

数字图像处理

第4章 图像复原

信息科学研究所

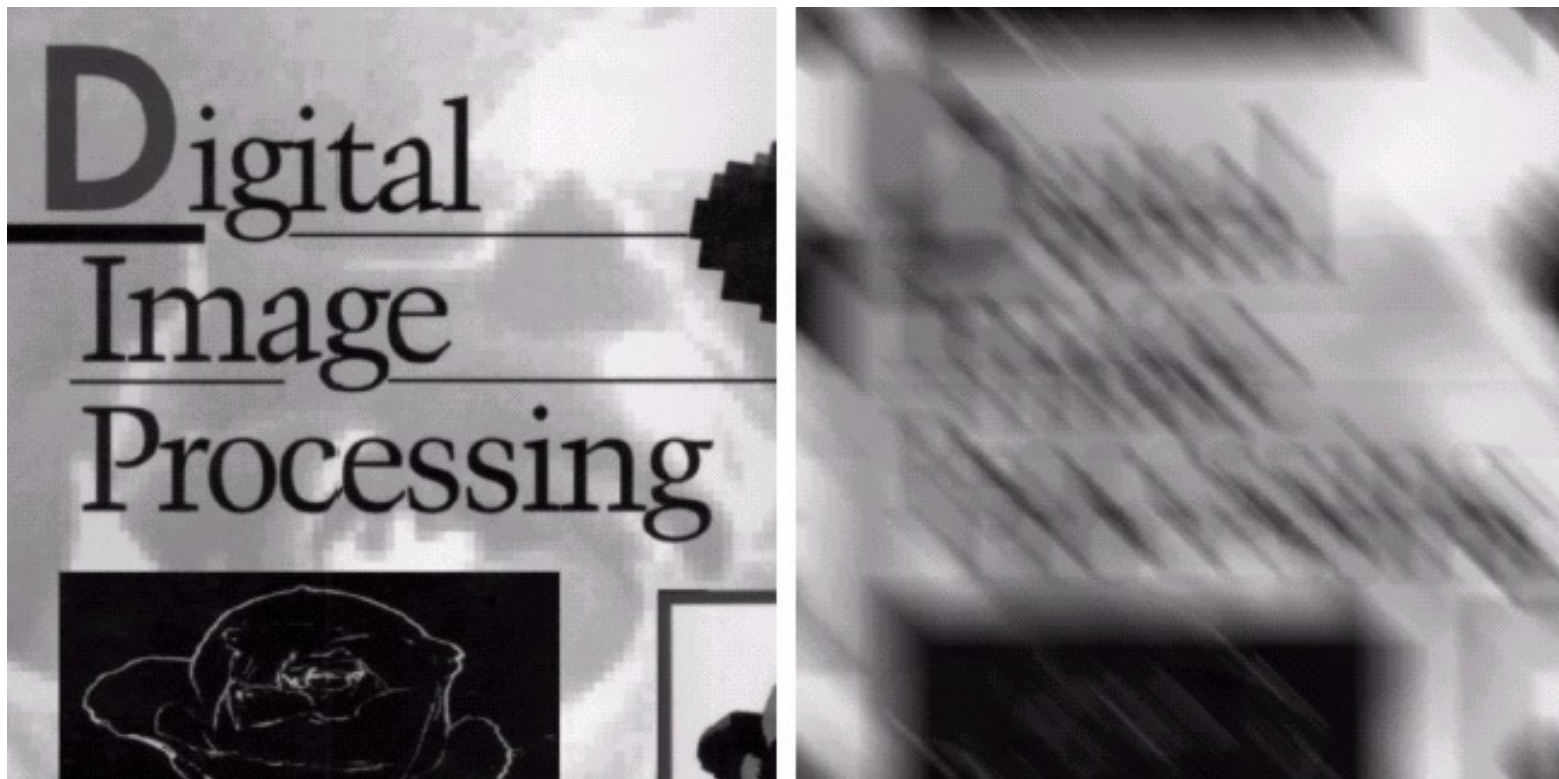
图像复原是图像处理的另一重要课题。它的主要目的是改善给定图像的质量。当给定了一幅退化的或者受到噪声污染的图像后，利用退化现象的某种先验知识来重建或恢复原有图像是图像复原处理的基本过程。

可能的退化有：

光学系统中的衍射，传感器非线性畸变，光学系统的像差，摄影胶片的非线性，大气湍流的扰动，图像运动造成的模糊，几何畸变等。



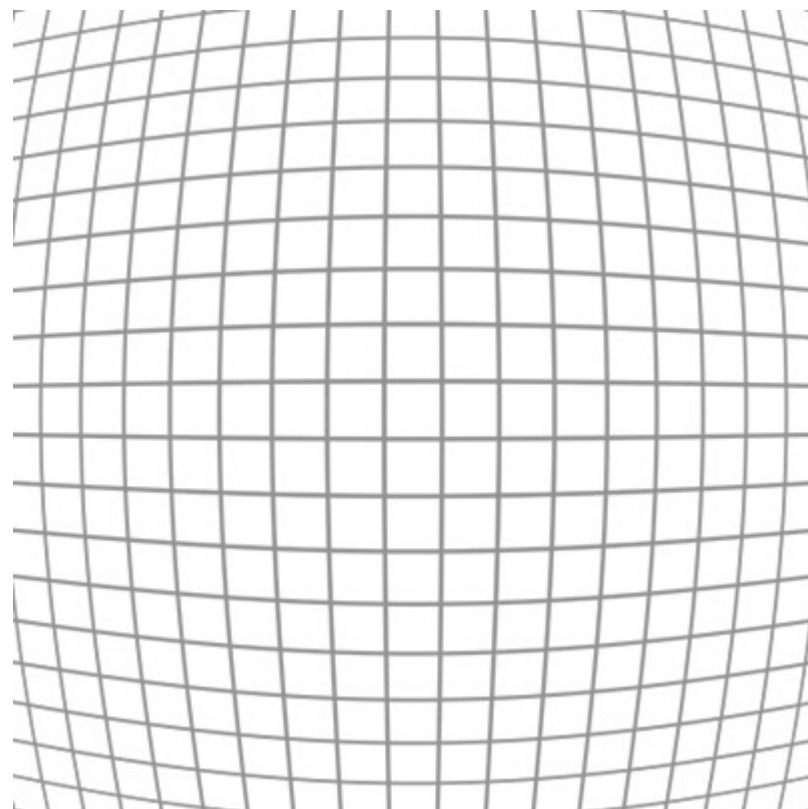
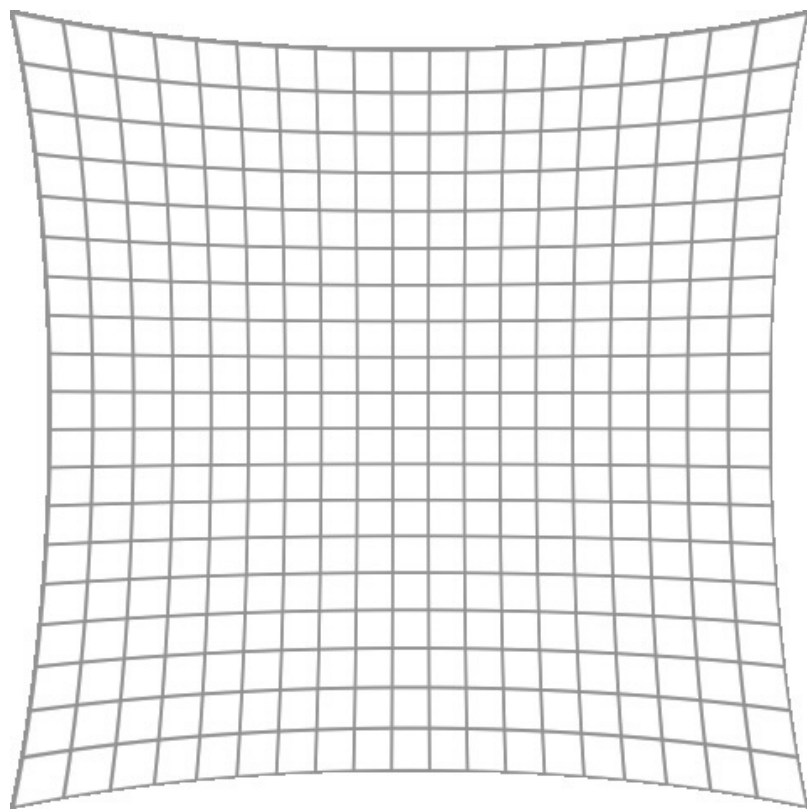
大气湍流造成的图像退化



运动模糊的图像



散焦造成的图像退化



枕形及桶形失真

噪声干扰可以由电子成像系统传感器、信号传输过程或者胶片颗粒性造成的。各种退化图像的复原都可归结为一种过程：

把退化模型化，然后采用相反的过程进行处理，以便恢复出原图像。

6.1 退化模型

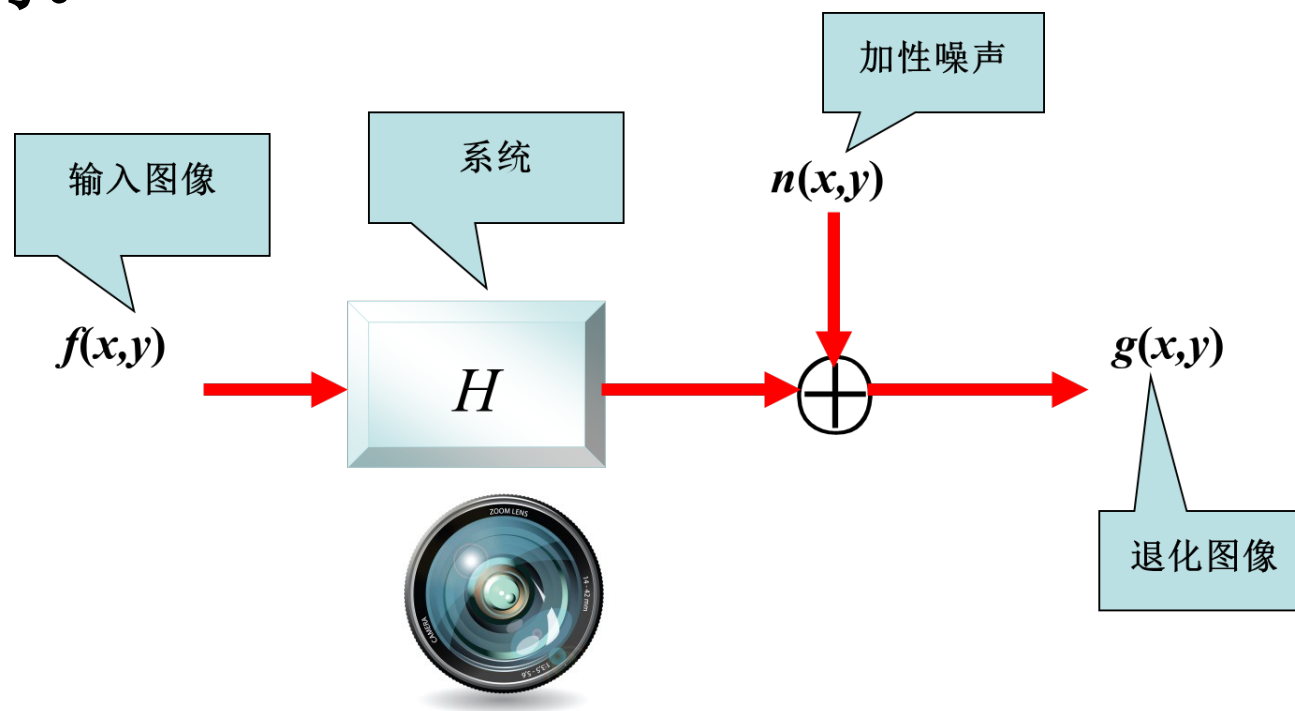
图像复原处理的关键问题在于建立退化模型。在用数学方法描述图像时，普遍的数学表达式为：

$$I = f(x, y, z, \lambda, t)$$

当研究的是静止的、单色的、平面的图像时，则其数学表达式就简化为：

$$I = f(x, y)$$

基于这样的数学表达式，可建立退化模型如图所示：
一幅纯净的图像 $f(x, y)$ 是由于通过了一个系统 H 及加入外来加性噪声 $n(x, y)$ 而使其退化为一幅图像 $g(x, y)$ 的。



图像复原可以看成是一个估计过程：如果已经给出了退化图像 $g(x,y)$ 并估计出系统参数 H ，从而可近似地恢复 $f(x,y)$ 。当然，为了对处理结果作出某种最佳的估计，一般应先明确一个质量标准。

6.1.1 系统 H 的基本定义

由图像的退化模型及复原的基本过程可见，复原处理的关键在于对系统 H 的基本了解。一般而言，系统是某些元件或部件以某种方式构造而成的整体，系统本身所具有的某些特性构成了通过系统的输入信号与输出信号的某种联系。

系统的分类方法很多。如：

线性系统和非线性系统；

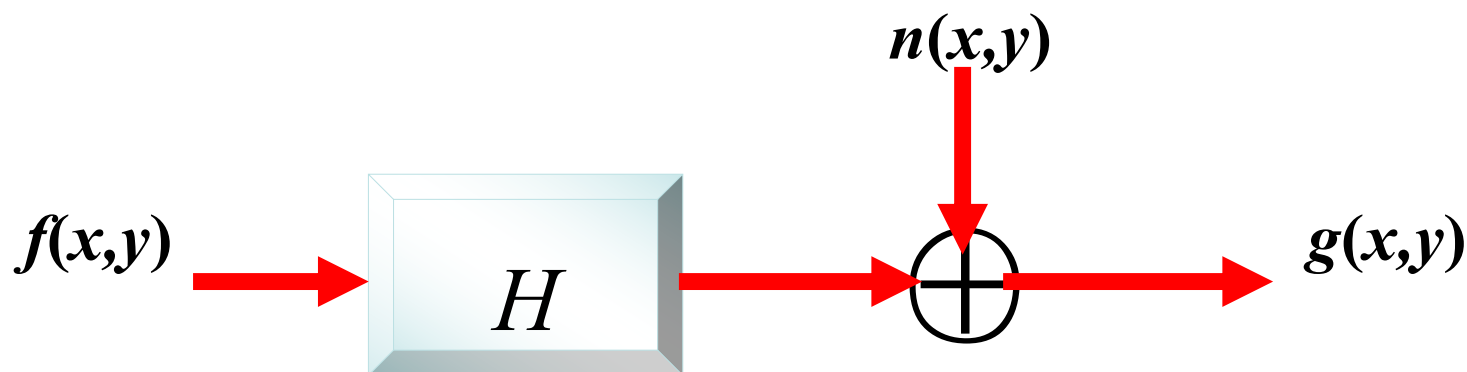
时变系统和非时变系统；

集中参数系统和分布参数系统；

连续系统和离散系统等。

线性系统就是具有均匀性和相加性的系统。对于如图所示的系统来说，可表示成下数学表达式：

$$g(x, y) = H \cdot [f(x, y)] + n(x, y) \quad (6-1)$$



如果暂不考虑加性噪声 $n(x,y)$ 的影响，而令 $n(x,y)=0$ 时，则：

$$g(x, y) = H \cdot [f(x, y)] \quad (6-2)$$

如果输入信号为 $f_1(x,y)$, $f_2(x,y)$, 对应的输出信号为 $g_1(x,y)$, $g_2(x,y)$, 通过系统后有下式成立

$$\begin{aligned} & H \cdot [k_1 f_1(x, y) + k_2 f_2(x, y)] \\ &= H \cdot [k_1 f_1(x, y)] + H \cdot [k_2 f_2(x, y)] \quad (6-3) \\ &= k_1 g_1(x, y) + k_2 g_2(x, y) \end{aligned}$$

那么，系统 H 是一个线性系统。其中 k_1, k_2 为一常数。如果 $k_1 = k_2 = 1$ ，则

$$\begin{aligned} & H \cdot [f_1(x, y) + f_2(x, y)] \\ &= H \cdot [f_1(x, y)] + H \cdot [f_2(x, y)] \quad (6-4) \\ &= g_1(x, y) + g_2(x, y) \end{aligned}$$

式(6—3)及式(6—4)说明，如果 H 为线性系统，
那么，两个输入之和的响应等于两个响应之和。

显然，线性系统的特性为求解多个激励情况下的
输出响应带来很大方便。

$$H \cdot [f(x - \alpha, y - \beta)] \\ = g(x - \alpha, y - \beta)$$

H 是空间不变系统（或称为位置不变系统），式中的 α 和 β 分别是空间位置的位移量。这说明图像中任一点通过该系统的响应只取决于该点的输入值，而与该点的位置无关。

由上述基本定义可见，如果系统 H 有式(6—3)和式(6—5)的关系，那么，系统就是线性的和空间位置不变的系统。

在图像复原处理中，尽管非线性和空间变化的系统模型更具普遍性和准确性，但是，它却给处理工作带来巨大的困难，常常没有解或者很难用计算机来处理。

因此，在图像复原处理中，往往用**线性和空间不变系统模型**加以近似。这种近似的优点是使线性系统理论中的许多理论可直接用来解决图像复原问题，所以图像复原处理特别是数字图像复原处理主要采用线性的、空间不变的复原技术。

6.1.2 连续函数形式的退化模型

在线性系统理论中，曾定义了单位冲激信号 $\delta(t)$ 。

它是一个振幅在零点之外所有时刻为零，在零点处振幅为无限大、宽度无限小，面积为1的窄脉冲。

$$\begin{cases} \int_{-\infty}^{\infty} \delta(t) dt = 1 & t = 0 \\ \delta(t) = 0 & t \neq 0 \end{cases} \quad (6-6)$$

如果冲激信号 $\delta(t)$ 有一个时刻 t_0 的延迟, 那么

$$\left\{ \begin{array}{ll} \int_{-\infty}^{\infty} \delta(t - t_0) dt = 1 & t = t_0 \\ \delta(t - t_0) = 0 & t \neq t_0 \end{array} \right. \quad (6-7)$$

冲激信号 $\delta(t)$ 的一个重要特性是取样特性。由于除了 $t=0$ 外，其它值均为零，所以有：

$$\begin{aligned}\int_{-\infty}^{\infty} \delta(t) f(t) dt &= \int_{-\infty}^{\infty} \delta(t) f(0) dt \\ &= f(0) \int_{-\infty}^{\infty} \delta(t) dt = f(0)\end{aligned}\tag{6—8}$$

同理，当 t 有 t_0 延时的时候有

$$\int_{-\infty}^{\infty} \delta(t-t_0)f(t)dt = \int_{-\infty}^{\infty} \delta(t-t_0)f(t_0)dt$$

(6—9)

$$=f(t_0)\int_{-\infty}^{\infty} \delta(t-t_0)dt = f(t_0)$$

冲激函数的另外一个取样公式就是卷积取样，即

$$\int_{-\infty}^{\infty} f(x-t)\delta(t)dt = f(x) \quad (6-10)$$

一维时域冲激函数 $\delta(t)$ 不难推广到二维空间域中。

如果推广至二维空间，那么可定义 $\delta(x,y)$ 为冲激函

数。 $\delta(x-\alpha,y-\beta)$ 就是有延迟的冲激函数。

显然根据卷积取样，可以把 $f(x,y)$ 写成下式形式：

$$f(x, y) = \int \int_{-\infty}^{\infty} f(\alpha, \beta) \delta(x - \alpha, y - \beta) d\alpha d\beta$$

(6—11)

$$\int_{-\infty}^{\infty} f(x - t) \delta(t) dt = f(x)$$

根据 $g(x, y) = H \cdot [f(x, y)] + n(x, y)$ 的关系,

如果令 $n(x, y) = 0$, 则有以下式成立

$$g(x, y) = H \cdot [f(x, y)]$$

$$= H \cdot \left[\int \int_{-\infty}^{\infty} f(\alpha, \beta) \delta(x - \alpha, y - \beta) d\alpha d\beta \right]$$

由于 H 是线性算子，所以

$$\begin{aligned} g(x, y) &= H \cdot [f(x, y)] \\ &= H \cdot \left[\int \int_{-\infty}^{\infty} f(\alpha, \beta) \delta(x - \alpha, y - \beta) d\alpha d\beta \right] \\ &= \int \int_{-\infty}^{\infty} H \cdot [f(\alpha, \beta) \delta(x - \alpha, y - \beta)] d\alpha d\beta \\ &= \int \int_{-\infty}^{\infty} f(\alpha, \beta) H \cdot \delta(x - \alpha, y - \beta) d\alpha d\beta \end{aligned}$$

(6—12)

令 $h(x, \alpha, y, \beta) = H \cdot \delta(x - \alpha, y - \beta)$

则

$$g(x, y) = \int \int_{-\infty}^{\infty} f(\alpha, \beta) h(x, \alpha, y, \beta) d\alpha d\beta$$

(6—13)

$h(x, \alpha, y, \beta) = H \cdot \delta(x - \alpha, y - \beta)$ 就是系统 H 的冲激响应。也就是说 $h(x, \alpha, y, \beta)$ 是系统 H 对坐标为 α, β 处的冲激函数 $\delta(x - \alpha, y - \beta)$ 的响应。在光学中，冲激为一光点，所以 $h(x, \alpha, y, \beta)$ 又称为点扩散函数（PSF）。

式(6—13)就是线性系统理论中非常重要的费雷德霍姆(*fredholm*)积分：如果系统 H 对冲激函数的响应为已知，则对任意输入 $f(\alpha, \beta)$ 的响应可用式(6—13)求得。换句话说，**线性系统 H 完全可由其冲激响应来表征。**

$$g(x, y) = \int \int_{-\infty}^{\infty} f(\alpha, \beta) h(x, \alpha, y, \beta) d\alpha d\beta$$

在空间位置不变的情况下

$$H \cdot \delta(x - \alpha, y - \beta) = h(x - \alpha, y - \beta) \quad (6-14)$$

在这种情况下，显然

$$g(x, y) = \int \int_{-\infty}^{\infty} f(\alpha, \beta) h(x - \alpha, y - \beta) d\alpha d\beta$$

(6-15)

这说明，系统 H 加入输入信号的响应就是系统输入信号与冲激响应的卷积积分。

在有加性噪声的情况下，前述的线性退化模型可表示为

$$g(x, y) = \int \int_{-\infty}^{\infty} f(\alpha, \beta) h(x - \alpha, y - \beta) d\alpha d\beta + n(x, y)$$

(6—16)

当然，在上述情况中，都假设噪声与图像中的像素位置无关。

6.1.3 离散形式的退化模型

连续函数的退化模型可由输入函数 $f(\alpha, \beta)$ 和点扩散函数相乘再积分表示。如把 $f(\alpha, \beta)$ 和 $h(x-\alpha, y-\beta)$ 进行均匀取样后就可引伸出离散的退化模型。

为了研究离散的退化模型，不妨用一维函数来说明基本概念，然后再推广至二维情况。

函数 $f(x)$ 和 $h(x)$ 被均匀取样后分别形成 A 维和 B 维的阵列： $f(x)$ 变成 $x=0, 1, 2, \cdots A-1$ 范围内的离散变量， $h(x)$ 变成 $x=0, 1, 2, \cdots B-1$ 范围内的离散变量。

由此，连续函数退化模型中的连续卷积关系就演变为离散卷积关系。

如果 $f(x)$, $h(x)$ 都是具有周期为 N 的序列, 那么,
它们的时域离散卷积可定义:

$$g(x) = \sum f(m)h(x - m) \quad (6-17)$$

显然， $g(x)$ 也是具有周期 N 的序列。周期卷积可用常规卷积法计算也可用卷积定理进行快速卷积计算。

如果 $f(x)$ 和 $h(x)$ 均不具备周期性，则可以用延拓的方法使其成为周期函数。为了避免折叠现象，可以令周期 $M \geq A+B-1$ ，使 $f(x)$ ， $h(x)$ 分别延拓为下列离散阵列

$$f_e(x) = \begin{cases} f(x) & 0 \leq x \leq A-1 \\ 0 & A-1 < x \leq M-1 \end{cases}$$

(6—18)

$$h_e(x) = \begin{cases} h(x) & 0 \leq x \leq B-1 \\ 0 & B-1 < x \leq M-1 \end{cases}$$

这样延拓后，可得到一个离散卷积退化模型

$$g_e(x) = \sum_{m=0}^{M-1} f_e(m)h_e(x-m) \quad (6-19)$$

式中 $x=0, 1, 2, \cdots, M-1$ 。显然， $g_e(x)$ 的周期也是 M 。

经过这样的延拓处理，一个非周期的卷积问题就变成了周期卷积问题了，因此也可以用快速卷积法进行运算。

如果用矩阵来表示上述离散退化模型，可写成下式之形式

$$[g] = [H] [f] \quad (6-20)$$

这里 $[g]$, $[H]$, $[f]$ 分别代表矩阵，其中

$$[f] = \begin{bmatrix} f_e(0) \\ f_e(1) \\ \vdots \\ \vdots \\ f_e(M-1) \end{bmatrix}$$

(6—21)

$$[g] = \begin{bmatrix} g_e(0) \\ g_e(1) \\ \vdots \\ \vdots \\ g_e(M-1) \end{bmatrix}$$

(6—22)

$[H]$ 是 $M \times M$ 阶矩阵

$$[H] = \begin{bmatrix} h_e(0) & h_e(-1) & h_e(-2) & \cdots h_e(-M+1) \\ h_e(1) & h_e(0) & h_e(-1) & \cdots h_e(-M+2) \\ h_e(2) & h_e(1) & h_e(0) & \cdots h_e(-M+3) \\ \cdots & \cdots \cdots & \cdots \cdots & \cdots \cdots \\ h_e(M-1) & h_e(M-2) & h_e(M-3) & \cdots \cdots h_e(0) \end{bmatrix}$$

(6—23)

由于 $h_e(x)$ 具有周期性，所以

$$h_e(x)=h_e(M+x) \text{ ,}$$

利用这一性质，式（6—23）又可以写成下式形式

$$[H] = \begin{bmatrix} h_e(0) & h_e(M-1) & h_e(M-2) & \cdots h_e(1) \\ h_e(1) & h_e(0) & h_e(M-1) & \cdots h_e(2) \\ h_e(2) & h_e(1) & h_e(0) & \cdots h_e(3) \\ \cdots \cdots & \cdots \cdots & \cdots \cdots & \cdots \cdots \\ h_e(M-1) & h_e(M-2) & h_e(M-3) & \cdots \cdots h_e(0) \end{bmatrix}$$

(6—24)

由于 $h_e(x)$ 的周期性，使得 $[H]$ 成为一个循环矩阵。

上述基本模型不难推广至二维情况。如果给出 $A \times B$ 大小的数字图像以及 $C \times D$ 大小的点扩散函数，可首先作成大小为 $M \times N$ 的周期延拓图像，即：

$$f_e(x, y) = \begin{cases} f(x, y) & 0 \leq x \leq A-1 \\ & 0 \leq y \leq B-1 \\ & A \leq x \leq M-1 \\ 0 & B \leq y \leq N-1 \end{cases} \quad (6-25)$$

$$h_e(x, y) = \begin{cases} h(x, y) & 0 \leq x \leq C-1 \\ & 0 \leq y \leq D-1 \\ & C \leq x \leq M-1 \\ 0 & D \leq y \leq N-1 \end{cases} \quad (6-26)$$

这样延拓后 $f_e(x,y)$ 和 $h_e(x,y)$ 分别成为二维周期函数。

它们在 x 和 y 方向上的周期分别为 M 和 N 。由此得到

二维退化模型为一个二维卷积形式

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

(6—27)

式中： $x=0, 1, 2, \cdots M-1$ ； $y=0, 1, 2, \cdots N-1$ ，卷积函数 $g_e(x,y)$ 也为周期函数，其周期与 $f_e(x,y)$ 和 $h_e(x,y)$ 一样。

为避免重叠，同样要按下式规则延拓

$$M \geq A + C - 1, N \geq B + D - 1 \quad (6-28)$$

式(6-27)的模型同样可用矩阵来表示

$$[g] = [H] [f] \quad (6-29)$$

其中 $[g]$, $[f]$ 代表 $M \times N$ 维列向量。这些列向量是由 $M \times N$ 维的函数矩阵 $[f_e(x,y)]$, $[g_e(x,y)]$ 的各列堆积而成的。

例如 $[f]$ 的第一组 N 个元素是 $[f_e(x,y)]$ 的第一列元素，第二组 N 个元素是由 $[f_e(x,y)]$ 的第二列元素得到的等等。因此，式(6—29)中的 $[g]$ 和 $[f]$ 是 MN 维向量矩阵，即 $[g], [f]$ 为 $(MN) \times 1$ 维矩阵。而 $[H]$ 为 $MN \times MN$ 维矩阵，即：

$$[H] = \begin{bmatrix} [H_0] & [H_{M-1}] & [H_{M-2}] & \cdots [H_1] \\ [H_1] & [H_0] & [H_{M-1}] & \cdots [H_2] \\ [H_2] & [H_1] & [H_0] & \cdots [H_3] \\ \dots\dots & \dots\dots & \dots\dots & \dots\dots \\ [H_{M-1}] & [H_{M-2}] & [H_{M-3}] & \cdots\dots\dots [H_0] \end{bmatrix}$$

(6—30)

每个部分 $[H_j]$ 是由延拓函数 $h_e(x,y)$ 的 j 行构成的，构成方法如下式：

$$[H_j] = \begin{bmatrix} h_e(j,0) & h_e(j,N-1) & h_e(j,N-2) & \cdots h_e(j,1) \\ h_e(j,1) & h_e(j,0) & h_e(j,N-1) & \cdots h_e(j,2) \\ h_e(j,2) & h_e(j,1) & h_e(j,0) & \cdots h_e(j,3) \\ \cdots & \cdots & \cdots & \\ \cdots & \cdots & \cdots & \\ h_e(j,N-1) & h_e(j,N-2) & h_e(j,N-3) & \cdots h_e(j,0) \end{bmatrix}$$

(6—31)

这里 $[H_j]$ 是一个循环矩阵, $[H]$ 的分块 $[H_j]$ 的下标也是循环方式标注。因此, $[H]$ 是一个分块循环矩阵。

一个更加完善的退化模型应加上噪声项。所以**离散退化模型**的完整形式如下式所示

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

(6—32)

其矩阵形式如下

$$[g] = [H] [f] + [n] \quad (6-33)$$

式中 $[n]$ 也是 MN 维列向量。

上述离散退化模型都是在线性空间不变的前提下推出的。目的是在给定了 $g(x,y)$ 并且知道 $h(x,y)$ 和 $n(x,y)$ 的情况下，估计出理想的原始图像 $f(x,y)$ 。但是，要想从式(6—33)得到 $f(x,y)$ ，对于实用大小的图像来说，处理工作是十分艰巨的。

$$[g] = [H] [f] + [n]$$

例如，对于一般精度的图像来说

$M=N=512$ ，此时 $[H]$ 的大小为：

$$\begin{aligned} MN \times MN &= (512)^2 \times (512)^2 \\ &= 262144 \times 262144 \end{aligned}$$

按8bit算，约500G

因此，要直接得到 $[f]$ 则需要求解262144个联立方程组。其计算量之浩大是不难想象的。为了解决这样的问题，必须研究一些简化算法，由于 $[H]$ 的循环性质，使得简化运算得以实现。

6.2 逆滤波

6.2.1 逆滤波的基本原理

逆滤波复原法也叫做反向滤波法。基本原理如下：

如果已知退化图像为 $g(x,y)$ ，求原始图像为 $f(x,y)$ ，在不考虑噪声的情况下，其退化模型用下式表示

$$g(x, y) = \int \int_{-\infty}^{\infty} f(\alpha, \beta) h(x - \alpha, y - \beta) d\alpha d\beta \quad (6-45)$$

上式显然是一卷积表达式。由傅立叶变换的卷积定理可知有下式成立

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

式中 $G(u,v)$ ， $H(u,v)$ ， $F(u,v)$ 分别是退化图像 $g(x, y)$ ，点扩散函数 $h(x,y)$ ，原始图像 $f(x,y)$ 的傅立叶变换。

由式(6—46)，可得

$$F(u, v) = \frac{G(u, v)}{H(u, v)} \quad (6—47)$$

$$f(x, y) = \mathbf{F}^{-1}[F(u, v)] = \mathbf{F}^{-1}\left[\frac{G(u, v)}{H(u, v)}\right] \quad (6—48)$$

这意味着，如果已知退化图像的傅立叶变换和“滤波”传递函数，则可以求得原始图像的傅立叶变换，经反傅立叶变换就可求得原始图像 $f(x,y)$ 。这里 $G(u,v)$ 除以 $H(u,v)$ 起到了反向滤波的作用。这就是逆滤波法复原的基本原理。

在有噪声的情况下，逆滤波原理可写成如下形式

$$G(u, v) = H(u, v)F(u, v) + N(u, v) \quad (6-49)$$

$$F(u, v) = \frac{G(u, v)}{H(u, v)} - \frac{N(u, v)}{H(u, v)} \quad (6-50)$$

式中 $N(u, v)$ 是噪声 $n(x, y)$ 的傅立叶变换。

利用式(6—47)和式(6—50)进行复原处理时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 $N(u,v)$ 的值小的多，因此由式（6—50）得到的噪声项可能会非常大，这样也会使 $f(x,y)$ 不能正确恢复。

一般来说，逆滤波法不能正确地估计 $H(u,v)$ 的零点，因此必须采用一个折衷的方法加以解决。实际上逆滤波不是用 $1/H(u,v)$ ，而是采用另外一个关于 u,v 的函数 $M(u,v)$ 。它的处理框图如图6—2所示。

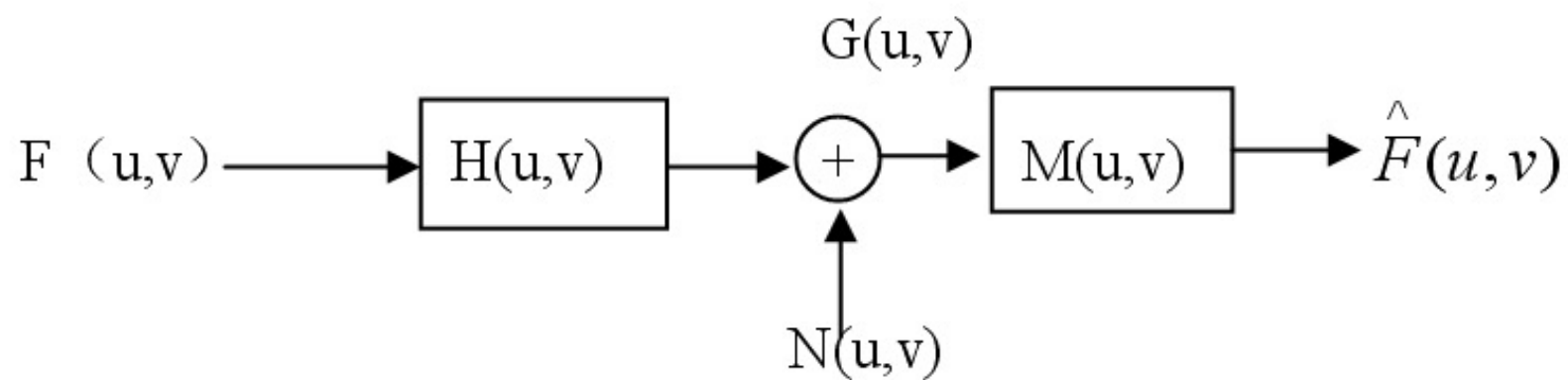


图 6—2 实际的逆滤波处理框图

在没有零点并且也不存在噪声的情况下

$$M(u, v) = \frac{1}{H(u, v)}$$

图6—2的模型包括了退化和恢复运算。退化和恢复总的传递函数可用 $H(u,v)M(u,v)$ 来表示。此时有：

$$\hat{F}(u, v) = [H(u, v)M(u, v)]F(u, v)$$

(6—51)

式中 $\hat{f}(x, y)$ 是 $f(x, y)$ 的估计值, $\hat{F}(u, v)$ 是其傅立叶变换。 $H(u, v)$ 叫做输入传递函数, $M(u, v)$ 叫做处理传递函数, $H(u, v)M(u, v)$ 叫做输出传递函数。

一般情况下， $H(u,v)$ 的幅度随着离 u,v 平面原点的距离的增加而迅速下降，而噪声项 $N(u,v)$ 的幅度变化是比较平缓的。在远离 u,v 平面的原点时 $N(u,v)/H(u,v)$ 的值就会变得很大，而对于大多数图像来说 $F(u,v)$ 却变小，

在这种情况下，噪声反而占优势，自然无法满意地恢复出原始图像。这一规律说明，应用逆滤波时仅在原点邻域内采用 $1/H(u,v)$ 方能奏效。

换句话说，应使 $M(u, v)$ 在下述范围内来选择

$$M(u, v) = \begin{cases} \frac{1}{H(u, v)} & u^2 + v^2 \leq \omega_0^2 \\ 1 & u^2 + v^2 > \omega_0^2 \end{cases}$$

(6—52)

ω_0 的选择应该将 $H(u, v)$ 的零点排除在此邻域之外。

6.3 中值滤波

对受到噪声污染的退化图像的复原可以采用线性滤波方法来处理，有许多情况下是很有效的。但是多数线性滤波具有低通特性，在去除噪声的同时也使图像的边缘变得模糊了。

中值滤波方法在某些条件下可以作到既去除噪声又保护图像边缘的较满意的复原。

中值滤波是一种去除噪声的非线性处理方法。它是由图基（Turky）在1971年提出的。开始，中值滤波用于时间序列分析，后来被用于图像处理，在去噪复原中得到了较好的效果。

6.3.1 中值滤波的基本原理

中值滤波的基本原理是把数字图像或数字序列中一点的值用该点的一个邻域中各点值的中值代替。

中值的定义如下：

一组数 $x_1, x_2, x_3 \cdots x_n$ ，把 n 个数按值的大小顺序排列：

$$x_{i1} \leq x_{i2} \leq x_{i3} \cdots \leq x_{in}$$

$$y = Med(x_1, x_2, x_3, \cdots x_n)$$

$$= \begin{cases} x_{i(\frac{n+1}{2})} & n \text{ 为奇数} \\ \frac{1}{2} \left[x_{i(\frac{n}{2})} + x_{i(\frac{n}{2}+1)} \right] & n \text{ 为偶数} \end{cases} \quad (6-144)$$

y 称为序列 $x_1, x_2, x_3 \dots x_n$ 的中值。例如有一序列为 (80, 90, 200, 110, 120)，这个序列的中值为110。

把一个点的特定长度或形状的邻域称作窗口。在一维情形下，中值滤波器是一个含有奇数个像素的滑动窗口。窗口正中间那个像素的值用窗口内各像素值的中值代替。

设输入序列为 $\{x_i, i \in I\}$, I 为自然数集合或子集, 窗口长度为 n 。则滤波器输出为

$$y_i = Med \{x_i\} = Med [x_{i-u} \cdots x_i \cdots x_{i+u}] \quad (6-145)$$

$$\text{其中 } i \in I, u = \frac{(n-1)}{2}$$

例如，有一输入序列如下

$$\{x_i\} = \{0\ 0\ 0\ 8\ 0\ 0\ 2\ 3\ 2\ 0\ 2\ 3\ 2\ 0\ 3\ 5\ 3\ 0\ 3\ 5\ 3\ 0\ 0\ 2\ 3\ 4\ 5\ 5\ 5\ 5\ 5\ 0\ 0\ 0\}$$

在此序列中前面的8是脉冲噪声，中间一段是一种寄生振荡，后面是希望保留的斜坡和跳变。在此采用长度为3的窗口，得到的结果为

$$\{y_i\} = \{0\ 0\ 0\ 0\ 0\ 0\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 3\ 3\ 3\ 3\ 3\ 3\ 3\ 0\ 0\ 2\ 3\ 4\ 5\ 5\ 5\ 5\ 0\ 0\ 0\}$$

显然，经中值滤波后，脉冲噪声8被滤除了，振荡被平滑掉了，斜坡和阶跃部分被保存了下来。

中值滤波的运算方法可以作些分析。例如常数 K 与序列 $f(i)$ 相乘的中值有如下关系存在

$$\text{Med}\{Kf(i)\} = K\text{Med}\{f(i)\} \quad (6-146)$$

而常数 K 与序列 $f(i)$ 相加的中值有如下关系

$$\text{Med}\{K + f(i)\} = K + \text{Med}\{f(i)\} \quad (6-147)$$

乘或加上常数 K 不改变原序列的中值

中值滤波的概念很容易推广到二维，此时可以利用某种形式的二维窗口。设 $\{x_{ij}, (i, j) \in I^2\}$ 表示数字图像各点的灰度值，滤波窗口为 A 的二维中值滤波可定义为

$$y_{ij} = Med_A \{x_{ij}\} = Med \{x_{(i+r), (j+s)} \mid (r, s) \in A, (i, j) \in I^2\}$$

(6—148)

二维中值滤波的窗口可以取方形，也可以取近似圆形或十字形。



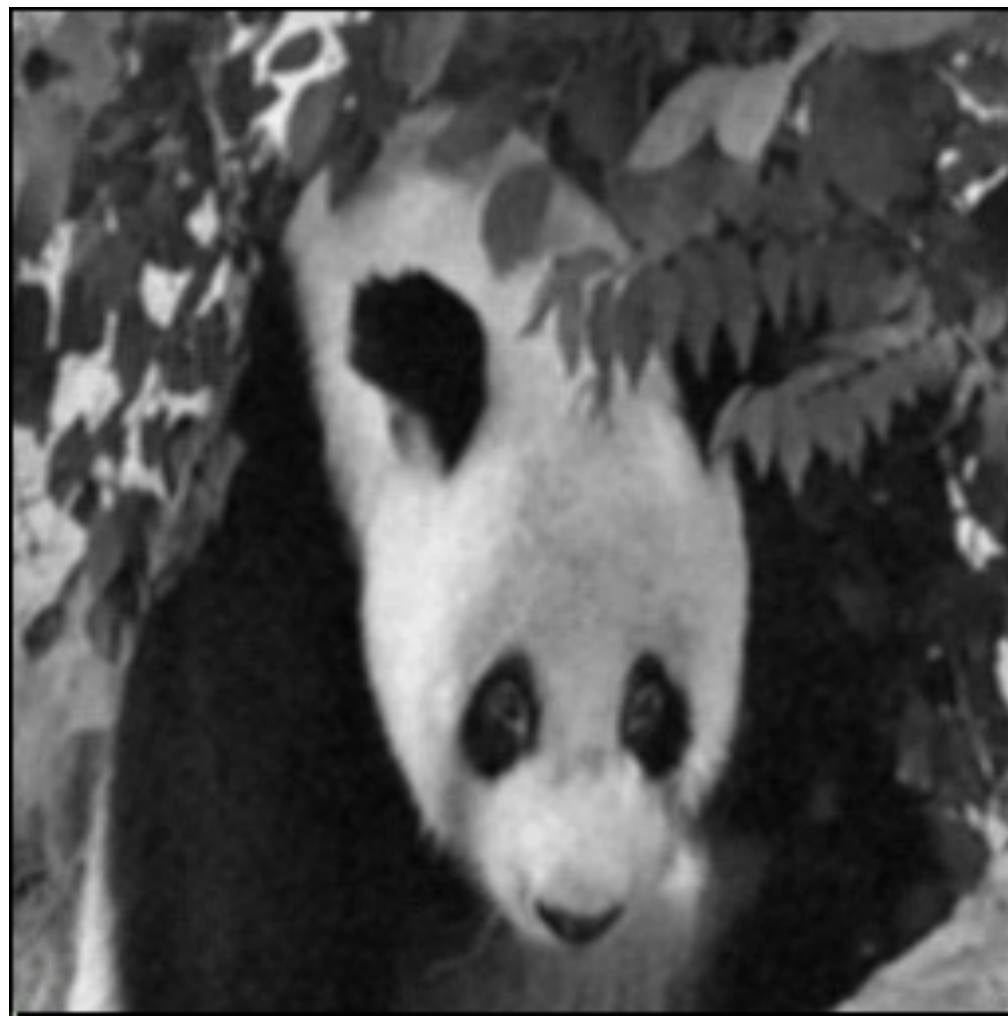
(a) 原像



(b)加有高斯白噪声图像



(c)中值滤波图像



(d) 均值滤波图像



(e) 加有椒盐噪声图像



(f) 中值滤波图像



(g) 均值滤波图像

6.3.2 加权的中值滤波

以上讨论中的中值滤波，窗口内各点对输出的作用是相同的。如果希望强调中间点或距中间点最近的几个点的作用，可以采用加权中值滤波法。加权中值滤波的基本原理是改变窗口中变量的个数，可以使一个以上的变量等于同一点的值，然后对扩张后的数字集求中值。

(1) 一维加权的中值滤波

以窗口为3的一维加权中值滤波为例，表示如下

$$\begin{aligned} y_i &= \textit{Weighted_Med}(x_{i-1}, x_i, x_{i+1}) \\ &= \textit{Med}(x_{i-1}, x_{i-1}, x_i, x_i, x_i, x_{i+1}, x_{i+1}) \end{aligned}$$

$$x_{i-1}$$

$$x_i$$

$$x_{i+1}$$

由公式（6—149）可见，在窗口内，中间点取奇数，两边点取对称数，也就是位于窗口中间的像素重复两次，位于窗口边缘的两个像素重复一次，形成新的序列，然后对新的序列在施以常规中值滤波处理。

(2) 二维的加权中值滤波

二维加权中值滤波与一维情况类似。如果适当地选取窗口内各点的权重，**加权中值滤波比简单中值滤波能更好地从受噪声污染的图像中恢复出阶跃边缘以及其他细节**。二维加权中值滤波以 3×3 窗口为例，表示如下

原始窗口为:

$$x_{i-1,j-1}$$

$$x_{i-1,j}$$

$$x_{i-1,j+1}$$

$$x_{i,j-1}$$

$$x_{i,j}$$

$$x_{i,j+1}$$

$$x_{i+1,j-1}$$

$$x_{i+1,j}$$

$$x_{i+1,j+1}$$

加权后的中值滤波如下式所示：

$$\begin{aligned} y_{ij} = & \textit{Weighted_Med}(x_{i-1,j-1}, x_{i-1,j}, x_{i-1,j+1} \\ & x_{i,j-1}, x_{i,j}, x_{i,j+1}, x_{i+1,j-1}, x_{i+1,j}, x_{i+1,j+1}) \\ = & \textit{Med}(x_{i-1,j-1}, x_{i-1,j}, x_{i-1,j}, x_{i-1,j+1}, x_{i,j-1}, x_{i,j-1}, x_{i,j}, \\ & x_{i,j}, x_{i,j+1}, x_{i,j+1}, x_{i+1,j-1}, x_{i+1,j}, x_{i+1,j}, x_{i+1,j+1}) \end{aligned}$$

(6—150)

即中间的点取三个值（重复两次），上、下、左、右的点各取两个（重复一次），对角线上的点取一个（不重复）。

中值滤波可有效地去除脉冲型噪声，而且对图像的边缘有较好的保护。但是它也有其固有的缺陷，如果使用不当，会损失许多图像细节。例如，采用 3×3 窗口对图6—6(a)所示的原始图像滤波。

1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	5	5	5	5	5	5	1	1
1	1	5	5	8	8	5	5	1	1
1	1	5	5	8	8	5	5	1	1
1	1	5	5	5	5	5	5	1	1
1	1	5	5	5	5	5	5	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1

(a)

1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	5	5	5	5	1	1	1
1	1	5	5	5	5	5	5	1	1
1	1	5	5	5	5	5	5	1	1
1	1	5	5	5	5	5	5	1	1
1	1	5	5	5	5	5	5	1	1
1	1	1	5	5	5	5	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1

(b)

图6—6 中值滤波的实例一

滤波结果如图(b)所示，其结果不但削去了方块的4个角，而且把中间的小方块也滤掉了。因此，中值滤波在选择窗口时要考虑其形状及等效带宽，以避免滤波处理造成的信息损失。

图6—7是中值滤波的另一实例。图(a)是一条细线条图像，经 3×3 窗口滤波后，图像中的细线条完全滤掉了，如图(b)所示。

0	0	1	0	0
0	0	1	0	0
0	0	1	0	0
0	0	1	0	0
0	0	1	0	0

(a)

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

(b)

图6—7 中值滤波的实例二

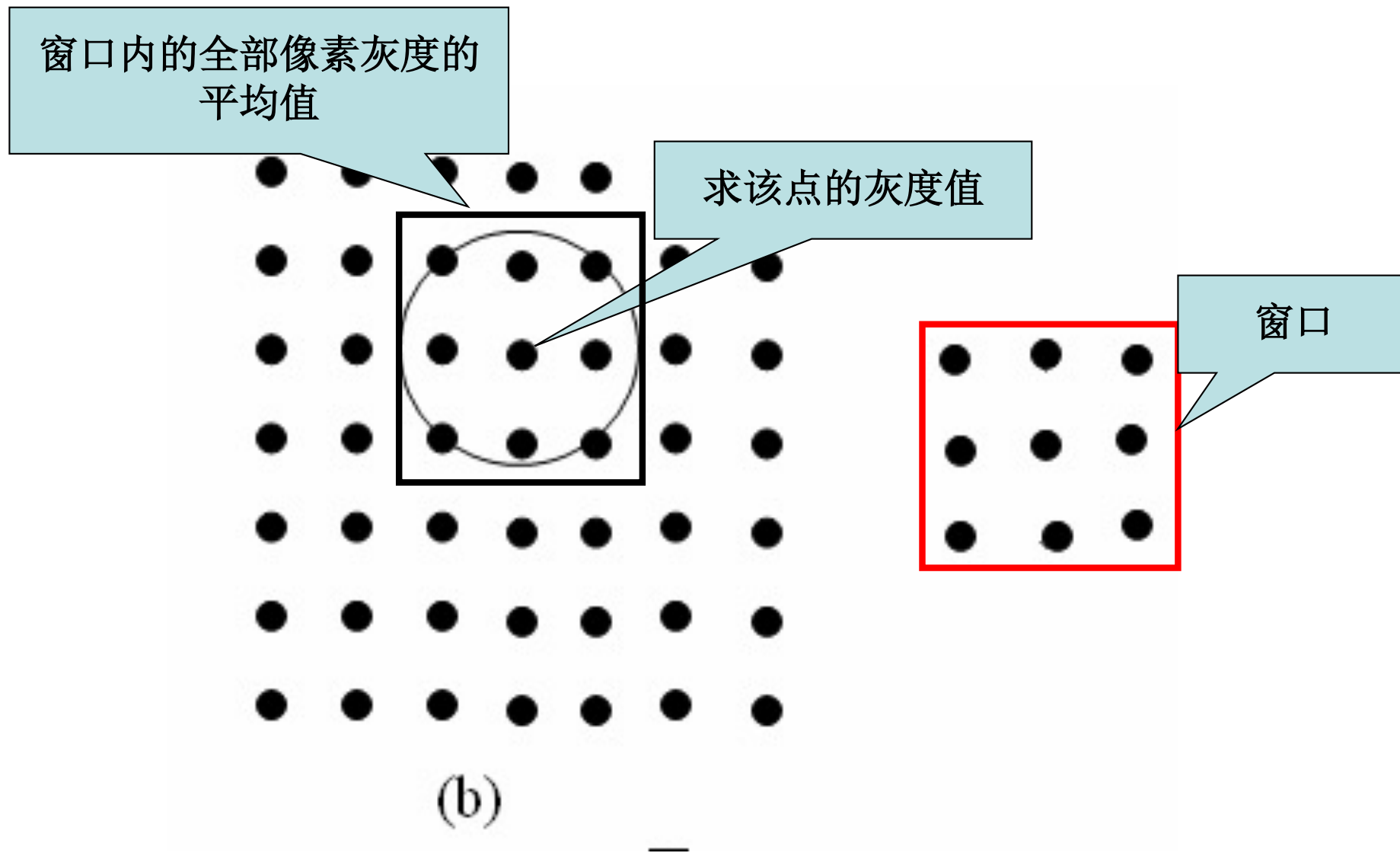
以上两例可以直观地看到，中值滤波对图像中的细节处很不理想，所以，中值滤波对所谓的椒盐噪声 (pepper salt Noise) 的滤除非常有效，但是它对点、线等细节较多的图像却不太适用。

前面图6—4中，为了比较中值滤波的效果，也给出了均值滤波的处理结果。均值滤波的滤波过程也是使一个窗口在图像（或序列）上滑动，窗中心位置的值用窗内各点值的平均值来代替。以二维均值滤波为例，它的定义如下：

设 $\{x_{ij}\}$ 表示数字图像各像素的灰度值, A 为一个 3×3 的窗口, 则二维均值滤波的定义为

$$\begin{aligned} y_{ij} &= Mean(x_{ij}) \\ &= \frac{1}{9} \{ x_{i-1,j-1} + x_{i-1,j} + x_{i-1,j+1} + x_{i,j-1} + x_{i,j} \\ &\quad + x_{i,j+1} + x_{i+1,j-1} + x_{i+1,j} + x_{i+1,j+1} \} \end{aligned}$$

(6—151)



一般均值滤波的边缘保护特性不如中值滤波。

6.4 几种其他空间复原技术

前边讨论了几种基本的图像复原技术。除此之外，尚有一些其他的空间图像复原方法，本节将对这些方法作一些简要介绍。

6.4.1 几何畸变校正

在图像的获取或显示过程中往往会产生几何失真。例如成像系统有一定的几何非线性。这主要是由于视像管摄像机及阴极射线管显示器的扫描偏转系统有一定的非线性，因此会造成如图6—8所示的枕形失真或桶形失真。图(a)为原始图像，图(b)和图(c)为失真图像。

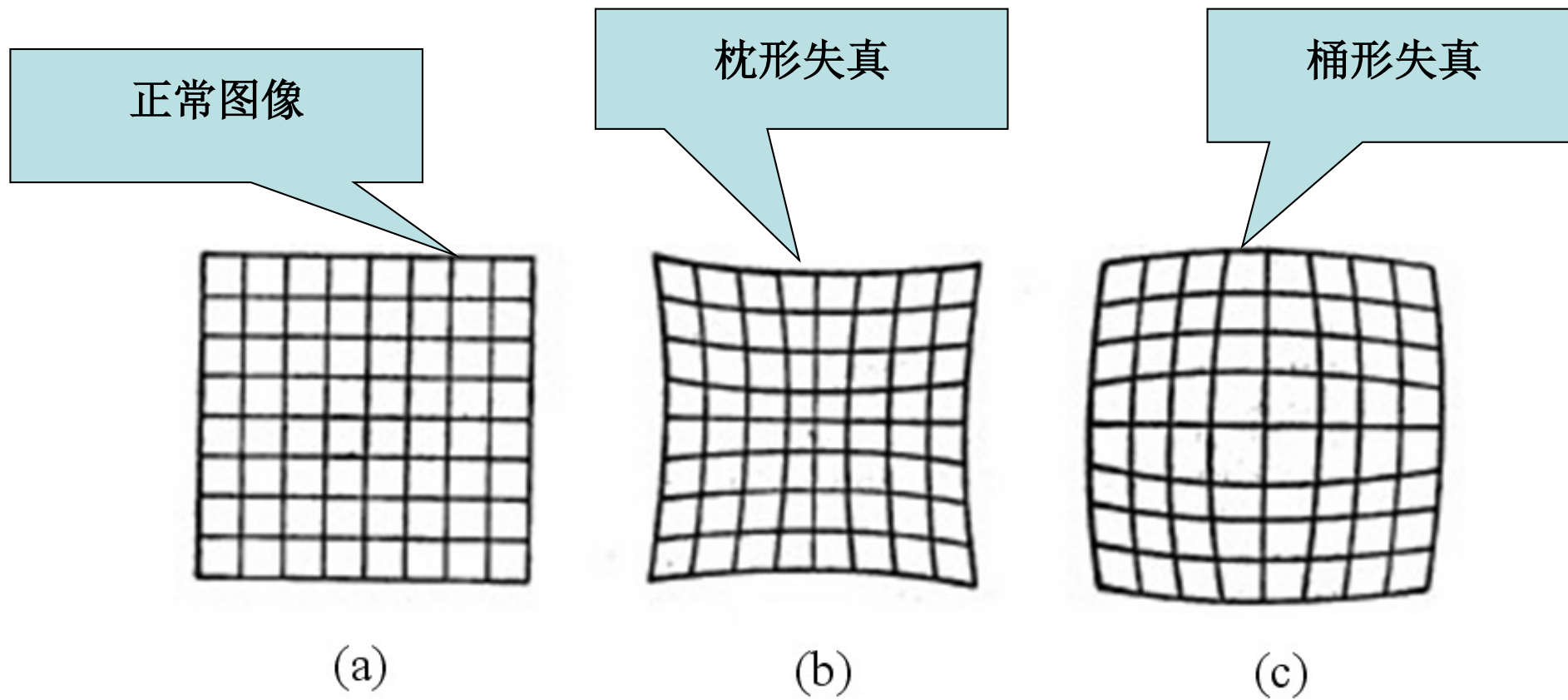
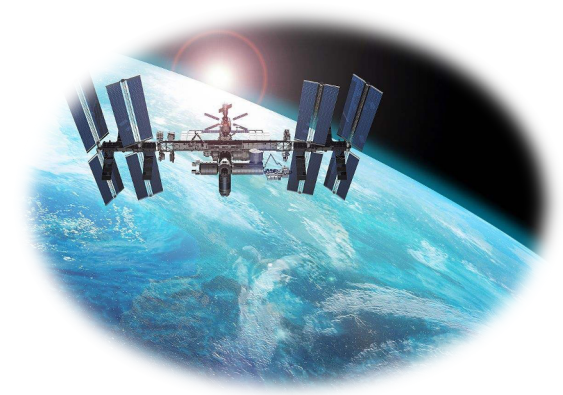


图6—8 几何畸变

除此之外还由于斜视角度获得的图像的透视失真。

另外，由卫星摄取的地球表面的图像往往覆盖较大的面积，由于地球表面呈球形，这样摄取的平面图像也将会有较大的几何失真。对于这些图像必须加以校正，以免影响分析精度。



由成像系统引起的几何畸变的校正有两种方法。一种是**预畸变法**，这种方法是采用与畸变相反的非线性扫描偏转法，用来抵消预计的图像畸变；

另一种是所谓的后验校正方法。这种方法是用多项式曲线在水平和垂直方向去拟合每一畸变的网线，然后求得反变化的校正函数。用这个校正函数即可校正畸变的图像。图像的空间几何畸变及其校正过程如图6—9所示。

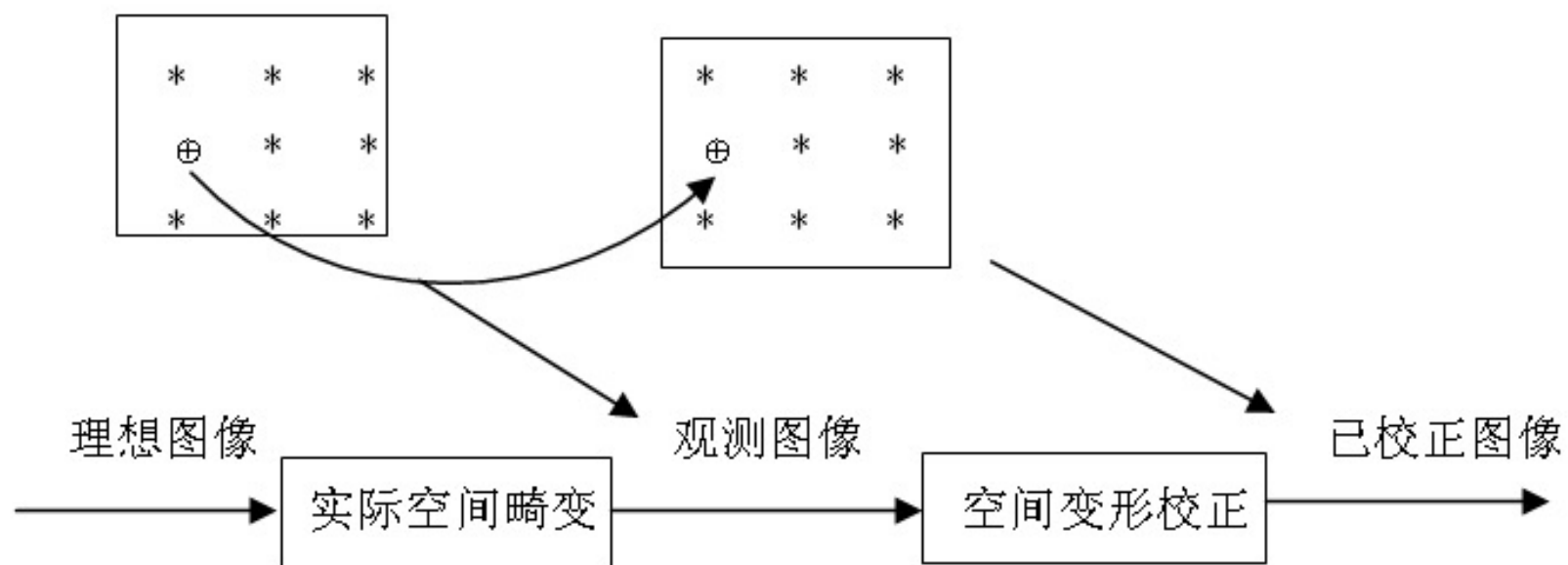


图6—9 空间几何畸变及校正的概念

任意几何失真都可由非失真坐标系 (x, y) 变换到失真坐标系 (x', y') 的方程来定义。方程的一般形式为

$$\begin{cases} x' = h_1(x, y) \\ y' = h_2(x, y) \end{cases} \quad (6-152)$$

在透视畸变的情况下，变换是线性的，即

$$\begin{cases} x' = ax + by + c \\ y' = dx + ey + f \end{cases} \quad (6-153)$$

设 $f(x,y)$ 是无失真的原始图像, $g(x',y')$ 是 $f(x,y)$ 畸变的结果, 这一失真的过程是已知的并且用函数 h_1 和 h_2 定义。于是有

$$g(x', y') = f(x, y) \quad (6-154)$$

这说明在图像中本来应该出现在像素 (x,y) 上的灰度值由于失真实际上却出现在 (x', y') 上了。这种失真的复原问题实际上是映射变换问题。在给定了 $h_1(x,y)$ ， $h_2(x,y)$ ， $g(x',y')$ 的情况下，其复原处理可如下进行：

(1) 对于 $f(x,y)$ 中的每一点 (x_0, y_0) ，找出在 $g(x',y')$ 中相应的位置 $(\alpha, \beta)=[h_1(x_0, y_0), h_2(x_0, y_0)]$ 。由于 α 和 β 不一定是整数，所以通常 (α, β) 不会与 $g(x',y')$ 中的任何点重合。

(2) 找出 $g(x',y')$ 中与 (α,β) 最靠近的点 (x_1',y_1') ，并且令 $f(x_0,y_0) = g(x_1',y_1')$ ，也就是把 $g(x_1',y_1')$ 点的灰度值赋于 $f(x_0,y_0)$ 。如此逐点作下去，直到整个图像，则几何畸变得到校正。

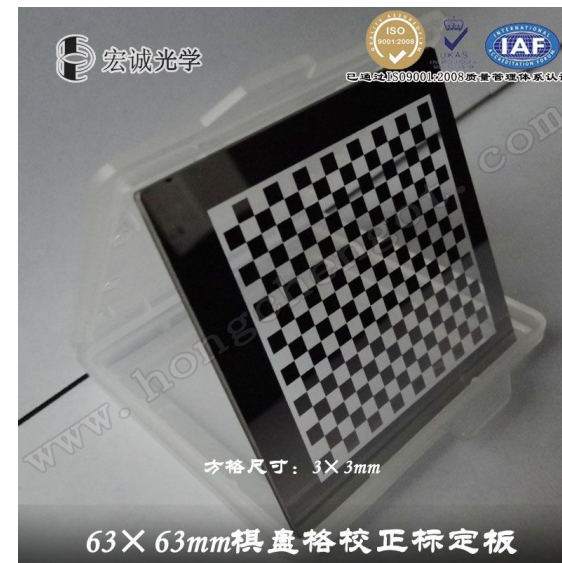
(3) 如果不采用 (2) 中的灰度值的代换方法也可以采用内插法。这种方法是假定 (α, β) 点找到后, 在 $g(x', y')$ 中找出包围着 (α, β) 的四个邻近的点并且有: $(x'_1, y'_1), (x'_{1+1}, y'_1), (x'_1, y'_{1+1}), (x'_{1+1}, y'_{1+1})$

$$\begin{cases} x'_1 \leq \alpha < x'_{1+1} \\ y'_1 \leq \beta < y'_{1+1} \end{cases} \quad (6-155)$$

$f(x,y)$ 中点 (x_0, y_0) 的灰度值由 $g(x', y')$ 中四个点的灰度值间的某种内插法来确定。

在以上方法的几何校正处理中，如果 (α, β) 处在图像 $g(x', y')$ 之外，则不能确定其灰度值，而且校正后的图像多半不能保持其原来的几何形状。

以上讨论的是 g, h_1, h_2 都知道的情况下几何畸变的校正方法。如果只知道 g ，而 h_1 和 h_2 都不知道，但是若有类似规则的网格之类的图案可供参考利用，那么就有可能通过测量 g 中的网格点的位置来决定失真变换的近似值。



如三个邻近网格点构成的小三角形，其在规则网格中的理想坐标为 $(r_1, s_1), (r_2, s_2), (r_3, s_3)$ ，并设这些点在 g 中的位置分别为 $(u_1, v_1), (u_2, v_2), (u_3, v_3)$ 。由线性变换关系

$$\begin{cases} x' = ax + by + c \\ y' = dx + ey + f \end{cases}$$

可认为它把三个点映射到它们失真后的位置，由此，
可构成如下六个方程。

$$\left\{ \begin{array}{l} u_1 = ar_1 + bs_1 + c \\ v_1 = dr_1 + es_1 + f \\ u_2 = ar_2 + bs_2 + c \\ v_2 = dr_2 + es_2 + f \\ u_3 = ar_3 + bs_3 + c \\ v_3 = dr_3 + es_3 + f \end{array} \right. \quad (6\text{---}156)$$

解这六个方程可求得 a, b, c, d, e, f 。这种变换
可用来校正 g 中被这三点连线包围的三角形部
分的失真。由此对每三个一组的网格点重复进
行，即可实现全部图像的几何校正。

6.4.2 盲目图像复原

多数的图像复原技术都是以图像退化的某种先验知识为基础，也就是假定系统的脉冲响应是已知的。

但是，在许多情况下难以确定退化的点扩散函数。

在这种情况下，必须从观察图像中以某种方式抽出退化信息，从而找出图像复原方法。

这种方法就是所谓的盲目图像复原。对具有加法性噪声的模糊图像作盲目图像复原的方法有两种：直接测量法和间接估计法。

直接测量法通常要测量图像的模糊脉冲响应和噪声功率谱或协方差函数。

在所观察的景物中，往往点光源能直接指示出冲激响应。另外，图像边缘是否陡峭也能用来推测模糊冲激响应。在背景亮度相对恒定的区域内测量图像的协方差可以估计出观测图像的噪声协方差函数。

间接估计法类似于多图像平均法处理。例如，在电视系统中，观测到的第 i 帧图像为

$$g_i(x, y) = f_i(x, y) + n_i(x, y) \quad (6-157)$$

式中 $f_i(x,y)$ 是原始图像, $g_i(x,y)$ 是含有噪声的图像, $n_i(x,y)$ 是加性噪声。如果原始图像在 M 帧观测图像内保持恒定, 对 M 帧观测图像求和, 得到下式之关系

$$f_i(x, y) = \frac{1}{M} \sum_{i=1}^M g_i(x, y) - \frac{1}{M} \sum_{i=1}^M n_i(x, y)$$

(6—158)

当 M 很大时，式(6—158)右边的噪声项的值趋向于它的数学期望值 $E\{n(x,y)\}$ 。一般情况下白色高斯噪声在所有 (x,y) 上的数学期望等于零，因此，合理的估计量是

$$\hat{f}_i(x, y) = \frac{1}{M} \sum_{i=1}^M g_i(x, y) \quad (6—159)$$

间接估计法也可以利用时间上平均的概念去掉图像中的模糊。如果有一成像系统，其中相继帧含有相对平稳的目标退化，这种退化是由于每帧有不同的线性位移不变冲激响应 $h_i(x,y)$ 引起的。

例如大气湍流对远距离物体摄影就会产生这种图像退化。只要物体在帧间没有很大移动并每帧取短时间曝光，那么第 i 帧的退化图像可表示为

$$g_i(x, y) = f_i(x, y) * h_i(x, y) \quad (6-160)$$

式中 $f_i(x, y)$ 是原始图像, $g_i(x, y)$ 是退化图像, $h_i(x, y)$ 是点扩散函数, $*$ 代表卷积。式中 $i = 1, 2, 3, \dots, M$ 。

退化图像的傅立叶变换为

$$G_i(u, v) = F_i(u, v)H_i(u, v) \quad (6-161)$$

利用同态处理方法把原始图像的频谱和退化传递函数分开，则可得到

$$\ln[G_i(u, v)] = \ln[F_i(u, v)] + \ln[H_i(u, v)]$$

如果帧间退化冲激响应是不相关的，则可得到下面的和式

$$\sum_{i=1}^M \ln[G_i(u, v)] = M \ln[F_i(u, v)] + \sum_{i=1}^M \ln[H_i(u, v)]$$

(6—163)

当 M 很大时，传递函数的对数和式接近于一恒定值，
即

$$K_H(u, v) = \lim_{M \rightarrow \infty} \sum_{i=1}^M \ln[H_i(u, v)] \quad (6-164)$$

因此，图像的估计量为

$$\hat{F}_i(u, v) = \exp\left\{\frac{K_H(u, v)}{M}\right\} \prod_{i=1}^M [G_i(u, v)]^{\frac{1}{M}} \quad (6-165)$$

对式(6—165)取傅立叶反变换就可得到空域估计是:

$$\hat{f}(x, y) = \mathcal{F}^{-1}(\hat{F}(u, v))$$

在上面分析中，并没考虑加性噪声分量。如果考虑加性噪声分量，则无法进行式(6—162)的分离处理，后边的推导也就不成立了。对于这样的问题，可以对观测到的每帧图像先进行滤波处理，去掉噪声，然后在图像没有噪声的假设下再进行上述处理。