华北电力大学

NORTH CHINA ELECTRIC POWER UNIVERSITY

5 图像复原

主要内容

- 图像复原概念
  - ✓ 图像退化的原因
  - ✓ 图像退化一般模型
  - ✓ 噪声模型
- 非约束复原
  - ✓ 离散函数的退化模型
  - ✓ 非约束复原
  - ✓ 代数方法
  - ✓ 逆滤波器方法

主要内容

- 有约束复原
  - ✓ 有约束的最小二乘法图像复原
  - ✓ 维纳滤波方法
  - ✓ 盲解卷积图像复原

5.1 图像复原的概念

- 图像复原是一种改善图像质量的处理技术

  ✓ 消除或减轻在图像获取及传输过程中造成的图像品质下降即退化现象。

- 图象恢复与图象增强的异同

  ✓ 相同点：改进输入图象的视觉质量 。

  ✓ 不同点：图象增强目的是取得较好的视觉结果(主观过程)； 图象恢复根据相应的退化模型和知识重建或恢复原始的图象(客观过程) 。

5.1.1 引起图像退化的原因

- 引起图像退化的原因：
  - ✓ 由成像系统光学特性造成的畸变
  - ✓ 相对运动造成的图像模糊
  - ✓ 源自电路和光度学因素的噪声等

由成像系统光学特性造成的畸变

镜头畸变会产生图像失真，
需要后期进行几何纠正。

由于孔径衍射产生的退化造成
原本比较清晰的图像变得模糊。

相对运动造成的图像模糊

拍摄过程中相机发生震动也会产生这种退化，目标的图案沿运动方向拖长,变得有叠影.

在实际拍摄过程中如果目标运动超过图像平面上一个以上像素的距离就会造成模糊。使用望远镜头的系统(视场较窄)对这类图像的退化非常敏感。

源自电路和光度学因素的噪声

可以看做一种随机性的退化，原本只有目标的图像叠加了许多随机的亮点和暗点，目标和背景都受到影响。

matlab中图像退化函数

- 运动模糊fspecial('motion', LEN, THETA);
  - ✓ LEN为运动模糊的长度，默认为9
  - ✓ THETA为运动模糊的角度，默认为0

matlab中图像退化函数

原图像

运动模糊，长度9，角度0

运动模糊，长度50，角度0

运动模糊，长度50，角度45

matlab中图像退化函数

```
I1 = imread('lena.jpg');
I1 = rgb2gray(I1);
I1 = im2double(I1);
H1 = fspecial('motion’);  %运动模糊
J1 = imfilter(I1, H1);
H2 = fspecial('motion', 50, 0); %运动模糊
J2 = imfilter(I1, H2);
H3 = fspecial('motion', 50, 45); %运动模糊
J3 = imfilter(I1, H3);
figure(1);
subplot(2,2,1);imshow(I1);title('原图像');
subplot(2,2,2);imshow(J1);title('运动模糊，长度9，角度0');
subplot(2,2,3);imshow(J2);title('运动模糊，长度50，角度0');
subplot(2,2,4);imshow(J3);title('运动模糊，长度50，角度45');
```

matlab中图像退化函数

- 高斯模糊fspecial(‘gaussian’, HSIZE, SIGMA);
  - ✓ HSIZE高斯低通滤波器的尺寸，默认5
  - ✓ SIGMA为均方差，默认为0.5

matlab中图像退化函数

原图像

高斯模糊，尺寸5，均方差0.5

高斯模糊，尺寸50，均方差5

高斯模糊，尺寸50，均方差50

matlab中图像退化函数

```
I1 = imread('lena.jpg');
I1 = rgb2gray(I1);
I1 = im2double(I1);
H1 = fspecial('gaussian'); %高斯模糊
J1 = imfilter(I1, H1);
H2 = fspecial('gaussian', 50, 5); %高斯模糊
J2 = imfilter(I1, H2);
H3 = fspecial('gaussian', 50, 50); %高斯模糊
J3 = imfilter(I1, H3);
figure(1);
subplot(2,2,1);imshow(I1);title('原图像');
subplot(2,2,2);imshow(J1);title('高斯模糊，尺寸5，均方差0.5');
subplot(2,2,3);imshow(J2);title('高斯模糊，尺寸50，均方差5');
subplot(2,2,4);imshow(J3);title('高斯模糊，尺寸50，均方差50');
```

matlab中图像退化函数

- 圆盘模糊fspecial(‘disk’,RADIUS);
  - ✓ RADIUS模糊半径，默认5

matlab中图像退化函数

原图像

圆盘模糊，半径5

圆盘模糊，半径10

圆盘模糊，半径50

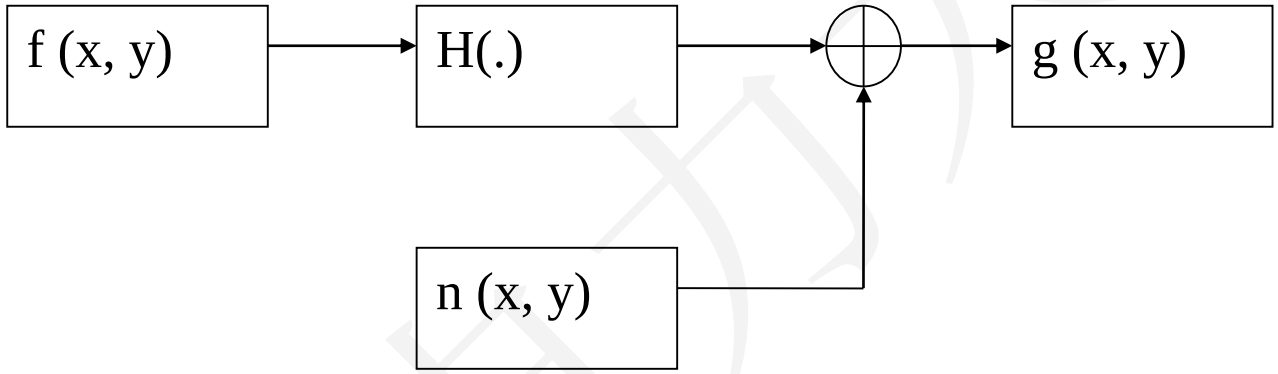
matlab中图像退化函数

```
I1 = imread('lena.jpg');
I1 = rgb2gray(I1);
I1 = im2double(I1);
H1 = fspecial('disk'); %圆盘模糊
J1 = imfilter(I1, H1);
H2 = fspecial('disk', 10); %圆盘模糊
J2 = imfilter(I1, H2);
H3 = fspecial('disk', 50); %圆盘模糊
J3 = imfilter(I1, H3);
figure(1);
subplot(2,2,1);imshow(I1);title('原图像');
subplot(2,2,2);imshow(J1);title('圆盘模糊，半径5');
subplot(2,2,3);imshow(J2);title('圆盘模糊，半径10');
subplot(2,2,4);imshow(J3);title('圆盘模糊，半径50');
```

5.1.2 图像退化一般模型

- 图像复原处理的关键问题在于建立退化模型。
- 利用已有的知识和经验对模糊或噪声等退化过程进行数学模型的建立及描述，并针对此退化过程的数学模型进行图像复原。
- 图像退化过程的先验知识在图像复原技术中起着重要作用。

5.1.2 图像退化一般模型



- 退化过程被模型化为一个系统(或算子)H(也称为点扩散函数PSF)，原始图像f(x,y)在经过该系统退化作用后与一个加性噪声n(x,y)相叠加而产生出最终的退化图像g(x,y)。

5.1.2 图像退化一般模型

- 如果系统H是一个线性、位置不变性的过程，退化图像可以表示为

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

空间域上的卷积等同于频率域上的乘积

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

h(x,y)表示退化函数的空间描述。

5.1.3 噪声模型

- 数字图像的噪声主要来源于图像的获取和传输过程

  ✓ 图像获取的数字化过程，如图像传感器的质量和环境条件

  ✓ 图像传输过程中传输信道的噪声干扰，如通过无线网络传输的图像会受到光或其它大气因素的干扰

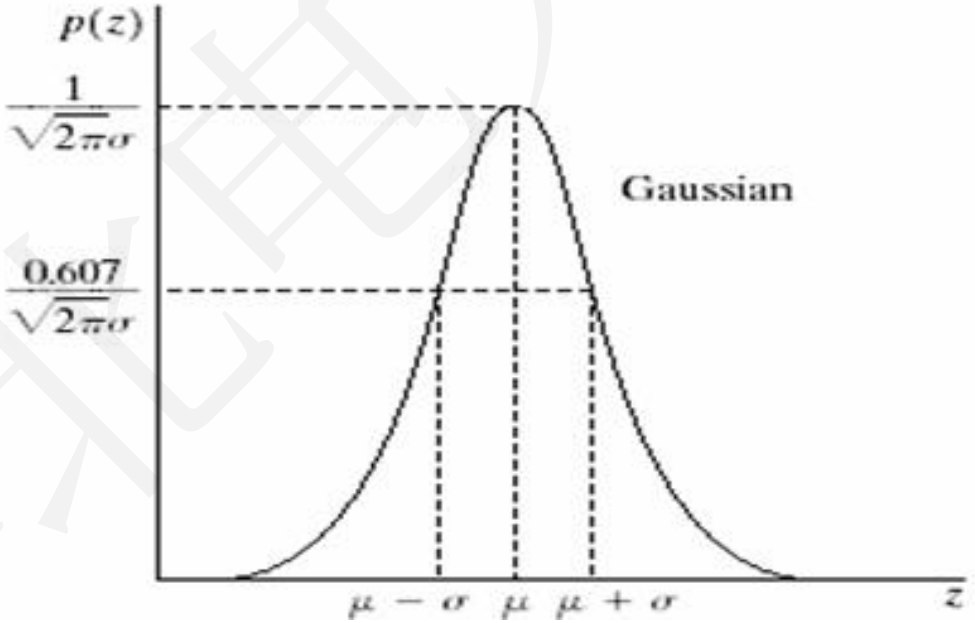
5.1.3 噪声模型

- 一些常见的、重要的噪声
  - ✓ 高斯噪声
  - ✓ 瑞利噪声
  - ✓ 伽马（爱尔兰）噪声
  - ✓ 指数分布噪声
  - ✓ 均匀分布噪声
  - ✓ 脉冲噪声（椒盐噪声）

5.1.3 噪声模型

- （1）高斯噪声的概率密度函数

$$p(z)=\frac{1}{\sqrt{2\pi\sigma}}e^{-(z-\mu)^2/2\sigma^2}$$



5.1.3 噪声模型

- （2）瑞利噪声的概率密度函数

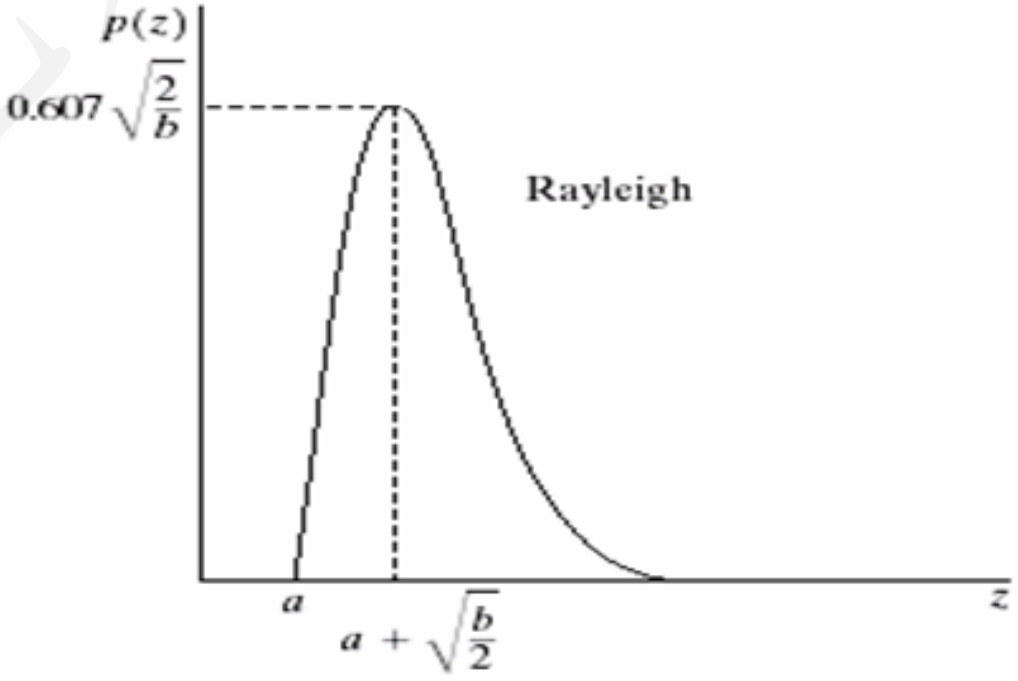
$$p(z)=\begin{cases}\dfrac{2}{b}(z-a)e^{-(z-a)^2/b} & z \geq a \\ 0 & z < a\end{cases}$$

$$\mu = a + \sqrt{\pi b / 4}$$

$$\sigma^2 = \frac{b(4-\pi)}{4}$$

距离原点的位移是a

函数曲线向右变形



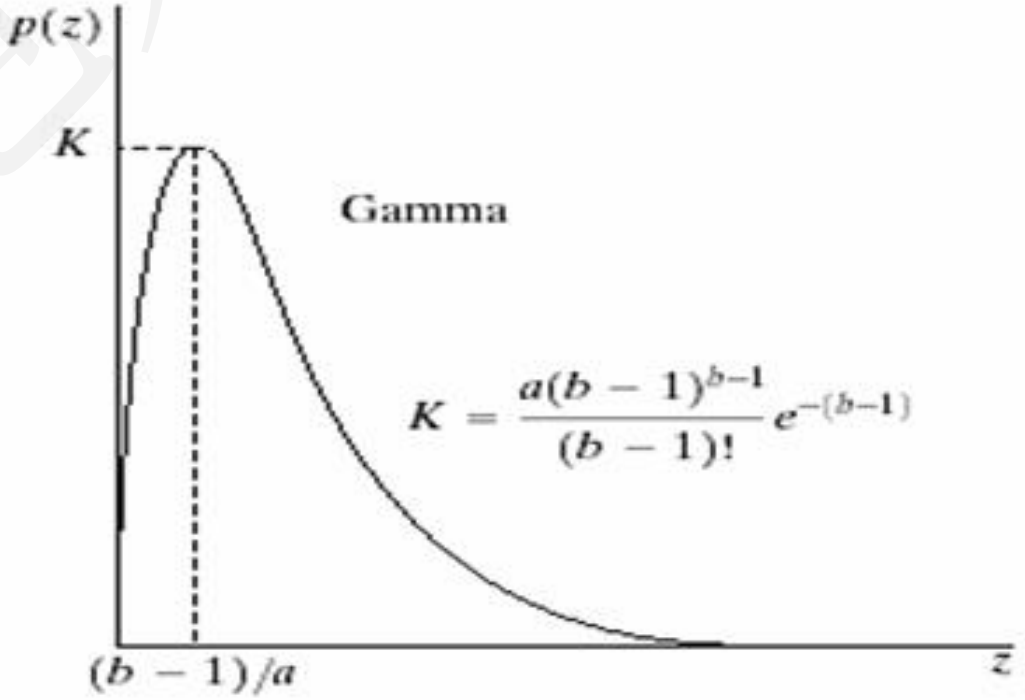
5.1.3 噪声模型

- （3）伽马噪声的概率密度函数

$$p(z)=\begin{cases}\dfrac{a^{b}z^{b-1}}{(b-1)!}e^{-az} & z\geq a\\ 0 & z<a\end{cases}$$

$$\mu=\frac{b}{a}$$

$$\sigma^{2}=\frac{b}{a^{2}}$$



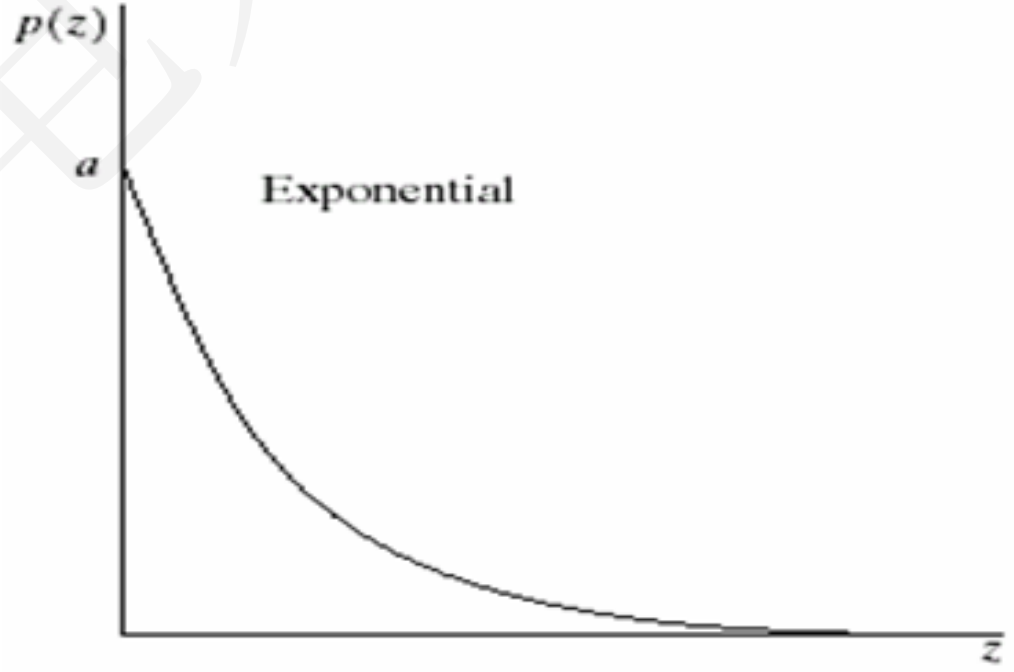
5.1.3 噪声模型

- （4）指数噪声的概率密度函数

$$p(z)=\begin{cases} ae^{-az} & z \geq a \\ 0 & z < a \end{cases}$$

$$\mu = \frac{1}{a}$$

$$\sigma^2 = \frac{1}{a^2}$$



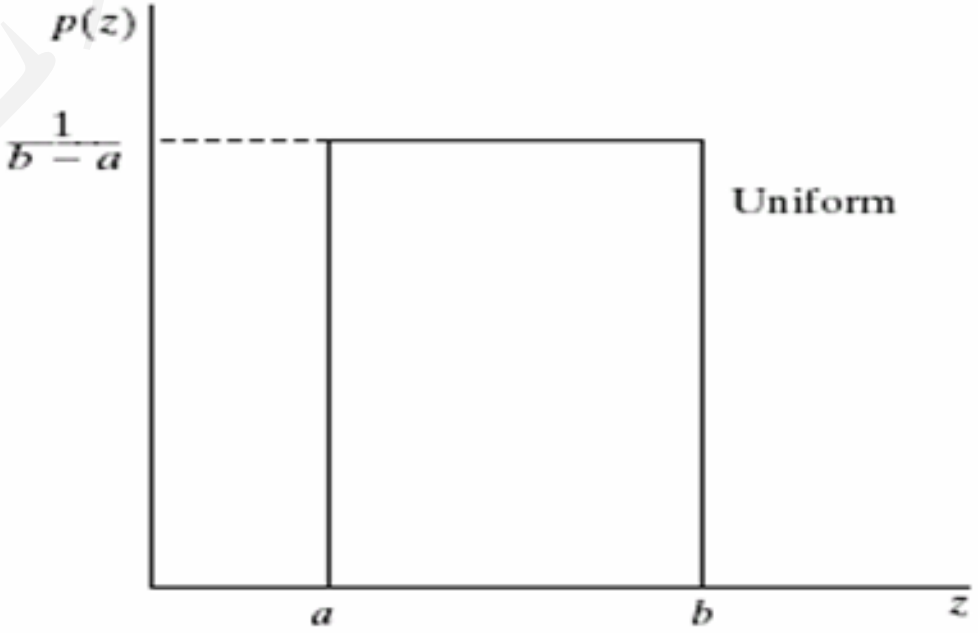
5.1.3 噪声模型

- （5）均匀分布噪声的概率密度函数

$$p(z)=\begin{cases}\dfrac{1}{b-a} & a \le z \le b \\ 0 & \text{其它}\end{cases}$$

$$\mu = \frac{a+b}{2}$$

$$\sigma^2 = \frac{(b-a)^2}{12}$$



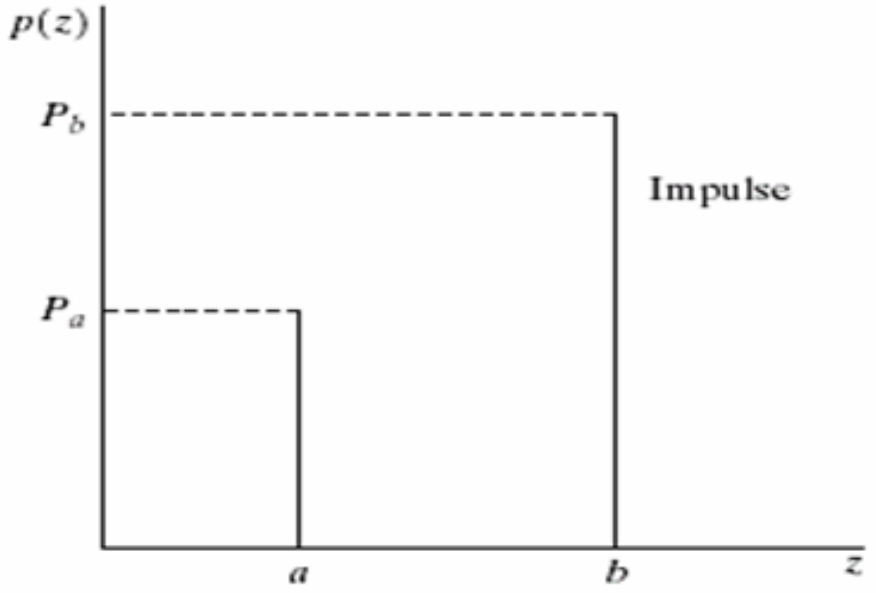
5.1.3 噪声模型

- （6）脉冲噪声（椒盐）的概率密度函数

$$p(z)=\begin{cases} P_a & z=a \\ P_b & z=b \\ 0 & \text{其它} \end{cases}$$

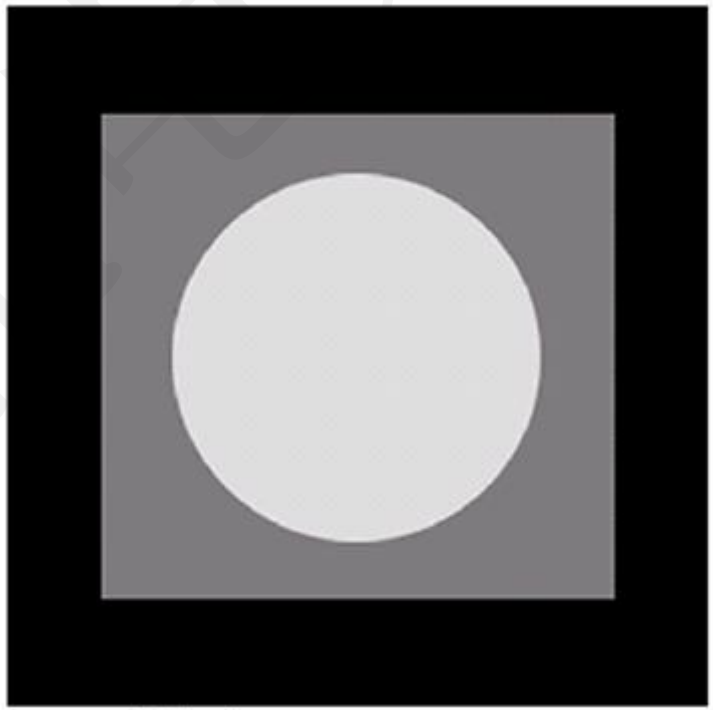
如果Pa或Pb为零，则脉冲噪声称为单极脉冲

如果Pa或Pb均不为零，则脉冲噪声称为双极脉冲噪声或椒盐噪声



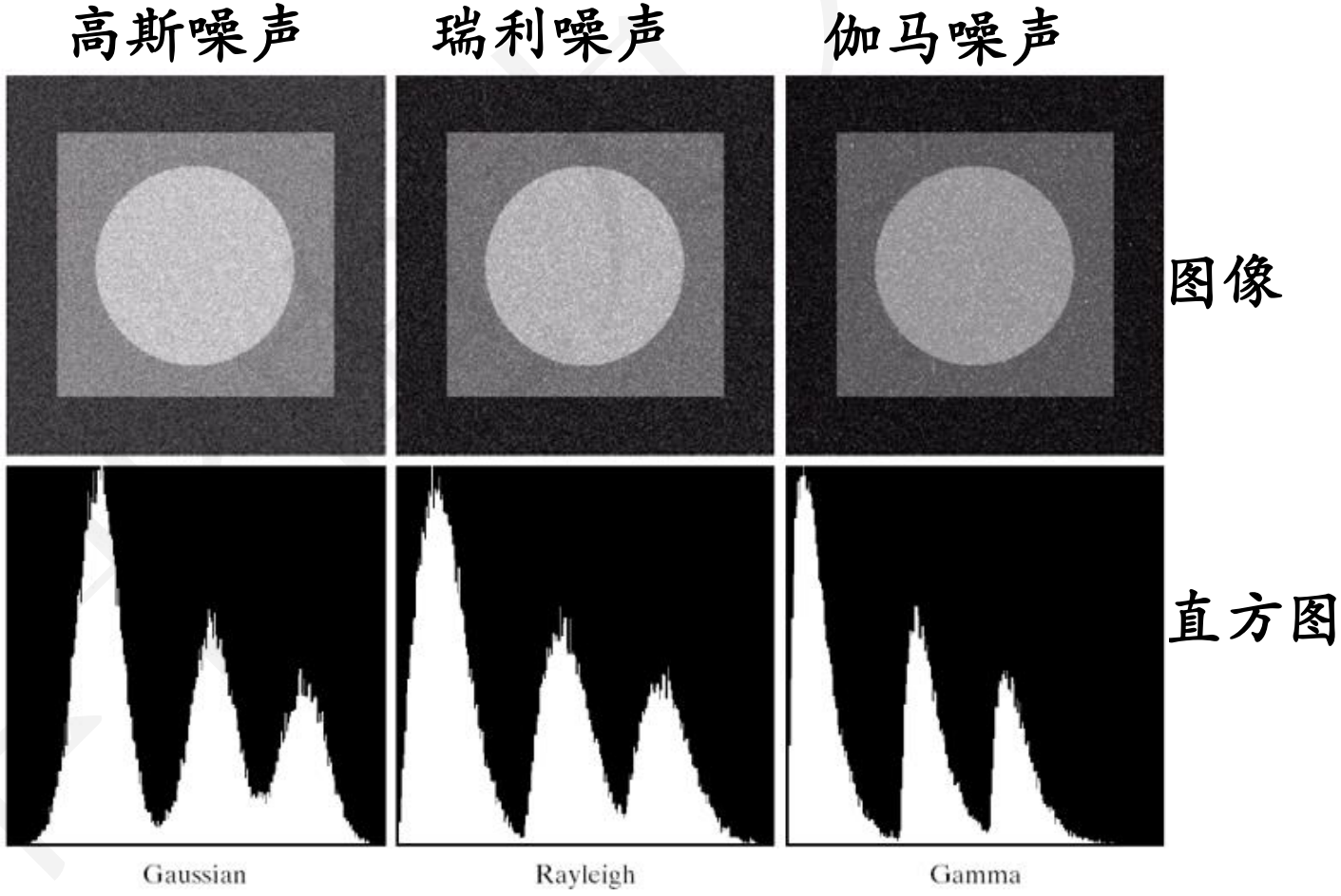
5.1.3 噪声模型

- 几种噪声的影响
  - ✓ 用于噪声模型的测试图
  - ✓ 由简单、恒定的区域组成
  - ✓ 仅仅有3个灰度级的变化



5.1.3 噪声模型

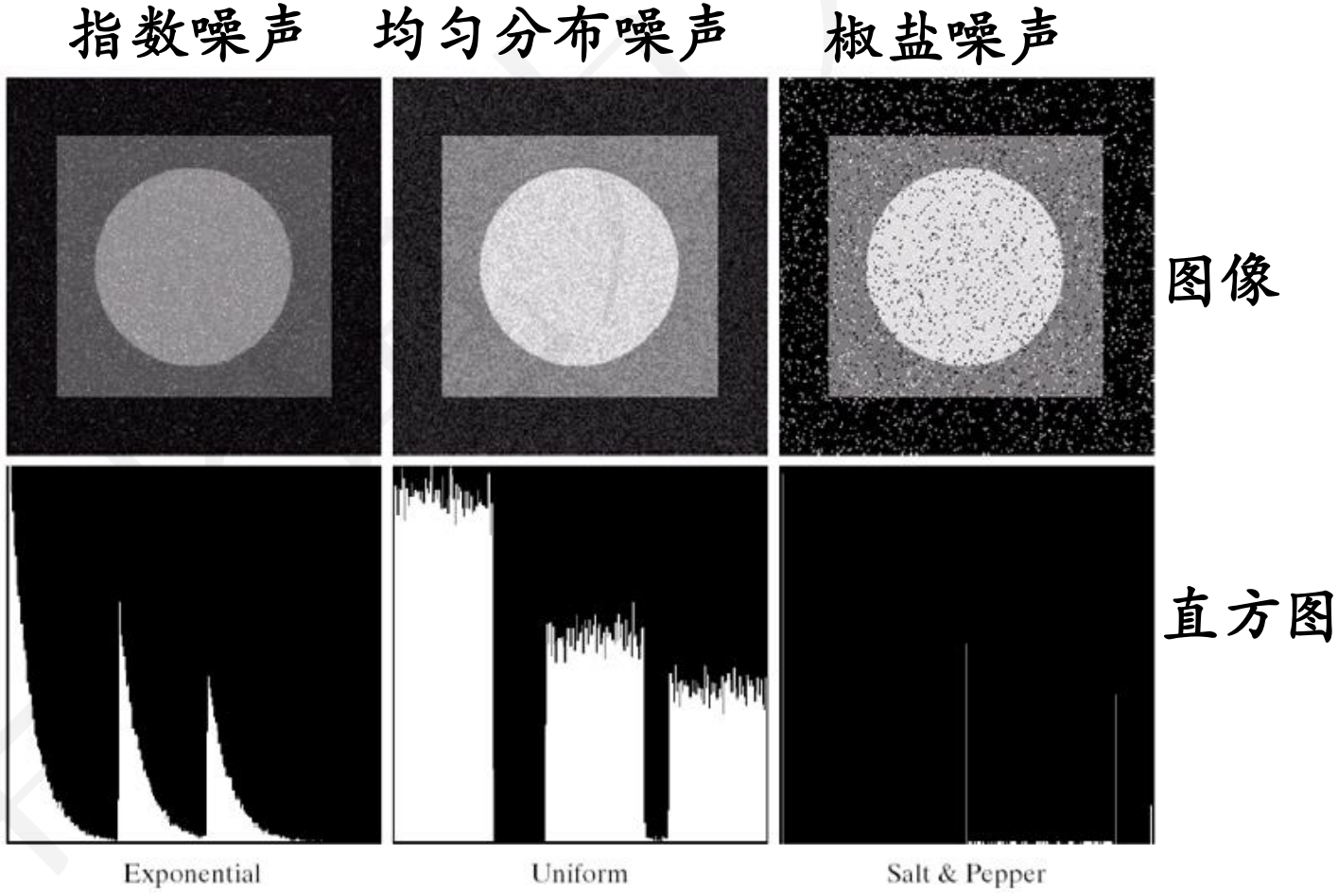
- 几种噪声的影响



5.1.3 噪声模型

- 几种噪声的影响

前面5种噪声的图像并没有显著不同
但它们的直方图具有明显的区别



5.1.3 噪声模型

- 几种噪声的运用
  - ✓高斯噪声源于电子电路噪声和由低照明度或高温带来的传感器噪声
  - ✓瑞利噪声对分布在图像范围内特征化噪声有用
  - ✓伽马分布和指数分布用于激光成像噪声
  - ✓均匀密度分布作为模拟随机数产生器的基础
  - ✓脉冲噪声用于成像中的短暂停留中，如错误的开关操作

5.1.4 噪声复原思想

- 当唯一退化是噪声时，

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

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

✓ 噪声项未知，不能从g(x,y)或G(u,v)减去噪声

✓ 可以选择空间域滤波和频率域方法进行图像复原

5.2 非约束复原

- 图像复原实际上就是已知g(x,y)的情况下，求f(x,y)的问题。

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

- 为方便计算机对退化图像进行恢复，必须对退化图像、噪声、要恢复的输入图像进行离散化。因此，需要引申出离散的退化模型。

5.2.1 离散函数的退化模型

- 在去除噪声之后，退化图像为：

$$g_e(x,y) = f_e(x,y) * h_e(x,y) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f_e(m,n)h_e(x-m,y-n)$$

✓ 式中x=0,1,2,…,M-1; y=0,1,2,…,N-1

✓ 二维离散退化模型同样可以采用矩阵表示形式

$$g=Hf$$

✓ 式中g,f是MN×1维列向量， H是MN×MN维矩阵.其方法是将g(x,y)和f(x,y)中的元素按列首尾相连排成列向量。

5.2.1 离散函数的退化模型

$$
\mathbf{f}=\begin{pmatrix}
f_e(0,0)\\ f_e(0,1)\\ \cdots\\ f_e(0,N-1)\\ f_e(1,0)\\ f_e(1,1)\\ \cdots\\ f_e(1,N-1)\\ \vdots\\ f_e(i,N-1)\\ \cdots\\ f_e(M-1,N-1)
\end{pmatrix}_{M\times N\text{行}}
\qquad
\mathbf{g}=\begin{pmatrix}
g_e(0,0)\\ g_e(0,1)\\ \cdots\\ g_e(0,N-1)\\ g_e(1,0)\\ g_e(1,1)\\ \cdots\\ g_e(1,N-1)\\ \vdots\\ g_e(i,N-1)\\ \cdots\\ g_e(M-1,N-1)
\end{pmatrix}_{M\times N\text{行}}
$$

5.2.1 离散函数的退化模型

H为$MN\times MN$阶的分块矩阵，这一矩阵为$M\times M$分块循环阵。H的任意元素H_j是由$h(x,y)$第j行循环构成，且H_j为$N\times N$循环阵。

$$\mathbf{H}=\begin{bmatrix} \mathrm{H}_0 & \mathrm{H}_{M-1} & \mathrm{H}_{M-2} & \dots & \mathrm{H}_1 \\ \mathrm{H}_1 & \mathrm{H}_0 & \mathrm{H}_{M-1} & \dots & \mathrm{H}_2 \\ \mathrm{H}_2 & \mathrm{H}_1 & \mathrm{H}_0 & \dots & \mathrm{H}_3 \\ \dots & \dots & \dots & \dots & \dots \\ \mathrm{H}_{M-1} & \mathrm{H}_{M-2} & \mathrm{H}_{M-3} & \dots & \mathrm{H}_0 \end{bmatrix} \quad M\times M\text{个分块}$$

$$\mathbf{H}_j=\begin{bmatrix} h_e(j,0) & h_e(j,N-1) & h_e(j,N-2) & \dots & h_e(j,1) \\ h_e(j,1) & h_e(j,0) & h_e(j,N-1) & \dots & h_e(j,2) \\ h_e(j,2) & h_e(j,1) & h_e(j,0) & \dots & h_e(j,3) \\ \dots & \dots & \dots & \dots & \dots \\ h_e(j,N-1) & h_e(j,N-2) & h_e(j,N-3) & \dots & h_e(j,0) \end{bmatrix} \quad N\times N\text{ 维}$$

5.2.1 离散函数的退化模型

- 如果考虑到噪声的影响，

$$g_e(x,y)=\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f_e(m,n)h_e(x-m,y-n)+n_e(x,y)$$

✓ 矩阵表示形式

$$g=Hf+n$$

✓ 要想得出f(x,y)，其计算工作是十分困难的。

例如，对于一般大小的图像来说，M=N=512，此时矩阵H的大小为MN*MN = 512*512*512*512 = 262144*262144，要直接得出f(x,y)，则需要求解262144个联立方程组，其计算量是十分惊人的。

5.2.2 非约束复原

- 图像复原的主要目的是在假设具备退化图像g及H和n的某些知识的前提下，估计出原始图像f 的估计值，估计值应使准则为最优。
- 如果仅仅要求某种优化准则为最小，不考虑其它任何条件约束，这种复原方法为非约束复原方法。
- 非约束复原方法仅将图像看做一个数字矩阵，从数学角度进行处理而不考虑恢复后图像应受到的物理约束。

✓代数方法

✓逆滤波器方法

5.2.3 代数方法

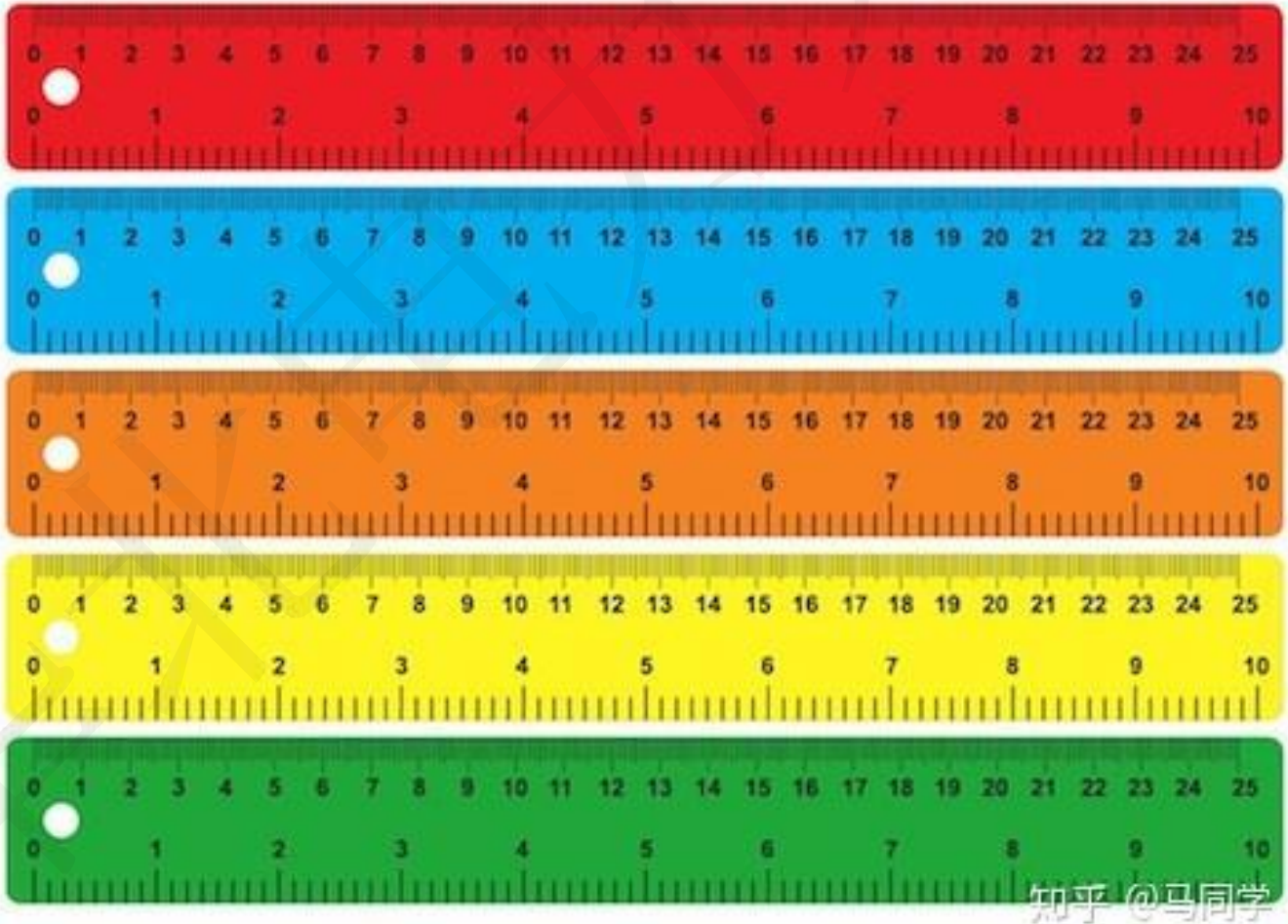
- 代数复原方法的中心是寻找一个估计，使事先确定的某种优化准则为最小。
- 如果退化模型是式 g=Hf+n 的形式，就可以用线性代数中的理论解决图像复原问题。
- 可以选择最小二乘法作为优化准则的基础。

5.2.3 代数方法

- 最小二乘法，是一种在误差估计、不确定度、系统辨识及预测、预报等数据处理诸多学科领域得到广泛应用的数学工具。

5.2.3 代数方法

- 来看一个生活中的例子。比如说，有五把尺子：



5.2.3 代数方法

- 用它们来分别测量一线段的长度，得到的数值分别为（颜色指不同的尺子）：

	长度
红	10.2
蓝	10.3
橙	9.8
黄	9.9
绿	9.8

5.2.3 代数方法

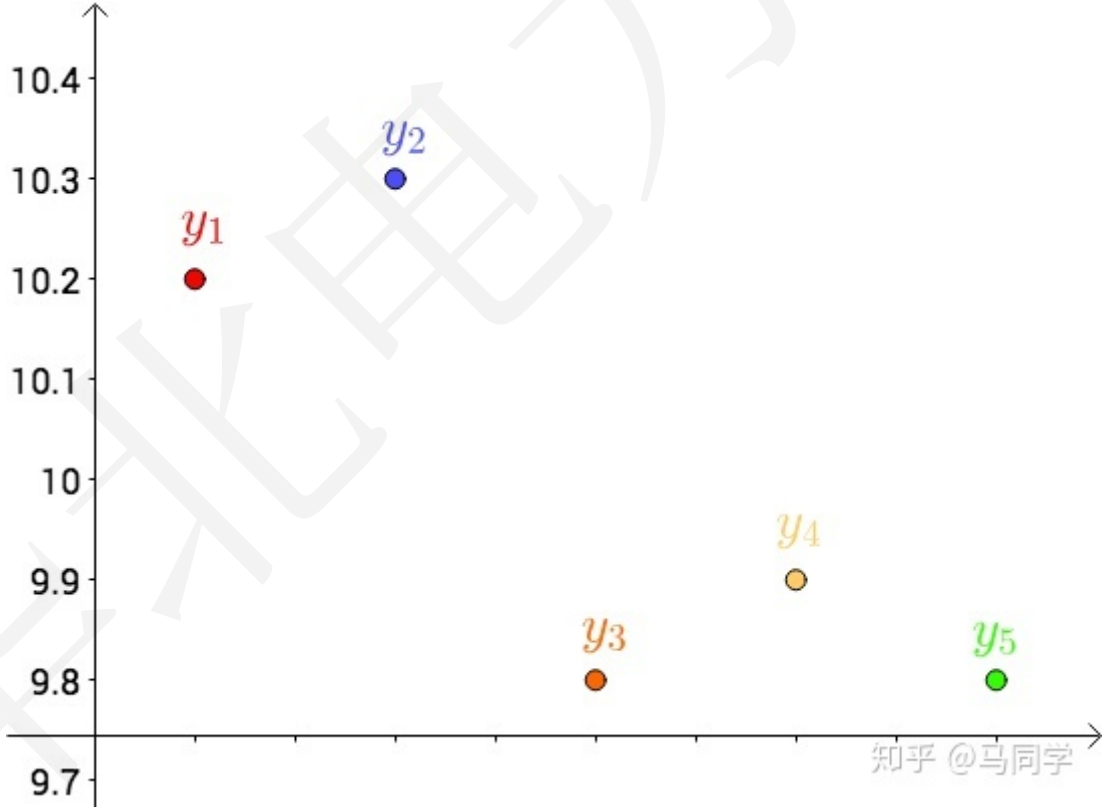
- 这种情况下，一般取平均值来作为线段的长度：

$$\overline{x} = \frac{10.2 + 10.3 + 9.8 + 9.9 + 9.8}{5} = 10$$

- 一些数学爱好者产生了疑问：
  - ✓ 这样做有道理吗?
  - ✓ 用调和平均数行不行?
  - ✓ 用中位数行不行?
  - ✓ 用几何平均数行不行?

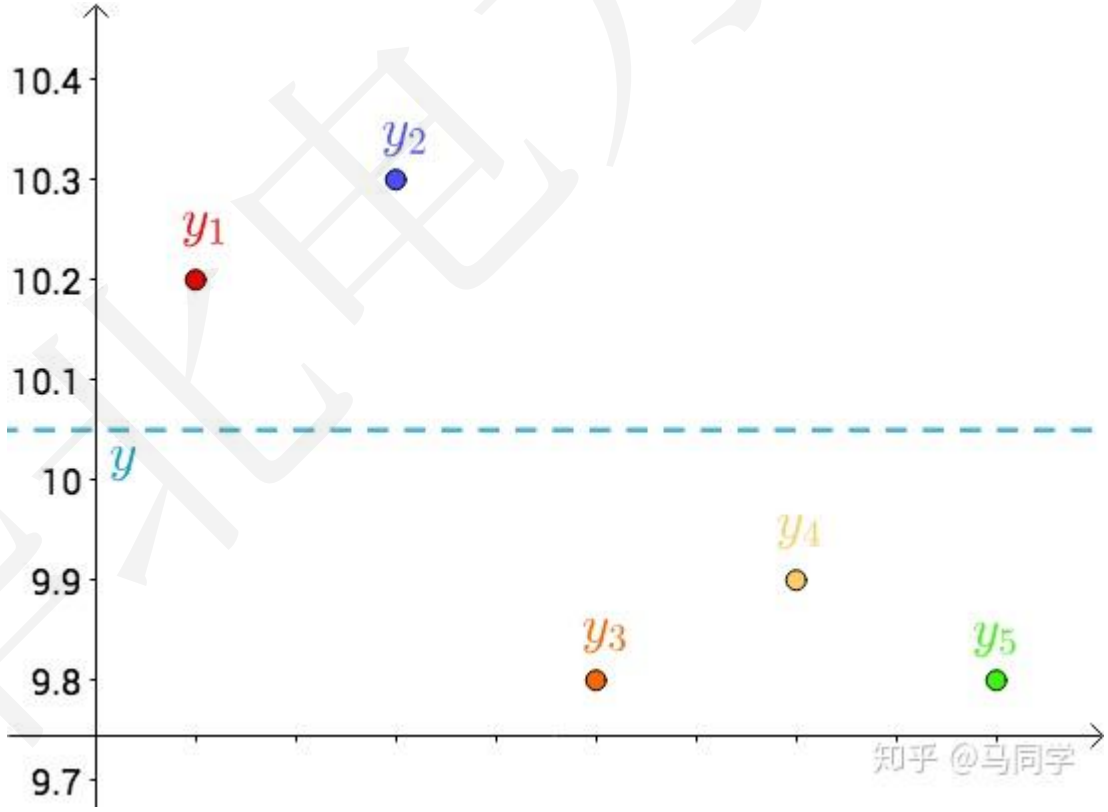
5.2.3 代数方法

- 换一种思路来思考刚才的问题。
  - ✓（1）把测试得到的值画在笛卡尔坐标系中



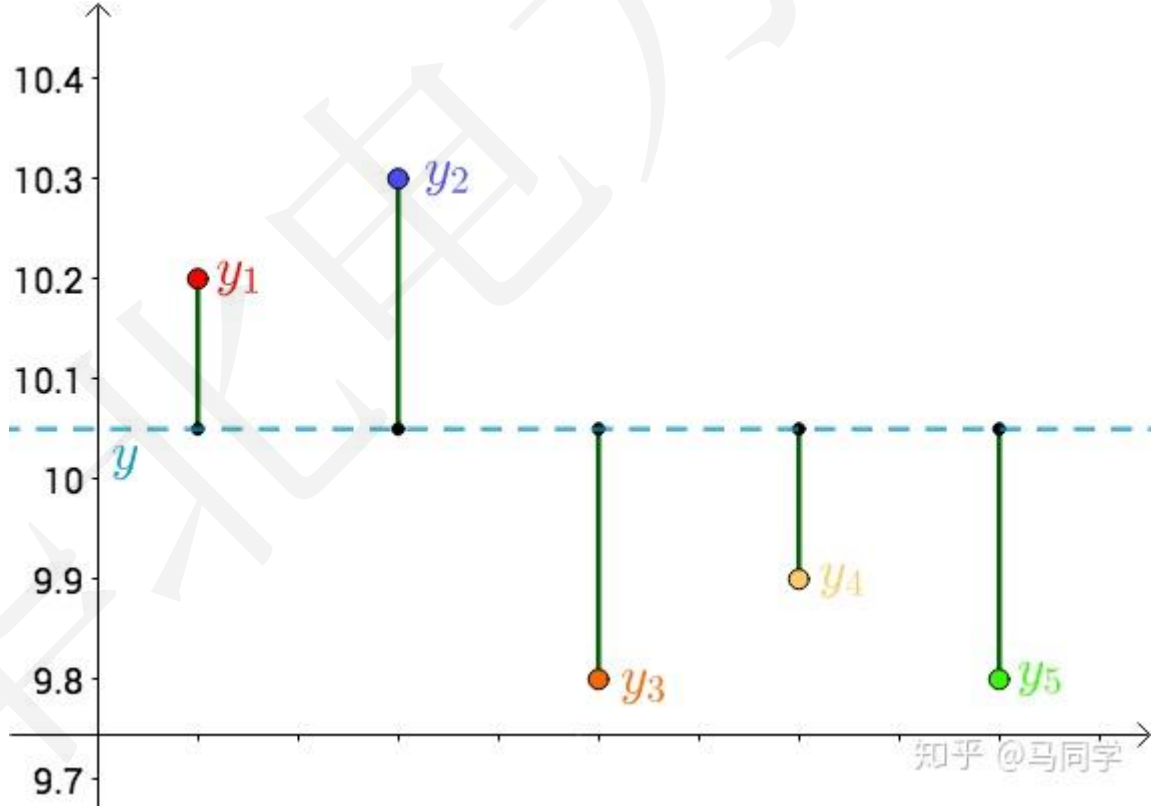
5.2.3 代数方法

- 换一种思路来思考刚才的问题。
  ✓（2）把要猜测的线段长度的真实值用平行于横轴的直线来表示（因为是猜测的，所以用虚线来画）



5.2.3 代数方法

- 换一种思路来思考刚才的问题。
  - ✓（3）每个点都向y做垂线，垂线的长度就是$|y-y_i|$，也可以理解为测量值和真实值之间的误差



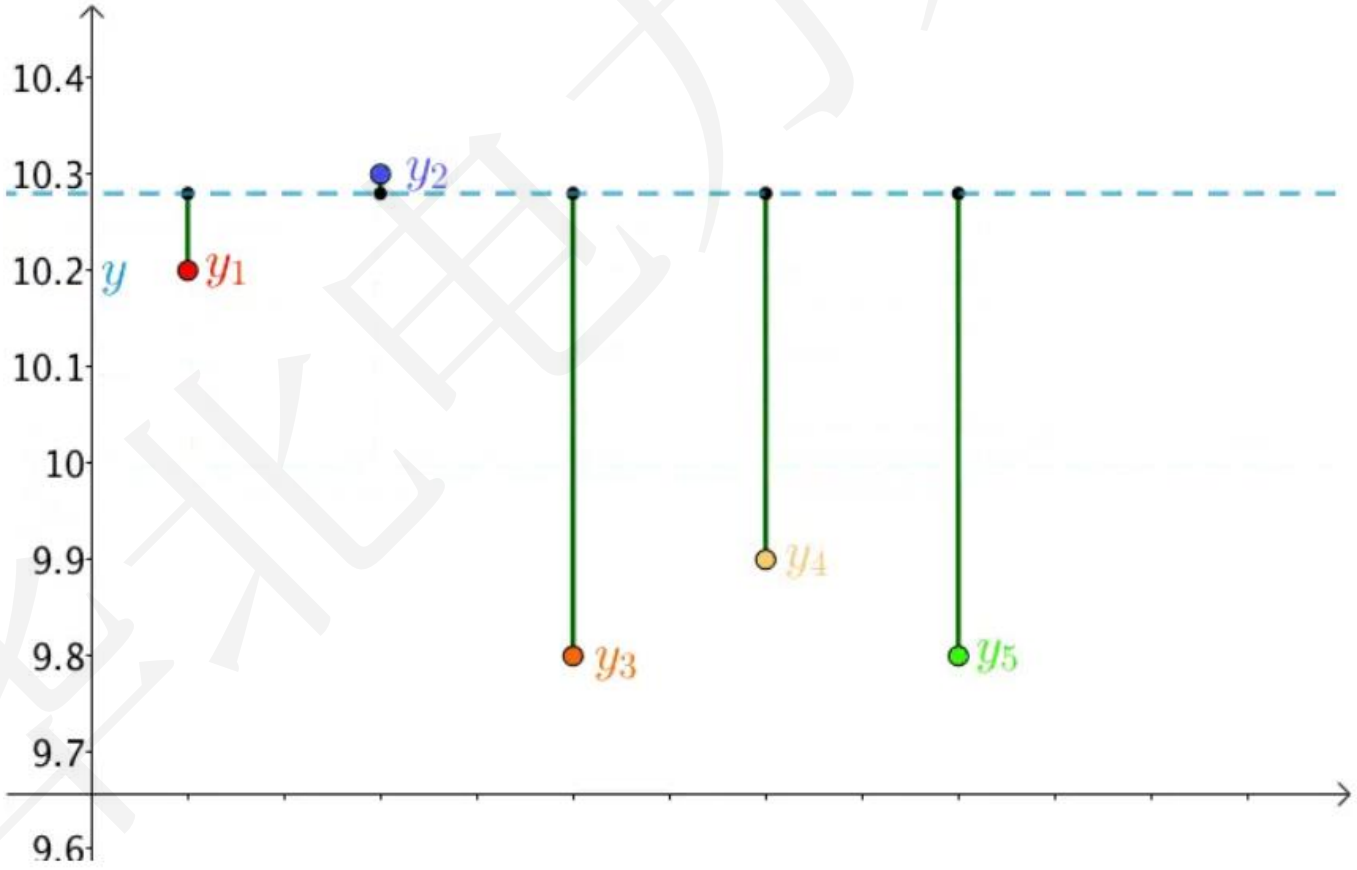
5.2.3 代数方法

- 换一种思路来思考刚才的问题。
  ✓（4）因为误差是长度，还要取绝对值，计算起来麻烦，就用平方来代表误差

$$S_{\epsilon^2} = \sum (y - y_i)^2$$

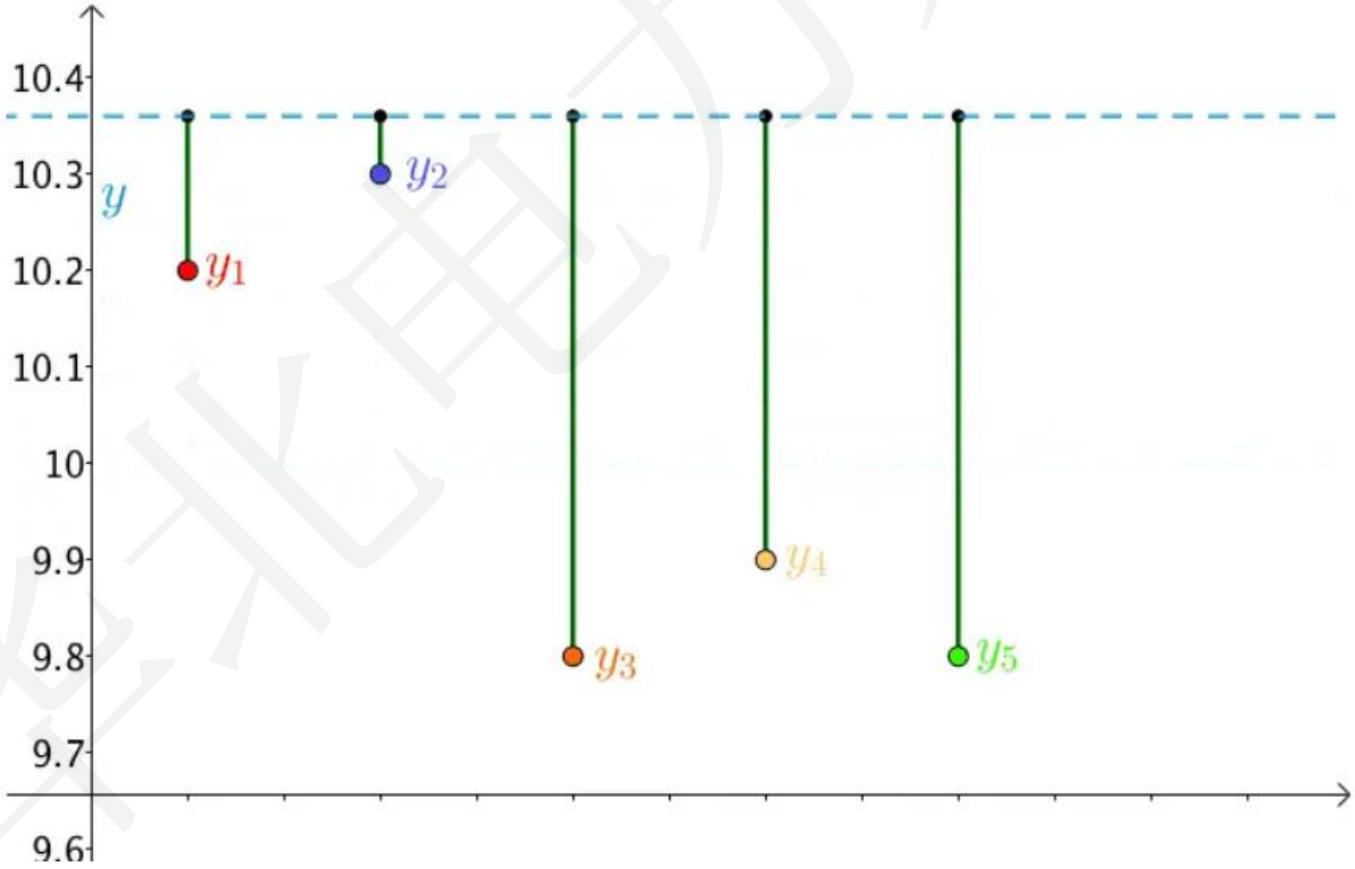
5.2.3 代数方法

- 换一种思路来思考刚才的问题。
  - ✓（5）因为y是猜测的，所以可以不断变换：



5.2.3 代数方法

- 换一种思路来思考刚才的问题。
  - ✓（5）因为y是猜测的，所以可以不断变换：



5.2.3 代数方法

- 换一种思路来思考刚才的问题。
  ✓（6）法国数学家阿德里安-马里-勒让德提出：总得误差平和最小的y就是真值。

$$S_{\epsilon^2} = \sum (y - y_i)^2 \text{最小} \Longrightarrow \text{真值} y$$

这是一个二次函数，对其求导，导数为0的时候取得最小值：

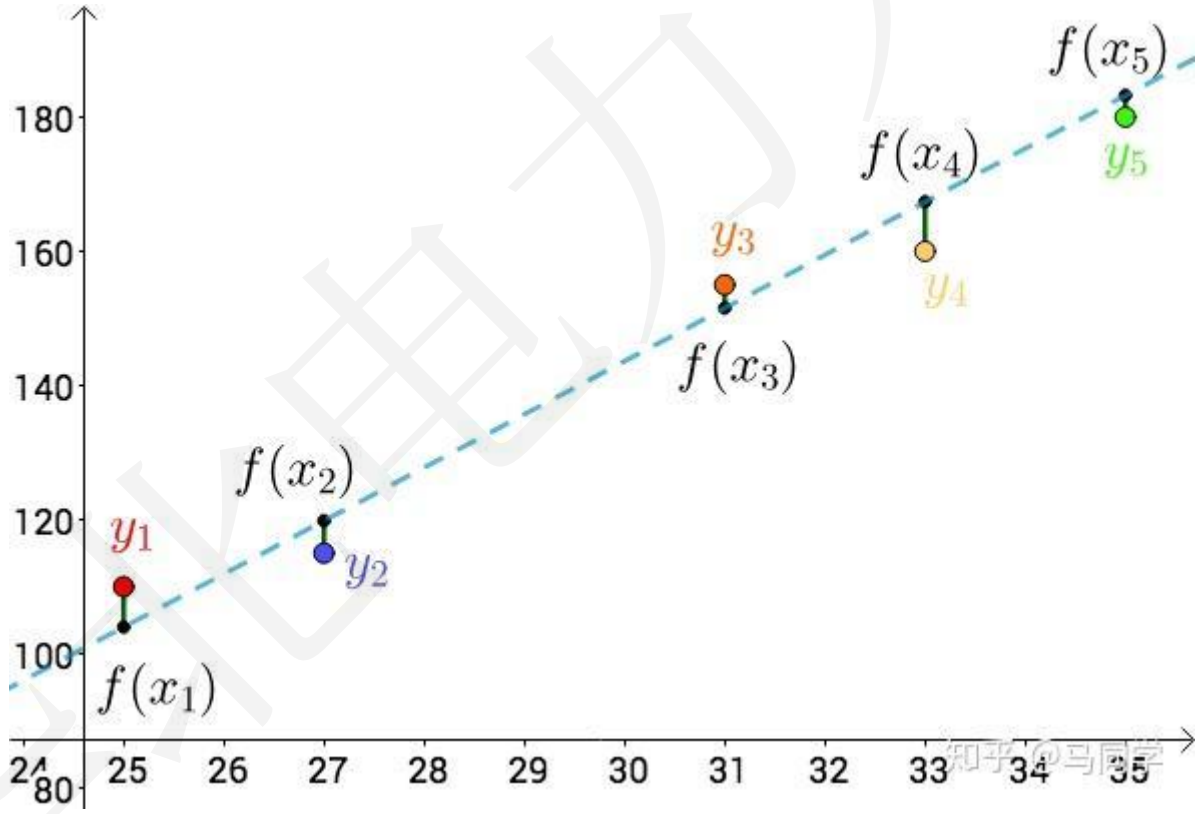
$$\frac{d}{dy} S_{\epsilon^2} = \frac{d}{dy} \sum (y - y_i)^2 = 2 \sum (y - y_i)$$

$$= 2((y - y_1) + (y - y_2) + (y - y_3) + (y - y_4) + (y - y_5)) = 0$$

$$5y = y_1 + y_2 + y_3 + y_4 + y_5 \Longrightarrow y = \frac{y_1 + y_2 + y_3 + y_4 + y_5}{5}$$

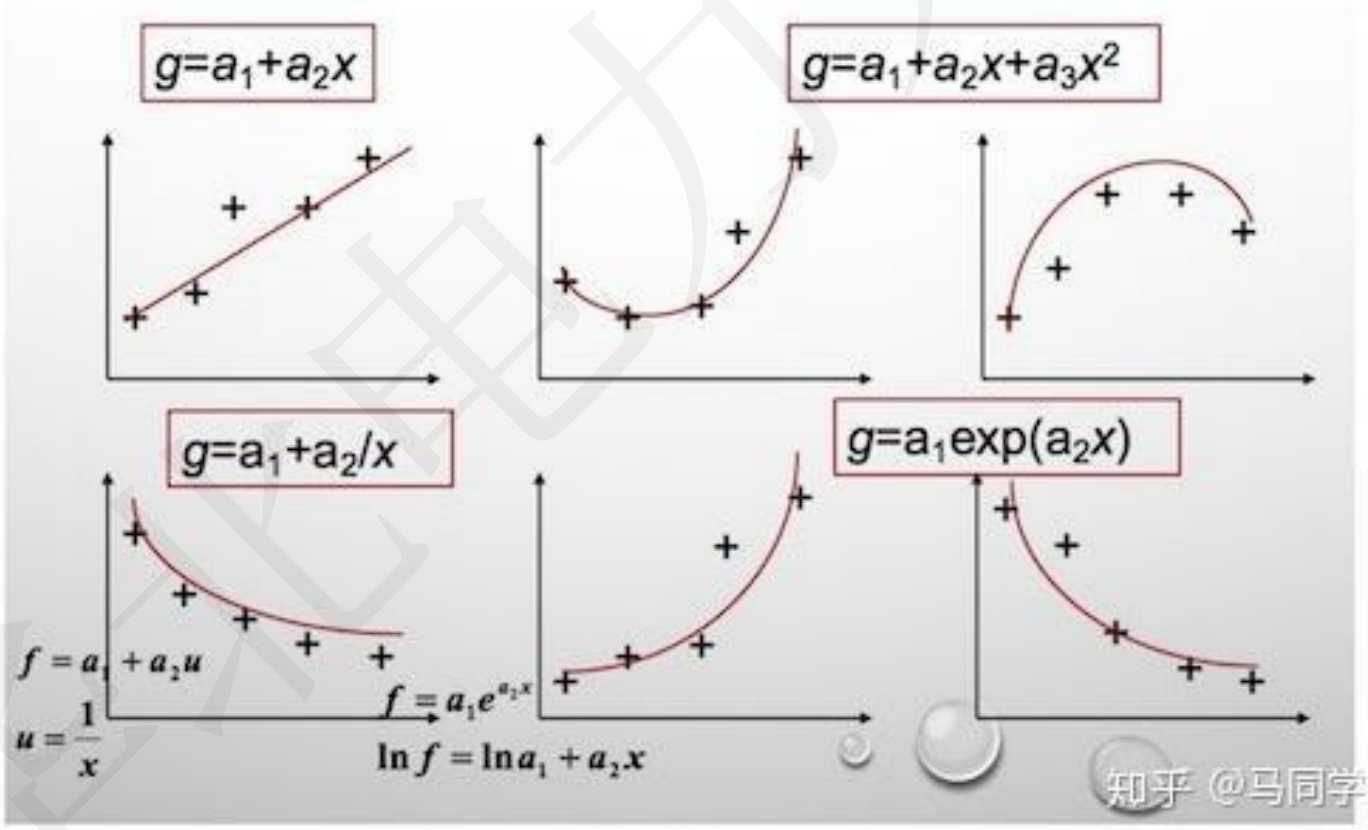
5.2.3 代数方法

- 最小二乘法的推广



5.2.3 代数方法

- 最小二乘法的推广



5.2.3 代数方法

- 由前面介绍的图像退化模型可知，其噪声项为：

$$n = g - Hf$$

在并不了解噪声项n的情况下，希望找到一个$\hat{f}$，使得$H\hat{f}$对在最小乘方意义上来说近似于g，也就是说，希望找到一个$\hat{f}$，使得下式最小

$$J(\hat{f}) = \|n\|^2 = \left\|g - H\hat{f}\right\|^2$$

5.2.3 代数方法

- 实际上是求极小值问题。

$$\frac{\partial J(\hat{f})}{\partial \hat{f}} = -2H^T(g - H\hat{f}) = 0$$

$$H^T H\hat{f} = H^T g$$

$$\hat{f} = (H^T H)^{-1} H^T g$$

当M=N，H为一方阵,并且设H^{-1}存在,则可求得:

$$\hat{f} = H^{-1}(H^T)^{-1} H^T g = H^{-1} g$$

5.2.4 逆滤波器方法

- 也叫做反向滤波法，其主要过程是首先将要处理的数字图像从空间域转换到傅立叶频率域中，进行反向滤波后再由频率域转回到空间域，从而得到复原的图像信号。

5.2.4 逆滤波器方法

- 在不考虑噪声的情况下，退化图像为g(x,y)，原始图像为f(x ,y)，其退化模型进行傅立叶变换得：

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

$$F(u,v) = G(u,v) / H(u,v)$$

$$\hat{f}(x,y) = \mathcal{F}^{-1}\left[\hat{F}(u,v)\right] = \mathcal{F}^{-1}\left[\frac{G(u,v)}{H(u,v)}\right] \qquad x, y = 0, 1, \cdots, M-1$$

5.2.4 逆滤波器方法

- 在考虑噪声的情况下，退化图像为g(x,y)，原始图像为f(x ,y)，其退化模型进行傅立叶变换得:

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

$$F(u,v) = G(u,v) / H(u,v)-N(u,v) / H(u,v)$$

✓ 由于H(u,v)处于分母的位置上，可能会发生下列情况，即在u,v平面上有些点或区域会产生H(u,v)=0或H(u,v)非常小的情况，在这种情况下，即使没有噪声也无法精确恢复f(x,y)。

✓ 另外，在有噪声存在时，在H(u,v)的邻域内，H(u,v)的值可能比N(u,v)的值小得多，上面的式子得到的噪声项可能会非常大，使f(x,y)不能正确恢复。

5.2.4 逆滤波器方法

原图像

退化后图像

0.1逆滤波复原

0.01逆滤波复原

5.2.4 逆滤波器方法

- matlab中声明一个函数的格式为：

  function $[y_1, \cdots, y_N]$=函数名$(x_1, \cdots, x_M)$

  ✓ 其中$x_1, \cdots, x_M$是参数，$y_1, \cdots, y_N$是返回值

  ✓ 一般一个函数放在一个.m文件里

```
function Z=fftifilter(X, H) %参数X为图像，H为滤波器，Z滤波后图像
F = fft2(X, size(H,1), size(H,2)); %傅里叶变换
F = fftshift(F); %平移
Z = H.*F; %滤波
Z = abs(ifft2(fftshift(Z))); %反变换
Z = Z(1:size(X,1), 1:size(X,2)); %恢复尺寸
end
```

5.2.4 逆滤波器方法

```
I1 = imread('lena.jpg');
I1 = rgb2gray(I1);
I1 = im2double(I1);

[m, n] = size(I1); %图像尺寸
u = -m/2:(m/2-1);
v = -n/2:(n/2-1);
[U, V] = meshgrid(u, v); %构建网格
D = sqrt(U.^2+V.^2);
D0 = 30;
H = exp(-(D.^2)./(2*(D0.^2))); %高斯低通滤波器

J = fftifilter(I1, H); %频率域滤波，J退化后的图像
```

5.2.4 逆滤波器方法

```
HC = zeros(m,n);
M1=H>0.1; %频率域范围
HC(M1) = 1./H(M1); %滤波器取倒数
K = fftifilter(J,HC); %逆滤波

HC = zeros(m,n);
M2 = H>0.01; %频率域范围
HC(M2) = 1./H(M2); %滤波器取倒数
L = fftifilter(J,HC); %逆滤波

figure(1);
subplot(2,2,1);imshow(I1);title('原图像');
subplot(2,2,2);imshow(J, []);title('退化后图像');
subplot(2,2,3);imshow(K, []);title('0.1逆滤波复原');
subplot(2,2,4);imshow(L, []);title('0.01逆滤波复原');
```

5.3 有约束复原

- 有约束图像复原技术是指除了要求了解关于退化系统的传递函数之外，还需要知道某些噪声的统计特性或噪声与图像的某些相关情况。
- 根据所了解的噪声的先验知识的不同，采用不同的约束条件，从而得到不同的图像复原技术。

5.3.1有约束的最小二乘法图像复原

- 在约束最小二乘法复原问题中,令Q为f的线性算子,要设法寻找一个最优估计$\hat{f}$,使形式为$\|Q\hat{f}\|$的、服从约束条件$\|n\|^2=\|g-H\hat{f}\|^2$的函数最小化。
- 求这类问题的最小化，采用拉格朗日乘子算法。也就是说，要寻找一个$\hat{f}$，使准则函数为最小：

$$J(\hat{f})=\|Q\hat{f}\|+\lambda(\|g-H\hat{f}\|^2-\|n\|^2)$$

✓式中λ为一常数，是拉格朗日系数。加上约束条件后，就可以按一般求极小值的方法进行求解。

5.3.1有约束的最小二乘法图像复原

- 在对 $\hat{f}$ 微分，并使结果为零，则有：

$$\frac{\partial J(\hat{f})}{\partial \hat{f}} = 2Q^T Q\hat{f} - 2\lambda H^T(g - H\hat{f}) = 0$$

$$Q^T Q\hat{f} + \lambda H^T H\hat{f} - \lambda H^T g = 0$$

$$\frac{1}{\lambda} Q^T Q\hat{f} + H^T H\hat{f} = H^T g$$

$$\hat{f} = (H^T H + \frac{1}{\lambda} Q^T Q)^{-1} H^T g$$

✓式中 1/λ必须调整到约束条件被满足为止。求解公式核心就是如何选用一个合适的变换矩阵Q。

5.3.1有约束的最小二乘法图像复原

- 在选择Q形式不同，就可得到不同类型的有约束的最小二乘方图像复原方法。
  - ✓如选用拉普拉斯算子形式，即使某个函数的二阶导数最小，就可推导出有约束最小二乘法恢复方法。
  - ✓如果选用图像f和噪声n的相关矩阵R_f和R_n表示Q就可以得到维纳滤波复原方法。

5.3.1有约束的最小二乘法图像复原

- J=deconvreg(I, PSF, NP)实现有约束最小二乘法
  - ✓I是退化图像，J是复原图像
  - ✓PSF点扩散函数，NP噪声强度，默认为0

5.3.1有约束的最小二乘法图像复原

- J=deconvreg(I, PSF, NP)实现有约束最小二乘法
  - ✓I是退化图像，J是复原图像
  - ✓PSF点扩散函数，NP噪声强度，默认为0

运动模糊退化图像

最小二乘法复原图像

高斯模糊退化图像

最小二乘法复原图像

5.3.1有约束的最小二乘法图像复原

```
I1 = imread('lena.jpg');
I1 = rgb2gray(I1);
I1 = im2double(I1);
PSF1 = fspecial('motion', 50, 45); %构建点扩散函数
J1 = imfilter(I1, PSF1, 'conv', 'circular'); % 退化图像
L1 = deconvreg(J1, PSF1); % 复原图像

PSF2 = fspecial('gaussian', 10, 6); %构建点扩散函数
J2 = imfilter(I1, PSF2, 'conv', 'circular'); % 退化图像
L2 = deconvreg(J2, PSF2);  % 复原图像
figure(1);
subplot(2,2,1);imshow(J1);title('运动模糊退化图像');
subplot(2,2,2);imshow(L1, []);title('最小二乘法复原图像');
subplot(2,2,3);imshow(J2, []);title('高斯模糊退化图像');
subplot(2,2,4);imshow(L2, []);title('最小二乘法复原图像');
```

5.3.2 维纳滤波方法

- 维纳滤波是假设图像信号可近似看成为平稳随机过程的前提下，按照使原图像和估计图像之间的均方误差达到最小的准则函数来实现图像复原的。

$$\frac{\partial J(\hat{f})}{\partial \hat{f}} = 2Q^T Q\hat{f} - 2\lambda H^T(g - H\hat{f}) = 0$$

改写为：

$$\hat{f} = (H^T H + \frac{1}{\lambda} Q^T Q)^{-1} H^T g$$

5.3.2 维纳滤波方法

- 维纳滤波就是选用变换矩阵，具有下列形式：

$$Q^T Q = R_f^{-1} R_n$$

$$R_f = E(ff^T)$$

$$R_n = E(nn^T)$$

其中R_f和R_n分别是图像f和噪声n的相关矩阵，因为图像f和噪声n的每个元素值都是实数，所以R_f和R_n都是实对称矩阵。E(.) 代表期望值计算.

5.3.2 维纳滤波方法

- 设W为变换矩阵，A和B分别对应于R_f和R_n相应的对角矩阵，根据循环矩阵对角化的性质可知，A和B中的诸元素分别为R_f和R_n中诸元素的傅立叶变换，后面A和B用功率谱$S_f(u,v)$和$S_n(u,v)$表示。

$$\boldsymbol{R}_f = \boldsymbol{W}\boldsymbol{A}\boldsymbol{W}^{-1} \qquad \boldsymbol{R}_n = \boldsymbol{W}\boldsymbol{B}\boldsymbol{W}^{-1}$$

考虑到H也是循环矩阵，也可以对角化，即可以写成：$H=WDW^{-1}$

$$\hat{f} = (WD^*DW^{-1} + \gamma WA^{-1}BW^{-1})^{-1}WD^*W^{-1}g$$

$$W^{-1}\hat{f} = (D^*D + \gamma A^{-1}B)^{-1}D^*W^{-1}g$$

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

5.3.2 维纳滤波方法

- J=deconvwnr(I, PSF, NSR)实现维纳滤波
  - ✓I是退化图像，J是复原图像
  - ✓PSF点扩散函数，NSR信噪比，默认为0

运动模糊退化图像

维纳滤波复原图像

高斯模糊退化图像

维纳滤波复原图像

5.3.2 维纳滤波方法

```
I1 = imread('lena.jpg');
I1 = rgb2gray(I1);
I1 = im2double(I1);
PSF1 = fspecial('motion', 50, 45); %构建点扩散函数
J1 = imfilter(I1, PSF1, 'conv', 'circular'); % 退化图像
L1 = deconvwnr(J1, PSF1); % 复原图像

PSF2 = fspecial('gaussian', 10, 6); %构建点扩散函数
J2 = imfilter(I1, PSF2, 'conv', 'circular'); % 退化图像
L2 = deconvwnr(J2, PSF2);  % 复原图像
figure(1);
subplot(2,2,1);imshow(J1);title('运动模糊退化图像');
subplot(2,2,2);imshow(L1, []);title('维纳滤波复原图像');
subplot(2,2,3);imshow(J2, []);title('高斯模糊退化图像');
subplot(2,2,4);imshow(L2, []);title('维纳滤波复原图像');
```

5.3.3 盲解卷积图像复原

- 实际情况，通常不知道PSF，此时需要结合一定的先验知识和约束条件来进行图像复原。

[J, PSF]=deconvblind(I, INITPSF, NUMIT)

✓ I是退化图像，J是复原图像
✓ INITPSF是预设的初始PSF
✓ NUMIT是迭代次数

5.3.3 盲解卷积图像复原

运动模糊退化图像(10,0)

运动模糊退化图像(30,90)

高斯模糊退化图像(10,6)

盲解复原图像

盲解复原图像

盲解复原图像

5.3.3 盲解卷积图像复原

```
I1 = imread('lena.jpg');
I1 = rgb2gray(I1);
I1 = im2double(I1);
PSF1 = fspecial('motion', 10, 0); %构建点扩散函数
J1 = imfilter(I1, PSF1, 'conv', 'circular'); % 退化图像
INITPSF1 = ones(size(PSF1)); %初始PSF
L1 = deconvblind(J1, INITPSF1, 10); % 复原图像
PSF2 = fspecial('motion', 30, 90); %构建点扩散函数
J2 = imfilter(I1, PSF2, 'conv', 'circular'); % 退化图像
INITPSF2 = ones(size(PSF2)); %初始PSF
L2 = deconvblind(J2, INITPSF2, 10); % 复原图像
PSF3 = fspecial('gaussian', 10, 6); %构建点扩散函数
J3 = imfilter(I1, PSF3,'conv', 'circular'); % 退化图像
INITPSF3 = ones(size(PSF3)); %初始PSF
L3 = deconvblind(J3, INITPSF3, 30);  % 复原图像
```