H(u, v)} = F(u, v) + \frac{N(u, v)}{H(u, v)}$
 当 $H(u, v) \rightarrow 0$ 时 $\hat{F}(u, v) \rightarrow \infty$ 但维纳滤波不会因为
 当 $H(u, v) \rightarrow 0$ 时 $\frac{G(u, v)}{H(u, v)} \rightarrow \infty$ 但 $\frac{|H(u, v)|^2}{|H(u, v)|^2 + S_y(u, v)/S_f(u, v)} \rightarrow 0$

② 当 $S_y(u, v) = 0$ 时 \rightarrow 即无噪声 \rightarrow 维纳退化到全通 (后面部分为 1)

③ $S_y(u, v) \ll S_f(u, v) \rightarrow$ 维纳滤波接近全通

④ $S_y(u, v) \gg S_f(u, v) \rightarrow$ 这部分就是 $H(u, v) \rightarrow 0$ 的部分 这部分被抑制

△ PS: 上述操作前提在于 $S_y(u, v)$ 即 $N(u, v)$ 已知 若未知时可以简易替代

$$\hat{F}(u, v) = \frac{G(u, v)}{H(u, v)} \cdot \frac{|H(u, v)|^2}{|H(u, v)|^2 + K} \quad \text{可以试}$$

比如: $S_y(u, v) = |N(u, v)|^2$ — 噪声

原图: $S_f(u, v) = |F(u, v)|^2$ 比如

比如: $\frac{S_y(u, v)}{S_f(u, v)}$ 如下:

CSDN @Karon_NeverAlone

建模估计: -

滤波: 已知 $H(u, v)$ 且 $\hat{F}(u, v) = F(u, v) + \frac{N(u, v)}{H(u, v)}$, 当 $H(u, v) \rightarrow 0$ 时 结果能

最小均方误差 (维纳) 滤波 \rightarrow 使 $e^2 = E\{ (t - \hat{f})^2 \}$ 最小

约束最小二乘滤波 \rightarrow 使 $c = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [v^2 f(x, y)]^2$ 最小

CSDN @Karon_NeverAlone

