目　录

基础篇

1　图像采集与量化 …… 1

1.1　确定型图像场取样 …… 1
1.2　随机图像场取样 …… 3
1.3　图像量化 …… 4
1.3.1　均匀量化 …… 4
1.3.2　非均匀量化 …… 6
1.3.3　矢量量化 …… 7
1.3.4　彩色图像的量化 …… 8
1.4　图像输入 …… 8
1.4.1　空间取样方式与最大限度取样分辨率 …… 9
1.4.2　图像采集系统 …… 9
1.4.3　图像输入设备 …… 11

2　图像变换 …… 16

2.1　预备知识 …… 16
2.1.1　矩阵与向量 …… 16
2.1.2　线性算子 …… 19
2.1.3　酉变换算子 …… 21
2.2　傅立叶变换(FT) …… 23
2.2.1　二维离散傅立叶变换(DFT) …… 23
2.2.2　快速傅立叶变换(FFT) …… 28
2.3　离散余弦变换(DCT) …… 31
2.3.1　关于轴对称 …… 32
2.3.2　偶对称余弦变换 …… 33
2.3.3　奇对称余弦变换 …… 34
2.4　沃尔什－哈达玛变换(WHT) …… 35
2.4.1　哈达玛变换对 …… 35
2.4.2　哈达玛变换矩阵 …… 37
2.4.3　向量表示的哈达玛变换 …… 38
2.4.4　定序哈达玛变换 …… 39

3 图像增强 …… 41
3.1 空间域单点增强 …… 41
3.1.1 灰度级校正 …… 42
3.1.2 灰度变换 …… 42
3.1.3 直方图修正 …… 44
3.2 空间域图像平滑 …… 49
3.2.1 噪声消除法 …… 49
3.2.2 邻域平均法 …… 49
3.2.3 中值滤波 …… 51
3.2.4 梯度倒数加权法 …… 52
3.2.5 选择式掩模平滑 …… 53
3.3 空间域图像锐化 …… 54
3.3.1 梯度法 …… 54
3.3.2 Laplacian算子 …… 56
3.3.3 高通滤波 …… 58
3.3.4 掩模匹配法 …… 59
3.3.5 统计差值法 …… 60
3.4 频率域增强处理 …… 61
3.4.1 低通滤波 …… 61
3.4.2 同态图像增强 …… 62
3.4.3 高通滤波 …… 64
3.5 彩色技术 …… 65
3.5.1 假彩色处理 …… 65
3.5.2 伪彩色增强 …… 66
4 图像分割 …… 68
4.1 图像边缘提取 …… 68
4.1.1 边缘算子法 …… 68
4.1.2 曲面拟合法 …… 72
4.1.3 模板匹配法 …… 74
4.1.4 门限化 …… 78
4.2 区域分割 …… 81
4.2.1 区域生长法 …… 81
4.2.2 特征空间分类法 …… 82
5 图像编码与压缩 …… 84
5.1 概述 …… 84
5.1.1 图像编码压缩的必要性 …… 84

5.1.2 图像编码压缩的可能性 …… 84
5.1.3 图像编码压缩的分类 …… 85
5.1.4 图像编码压缩术语简介 …… 85
5.2 预测编码 …… 86
5.2.1 线性预测编码 …… 87
5.2.2 非线性预测编码 …… 88
5.3 正交变换编码 …… 90
5.3.1 变换编码的基本原理 …… 90
5.3.2 正交变换编码特性评价 …… 91
5.3.3 变换编码 …… 91
5.4 统计编码 …… 94
5.4.1 哈夫曼编码(Huffman Coding) …… 94
5.4.2 香农编码(Shannon Coding) …… 97
5.4.3 算术编码(Arithmetics Coding) …… 99
5.5 轮廓编码(Contour Coding) …… 100
5.5.1 T(Trace)算法 …… 100
5.5.2 IP(Initiat Point)算法 …… 101
5.5.3 轮廓编码 …… 101
5.6 二值图像编码 …… 103
5.6.1 直接编码 …… 103
5.6.2 跳跃空白编码 …… 104
5.6.3 游程长度编码 …… 104
5.7 多媒体图像编码压缩技术 …… 109
5.7.1 经典编码压缩技术 …… 109
5.7.2 分形编码 …… 110
5.7.3 自适应网格编码 …… 112

6 图像恢复与重建 …… 113

6.1 图像恢复的基本概念 …… 113
6.2 图像退化模型 …… 114
6.3 图像恢复方法 …… 117
6.3.1 反向滤波器法 …… 117
6.3.2 约束还原 …… 119
6.4 点冲激函数的退化模型 …… 121
6.5 几何畸变校正 …… 123
6.6 图像重建的基本概念 …… 124
6.7 图像重建的方法 …… 124
6.7.1 平行投影 …… 124
6.7.2 获得平行投影的物理方法 …… 126

6.7.3 图像重建的傅立叶变换法 …… 126
6.7.4 滤波逆投影法 …… 128

7 图像特征与分析 …… 129

7.1 纹理特征与分析 …… 129
7.1.1 纹理分析的基本概念 …… 129
7.1.2 纹理分析中的直方图方法 …… 130
7.1.3 纹理分析的自相关函数方法 …… 131
7.1.4 纹理的灰度分布统计特征分析 …… 132
7.2 形状特征与分析 …… 138
7.2.1 区域边界与骨架的提取 …… 138
7.2.2 区域边界的形状特征描述 …… 139
7.2.3 区域的变换域的形状分析 …… 142
7.3 时间序列特征分析 …… 144
7.3.1 视差、光流场 …… 144
7.3.2 序列图像中的特征点选择与匹配 …… 145
7.3.3 时间序列图像的运动估计和分析 …… 146
7.4 二维图像特征提取与识别 …… 148
7.4.1 二维图像特征的作用 …… 148
7.4.2 二维图像识别的目的 …… 149

工 具 篇

8 图像文件格式 …… 150

8.1 MacPaint 图像文件 …… 150
8.1.1 文件头(Mac Binary) …… 150
8.1.2 图案数据 …… 151
8.1.3 位图数据 …… 151
8.1.4 创建 MacPaint 图像文件 …… 152
8.2 IMG 图像文件 …… 152
8.2.1 IMG 的文件头 …… 152
8.2.2 IMG 图像数据文件 …… 153
8.3 GIF 图像文件 …… 154
8.3.1 GIF 的文件头 …… 154
8.3.2 GIF 的图像数据 …… 154
8.3.3 GIF 的图像编码压缩算法:LZW …… 155
8.4 TIFF 的图像文件 …… 155

8.4.1 TIFF 的文件头 …… 155
8.4.2 TIFF 图像数据文件 …… 156
8.5 其他图像数据文件(PCX 和 BMP) …… 156
8.5.1 PCX 图像文件 …… 156
8.5.2 BMP 图像文件 …… 157
8.6 图像文件格式转换 …… 157

9 PostScript 文件 …… 165

9.1 矢量类型图形 …… 165
9.2 灰度位图图像 …… 166
9.3 彩色位图图像 …… 166
9.4 PostScript 程序的执行 …… 167
9.5 PostScript 文件硬拷贝输出 …… 167

应 用 篇

10 生物医学图像处理 …… 168

10.1 电子显微图像特点 …… 168
10.2 电子显微图像的计算机处理 …… 169
10.2.1 二维显微图像的处理 …… 169
10.2.2 显微图像的三维重建 …… 169
10.2.3 显微图像的识别分类 …… 169

11 金属结构图像的处理 …… 170

11.1 系统设备及工作原理 …… 170
11.2 图像预处理 …… 170
11.2.1 灰度变换 …… 171
11.2.2 滤波 …… 171
11.2.3 图像二值化 …… 171
11.2.4 焊缝缺陷的轮廓线跟踪 …… 171
11.3 物理量计算 …… 171
11.4 计算机自动识别 …… 172

12 遥感图像处理 …… 173

12.1 遥感技术与遥感信息的获取 …… 173
12.2 卫星遥感图像的特点和分类 …… 173
12.2.1 美国陆地卫星的图像资料 …… 174
12.2.2 法国 SPOT 卫星图像资料 …… 174

12.2.3 我国国土卫星的遥感图像 …… 174
12.3 遥感信息处理 …… 175
12.3.1 辐射校正 …… 175
12.3.2 几何校正 …… 175
12.4 遥感图像的解译与分类 …… 176
12.5 遥感图像的应用 …… 176
12.5.1 地质遥感图像处理与应用 …… 176
12.5.2 森林遥感图像处理与应用 …… 176
12.5.3 国土资源遥感图像处理与应用 …… 177
12.5.4 海洋遥感图像处理与应用 …… 177
12.5.5 气象卫星云图的处理与应用 …… 177
13 3S 系统与图像处理 …… 181
13.1 地理信息系统(GIS)和遥感图像处理 …… 181
13.2 全球定位系统(GPS)和遥感图像处理 …… 181

发 展 篇

14 马尔可夫随机场及其在图像分析与处理中的应用 …… 183
14.1 马尔可夫随机场的定义和基本性质 …… 183
14.2 模拟退火算法 …… 184
14.3 MRF 在图像处理中的应用 …… 186
14.3.1 用于图像恢复的 MRF 模型 …… 186
14.3.2 用于图像噪声处理的 MRF 模型 …… 186
14.3.3 用于纹理分析的 MRF 模型 …… 187
15 小波分析及其在图像处理中的应用 …… 189
15.1 连续小波变换及其性质 …… 189
15.2 二进小波变换及其性质 …… 191
15.3 多尺度分析的概念及其性质 …… 191
15.4 二维连续小波变换 …… 192
15.5 图像的小波分解编码 …… 193
15.6 图像纹理的小波分析 …… 195
15.7 图像小波分析中的自适应处理 …… 196
15.8 小波变换在图像编码压缩中的应用 …… 198
参考文献 …… 200

1 图像采集与量化

●连续图像　●空间取样　●样点阵列
●多形式量化　●采集系统　●输入设备

从广义上说，图像是自然界景物的客观反映。以照片形式或视频记录介质保存的图像是连续的，计算机无法接收和处理这种空间分布和亮度取值均连续分布的图像。图像数字化就是将连续图像离散化，其工作包括两个方面：取样和量化。

所谓取样，就是把一幅连续图像在空间上分割成 $M \times N$ 个网格，每个网格用一亮度值来表示。由于结果是一个样点值阵列，故又叫点阵取样。

取样使连续图像在空间上离散化，但取样点上图像的亮度值还是某个幅度区间内的连续分布。根据取样定义，每个网格上只能用一个确定的亮度值表示。把取样点上对应的亮度连续变化区间转换为单个特定数码的过程，称之为量化，即样点亮度的离散化。

在取样和量化过程中，取样密度取多大合适？以多少个等级表示样本的亮度值为最好？这些都将影响到离散图像能否保持连续图像信息的问题。原则上，$M \times N$ 的取值主要取决于其是否满足取样定理，因为在满足取样定理的情况下，重建图像就不会产生失真。灰度级的确定将根据图像的内容和要求来考虑。

1.1 确定型图像场取样

令 $f(x,y)$ 为一个有限带宽的二维连续图像函数，$f(x,y)$ 的傅立叶变换(Fourier Transform)对为

$$f(x,y) \Leftrightarrow F(u,v)$$

$F(u,v)$ 具有以下特点：当 u,v 方向上的带宽分别为 L_u 和 L_v 时，$f(x,y)$ 的能量集中在半径为 L_u 或 L_v 所包含的区域内(见图 1.1)。

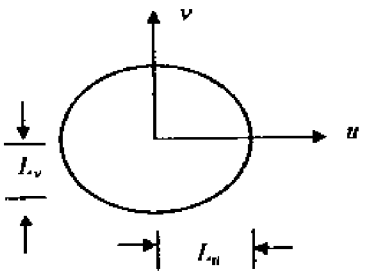


图 1.1　$u \sim v$ 平面上 $f(x,y)$ 对应的能量区域

又知二维取样函数

$$s(x,y)=\sum_{i=-\infty}^{+\infty}\sum_{j=-\infty}^{+\infty}\delta(x-i\Delta x,y-j\Delta y) \tag{1.1}$$

式中，Δx、Δy 为 x、y 方向的取样间隔。(1.1)式是脉冲函数 $\delta(x,y)$在 x、y 方向以 Δx、Δy 为间隔的展开。见图 1.2。

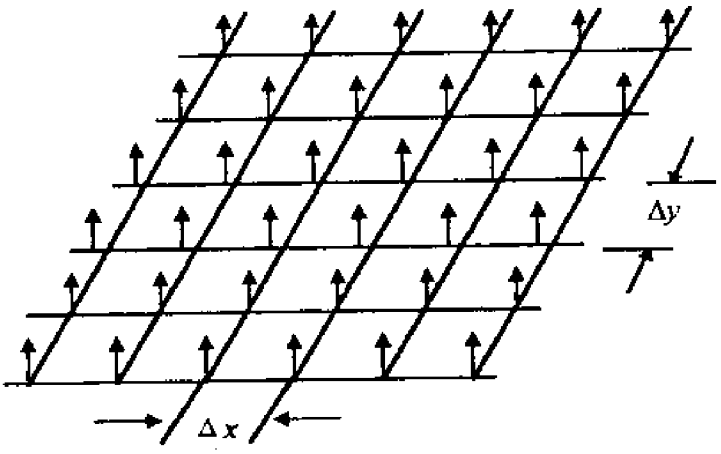


图 1.2　脉冲函数阵列

对图像 $f(x,y)$取样，就是将 $f(x,y)$乘以取样函数 $s(x,y)$。取样后的图像

$$\begin{aligned}f_s(x,y)&=f(x,y)\cdot s(x,y)\\&=f(x,y)\sum_{i=-\infty}^{+\infty}\sum_{j=-\infty}^{+\infty}\delta(x-i\Delta x,y-j\Delta y)\\&=\sum_{i=-\infty}^{+\infty}\sum_{j=-\infty}^{+\infty}f(i\Delta x,j\Delta y)\delta(x-i\Delta x,y-j\Delta y)\end{aligned} \tag{1.2}$$

在上式中，连续图像函数 $f(x,y)$移入求和式内变成离散形式 $f(i\Delta x,j\Delta y)$，这是因为图像函数只在$(i\Delta x,j\Delta y)$取样点上取值。

取样函数 $s(x,y)$的傅立叶谱为

$$\begin{aligned}s(u,v)&=\frac{1}{\Delta x\Delta y}\sum_{i=-\infty}^{+\infty}\sum_{j=-\infty}^{+\infty}\delta\left(u-i\frac{1}{\Delta x},v-j\frac{1}{\Delta y}\right)\\&=\frac{4\pi^2}{\Delta x\Delta y}\sum_{i=-\infty}^{+\infty}\sum_{j=-\infty}^{+\infty}\delta(u-i\Delta u,v-j\Delta v)\end{aligned} \tag{1.3}$$

$\Delta u=2\pi/\Delta x$，$\Delta v=2\pi/\Delta y$。根据取样定义和傅立叶变换卷积定理，$f(x,y)$、$f_s(x,y)$、$s(x,y)$三者之间的频域关系为

$$F_s(u,v)=\frac{1}{4\pi^2}F(u,v)*s(u,v) \tag{1.4}$$

将(1.3)式代入(1.4)式，展开卷积式并进一步整理，得

$$\begin{aligned}F_s(u,v)&=\frac{F(u,v)}{\Delta x\Delta y}\sum_{i=-\infty}^{+\infty}\sum_{j=-\infty}^{+\infty}\delta(u-i\Delta u,v-j\Delta v)\\&=\frac{1}{\Delta x\Delta y}\sum_{i=-\infty}^{+\infty}\sum_{j=-\infty}^{+\infty}F(u-i\Delta u,v-j\Delta v)\end{aligned} \tag{1.5}$$

由(1.5)式不难看出，取样图像 $f_s(x,y)$的频谱是连续图像谱在(u,v)方向上以一定间隔的

分布。当 Δx、Δy 选择适当，使得 Δu、Δv 大于或等于原图像覆盖频率间隔 L_u、L_v 两倍时，取样就不会出现重叠现象(见图 1.3)，就可以获得期望的取样点阵。

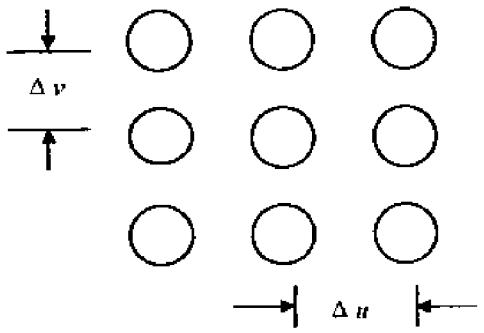


图 1.3　取样图像谱

依据以上分析，确保正确取样的条件是

$$\left.\begin{aligned}\Delta u &\geqslant 2L_u\\ \Delta v &\geqslant 2L_v\\ \Delta x &\leqslant \pi/L_u\\ \Delta y &\leqslant \pi/L_v\end{aligned}\right\} \tag{1.6}$$

上式中，“等于”条件的取样是著名的奈奎斯特取样定理，即在这样的通带宽度下，由 $f_s(x,y)$ 可以不失真地重现 $f(x,y)$。

1.2　随机图像场取样

图像场的另一种模型是随机场。随机图像场取样后其功率谱如何分布呢？试作一简单介绍。

令 $f(x,y)$ 是连续平稳的二维随机图像场，其自相关函数为

$$R_f(x_1-x_2,y_1-y_2)=E\{f(x_1,y_1)f^*(x_2,y_2)\} \tag{1.7}$$

f^* 是 f 的共轭值。用脉冲函数 $\delta(x,y)$ 对 $f(x,y)$ 取样，所得图像场为 $f_s(x,y)$，

$$f_s(x,y)=f(x,y)\cdot s(x,y) \tag{1.8}$$

由(1.1)式有

$$f_s(x,y)=f(x,y)\sum_{i=-\infty}^{+\infty}\sum_{j=-\infty}^{+\infty}\delta(x-i\Delta x,y-j\Delta y) \tag{1.9}$$

$f_s(x,y)$ 的自相关函数 $R_{fs}(x_1-x_2,y_1-y_2)$ 的表达式如下：

$$R_{fs}(x_1-x_2,y_1-y_2)=E\{f_s(x_1,y_1)f_s^*(x_2,y_2)\}$$

结合(1.8)式

$$\begin{aligned}R_{fs}(x_1-x_2,y_1-y_2)&=E\{f(x_1,y_1)\cdot s(x_1,y_1)f_s^*(x_2,y_2)\}\\&=R_f(x_1-x_2,y_1-y_2)\cdot s(x_1-x_2,y_1-y_2)\end{aligned} \tag{1.10}$$

令 $\tau_x=x_1-x_2$，$\tau_y=y_1-y_2$，进一步改写上式，有

$$R_{fs}(\tau_x,\tau_y)=R_f(\tau_x,\tau_y)\cdot s(\tau_x,\tau_y) \tag{1.11}$$

上式右边第一项为随机图像场的自相关函数，第二项是间隔为 τ_x、τ_y 的取样函数。可见，取

现在用均方误差来讨论量化误差，因为这种测度易于处理，并且与主观标准有比较密切的关系。显然，用 C 个层次进行量化时，均方误差

$$\varepsilon^2 = \int_{a_0}^{a_k} (f - \hat{f}) P(f) \mathrm{d}f$$

$$= \sum_{i=0}^{C-1} \int_{d_i}^{d_{i+1}} (f - N_i)^2 P(f) \mathrm{d}f \tag{1.14}$$

$P(f)$为样本在$[a_0, a_k]$区间的概率密度。因为是均匀量化，所以 $P(f)$可以认为是常数。这样，均方误差

$$\varepsilon^2 = \sum_{i=0}^{C-1} P(N_i) \int_{d_i}^{d_{i+1}} (f - N_i)^2 \mathrm{d}f$$

$$= \frac{1}{3} \sum_{i=0}^{C-1} P(N_i) [(d_{i+1} - N_i)^3 - (d_i - N_i)^3] \tag{1.15}$$

知道了 N_i，就不难求出 ε^2。

N_i 由区间$[d_i, d_{i+1}]$决定，在这个区间内 N_i 有多个值，从而就有了$\{\varepsilon^2\}$。选择怎样的 N_i 才能使ε 最小呢？令

$$\frac{\partial \varepsilon^2}{\partial N_i} = 0$$

得

$$N_i = \frac{1}{2}(d_{i+1} + d_i) \tag{1.16}$$

当量化输出 N_i 等于判断区间上、下限之中点时，量化误差最小。将(1.16)式代入(1.15)式，得

$$\hat{\varepsilon}^2 = \frac{1}{3} \sum_{i=0}^{C-1} P(N_i) [(d_{i+1} - \frac{1}{2} d_{i+1} - \frac{1}{2} d_i)^3 - (d_i - \frac{1}{2} d_{i+1} - \frac{1}{2} d_i)^3]$$

$$= \frac{1}{12} \sum_{i=0}^{C-1} P(N_i) [d_{i+1} - d_i]^3 \tag{1.17}$$

进一步，设子区间$[d_i, d_{i+1}]$的长度为 L，则

$$\varepsilon^2 = \frac{L^2}{12} \tag{1.18}$$

或

$$\varepsilon^2 = \frac{1}{12C^2} \tag{1.19}$$

加大量化层次 C 对保持原图像的信息十分有利。但无疑也会带来负面影响，如图像数据会急剧膨胀，给存储、传输和处理带来压力。怎样选择量化层次和取样点数，是取样和量化中必须妥善考虑的问题。原则上说，在图像数据长度要求比较宽容的场合，量化级数可尽量多些，以利改善图像重建质量。当然，这不是说可以任意地大，因为任何处理系统在目标实现的前提下，总是希望图像数据少一些，以减少资源开销。同时，人眼对灰度的分辨能力有限，一般情况下有 8 比特量化级数就可以了。对于某些卫星照片，为了能分辨一些细微目标，可选择 8～14 比特的量化级数。至于这时的取样点数 $M \times N$，则以在满足取样定理情况下，重建图像不失真为度。

如果对图像数据量($M \times N \times$ 比特数/像素)要求给定，那对取样点数 $M \times N$ 和量化比特数的选择就要视图像内容和系统要求来决定。如图像中灰度变化平缓的区域，可减少取样

点数，增大量化层次，这样，重建图像层次丰满。否则，在灰度平缓区域会出现“假轮廓”现象。反之，细节丰富的地方，取样点数密一些，量化级数可少些，这样方能保持图像细节。

1.3.2 非均匀量化

在某些情况下，样本值在$[a_0, a_k]$范围内呈不均匀分布。为了让数字图像与连续图像保持良好的匹配，有必要在样本频率高的区域采用小量化区间的密集量化，而在其它地方则采用相对粗糙的量化，这种不等间隔的量化属于非均匀量化(见图 1.5)。

在下列前提下：

(1) $f=f(m,n)$；

(2) $f\in(d_i,d_{i+1})$，量化层次为 C；

(3) 各子区间的量化数码为 $N_i, i=0,1,2,\cdots,C-1$。量化均方误差

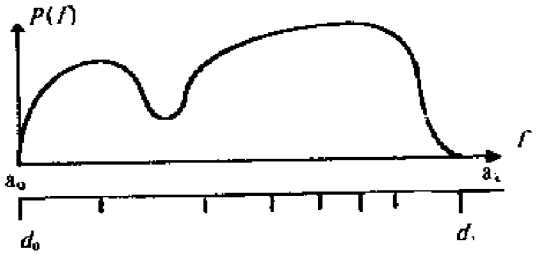


图 1.5 非均匀量化示意图

$$\begin{aligned}\varepsilon^2 &= \int_{d_0}^{d_1}(f-N_0)^2P(f)\mathrm{d}f \\ &+\int_{d_1}^{d_2}(f-N_2)^2P(f)\mathrm{d}f+\cdots \\ &+\int_{d_{C-1}}^{d_C}(f-N_{C-1})^2P(f)\mathrm{d}f\end{aligned} \tag{1.20}$$

将均方误差分别对 d_i 和 N_i 求偏导，并使之为零，得

$$\frac{\partial\varepsilon^2}{\partial d_i}=(d_i-N_{c-1})^2P(d_i)-(d_i-N_i)^2P(d_i)=0$$

$$\frac{\partial\varepsilon^2}{\partial N_i}=-2\int_{d_i}^{d_{i+1}}(f-N_i)P(f)\mathrm{d}f=0$$

化简后得

$$d_i=\frac{1}{2}(N_i+N_{i+1}) \tag{1.21}$$

$$N_i=\frac{\int_{d_i}^{d_{i+1}}f\cdot P(f)\mathrm{d}f}{\int_{d_i}^{d_{i+1}}P(f)\mathrm{d}f} \tag{1.22}$$

每一个量化子区间的输出 N_i 是$[d_i,d_{i+1}]$上 $P(f)$的质量中心。如果令 $P(f)$ = 常数，则有

$$N_i=\frac{1}{2}(d_{i+1}+d_i)$$

可见均匀量化是非均匀量化的特例。

非均匀量化是一个很重要的概念。大多数图像的灰度分布都是不均匀的，在使用直方图取得 $P(f)$后，通过反复迭代，可由(1.21)、(1.22)式求出$\{d_i\}$、$\{N_i\}$。

求解非均匀量化判断区间及其灰度代码的另一种方案是压扩量化法。其过程(见图 1.6)是先将图像信号 f 进行非线性变换，得

$$g=T\{f\} \tag{1.23}$$

变换后的图像信号 g 的概率密度 $P(g)$呈均匀分布；对 g 进行均匀量化，得到 $\hat{g}$；将量化结果作相应的非线性反变换，即

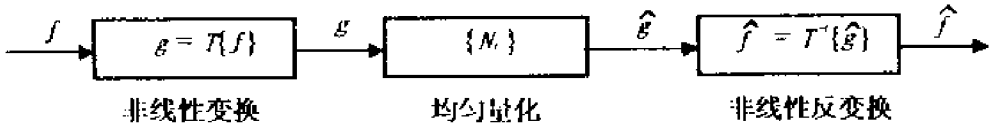


图 1.6 压扩量化示意图

$$\hat{f} = T^{-1}\{\hat{g}\} \tag{1.24}$$

对不同的 $P(g)$形式，g、$\hat{f}$的变换表达式是有差异的。如瑞利(Ragleigh)分布

正变换
$$g = \frac{1}{2} - \exp\left[-\frac{f^2}{\sigma^2}\right] \tag{1.25}$$

反变换
$$\hat{f} = \left\{\sqrt{2}\sigma^2 \ln\left[\frac{1}{\frac{1}{2} - \hat{g}}\right]\right\}^{\frac{1}{2}} \tag{1.26}$$

拉普拉斯(Laplacian)分布

正变换
$$g = \frac{1}{2}[1 - \exp(-\alpha f)]; \quad f \geqslant 0 \tag{1.27}$$
$$g = \frac{1}{2}[1 - \exp(\alpha f)]; \quad f < 0$$

反变换
$$\hat{f} = \frac{1}{\alpha}\ln[1 - 2\hat{g}]; \quad \hat{g} \geqslant 0 \tag{1.28}$$
$$\hat{f} = \frac{1}{\alpha}\ln[1 + 2\hat{g}]; \quad \hat{g} < 0$$

式中 σ^2 为样本方差；

$\hat{g}$ 为均匀量化后的非线性变换信号；

$\alpha = 1.44\sigma^{-1}$。

1.3.3 矢量量化

均匀量化和非均匀量化称之为标量量化。标量量化后图像信息长度为

$$L_b = M \times N \times N_b \tag{1.29}$$

单位为比特。N_b 是每个量化数码的比特数，$M \times N$ 为离散化图像阵列样本数。

矢量量化是 50 年代由香农提出的，其处理对象不是取样图像阵列中的单个样本，而是由 N 个样本组成的向量 $\boldsymbol{f}$。具体地说，就是将$[(f(m,n)]_{M\times N}$中的 N 个样本组合成向量 $\boldsymbol{f}$，令 $\boldsymbol{f}$ 是随机过程的样本，该随机过程的 n 阶概率密度函数 $P(\boldsymbol{f})$为

$$P(\boldsymbol{f}) = P\{f_1, f_2, \cdots, f_n\} \tag{1.30}$$

量化时将 N 维空间细分为 J 个判决区 D_j，每个判决区对应一个矢量形式的量化值 $\boldsymbol{r}_j$。如果取样图像向量 $\boldsymbol{f}$ 在判决区 D_j 内，$\boldsymbol{r}_j$ 即作为 $\boldsymbol{f}$ 在 D_j 区内的量化输出。

矢量量化的均方误差如下式表示：

$$\varepsilon^2 = \sum_{j=0}^{J-1}\int_{D_j} \mathrm{tr}[(\boldsymbol{f} - \boldsymbol{r}_j)(\boldsymbol{f} - \boldsymbol{r}_j)^T]P(\boldsymbol{f})\mathrm{d}\boldsymbol{f} \tag{1.31}$$

将(1.31)式对 $\boldsymbol{r}_j$ 求偏导，并令其为零，可获得最佳矢量量化输出，进一步求得

$$\frac{\partial \varepsilon^2}{\partial r_j} = \int_{D_j} (f - r_j) P(f) \mathrm{d}f \tag{1.32}$$

$$r_j = \frac{\int_{D_j} f P(f) \mathrm{d}f}{\int_{D_j} P(f) \mathrm{d}f} \tag{1.33}$$

(1.33)式是 f 的条件平均估值。

矢量量化的基本原理是容易理解的，但实际应用可不是一件轻松的事，因为概率密度 $P(f)$ 计算、判决区 D_j 的选取等，均要涉及到复杂的数学问题。限于篇幅，不予深论，从事这方面设计的读者请进一步参考其他有关著作。

1.3.4 彩色图像的量化

随着电视技术、遥感技术、多媒体技术的日益发展与普及，彩色图像已司空见惯。根据三色标记法，彩色可用红(R)、绿(G)、蓝(B)三基色来表示。

针对彩色的这一特点，我们采用单色图像量化法对红、绿、蓝三基色分别量化。由于视觉对 R、G、B 三基色的灵敏度各不相同，所以在量化过程中区别对象使用不同特性的压扩变换方法，来取得更佳的量化效果。图 1.7 是彩色图像量化的原理方框图。彩色图像输入后被分解为 R、G、B 三种基色；彩色坐标处理使 R、G、B 变成 $x(1)$、$x(2)$、$x(3)$三个分量；然后量化，三个分量被赋予二进制码组；再经彩色坐标反变换，将 $\hat{x}(1)$、$\hat{x}(2)$、$\hat{x}(3)$变回到彩色坐标系；最后重新合成彩色图像。

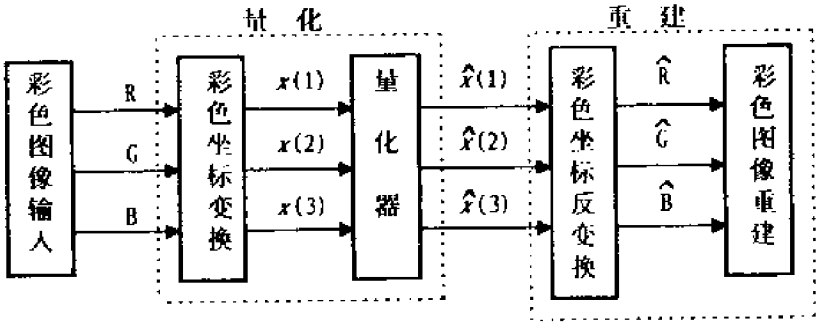


图 1.7 彩色图像量化原理框图

彩色图像量化编码的总比特数等于 $x(1)$、$x(2)$、$x(3)$三个分量量化编码比特数之和。即

$$B = b(1) + b(2) + b(3) \tag{1.34}$$

而不是简单的单个分量比特数的 3 倍，因为各个分量量化时的层次不尽相同。

1.4 图像输入

通过前面的介绍，我们已经了解了取样和量化的原理和方法。下面再谈谈图像输入问题。

1.4.1 空间取样方式与最大限度取样分辨率

空间取样方式有随机扫描和顺序扫描两种。随机扫描适用于截取一帧图像内某些关心区域的场合，即跳来跳去扫描那些必要部分。顺序扫描遵循一定的次序进行，如自左向右、自上而下逐点扫描，构成 $M \times N$ 点阵(图 1.8(a))；首行自左起步，后续尾首相接，形成一维序列(图 1.8(b))；隔行扫描(图 1.8(c))。一般情况下，普遍采用形式(b)，因为这种格式的数据便于后续处理。电视系统中有时采用方式(c)。

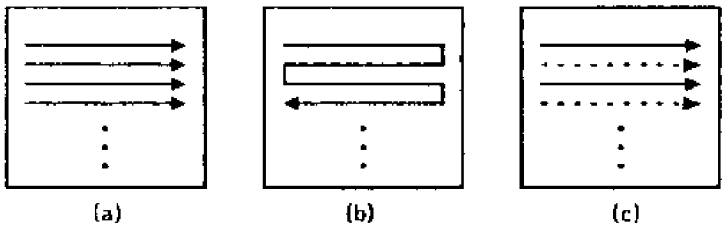


图 1.8 顺序扫描方式

1.4.2 图像采集系统

图 1.9 是图像采集系统组成框图，其大体上由同步信号产生系统、扫描系统、光电变换系统、模/数转换系统、照明系统五部分构成。同步信号通常由标准振荡器和分频器组成，这一信号使所有部件同步动作。扫描系统是图像采集系统的固有组成部分，在计算机自动控制下，将图像的光强信息转变为模拟电信号，目前使用的扫描系统有电子光学扫描系统和机械扫描系统两大类。光电变换部分负责把扫描系统输出的与像素属性对应的光能转换成电能。放大电路是为了使模拟信号的幅度与模/数转换器(A/D)匹配而设置的。从光电变换系统输出的电信号进入模/数转换系统转换成数字信号，供存储、显示、传输或后续处理。

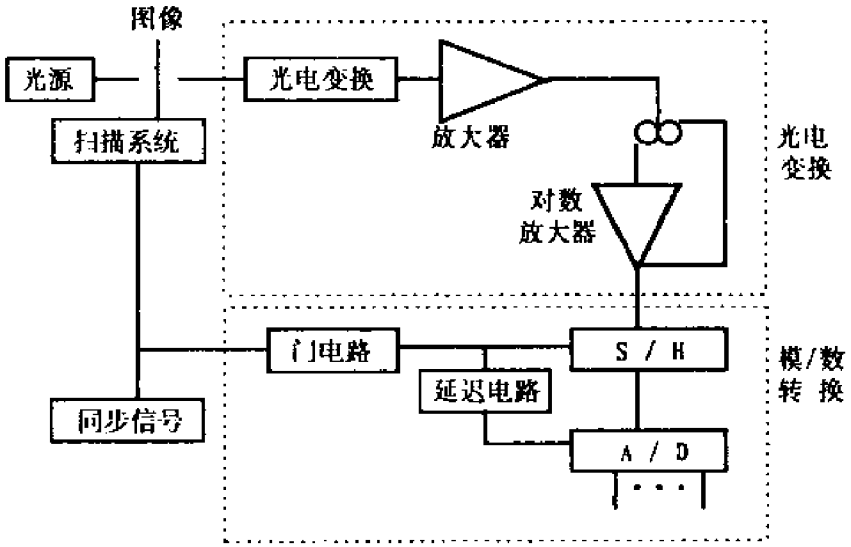


图 1.9 图像采集系统原理框图

为加深对采集系统的理解，再作两点说明：

1）光电变换特性

扫描装置把黑白图像的亮度信息转换成电信号。在一定的照明系统下，如果图像亮度低于某一个值时，扫描系统对该亮度的光没有反应，称为无感受区域；如果再增加图像亮度，扫描系统输出也不发生变化，称之为饱和区域；介于这两者之间的称之为动态区域，扫描输出与图像亮度基本呈线性关系（图 1.10，纵坐标 u_o 为输出电压）。

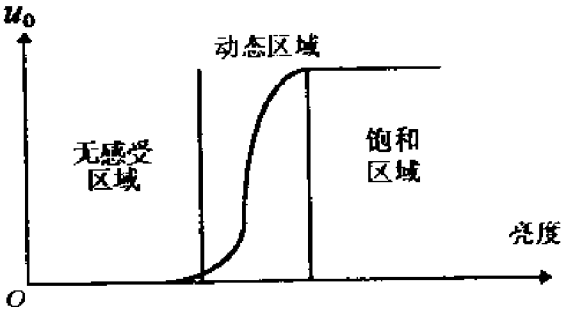


图 1.10　输入/输出特性曲线

在有些负片扫描系统中，底片黑度（D）与输出电压（u_o）之间出现类指数关系，D 在0～1范围内，输出电压变化很大，控制范围宽；D 大于 4 以后，输出电压变化极小，控制范围窄（图 1.11 实线）。如果将这种关系下的电压信号直接进入后续电路，则黑度大的部分采集到的灰度信息会严重丢失，而在有些场合许多图像特征恰恰又在这些黑度区域（如焊缝缺陷的 x 光底片）。为此，可在光电变换后，增加一级由前置运算放大器电路和对数运算放大器电路组成的转换放大单元。前置运算放大器是微弱输入信号的直流放大器，要求具有高输入阻抗、低噪声、低温度漂移、高灵敏度且性能稳定。对数放大器使输出电压与黑度变为近似线性（图 1.11 虚线），从而提高了系统的黑度分辨率，使黑度很大的底片也能进行有效的处理。

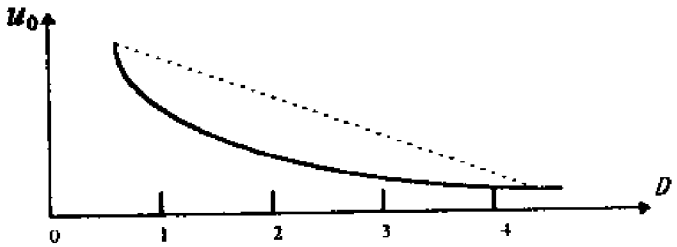


图 1.11　负片黑度与输出电压关系

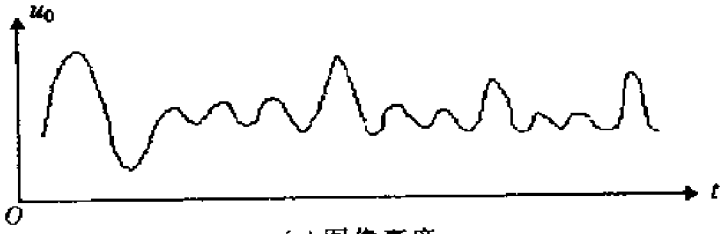
彩色图像输入时，要用滤色片（或分色镜、棱镜、衍射光栅等）对彩色信号进行分解，取出红、绿、蓝三基色，再对三个颜色分量进行光电转换。

2）取样/保持电路

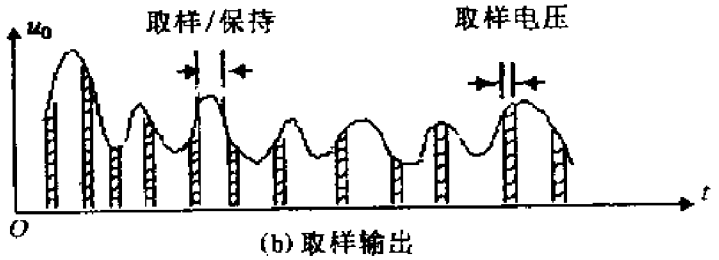
在图 1.9 中，模/数转换系统由门电路、延迟电路、取样/保持、A/D 变换等部分组成。所谓取样/保持就是在 A/D 变换期间使电压保持一定值的部分。之所以要这样做，是因为在某一瞬间对光电变换系统输出的信号进行 A/D 变换时，如果在转换过程中信号又变了，则产生出的结果就会杂乱无章。取样/保持时间与取样间隔是一致的，所以 A/D 输出的数字图像阵列与取样阵列相一致。图 1.12 形象地表示了取样、取样/保持、A/D 转换输出三者之

742293

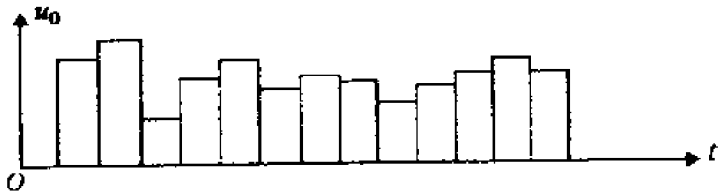
间的关系。



(a) 图像亮度



(b) 取样输出



(c) A/D 转换器

图 1.12 A/D 转换关系

1.4.3 图像输入设备

图像输入设备是图像处理系统中的先导硬件，它具有图像输入和数字化双重功能。随着科学技术的发展，图像输入设备日益向高速度、高分辨率、多功能、智能化方向发展。目前，用得较多的图像输入设备是扫描仪和电视摄像机。

1) 扫描仪

扫描仪是当今流行的一种静态图形图像(照片、卫片、航片、图纸等)输入设备，可广泛用于图形图像处理、电脑桌面排版、计算机辅助设计、多媒体等领域。

从工作方式考虑，扫描仪可分为平板式、手持式、滚筒式三类。平板式扫描仪工作时，图像放置在平板上，从氙灯等光源发出的光经聚光后入射到被摄图像上，透过被摄图像的光经光电倍增管放大后便成为图像信号，而扫描则通过平台沿两轴方向的机械运动来实现。平板式扫描仪分辨率高，1995 年市售平板扫描仪的最高分辨率已达 4800dpi(每英寸点数，如 TH－4800SP 型扫描仪)，颜色深度为 36 比特，被誉之为超彩色。

手持式扫描仪主要用于要求不高的应用环境。其主要特点是携带方便，价格便宜。1994 年以前的手持式扫描仪一般是用手控制垂直方向的扫描，图像易产生失真，分辨率低，多为几十～几百 dpi。现在已生产"自走式"手持扫描仪(如 PRIMA×P800M 型)，这种新型手持扫描仪消除了以往操作者手的抖动造成的失误，最高分辨率已达到 800～1200dpi，颜色深度 24 比特，性能逐渐逼近中期平板扫描仪。

滚筒式扫描仪常用于大幅图像输入，如大型工程图，CAD 图等。它通过滚筒带动图像旋转和扫描头相对位移实现扫描。

扫描仪一般都配有相应的软件，这些软件可用来选择扫描时的工作参数，如扫描区域、对比度、分辨率、图像深度等。此外，有些扫描仪的配套软件还具有平滑、放大、缩小、旋转、编辑等功能。

扫描得到的图像在计算机显示器上显示的色彩及尺寸与系统的设置和能力有关。可以用软件调整显示器的色度，使其颜色深度与扫描仪输出一致或接近，这样色彩比较逼真。在 640×480 图形方式下，14 英寸屏幕的分辨率为每英寸 72 个像素。如果扫描仪扫描时分辨率也设置为 72dpi，则屏幕显示的图像与原图像的大小比例为 1:1。如果扫描仪的分辨率大于 72dpi，则显示图像比原图像大；否则，比原图像小。

2) 摄像机

摄像机是最常用的图像输入装置，这不仅是因为它面向对象广泛，如照片、胶片、场景等，而且输入速度快、灵敏度高、使用方便，从而倍受青睐。

摄像机通常由摄像镜头、摄像器件、同步信号发生电路、偏转电路、放大电路、电源等部分组成(图 1.13)。来自被摄物体的光通过光学系统经由摄像器件形成光学图像，进一步放大后变成视频信号输出。

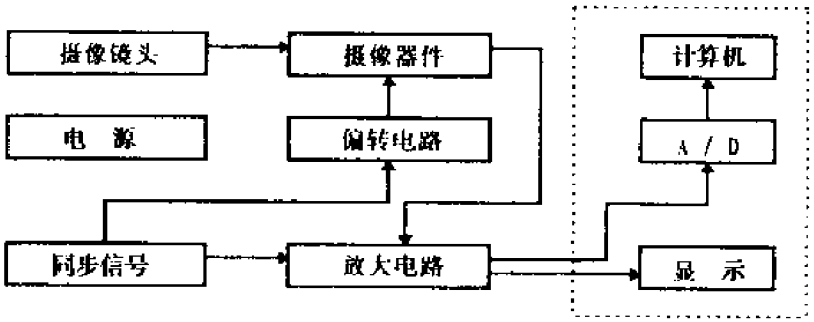


图 1.13　电视摄像机逻辑结构框图

从摄像器件的组成出发，可将电视摄像机分成两大类，即电子管摄像机和固体器件摄像机。再细分，则可列出以下十余种：

- 摄像机
  - 电子管摄像机
    - 外光电效应摄像机
      - 光电摄像管
      - 移像光电摄像管
      - 正析像管
      - 超正析像管
    - 内光电效应摄像机
      - 视像管
      - 氧化铅管
      - 硒化镉管
      - 硒砷碲管
      - 硅光导摄像管
  - 固件器件摄像机
    - CCD(Charge Couple Device)
    - BBD (Bucket Brigade Device)

其中,外光电效应类摄像机目前几乎已不使用,从略。内光电效应摄像机因体积小、重量轻、成本低、调整方便而受到普遍欢迎。固体器件摄像机耗电省、寿命长、可靠性高得到了充分重视。下面,分别以视像管和 CCD(电荷耦合器件)为例,讨论电子管和固体器件摄像机的基本工作过程。

(1) 视像管工作原理

图 1.14 给出了视像管构造略图。光电靶很薄,分三层:向着图像的一侧为玻璃板;接着是信号板,这是均匀喷涂在玻璃板上的一层透明金属导电层;光电靶的另一侧为光电阴极,它由蒸镀在信号板上具有内光电效应的一层材料所构成。紧靠光电靶的网电极形成均匀电场,使扫描电子束垂直上靶。

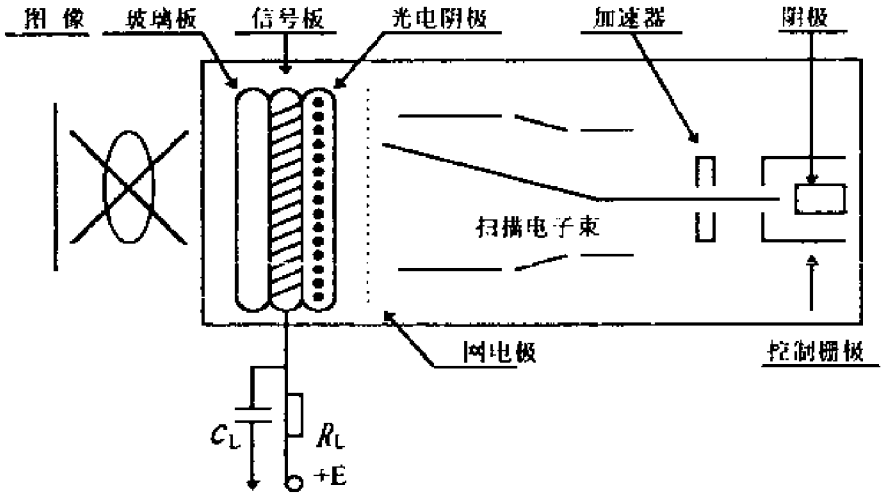


图 1.14 视像管构造略图

当电子束按一定规律扫描时,靶电阻变小,产生电流,流过负载电阻 R_L,电容 C_L 被充电。由于靶面上各点因图像或景物光照不同而有不同的电导,所以流经 R_L 的电流也将随着电导值的不同而变化。这样,电子束扫描时,负载 R_L 上所形成的电信号就反映了图像或景物的亮度变化,"光像"映射为随时间变化的"电图像"信号。

(2) 视像管摄像机方式

面对色彩斑斓的对象图或景物,摄像机如何分离出三基色信号,这是获得高质量图像信息的关键。

① 单管式摄像方式

顾名思义,就是用一个摄像管取出 R、G、B 三基色信号。其原理是在摄像机的靶面上配置非常细的竖条三色条纹滤色片,用摄像管取出 R、G、B 混合信号,该混合信号是根据分时原则,由 R、G、B 三分量依次排列的特殊信号。当显示或量化时,再用频率分离、相位分离、能量分级解调等方法,取出 R、G、B 分量值。

② 双管式摄像方式

一只管提供亮度信号,另一只管子如同单管式那样取出 R、G、B 分量。

③ 三管式摄像方式

彩色图像或景物光像经摄像镜头进入分光系统(图 1.15),第一个分色镜面反射出蓝色

光分量(B),其余光色成分则穿透过去。第二个分色镜反射出红色光成分(R),而让绿色光分量(G)穿透过去。G分量直接经中继透镜到达G摄像管,R、B分量则经反射镜反射后由透镜到达R、B摄像管。这样,三个摄像管分别将各自接收到的单色光信号转换成电信号,预放大后送控制台处理。

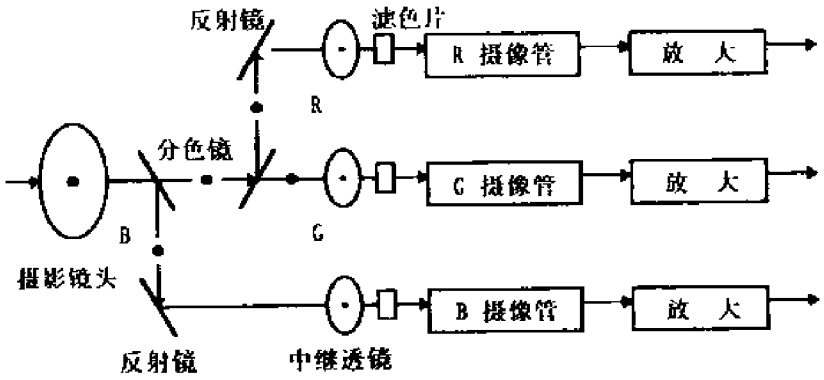


图 1.15 分色镜分光系统

(3) CCD 固体器件摄像机

CCD 中文的意思是电荷耦合器件,其工作原理是:

在硅单晶片上,生成很薄的一层二氧化硅(SiO_2),再于其上蒸发一层间距很小的铝电极条。加电压于电极上后,电场穿透 SiO_2 薄层,并排斥硅中多数载流子,从而在电极下的硅单晶片上形成电荷耗尽层(又称势阱)。当硅改变电极电压,使势阱内的电荷从一个电极传送到另一个电极,取出这移动的电荷就形成输出信号。

扫描系统顺序扫描各像素,在一个扫描周期内,用较长的时间感光、积累电荷,用极短的时间读取图像信号。由于图像各处的光亮不同,耗尽层内"捕捉"的电荷量形成差异,这就完成了图像的光电转换。

采用 CCD 固体器件的彩色摄像机其工作方式分为三种:单板式、二板式、三板式。

① 单板式

使用一个摄像器件,外加滤色片取出三基色。

单板式一般用于家庭摄像机。当前有代表性的单板式彩色摄像机列于表 1.1。

表 1.1 单板式彩色摄像机

性 能	VK-C2000	A	B	C	D
水平分辨率(TV 线)	260	350	260	240	250
垂直分辨率(TV 线)	350	350	350	350	490
灵 敏 度(lx)	500	500	200	150	250
功 耗(W)	5.3	–	8	4.5	7.0

② 二板式

来自摄像镜头的光线,首先经分色镜取出绿色。该绿色分量通过绿色滤色片到达第一个 CCD 摄像器件转换成电信号。穿过分色镜的红、蓝混合信号被红、蓝条形滤色片分时式

分离出红色和蓝色，再顺序送到第二个摄像器件去改变势阱电荷，完成红、蓝色的光电转换。两个摄像器件取得的共三种信号都要送到彩色编码电路，形成 NTSC(National Television System Committee 的缩写)信号。

简言之，二板式就是一个摄像器件用于绿色，一个摄像器件用于红色和蓝色。二板式一般用于专业摄像机。

③ 三板式

有了上面的知识，三板式就不难理解了。穿过镜头的光信号，经过去红外线滤色片去除红外线后，入射到分色镜。两枚分色镜分色得到红、绿、蓝三基色，后由三个 CCD 固体摄像器件分别转换成独立的电信号。这些信号再经放大电路、白平衡电路、信号处理电路和编码器电路而变成 NTSC 信号。

至此，我们已提纲挈领地介绍了扫描仪和电视摄像机的工作原理、组成情况及应用场合。作为图像输入设备，其品种可谓众多，尚未讨论的如飞点扫描器、微密度计、扫描鼓、激光扫描器等，这里不一一论述。

2 图像变换

●傅立叶变换 ●离散余弦变换 ●沃尔什－哈达玛变换
● K－L 变换 ●改变表达式形式 ●显示重要特性

这里所说的图像变换指二维正交变换，它在图像处理中扮演着重要的角色。如傅立叶变换后平均值正比于图像灰度的平均值，高频分量指示图像中目标的边缘信息，利用这些性质可以从图像中抽取特征。又如在变换域中，图像能量往往集中在少数项上，或者说能量主要集中在低频分量上，这时对低频成分分配较多的比特数，对高频成分分配较少的比特数，即可实现图像数据的压缩编码(具体见第 5 章)。再者，舍弃变换系数矩阵中某些幅度小的系数，可缩减计算维数，提高计算的速度。

图像变换算法纷繁，本章将讨论常用的傅立叶变换(Fourier Transform，简写为 FT)、离散余弦变换(Discret Cosine Transform，简写为 DCT)、沃尔什—哈达玛变换(Walsh Hadamard Transform，简写为 WHT)。

2.1 预备知识

为方便读者对本章内容的理解，先简要给出图像变换中可能涉及的一些数学知识。

2.1.1 矩阵与向量

1) 矩阵与分块矩阵

$M \times N$ 矩阵 $\boldsymbol{F}$ 是由元素 $F(m,n)$ 构成的二维阵列。即

$$\boldsymbol{F} = \begin{bmatrix} F(1,1) & F(1,2) & \cdots & F(1,N) \\ F(2,1) & F(2,2) & \cdots & F(2,N) \\ \cdots & \cdots & \cdots & \cdots \\ F(M,1) & F(M,2) & \cdots & F(M,N) \end{bmatrix} \tag{2.1}$$

$m = 1,2,\cdots,M$；$n = 1,2,\cdots,N$。如果 $M = N$，$m \neq n$ 时 $F(m,n) = 0$(非对角线上阵元为零)，则(2.1)式对应的矩阵称之为对角线矩阵。对角线上阵元均为 1 的对角线矩阵叫单位矩阵，用 I 表示。

将阶数比较高的矩阵 $\boldsymbol{F}$ 用若干条纵线和横线分成许多块低阶矩阵，每一块低阶矩阵 $\boldsymbol{F}_{ij}$ 称为 $\boldsymbol{F}$ 的子块，以子块为元素的矩阵称为分块矩阵。如

$$\boldsymbol{F} = \left[\begin{array}{cc:cc} F(1,1) & F(1,2) & F(1,3) & F(1,4) \\ F(2,1) & F(2,2) & F(2,3) & F(2,4) \\ \hdashline F(3,1) & F(3,2) & F(3,3) & F(3,4) \\ F(4,1) & F(4,2) & F(4,3) & F(4,4) \end{array}\right]$$

$$\boldsymbol{F}=\begin{bmatrix}\boldsymbol{F}_{11} & \boldsymbol{F}_{12}\\ \boldsymbol{F}_{21} & \boldsymbol{F}_{22}\end{bmatrix} \tag{2.2}$$

令矩阵 $\boldsymbol{A}$ 和矩阵 $\boldsymbol{B}$ 的行、列数相同，同样的方式分块，$\boldsymbol{A}$ 的子块以 $\boldsymbol{A}_{mn}$ 表示，$\boldsymbol{B}$ 的子块以 $\boldsymbol{B}_{mn}$ 表示。$m=1,2,\cdots,r;n=1,2,\cdots s$。分块矩阵之和为

$$\boldsymbol{A}+\boldsymbol{B}=\begin{bmatrix}\boldsymbol{A}_{11}+\boldsymbol{B}_{11} & \boldsymbol{A}_{12}+\boldsymbol{B}_{12} & \cdots & \boldsymbol{A}_{1s}+\boldsymbol{B}_{1s}\\ \boldsymbol{A}_{21}+\boldsymbol{B}_{21} & \boldsymbol{A}_{22}+\boldsymbol{B}_{22} & \cdots & \boldsymbol{A}_{2s}+\boldsymbol{B}_{2s}\\ \cdots & \cdots & \cdots & \cdots\\ \boldsymbol{A}_{r1}+\boldsymbol{B}_{r1} & \boldsymbol{A}_{r2}+\boldsymbol{B}_{r2} & \cdots & \boldsymbol{A}_{rs}+\boldsymbol{B}_{rs}\end{bmatrix} \tag{2.3}$$

2) 矩阵运算与变换

(1) 矩阵加法

两个同型的矩阵 $\boldsymbol{A}$、$\boldsymbol{B}$ 的和是一个同型的矩阵 $\boldsymbol{C}$，记作

$$\boldsymbol{C}=\boldsymbol{A}+\boldsymbol{B} \tag{2.4}$$

阵元 $C(m,n)=A(m,n)+B(m,n)$。

(2) 矩阵乘法

$M\times P$ 矩阵 $\boldsymbol{A}$ 和 $P\times N$ 矩阵 $\boldsymbol{B}$（前一个矩阵的列数与后面一个矩阵的行数相等）之乘积为 $M\times N$ 矩阵 $\boldsymbol{C}$。

$$\boldsymbol{C}=\boldsymbol{AB} \tag{2.5}$$

阵元 $C(m,n)$ 的一般形式为

$$\begin{aligned}C(m,n)&=\sum_{p=1}^{P}A(m,p)B(p,n)\\&=A(m,1)B(1,n)+A(m,2)B(2,n)+\cdots+A(m,p)B(p,n)\end{aligned} \tag{2.6}$$

式中 $m=1,2,\cdots,M;\ n=1,2,\cdots,N;p=1,2,\cdots,P$。如

$$\boldsymbol{A}=\begin{bmatrix}A_{11} & A_{12} & A_{13}\\ A_{21} & A_{22} & A_{23}\end{bmatrix}$$

$$\boldsymbol{B}=\begin{bmatrix}B_{11} & B_{12}\\ B_{21} & B_{22}\\ B_{31} & B_{32}\end{bmatrix}$$

$$\boldsymbol{AB}=\begin{bmatrix}A_{11}B_{11}+A_{12}B_{12}+A_{13}B_{31} & A_{11}B_{12}+A_{12}B_{22}+A_{13}B_{32}\\ A_{21}B_{11}+A_{22}B_{21}+A_{23}B_{31} & A_{21}B_{12}+A_{22}B_{22}+A_{23}B_{32}\end{bmatrix}$$

(3) 矩阵的数乘

一个数 λ 乘以矩阵 $\boldsymbol{A}$，其积等于 $\boldsymbol{A}$ 的每一个阵元 $A(m,n)$ 乘以该数。即

$$\lambda\boldsymbol{A}=[\lambda A(m,n)] \tag{2.7}$$

(4) 矩阵的左直积

$P\times Q$ 矩阵 $\boldsymbol{A}$ 与 $M\times N$ 矩阵 $\boldsymbol{B}$ 的左直积是 $PM\times QN$ 矩阵 $\boldsymbol{C}$，即

$$\boldsymbol{C}=\boldsymbol{A}\otimes\boldsymbol{B}=\left[\begin{array}{c:c:c:c}\boldsymbol{B}(1,1)\boldsymbol{A} & \boldsymbol{B}(1,2)\boldsymbol{A} & \cdots & \boldsymbol{B}(1,N)\boldsymbol{A}\\ \hdashline \boldsymbol{B}(2,1)\boldsymbol{A} & \boldsymbol{B}(2,2)\boldsymbol{A} & \cdots & \boldsymbol{B}(2,N)\boldsymbol{A}\\ \hdashline \cdots & \cdots & \cdots & \cdots\\ \hdashline \boldsymbol{B}(M,1)\boldsymbol{A} & \boldsymbol{B}(M,2)\boldsymbol{A} & \cdots & \boldsymbol{B}(M,N)\boldsymbol{A}\end{array}\right] \tag{2.8}$$

(5) 矩阵求逆

方矩阵 A 的逆矩阵用 A^{-1} 表示。逆矩阵具有以下性质：

① $AA^{-1}=A^{-1}A=I$(单位矩阵)。

② 如果逆矩阵 A^{-1} 存在，则称矩阵 A 是非奇异的，否则 A 是奇异的。

③ 若矩阵有逆，则逆是唯一的。逆矩阵的逆是原矩阵，即

$$[A^{-1}]^{-1}=A \tag{2.9}$$

④ 若矩阵 A 和 B 是非奇异矩阵，标量 $k\neq 0$，则

$$[AB]^{-1}=B^{-1}A^{-1} \tag{2.10}$$

$$[kA]^{-1}=k^{-1}A^{-1} \tag{2.11}$$

(6) 矩阵转置

把矩阵 A 的行与列互换，得到的矩阵称为 A 的转置矩阵，记作 A^T。对任何矩阵 A 和 B，有

$$[AB]^T=B^TA^T \tag{2.12}$$

如

$$AB=\begin{bmatrix}A_{11} & A_{12}\\ A_{21} & A_{22}\\ A_{31} & A_{32}\end{bmatrix}\begin{bmatrix}B_{11} & B_{12}\\ B_{21} & B_{22}\end{bmatrix}$$

$$=\begin{bmatrix}A_{11}B_{11}+A_{12}B_{21} & A_{11}B_{12}+A_{12}B_{22}\\ A_{21}B_{11}+A_{22}B_{21} & A_{21}B_{12}+A_{22}B_{22}\\ A_{31}B_{11}+A_{32}B_{21} & A_{31}B_{12}+A_{32}B_{22}\end{bmatrix}$$

$$B^TA^T=\begin{bmatrix}B_{11} & B_{21}\\ B_{12} & B_{22}\end{bmatrix}\begin{bmatrix}A_{11} & A_{21} & A_{31}\\ A_{12} & A_{22} & A_{32}\end{bmatrix}$$

$$=\begin{bmatrix}B_{11}A_{11}+B_{21}A_{12} & B_{11}A_{21}+B_{21}A_{22} & B_{11}A_{31}+B_{21}A_{32}\\ B_{12}A_{11}+B_{22}A_{12} & B_{12}A_{21}+B_{22}A_{22} & B_{12}A_{31}+A_{32}\end{bmatrix}$$

当方矩阵 A 中的元素 $A(m,n)=A(n,m)$ 时，A 叫做对称矩阵。若 $A(m,n)=-A(n,m)$，则 A 为反对称矩阵(对角元素必为零)。例如

对称矩阵

$$A=\begin{bmatrix}2 & 0 & 1\\ 0 & 3 & 5\\ 1 & 5 & 4\end{bmatrix}$$

反对称矩阵

$$A=\begin{bmatrix}0 & 2 & -3\\ -2 & 0 & -5\\ 3 & 5 & 0\end{bmatrix}$$

对称矩阵与其转置矩阵相等，即

$$A=A^T \tag{2.13}$$

若 A 是非奇异的，则 A^T 也是非奇异的，并且

$$[A^T]^{-1}=[A^{-1}]^T \tag{2.14}$$

(7) 矩阵的迹

$N \times N$ 方矩阵 F 的迹是对角线阵元的和，可表示为

$$\mathrm{tr}[F] = \sum_{n=1}^{N} F(n,n) \tag{2.15}$$

若 A 和 B 是方矩阵，则

$$\mathrm{tr}[AB] = \mathrm{tr}[BA] \tag{2.16}$$

3) 向量

对一幅 $N_1 \times N_2$ 的数字图像矩阵 F，其对应的向量形式为

$$f = \sum_{n=1}^{N_2} N_n F V_n \tag{2.17}$$

这里 V_n 和 N_n 分别被定义为

$$V_n = \begin{bmatrix} 0 \\ \cdots \\ 0 \\ 1 \\ 0 \\ \cdots \\ 0 \end{bmatrix} \begin{matrix} 1 \\ \cdots \\ n-1 \\ n \\ n+1 \\ \cdots \\ N_2 \end{matrix} \qquad N_n = \begin{bmatrix} 0 \\ \cdots \\ 0 \\ I \\ 0 \\ \cdots \\ 0 \end{bmatrix} \begin{matrix} 1 \\ \cdots \\ n-1 \\ n \\ n+1 \\ \cdots \\ N_2 \end{matrix} \tag{2.18}$$

式中 V_n 为 $N_2 \times 1$ 运算向量；

N_n 为 $N_1 N_2 \times N_1$ 矩阵；

0 为零矩阵；

I 为单位矩阵。

2.1.2 线性算子

1) 离散图像的统计特性

对 $N_1 \times N_2$ 离散图像矩阵 F，具有下列统计特性：

(1) 平均值

$$E\{F\} = [E\{F(n_1,n_2)\}] \tag{2.19}$$

其中 $1 \leqslant n_i \leqslant N_i$。如果将图像阵列写成列扫描向量，那么图像向量 f 的平均值是

$$\eta_f = E\{f\} = \sum_{n=1}^{N_2} N_n E\{F\} V_n \tag{2.20}$$

(2) 图像矩阵 F 的自相关函数

$$R(n_1,n_2;n_3,n_4) = E\{F(n_1,n_2)F^*(n_3,n_4)\} \tag{2.21}$$

如果图像阵列 F 是广义平稳的，则

$$R(n_1,n_2;n_3,n_4) = R(n_1-n_3,n_2-n_4) = R(j,k) \tag{2.22}$$

式中 $j = n_1 - n_3$；$k = n_2 - n_4$。

(3) 图像矩阵 F 的协方差函数

$$K(n_1,n_2;n_3,n_4) = E\{[F(n_1,n_2) - E\{F(n_1,n_2)\}] \cdot [F^*(n_3,n_4) - E\{F^*(n_3,n_4)\}]\} \tag{2.23}$$

(4) 图像矩阵 F 的方差函数

$$\sigma^2(n_1,n_2)=K(n_1,n_2;n_1,n_2) \tag{2.24}$$

(5) 相关矩阵

如果图像矩阵 $\boldsymbol{F}$ 用向量 $\boldsymbol{f}$ 表示，则 $\boldsymbol{f}$ 的相关矩阵定义为

$$\boldsymbol{R}_f=\sum_{m=1}^{N_2}\sum_{n=1}^{N_2}\boldsymbol{N}_m\boldsymbol{R}_{m,n}\boldsymbol{N}_n^T \tag{2.25}$$

式中 $\boldsymbol{R}_{m,n}$ 是 $\boldsymbol{F}$ 中第 m 列和第 n 列的 $N_1\times N_2$ 相关矩阵。因此，$\boldsymbol{R}_f$ 可用子块形式表示成

$$\boldsymbol{R}_f=\begin{bmatrix}\boldsymbol{R}_{1,1} & \boldsymbol{R}_{1,2} & \cdots & \boldsymbol{R}_{1,N_2}\\ \boldsymbol{R}_{2,1} & \boldsymbol{R}_{2,2} & \cdots & \boldsymbol{R}_{2,N_2}\\ \cdots & \cdots & \cdots & \cdots\\ \boldsymbol{R}_{N_2,1} & \boldsymbol{R}_{N_2,2} & \cdots & \boldsymbol{R}_{N_2,N_2}\end{bmatrix} \tag{2.26}$$

(6) $\boldsymbol{f}$ 的协方差矩阵

可由相关矩阵 $\boldsymbol{R}_f$ 与平均值向量 $\boldsymbol{\eta}_f$ 的积的差来求出：

$$\boldsymbol{K}_f=\boldsymbol{R}_f-\boldsymbol{\eta}_f\boldsymbol{\eta}_f^{*T} \tag{2.27}$$

对于广义平稳图像 $\boldsymbol{F}$，存在以下形式的 Toeplitz 分块矩阵：

$$\boldsymbol{K}_f=\begin{bmatrix}\boldsymbol{K}_1 & \boldsymbol{K}_2 & \boldsymbol{K}_3 & \cdots & \boldsymbol{K}_{N_2}\\ \boldsymbol{K}_2^* & \boldsymbol{K}_1 & \boldsymbol{K}_2 & \cdots & \boldsymbol{K}_{N_2-1}\\ \cdots & \cdots & \cdots & \cdots & \cdots\\ \boldsymbol{K}_{N_2}^* & \boldsymbol{K}_{N_2-1}^* & \boldsymbol{K}_{N_2-2}^* & \cdots & \boldsymbol{K}_1\end{bmatrix} \tag{2.28}$$

2) 广义线性算子

令广义线性运算后，输出图像 $P(m_1,m_2)$ 与输入图像 $F(n_1,n_2)$ 之间存在以下关系：

$$P(m_1,m_2)=\sum_{n_1=1}^{N_1}\sum_{n_2=2}^{N_2}F(n_1,n_2)O(n_1,n_2;m_1,m_2) \tag{2.29}$$

其中，输入图像的阶数为 $N_1\times N_2$；输出图像的阶数为 $M_1\times M_2$；核 $O(n_1,n_2;m_1,m_2)$ 为一种加权因子，是输入和输出图像坐标的函数。

令 $\boldsymbol{f}$ 为对 $[F(n_1,n_2)]$ 进行列扫描所得的向量；$\boldsymbol{P}$ 是对输出图像 $[P(m_1,m_2)]$ 作列扫描的向量；$N_1=N_2=N$；$M_1=M_2=M$；n_1、n_2 分别为 $1,2,\cdots,N$；m_1、m_2 分别为 $1,2,\cdots,M$；$\boldsymbol{T}$ 是一个 $M^2\times N^2$ 阶的线性变换矩阵。则对 $N^2\times 1$ 输入图像向量 $\boldsymbol{f}$ 进行线性变换后，$M^2\times 1$ 输出图像向量

$$\boldsymbol{P}=\boldsymbol{T}\boldsymbol{f} \tag{2.30}$$

线性变换矩阵 $\boldsymbol{T}$ 的子矩阵形式为

$$\boldsymbol{T}=\begin{bmatrix}\boldsymbol{T}(1,1) & \boldsymbol{T}(1,2) & \cdots & \boldsymbol{T}(1,N)\\ \boldsymbol{T}(2,1) & \boldsymbol{T}(2,2) & \cdots & \boldsymbol{T}(2,N)\\ \cdots & \cdots & \cdots & \cdots\\ \boldsymbol{T}(M,1) & \boldsymbol{T}(M,2) & \cdots & \boldsymbol{T}(M,N)\end{bmatrix} \tag{2.31}$$

3) 线性算子的统计表示

在把输入图像阵列 $\boldsymbol{F}$ 看成是随机抽像阵列，其一阶矩和二阶矩已知的情况下，输出图

像 P 的一阶矩和二阶矩也是可以确定的。

(1) 输出图像均值

$$E\{P(m_1,m_2)\} = E\{\sum_{n_1=1}^{N_1}\sum_{n_2=1}^{N_2}\{F(n_1,n_2)\}O(n_1,n_2;m_1,m_2)\} \tag{2.32}$$

由于期望算子是线性的,所以

$$E\{P(m_1,m_2)\} = E\{\sum_{n_1=1}^{N_1}\sum_{n_2=1}^{N_2}E\{F(n_1,n_2)\}O(n_1,n_2;m_1,m_2)\} \tag{2.33}$$

(2) 输出图像的相关函数

$$\begin{aligned}R_p(m_1,m_2;m_3,m_4) &= E\{P(m_1,m_2)P^*(m_3,m_4)\}\\ &= E\{[\sum_{n_1=1}^{N_3}\sum_{n_2=1}^{N_2}(F(n_1,n_2)O(n_1,n_2;m_1,m_2)\\ &\quad\cdot\sum_{n_3=1}^{N_1}\sum_{n_4=1}^{N_2}F^*(n_3,n_4)O^*(n_3,n_4;m_3,m_4)]\}\end{aligned}$$

在随机分量相乘后再作有关运算,得

$$\begin{aligned}R_p(m_1,m_2;m_3,m_4) &= \sum_{n_1=1}^{N_1}\sum_{n_2=1}^{N_2}\sum_{n_3=1}^{N_3}\sum_{n_4=1}^{N_4}R_F(n_1,n_2;n_3,n_4)\\ &\quad\cdot O(n_1,n_2;m_1,m_2)O^*(n_3,n_4;m_3,m_4)\end{aligned} \tag{2.34}$$

式中 $R_F(n_1,n_2;n_3,n_4)$为输入图像的相关函数。

(3) 输出图像的协方差

$$\begin{aligned}K_p(m_1,m_2;m_3,m_4) &= \sum_{n_1=1}^{N_1}\sum_{n_2=1}^{N_2}\sum_{n_3=1}^{N_3}\sum_{n_4=1}^{N_4}K_p(n_1,n_2;n_3,n_4)\\ &\quad\cdot O(n_1,n_2;m_1,m_2)O^*(n_3,n_4;m_3,m_4)\end{aligned} \tag{2.35}$$

若输入和输出图像分别以向量 f 和 P 表示,则(2.33)、(2.34)、(2.35)式将分别如(2.36)、(2.37)、(2.38)式所示:

均值

$$\boldsymbol{\eta}_p = E\{\boldsymbol{P}\} = \boldsymbol{T}\boldsymbol{\eta}_f \tag{2.36}$$

相关矩阵

$$\boldsymbol{R}_p = E\{\boldsymbol{P}\boldsymbol{P}^{*T}\} = \boldsymbol{T}\boldsymbol{R}_f\boldsymbol{T}^{*T} \tag{2.37}$$

协方差矩阵

$$\boldsymbol{K}_p = \boldsymbol{T}\boldsymbol{K}_f\boldsymbol{T}^{*T} \tag{2.38}$$

2.1.3 酉变换算子

酉变换是一种准可逆的线性正交变换。$N_1 \times N_2$ 的离散图像阵列经正向变换后生成 $N_1 \times N_2$ 的变换域图像阵列。按定义

正变换

$$\tilde{f}(m_1,m_2) = \sum_{n_1=1}^{N_1}\sum_{n_2=1}^{N_2}F(n_1,n_2)A(n_1,n_2;m_1,m_2) \tag{2.39}$$

反变换

$$F(n_1,n_2)=\sum_{m_1=1}^{N_1}\sum_{m_2=1}^{N_2}\tilde{f}(m_1,m_2)B(n_1,n_2;m_1,m_2) \tag{2.40}$$

式中 $\tilde{f}(m_1,m_2)$为变换域阵列；

$F(n_1,n_2)$为从变换域恢复的原图像阵列；

$A(n_1,n_2;m_1,m_2)$为正变换核；

$B(n_1,n_2;m_1,m_2)$反变换核。

如果正变换核 $A(n_1,n_2;m_1,m_2)$及反变换核 $B(n_1,n_2;m_1,m_2)$满足下列条件，则上述变换称之为酉变换：

$$\sum_{m_1}\sum_{m_2}A(n_1,n_2;m_1,m_2)A^*(j_1,j_2;m_1,m_2)=\delta(n_1-j_1,n_2-j_2) \tag{2.41}$$

$$\sum_{m_1}\sum_{m_2}B(n_1,n_2;m_1,m_2)B^*(j_1,j_2;m_1,m_2)=\delta(n_1-j_1;n_2-j_2) \tag{2.42}$$

$$\sum_{n_1}\sum_{n_2}A(n_1,n_2;m_1,m_2)A^*(n_1,n_2;k_1,k_2)=\delta(m_1-k_1;m_2-k_2) \tag{2.43}$$

$$\sum_{n_1}\sum_{n_2}B(n_1,n_2;m_1,m_2)B^*(n_1,n_2;k_1,k_2)=\delta(m_1-k_1,m_2-k_2) \tag{2.44}$$

如果变换核能写成下列形式，就定义变换可分

$$A(n_1,n_2;m_1,m_2)=A_c(n_1,m_1)A_R(n_2,m_2) \tag{2.45}$$

$$B(n_1,n_2;m_1,m_2)=B_c(n_1,m_1)B_R(n_2,m_2) \tag{2.46}$$

下标 C、R 表示变换的方向，C 指示列方向，R 指示行方向。在可分离的情况下，二维酉变换可通过两步完成。先对图像 $F(n_1,n_2)$的每一列作一维变换，取得

$$F_c(m_1,n_2)=\sum_{n_1=1}^{N_1}F(n_1,n_2)A_c(n_1,m_1) \tag{2.47}$$

然后再沿 F_c 的每一行作另一次一维变换，得

$$\tilde{f}(m_1,m_2)=\sum_{n_2=1}^{N_2}F_c(m_1,n_2)A_R(n_2,m_2) \tag{2.48}$$

同样，反变换时也可由两步完成。即

$$\tilde{f}_c(m_1,n_2)=\sum_{n_1=1}^{N_1}\tilde{f}(n_1,n_2)B_c(n_1,m_1) \tag{2.49}$$

$$F(n_1,n_2)=\sum_{n_2=1}^{N_2}\tilde{f}_c(m_1,n_2)B_c(n_2,m_2) \tag{2.50}$$

下面给出酉变换的向量表示。设 F 和 f 分别为原图像矩阵和向量表示，F 和 q 分别为变换域图像矩阵和向量形式。于是，正、反变换可表示成

$$\boldsymbol{q}=\boldsymbol{A}\boldsymbol{f} \tag{2.51}$$

$$\boldsymbol{f}=\boldsymbol{B}\boldsymbol{q} \tag{2.52}$$

A 为正交变换矩阵，B 为逆变换矩阵。显然

$$\boldsymbol{B}=\boldsymbol{A}^{-1} \tag{2.53}$$

对于酉变换，逆矩阵由下式求得：

$$\boldsymbol{A}^{-1}=\boldsymbol{A}^{*T} \tag{2.54}$$

令 A 为实正交矩阵，则

$$\boldsymbol{A}^{-1} = \boldsymbol{A}^{T} \tag{2.55}$$

若变换核可分，正变换与反变换矩阵将分别为

$$\bar{\boldsymbol{f}} = \boldsymbol{A}_c \boldsymbol{F} \boldsymbol{A}_R^{T} \tag{2.56}$$

$$\boldsymbol{F} = \boldsymbol{B}_c \bar{\boldsymbol{f}}\, \boldsymbol{B}_R^{T} \tag{2.57}$$

式中 A_R、A_c 为行和列变换矩阵；

$$\boldsymbol{B}_c = \boldsymbol{A}_c^{-1};$$

$$\boldsymbol{B}_R = \boldsymbol{A}_R^{-1}。$$

在后面的讨论中，为了简化二维正交变换的分析，将(2.39)、(2.40)式的条件作某些修正。即令 $N_1 = N_2 = N$，输入图像阵列为 $F(j,k)$，变换后的图像阵列用 $\bar{f}(u,v)$ 或 $F(u,v)$ 表示，变量 j,k,u,v 分别为 $0,1,2\cdots,N-1$。于是，酉变换关系式变成：

正变换

$$\bar{f}(u,v) = \sum_{j=0}^{N-1}\sum_{k=0}^{N-1} F(j,k)A(j,k;u,v) \tag{2.58}$$

反变换

$$F(j,k) = \sum_{u=0}^{N-1}\sum_{v=0}^{N-1} \bar{f}(u,v)B(j,k;u,v) \tag{2.59}$$

2.2 傅立叶变换(FT)

傅立叶变换应用十分广泛，如图像特征提取、空间频率域滤波、图像恢复、纹理分析等。

本节先介绍二维离散傅立叶变换(Discrete Fourier Transform，简写为 DFT)，然后再讨论快速傅立叶变换(Fast Foureier Transform，简写为 FFT)。

2.2.1 二维离散傅立叶变换(DFT)

如果连续图像 $f(x,y)$ 可积，则其傅立叶变换为

$$f(u,v) = \int_{-\infty}^{+\infty}\int f(x,y)\exp[-i2\pi(ux+vy)]\mathrm{d}x\mathrm{d}y \tag{2.60}$$

对应的反变换为

$$f(x,y) = \int_{-\infty}^{+\infty}\int \bar{f}(u,v)\exp[i2\pi(ux+vy)]\mathrm{d}u\mathrm{d}v \tag{2.61}$$

(u,v) 为傅立叶变换域的空间频率分量。

对 $N\times N$ 数字图像阵列 $F(j,k)$，其二维离散傅立叶变换的级数形式定义为

$$\bar{f}(u,v) = \frac{1}{N}\sum_{j=0}^{N-1}\sum_{k=0}^{N-1} F(j,k)\exp[-i2\pi(uj+vk)/N] \tag{2.62}$$

式中 $i=\sqrt{-1}$。其反变换为

$$F(j,k) = \frac{1}{N}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1} \bar{f}(u,v)\exp[i2\pi(uj+vk)/N] \tag{2.63}$$

(2.62)、(2.63)式并不是二维离散傅立叶变换式的唯一形式，如有些作者采用相反正负号的变换核，有的系数项取 N^{-2}。根据前面的讨论，变换核是可分的，复指数形式的变换核可分解为正弦分量和余弦分量两部分：

$$A(j,k;u,v) = \exp[-i2\pi(uj+vk)/N]$$
$$= \cos[2\pi(uj+vk)/N] - i\sin[2\pi(uj+vk)/N] \tag{2.64}$$

$$B(j,k;u,v) = \exp[i2\pi(uj+vk)/N]$$
$$= \cos[2\pi(uj+vk)/N] + i\sin[2\pi(uj+vk)/N] \tag{2.65}$$

DFT 有许多重要的性质：

(1) 一维正向傅立叶变换是二维正、反变换的基础

已知

$$\exp[\pm i2\pi(uj+vk)/N] = \exp[\pm i2\pi(uj/N)]\cdot\exp[\pm i2\pi(vk/N)]$$

使用变换核的可分性质，由(2.62)式可得正变换表达式

$$\tilde{f}(u,v) = \frac{1}{N}\sum_{j=0}^{N-1}\exp[-i2\pi uj/N]\cdot\{\sum_{k=0}^{N-1}F(j,k)\exp[-i2\pi uk/N]\}$$
$$= \frac{1}{N}\sum_{j=0}^{N-1}F(j,v)\exp[-i2\pi uj/N] \tag{2.66}$$

式中

$$F(j,v) = N\{\frac{1}{N}\sum_{k=0}^{N-1}F(j,k)\exp[-i2\pi vk/N]\}$$

类似地，反变换表达式可写成

$$F(j,k) = \frac{1}{N}\sum_{u=0}^{N-1}\exp[i2\pi uj/N]\cdot\{\sum_{v=0}^{N-1}F(u,v)\exp[i2\pi vk/N]\}$$
$$= \frac{1}{N}\sum_{u=0}^{N-1}F(u,k)\exp[i2\pi uj/N] \tag{2.67}$$

式中

$$F(u,k) = N\{\frac{1}{N}\sum_{v=0}^{N-1}F(u,v)\exp[i2\pi vk/N]\}$$

$F(u,v)=f(u,v)$，是 DFT 变换域的另一种表示符号。

由(2.64)式可以看出，二维离散傅立叶变换可以用两步一维的离散傅立叶变换来进行，先沿着 $F(j,k)$的诸列取变换，然后沿着所得结果的行取变换(图 2.1(a))。同样，由(2.65)式知，二维反变换也可以通过两步一维的反傅立叶变换来实现(图 2.1(b))。

(2) 傅立叶变换域原点的频谱分量 $\tilde{f}(0,0)$ 是图像面的空间平均值的 N 倍

$$\tilde{f}(u,v) = \frac{1}{N}\sum_{j=0}^{N-1}\sum_{k=0}^{N-1}F(j,k)\exp[-i2\pi(uj+vk)/N)]$$

$$\tilde{f}(0,0) = \frac{1}{N}\sum_{j=0}^{N-1}\sum_{k=0}^{N-1}F(j,k)$$
$$= N[\frac{1}{N^2}\sum_{j=0}^{N-1}\sum_{k=0}^{N-1}F(j,k)] = N\bar{F}(j,k) \tag{2.68}$$

(3) 变换域是周期性的

以 $u=u+mN$，$v=v+nN$ 代入级数形式表示的 DFT 表达式(2.62)式，得

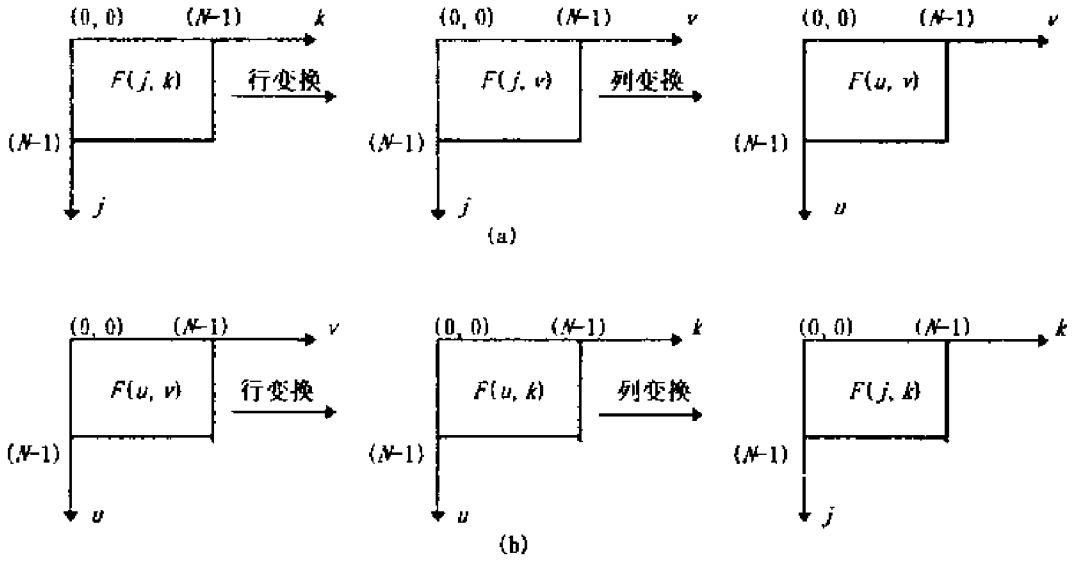


图 2.1　先列后行两步式变换流程

(a)二维傅立叶正变换；　(b)二维反傅立叶变换

$$\begin{aligned}\tilde{f}(u+mN,v+nN) &= \frac{1}{N}\sum_{j=0}^{N-1}\sum_{k=0}^{N-1}F(j,k)\\ &\quad\cdot\exp\{-i2\pi[(u+mN)j+(v+nN)k]/N\}\\ &= \frac{1}{N}\sum_{j=0}^{N-1}\sum_{k=0}^{N-1}F(j,k)\cdot\exp\{-i2\pi(uj+vk)/N\}\\ &\quad\cdot\exp\{-i2\pi(mj+nk)\}\end{aligned}\tag{2.69}$$

当 m,n 为整数时，$\exp\{-i2\pi(mj+nk)\}$ 为单位值，从而证明变换域是循环性的。也就是说，在上述条件下，有

$$\tilde{f}(u,v)=\tilde{f}(u+mN,v+nN)\tag{2.70}$$

从直观上看，实现周期性变化的图像阵列，在水平和垂直方向上呈周期性变换，即图像的右边和左边邻接，上顶和下底毗连，其中包括空间域 $F(j,k)$。见图 2.2 所示。

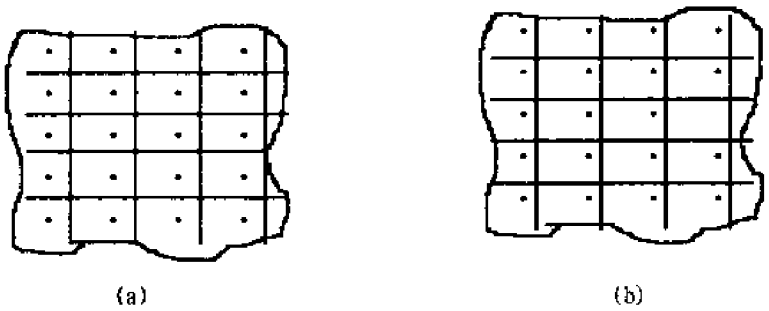


图 2.2　循环图像阵列和 DFT 变换域

(a)DFT 变换域；　(b)空间域

(4) 共轭对称性

图像函数 $F(j,k)$ 一般是正实数，其傅立叶变换是复数。在变换域内，每个变换系数有

实数和虚数两个分量，整个变换场将具有 $2N^2$ 个分量。这样，我们会自然想到傅立叶变换要增加维数。实际情况并非如此，因为傅立叶变换具有共轭对称性，只要求出一半阵元，整个变换域便可知道。已知，在 m,n 分别为 $0,1,2,\cdots$ 时，$\exp\{-i2\pi(mj+nk)\}$ 为单位值。这时

$$\tilde{f}(u+mN,v+nN)=\frac{1}{N}\sum_{j=0}^{N-1}\sum_{k=0}^{N-1}F(j,k)\cdot\exp[-i2\pi(uj+vk)/N]$$

从而共轭

$$\tilde{f}^*(u+mN,v+nN)=\frac{1}{N}\sum_{j=0}^{N-1}\sum_{k=0}^{N-1}F(j,k)\cdot\exp[i2\pi(uj+vk)/N]$$

以 $u=-u,v=-v$ 代入上式，有

$$\begin{aligned}\tilde{f}^*(-u+mN,-v+nN)&=\frac{1}{N}\sum_{j=0}^{N-1}\sum_{k=0}^{N-1}F(j,k)\cdot\exp[i2\pi(-uj-vk)/N]\\&=\frac{1}{N}\sum_{j=0}^{N-1}\sum_{k=0}^{N-1}F(j,k)\exp[-i2\pi(uj+vk)/N]\end{aligned}$$

所以，共轭表示式为

$$\tilde{f}^*(-u+mN,-v+nN)=\tilde{f}(u,v)\tag{2.71}$$

(5) 变换域频移性

由(2.63)式知，反傅立叶变换为

$$F(j,k)=\frac{1}{N}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1}\tilde{f}(u,v)\exp[i2\pi(uj+vk)/N]$$

$$\begin{aligned}\tilde{f}^{-1}\{\tilde{f}(u-u_0,v-v_0)\}&=\frac{1}{N}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1}\tilde{f}(u-u_0),v-v_0)\cdot\exp\{i2\pi[(u-u_0+u_0)j\\&\quad+(v-v_0+v_0)k]/N\}\\&=\frac{1}{N}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1}\tilde{f}(u',v')\cdot\exp\{i2\pi[(u'+u_0)j+(v'+v_0)k]/N\}\\&=F(j,k)\exp[i2\pi(u_0j+v_0k)/N]\end{aligned}\tag{2.72}$$

式中 $u'=u-u_0,v'=v-v_0$。频域中位移(u_0,v_0)相当于空间域乘以一复正弦函数。

令 $u_0=\frac{N}{2},v_0=\frac{N}{2}$，则

$$\begin{aligned}\exp[i2\pi(u_0j+v_0k)/N]&=\exp[i2\pi(\frac{N}{2}j+\frac{N}{2}k)/N]\\&=\exp[i\pi(j+k)]=(-1)^{j+k}\end{aligned}\tag{2.73}$$

结合(2.72)、(2.73)式得

$$\tilde{f}(u-\frac{N}{2},v-\frac{N}{2})\Leftrightarrow F(j,k)(-1)^{j+k}\tag{2.74}$$

符号$\Leftrightarrow$表示空间域与频率域之间的对应变换。上式表明，原图像乘以$(-1)^{j+k}$，则傅立叶变换的原点将从(0,0)移到 $N\times N$ 频率域的中心 $(\frac{N}{2},\frac{N}{2})$。

(6) 空间域平移性

空间域内图像 $F(j,k)$原点平移到(j_0,k_0)时，其对应的频谱变换关系为

$$F(j-j_0,k-k_0)\Leftrightarrow\tilde{f}(u,v)\exp[-i2\pi(uj_0+vk_0)/N]\tag{2.76}$$

即频谱要乘上一个负的指数项，造成相位平移，而幅度不改变。因为

$$|\bar{f}(u,v)\exp[-i2\pi(uj_0+vk_0)/N]|=|\bar{f}(u,v)|$$

(7) 旋转不变性

在空间域以极坐标 r，θ 取代 j,k；在变换域以 ω,Φ 代替 u,v。致使

$$\begin{cases}j=r\cos\theta\\k=r\sin\theta\end{cases}$$

$$\begin{cases}u=\omega\cos\Phi\\v=\omega\sin\Phi\end{cases}$$

显然，在 DFT 变换前图像为 $F(r,\theta)$，DFT 变换后图像为 $\bar{f}(\omega,\Phi)$。可以证明存在以下变换对：

$$F(r,\theta+\theta_0)\Leftrightarrow\bar{f}(\omega,\Phi+\theta_0) \tag{2.77}$$

这表明，图像阵列 $F(j,k)$在空间域中旋转 θ_0 角度后，变换系数矩阵在频率域中也旋转同样的角度。进一步，如果变换系数阵列在频域中旋转 θ_0 角度，则反变换后获得的空间域图像 $F(j,k)$必然旋转 θ_0 角度。

(8) 卷积定理

二维连续函数 $F(j,k)$、$G(j,k)$的卷积记作 $F(j,k)*G(j,k)$，其定义为

$$F(j,k)*G(j,k)=\int_{-\infty}^{+\infty}\int F(\alpha,\beta)G(j-\alpha,k-\beta)\mathrm{d}\alpha\mathrm{d}\beta \tag{2.78}$$

α、β 是伪变量。对应的卷积定理由下式表示：

$$F(j,k)*G(j,k)\Leftrightarrow F(u,v)\cdot G(u,v) \tag{2.79}$$

$$F(j,k)\cdot G(j,k)\Leftrightarrow F(u,v)*G(u,v) \tag{2.80}$$

$F(u,v)$、$G(u,v)$分别是 $F(j,k)$、$G(j,k)$的变换域。

这表明，两个二维图像函数在空间域中的卷积可以由相应的 DFT 之积的反变换得到。而频率域中的卷积则可用空间域中两函数乘积的 DFT 操作而得。

卷积定理同样适用于二维离散函数。稍有差别的是由于二维离散函数的傅立叶变换及其反变换具有周期性，所以需对二维离散函数的定义域加以扩展，以防止卷积后产生交叠误差。现在假定离散函数 $F(j,k)$、$G(j,k)$是在 j 方向和 k 方向上周期分别为 M 和 N 的周期函数，则卷积也是具有同样周期的周期函数。

令 $F(j,k)$、$G(j,k)$的大小分别为 $A\times B$ 和 $C\times D$，它们的定义域是：

$F(j,k)$：　　$0\leqslant j\leqslant A-1$　，　$0\leqslant k\leqslant B-1$

$G(j,k)$：　　$0\leqslant j\leqslant C-1$　，　$0\leqslant k\leqslant D-1$

可以证明，只有周期 M 和 N 的取值为

$$M\geqslant A+C-1$$

$$N\geqslant B+D-1$$

时，其卷积周期方能避免交叠误差。

将 $F(j,k)$、$G(j,k)$用增补零的方法扩充为以下的周期阵列：

$$F_e(j,k) = \begin{cases} F(j,k); & 0 \leqslant j \leqslant A-1 \quad 0 \leqslant k \leqslant B-1 \\ 0 \quad\quad ; & A \leqslant j \leqslant M-1 \quad B \leqslant k \leqslant N-1 \end{cases}$$

$$G_e(j,k) = \begin{cases} G(j,k); & 0 \leqslant j \leqslant C-1 \quad 0 \leqslant k \leqslant D-1 \\ 0 \quad\quad ; & C \leqslant j \leqslant M-1 \quad D \leqslant k \leqslant N-1 \end{cases}$$

在这样的前提下，$F_e(j,k)$、$G(j,k)$之间的卷积关系如下：

$$F_e(j,k) * G_e(j,k) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} F_e(m,n)G_e(j-m,k-n) \tag{2.81}$$

式中 $j=0,1,2,\cdots,M-1$；

$k=0,1,2,\cdots,N-1$。

相应的卷积定理为

$$F_e(j,k) * G_e(j,k) \Leftrightarrow F_e(u,v) \cdot G_e(u,v) \tag{2.82}$$

$$F_e(j,k) \cdot G_e(j,k) \Leftrightarrow F_e(u,v) * G_e(u,v) \tag{2.83}$$

从形式上看，连续图像函数与离散图像函数的卷积定理没有差异，但 $F_e(j,k)$、$G_e(j,k)$是扩充函数，离散图像函数卷积的计算过程中所用的变量是离散量。

二维连续图像函数 $F(j,k)$和 $G(j,k)$的相关用符号 $F(j,k)oG(j,k)$表示，并定义为

$$F(j,k)oG(j,k) = \int_{-\infty}^{+\infty}\int F(\alpha,\beta)G(j+\alpha+k+\beta)\mathrm{d}\alpha\mathrm{d}\beta \tag{2.84}$$

在离散情况下，有

$$F_e(j,k)oG_e(j,k) = \sum_{m=0}^{M}\sum_{n=0}^{N-1} F_e(m,n)G_e(j+m,k+n) \tag{2.85}$$

如同卷积那样，$F_e(j,k)$、$G_e(j,k)$是扩充了的二维离散图像函数，j方向以 M 为周期，k 方向以 N 为周期，选取

关函数在周期内不产生交叠现象。其中 $A \times B$ 和 $C \times D$ 分别为 $F_e(j,k)$、$G_e(j,k)$的阵元数。

二维连续和离散图像函数的相关定理表示成：

$$F(j,k)oG(j,k) \Leftrightarrow F(u,v)G^*(u,v) \tag{2.86}$$

$$F(j,k) \cdot G^*(j,k) \Leftrightarrow F(u,v)oG^*(u,v) \tag{2.87}$$

$$F_e(j,k)oG_e(j,k) \Leftrightarrow F_e(u,v)G_e{}^*(u,v) \tag{2.88}$$

$$F_e(j,k) \cdot G_e{}^*(j,k) \Leftrightarrow F_e(u,v)oG_e(u,v) \tag{2.89}$$

上角标“*”表示共轭。

2.2.2 快速傅立叶变换(FFT)

2.2.1 节已经证明，二维离散傅立叶变换可以由两步一维离散傅立叶变换来实现。基于这样的考虑，对 FFT 的讨论将以一维离散傅立叶变换为模型。

一维 DFT 变换对为

正变换

$$F(u) = \frac{1}{N}\sum_{j=0}^{N-1} F(j)\exp[-i2\pi uj/N] \tag{2.90}$$

反变换

$$F(j) = \frac{1}{N}\sum_{u=0}^{N-1} F(u)\exp[i2\pi uj/N] \tag{2.91}$$

很容易看出,求一维 DFT 时,对 u 的 N 个值中的每一个,求和操作都要做 N 次的复数乘法和($N-1$)次的复数加法(以下简称乘法/加法运算)。对 N 个 u 值,一维 DFT 运算将要进行 N^2 次乘法/加法运算。显然,当 N 很大时,计算是十分费时的,特别当计算机的性能不很理想的情况下更是如此。例如,在 $N>8000$,计算工具为 IBM7094 的环境下,耗时达到 40 分钟以上。FFT 算法旨在通过适当分解使计算次数大幅度降低,如 Cooley 与 Tukey 在 1965 年提出的一种 FFT 算法,其乘法/加法次数与 $N\log_2 N$ 成比例。相对于 DFT 而言,它的计算收益率十分可观(见表 2.1)。

表 2.1 FFT 相对于 DFT 的计算收益

N	N^2(DFT)	$N\log_2 N$(FFT)	$N^2/\log_2 N$
2	4	2	2.0
4	16	8	2.0
8	64	24	2.7
16	256	64	4.0
32	1024	160	6.4
64	4096	394	10.7
128	16384	896	18.3
256	65536	2048	32.0
512	262144	4608	56.9
1024	1048576	10240	102.4
2048	4194304	22528	186.2

1) FFT 算法

FFT 算法以 N 的组成状况可以分成 N 为 2 的整数幂的算法;N 为高复合数的算法;N 为素数的算法三种情况。这里介绍第一种算法。

令 $$W_N = \exp[-i2\pi/N] \tag{2.92}$$

由(2.90)式,一维 DFT 变为

$$F(u) = \frac{1}{N}\sum_{j=0}^{N-1} F(j)\exp[(-i2\pi/N)uj] = \frac{1}{N}\sum_{j=0}^{N-1} F(j)W_N^{uj} \tag{2.93}$$

u,j 分别为 $0,1,2,\cdots,N-1$。再令

$$N = 2^n \tag{2.94}$$

$n=0,1,2,\cdots$。在此基础上,将 $F(j)$ 分解成 $F(2j)$ 和 $F(2j+1)$ 对应的偶数和奇数两部分,j 的取值范围由原来的 $0\sim N-1$ 改为 $0\sim \frac{N}{2}-1$。为书写方便,用 C 代替 $\frac{N}{2}$,即 $N=2C$。

于是

$$F(u) = \frac{1}{2}\left\{\frac{1}{C}\sum_{j=0}^{C-1} F(2j)W_{2C}^{u(2j)} + \frac{1}{C}\sum_{j=0}^{C-1} F(2j+1)W_{2C}^{u(2j+1)}\right\}$$

$$= \frac{1}{2}\left\{\frac{1}{C}\sum_{j=0}^{C-1} F(2j)W_C^{uj} + \frac{1}{C}\sum_{j=0}^{C-1} F(2j+1)W_C^{uj}W_{2C}^{u}\right\} \tag{2.95}$$

再定义

$$F_O(u) = \frac{1}{C}\sum_{j=0}^{C-1} F(2j)W_C^{uj} \tag{2.96}$$

$$F_J(u) = \frac{1}{C}\sum_{j=0}^{C-1} F(2j+1)W_C^{uj} \tag{2.97}$$

式中　下标 O、J 分别表示偶数和奇数，$u=0,1,2,\cdots,C-1$。由此，(2.95)式可写成

$$F(u) = \frac{1}{2}\{F_O(u) + F_J(u)W_{2C}^{u}\} \tag{2.98}$$

又因为

$$W_{2C}^{C} = W_N^{N} = -1$$

$$W_{2C}^{u+C} = W_{2C}^{u} \cdot W_{2C}^{C} = -W_{2C}^{u}$$

故根据(2.96)～(2.98)式，可得

$$F(u+C) = \frac{1}{2}\{F_O(u) - F_J(u)W_{2C}^{u}\} \tag{2.99}$$

稍作分析可以发现，N 点变换可以变成把原式分成(2.98)、(2.99)式两部分来分别加以计算的方法予以实现。即由(2.96)、(2.97)式得到 $F_O(u)$、$F_J(u)$后代入(2.98)式，求出 $u=0,1,2,\cdots,(N-1)$时 $F(u)$。另一半则从(2.99)式求取。

2) FFT 的实现

先了解一下“逆序”的概念。令 j 为数组 $F(j)$的宗量值，j 的二进制形式为

$B_j = n_{b-1}n_{b-2}\cdots n_1 n_0$　其逆序形式为

$B'_j = n_0 n_1 \cdots n_{b-2}n_{b-1}$

数组 $F(j)$由 $F(B_j)$形式变成 $F(B'_j)$形式称为“逆序”。如已知 $N=2^3$，则 $F(j)$的“逆序”排列将如表 2.2 所示。

表 2.2　$N=8$ 时数组的“逆序”排列

B_j	原始数组	B'_j	“逆序”数组
000	$F(0)$	000	$F(0)$
001	$F(1)$	100	$F(4)$
010	$F(2)$	010	$F(2)$
011	$F(3)$	110	$F(6)$
100	$F(4)$	001	$F(1)$
101	$F(5)$	101	$F(5)$
110	$F(6)$	011	$F(3)$
111	$F(7)$	111	$F(7)$

有了“逆序”的概念，就可以着手讨论 FFT 的实现。我们仍以 $N=2^3$ 为例，计算时，先将原始数组按宗量值“逆序”顺序，重新排列(见表 2.2)，然后自左向右，每两个相邻元素为一组，共 4 组，具体是：$F(0)$、$F(4)$；$F(2)$、$F(6)$；$F(1)$、$F(5)$；$F(3)$、$F(7)$。在每一组中，第

个元素作为偶元素，第二个元素作为奇元素（注意对所有组都一样）。在此基础上，用(2.96)～(2.98)式完成四个“两点变换”。下一步是利用“两点变换”结果形成两个“四点变换”，最后是建立在“四点变换”基础上的一个“八点变换”。具体见图 2.3。

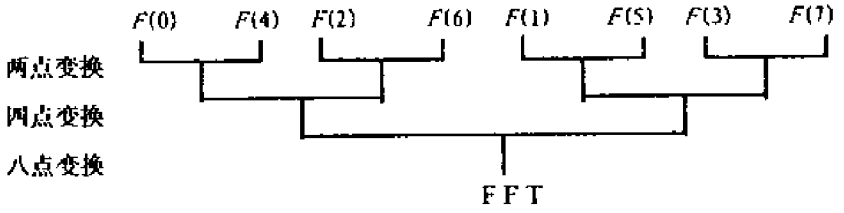


图 2.3 FFT 算法流程($N=8$)

以上 FFT 算法流程简称为逐次加倍法，因为“两点变换”由两个“一点变换”算出，“四点变换”由两个“两点变换”算出，“八点变换”由两个“四点变换”算出，依此类推。逐次加倍法求取 $N=2^n$ 的 FFT 运算的复数乘法与加法的次数分别为：

$$\text{乘法} \qquad N_m = \frac{1}{2}N\log_2 N = \frac{1}{2}N_n \tag{2.100}$$

$$\text{加法} \qquad N_a = N\log_2 N = N_n \tag{2.101}$$

与 DFT 相比，计算次数相对节省率为

$$R_a = |[(\frac{1}{2}N\log_2 N + N\log_2 N) - N^2]/N^2|$$

$$= |[\frac{3}{2}N\log_2 N - N^2]/N^2| = |1 \cdot 5\frac{n}{N} - 1| \tag{2.102}$$

图 2.4 给出了 $n=0,1,2,\cdots,8$ 时 R_a 的变化情况。

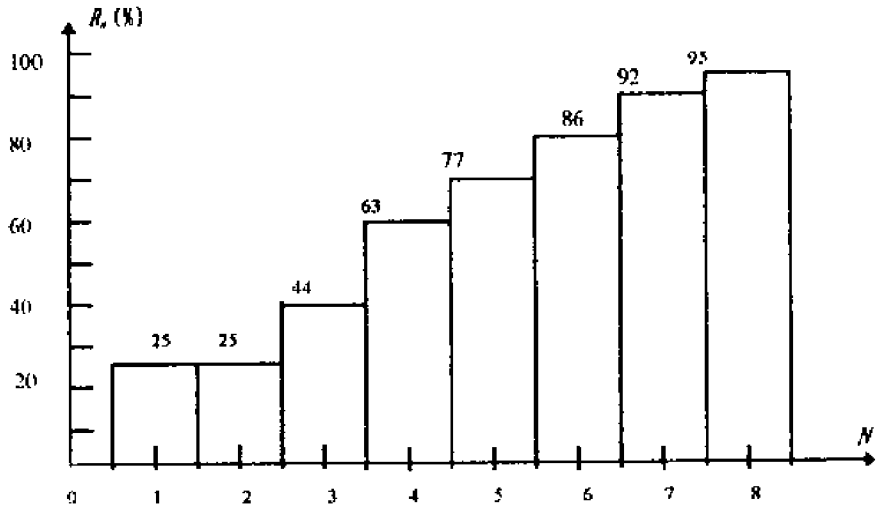


图 2.4 FFT 计算效率

2.3 离散余弦变换(DCT)

在傅立叶级数展开式中，如果函数对称于原点，则其级数中将只有余弦函数项。从这一现象受到启示，人们提出了另一种图像变换方法——离散余弦变换。

2.3.1 关于轴对称

有一离散图像序列 $f(j)=\{f(0),f(1),\cdots,f(N-1)\}$(图 2.5(a))。现将 $f(j)$以 $j=-\frac{1}{2}$ 为对称轴(相当于移动坐标原点到 $-\frac{1}{2}$ 处)向 $-j$ 方向折叠(图 2.5(b)),取得新的序列 $f_s(j)$,$f_s(j)$的长度为 $2N$,$f_s(j)$与 $f(j)$的关系为

$$f_s(j)=\begin{cases}f(j) & ;\ 0\leqslant j\leqslant N-1\\ f(-m-1) & ;\ -N\leqslant j\leqslant -1\end{cases} \tag{2.103}$$

这就是一维离散函数 $f(j)$ 关于 $j=-\frac{1}{2}$ 的偶对称序列。现在,j_s 轴的原点置于 $j=-\frac{1}{2}$ 处,方向与 j 轴相同,于是就存在 $j_s=j+\frac{1}{2}$ 的变换关系。

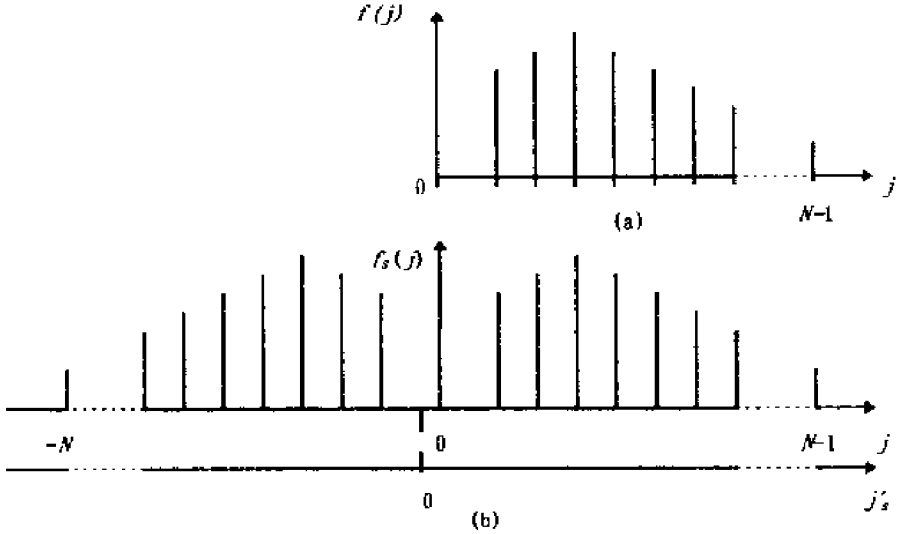


图 2.5 一维 DCT 偶对称示意图

类似于一维情况,对大小为 $2N\times 2N$ 的二维离散图像 $F(j,k)$,将其沿水平、垂直边界翻折,形成四个 $N\times N$ 的图像子块,(图 2.6(a)),它称之为偶对称,所对应的 DCT 变换称之为

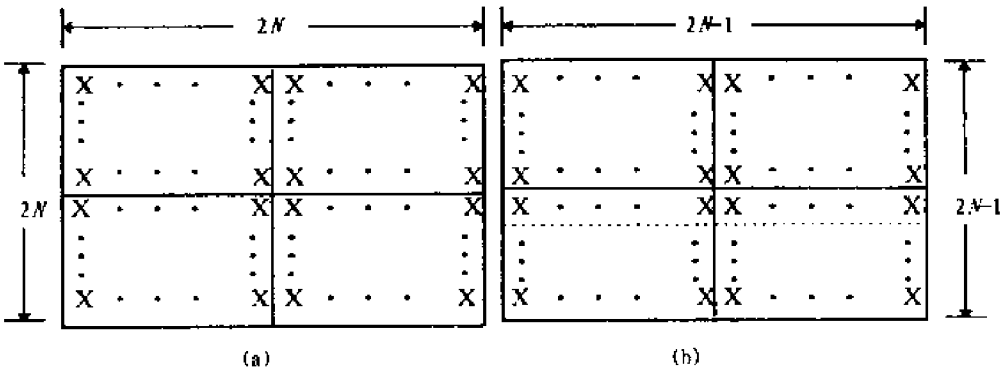


图 2.6 余弦变换对称示意图

(a)边沿折叠(偶对称); (b)像素折叠(奇对称)

偶对称余弦变换。如果图像翻折时重叠一个像素(图 2.6(b)),称之为奇对称,所进行的DCT 变换称为奇对称变换。

2.3.2 偶对称余弦变换

在偶对称变换场,有

$$F_s(j,k)=\begin{cases}F(j,k) & ;\quad j\geqslant 0\quad k\geqslant 0\\ F(-1-j,k) & ;\quad j<0\quad k<0\\ F(j,-1-k) & ;\quad j\geqslant 0\quad k<0\\ F(-1-j,-1-k) & ;\quad j<0\quad k<0\end{cases}\tag{2.104}$$

这里,$F_s(j,k)$对 $j=-\frac{1}{2},k=-\frac{1}{2}$ 的虚拟点对称。j,k 的取值范围为 $-N,-N+1,\cdots,N-2,N-1$。取对称点的傅立叶变换,得

$$\tilde{f}_s(u,v)=\frac{1}{2N}\sum_{j=-N}^{N-1}\sum_{k=-N}^{N-1}F_s(j,k)\cdot\exp\{-i2\pi[u(j+\tfrac{1}{2})+v(k+\tfrac{1}{2})]/2N\}\tag{2.105}$$

此时 u,v 分别为 $-N,-N+1,\cdots,N-2,N-1$。该式的成立是因为已知对 $N\times N$ 的离散图像阵列,其傅立叶变换为

$$\tilde{f}(u,v)=\frac{1}{N}\sum_{j=0}^{N-1}\sum_{k=0}^{N-1}F(j,k)\exp[-i2\pi(uj+vk)/N]\tag{2.106}$$

在图 2.6(a)中 $F_s(j,k)$的大小为 $2N\times 2N$,坐标原点移到了 $(j-\frac{1}{2},k-\frac{1}{2})$ 处,

$$j_s=j+\frac{1}{2},k_s=k+\frac{1}{2}。$$

于是 $$\tilde{f}_s(u,v)=\frac{1}{N_S}\sum_{j=-N}^{N-1}\sum_{k=-N}^{N-1}F_s(j,k)\cdot\exp[-i2\pi(uj_s+vk_s)/n]$$

将有关参数代入即可得到(2.105)式。

傅立叶变换具有共轭对称性,即

$$\tilde{f}_s(u,v)=\tilde{f}_s^*(-u,-v)$$

因此只需对非负的(u,v)计算 $\tilde{f}_s(u,v)$。因此,偶对称余弦变换当可用下式描述:

$$\tilde{f}_s(u,v)=\frac{1}{N}\sum_{j=0}^{N-1}\sum_{k=0}^{N-1}F(j,k)\cdot\exp\{-i2\pi[u(j+\tfrac{1}{2})+v(k+\tfrac{1}{2})]/N\}\tag{2.107}$$

再考虑到 $F_s(j,k)$是实对称实数,所以(2.107)式可简化为

$$\tilde{f}_s(u,v)=\frac{2}{N}\sum_{j=0}^{N-1}\sum_{k=0}^{N-1}F(j,k)\cdot\cos[\pi u(j+\tfrac{1}{2})/N]\cos[\pi v(k+\tfrac{1}{2})/N]\tag{2.108}$$

将其规范化

$$\tilde{f}(u,v)=\frac{2}{N}C(u)C(v)\sum_{j=0}^{N-1}\sum_{k=0}^{N-1}F(j,k)\cdot\cos[\pi u(j+\tfrac{1}{2})/N]\cos[\pi v(k+\tfrac{1}{2})/N]\tag{2.109}$$

相应的反变换定义为

$$F(j,k) = \frac{2}{N}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1} C(u)C(v)\bar{f}(u,v)\cdot\cos[\pi u(j+\frac{1}{2})/N]\cos[\pi v(k+\frac{1}{2})/N] \tag{2.110}$$

式中 $C(0) = 2^{-\frac{1}{2}}$；

$C(W)=1$，$W=1,2,\cdots,N-1$。

偶对称余弦变换的变换核是一种离散的 Chebyshev 多项式。

2.3.3 奇对称余弦变换

在奇对称余弦变换场，图像阵列可定义为(2.111)式：

$$F_S(j,k) = \begin{cases} F(j,k) & ; \quad j \geqslant 0 \quad k \geqslant 0 \\ F(-j,k) & ; \quad j < 0 \quad k > 0 \\ F(j,-k) & ; \quad j \geqslant 0 \quad k < 0 \\ F(-j,-k); & \quad j < 0 \quad k < 0 \end{cases} \tag{2.111}$$

相应的傅立叶表达式如下：

$$\tilde{f}_s(u,v) = \frac{1}{2N-1}\sum_{j=-N+1}^{N-1}\sum_{k=-N+1}^{N-1} F_S(j,k)\cdot\exp[-i2\pi(ju+kv)/(2N-1)] \tag{2.112}$$

式中 u,v 分别为 $-N+1,-N+2,\cdots,-1,0,1,\cdots,N-1$。基于傅立叶变换的共轭性，有

$$\tilde{f}_s(u,v) = \frac{1}{2N-1}\sum_{j=0}^{N-1}\sum_{k=0}^{N-1} F'(j,k)\cdot\exp[-i2\pi(ju+kv)/(2N-1)] \tag{2.113}$$

同前，$F'(j,k)$是实对称函数，故

$$f_s(u,v) = \frac{4}{2N-1}\sum_{j=0}^{N-1}\sum_{k=0}^{N-1} F'(j,k)\cdot\cos[2\pi ju/(2N-1)]\cos[2\pi kv/(2N-1)] \tag{2.114}$$

式中 F' 按下式加权求得：

$$F'(j,k) = \begin{cases} \frac{1}{4}F(j,k); & \quad j=0 \quad k=0 \\ \frac{1}{2}F(j,k); & \quad j=0 \quad k\neq 0 \\ \frac{1}{2}F(j,k); & \quad j\neq 0 \quad k=0 \\ F(j,k); & \quad \text{其他} \end{cases} \tag{2.115}$$

作规格化处理，结果是：

$u=v=0$，则

$$\tilde{f}(u,v) = \frac{1}{N}\sum_{j=0}^{N-1}\sum_{k=0}^{N-1} F'(j,k) \tag{2.116}$$

$u=v\neq 0$，则

$$\tilde{f}(u,v) = \frac{2}{N}\sum_{j=0}^{N-1}\sum_{k=0}^{N-1} F'(j,k)\cdot\cos[2\pi ju/(2N-1)]\cos[2\pi kv/(2N-1)] \tag{2.117}$$

原图像阵列可由下式重建：

$$F(j,k) = \begin{cases} 4F'(j,k); & j = 0 \quad k = 0 \\ 2F'(j,k); & j = 0 \quad k \neq 0 \\ 2F'(j,k); & j \neq 0 \quad k = 0 \\ F'(j,k); & \text{其他} \end{cases} \tag{2.118}$$

奇对称余弦变换的反变换定义为

$$F'(j,k) = \frac{1}{N}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1}\tilde{f}(u,v), \quad j = k = 0 \tag{2.119}$$

$$F'(j,k) = \frac{1}{N}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1}\tilde{f}(u,v)\cdot\cos[2\pi ju/(2N-1)]\cos[2\pi kv/(2N-1)]$$

$$j \neq 0 \quad k \neq 0 \tag{2.120}$$

2.4 沃尔什—哈达玛变换(WHT)

严格地说,沃尔什(Walsh)变换与哈达玛(Hadamard)变换是两种变换。而事实上两者十分相似,如变换对形式一致;变换方阵均具备正交特性,矩阵元素均由"-1"和"+1"组成,在阶 $N=2^n$ 时,两个变换矩阵除行、列顺序有些不同外,彼此没有任何本质上的差异(见表2.3,表2.4)等。因此,大家习惯把沃尔什变换与哈达玛变换统称为"沃尔什-哈达玛变换"。不过,由于哈达玛变换矩阵拥有简单递推的关系(即高阶矩阵可由两个低阶矩阵之直积求得),所以在实际中人们更乐意使用它。下面,我们将以哈达玛变换为线索,进行此类变换的讨论。

2.4.1 哈达玛变换对

当 $N=2^n$(n 为正整数),u,j 分别为 $0,1,2,\cdots,N-1$ 时,一维离散函数

$$f(j) = \{f(0),f(1),\cdots,f(N-1)\} \tag{2.121}$$

的哈达玛变换对为

$$H(u) = \sum_{j=0}^{N-1} f(j)g(j,u) \tag{2.122}$$

$$f(j) = \sum_{u=0}^{N-1} H(u)h(j,u) \tag{2.123}$$

由于

$$g(j,u) = \frac{1}{N}(-1)^p$$

$$h(j,u) = (-1)^p$$

$$p = \sum_{i=0}^{n-1} b_i(j)b_i(u) \tag{2.124}$$

所以有

$$H(u) = \frac{1}{N}\sum_{j=0}^{N-1} f(j)(-1)^p \tag{2.125}$$

$$f(j) = \sum_{u=0}^{N-1} H(u)(-1)^p \tag{2.126}$$

式中 $b_k(z)$ 是 z 的二进制形式表示的相应的位(bit),如 $z=5$,则 $b_0=1, b_1=0, b_2=1$。

类似地，二维离散函数 $F(j,k)$的哈达玛变换对通式及级数式分别为

通式

$$H(u,v) = \sum_{j=0}^{N-1}\sum_{k=0}^{N-1} F(j,k)g(j,k,u,v) \tag{2.127}$$

$$F(j,k) = \sum_{u=0}^{N-1}\sum_{v=0}^{N-1} H(u,v)h(j,k,u,v) \tag{2.128}$$

级数式

$$H(u,v) = \frac{1}{N}\sum_{j=0}^{N-1}\sum_{k=0}^{N-1} F(j,k)(-1)^{Q} \tag{2.129}$$

$$F(j,k) = \frac{1}{N}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1} H(u,v)(-1)^{Q} \tag{2.130}$$

式中

$$Q = \sum_{i=0}^{n-1}[b_i(j)b_i(u) + b_i(k)b_i(v)] \tag{2.131}$$

注意，在由(2.124)式、(2.131)式计算变换核指数 P、Q 时，求和是按模 2 进行的。$N=8(n=3)$时，一维哈达玛变换核对应的数值矩阵(简单起见，略去了常数项 $\frac{1}{N}$)列于表 2.3。作为对照，相应的沃尔什矩阵列于表 2.4。作一比较，可以发现两表的表元素从总体上来说是相同的，但同类表列所处的位置不尽一致。

哈达玛变换核是可分离和对称的。也就是说，有

表 2.3 $N=8$ 时哈达玛变换核的值

u \ j	0	1	2	3	4	5	6	7	变号次数
0	1	1	1	1	1	1	1	1	0
1	1	-1	1	-1	1	-1	1	-1	7
2	1	1	-1	-1	1	1	-1	-1	3
3	1	-1	-1	1	1	-1	-1	1	4
4	1	1	1	1	-1	-1	-1	-1	1
5	1	-1	1	-1	-1	1	-1	1	6
6	1	1	-1	-1	-1	-1	1	1	2
7	1	-1	-1	1	-1	1	1	-1	5

表 2.4 $N=8$ 时沃尔什变换核的值

u \ j	0	1	2	3	4	5	6	7	变号次数
0	1	1	1	1	1	1	1	1	0
1	1	1	1	1	-1	-1	-1	-1	1
2	1	1	-1	-1	1	1	-1	-1	3
3	1	1	-1	-1	-1	-1	1	1	2
4	1	-1	1	-1	1	-1	1	-1	7
5	1	-1	1	-1	-1	1	-1	1	6
6	1	-1	-1	1	1	-1	-1	1	4
7	1	-1	-1	1	-1	1	1	-1	5

$$
\begin{aligned}
g(j,k,u,v) &= h(j,k,u,v) \\
&= g_1(j,u)g_1(k,v) \\
&= h_1(j,u)h_1(k,v) \\
g(j,k,u,v) &= [N^{-\frac{1}{2}}(-1)^{S_1}][N^{-\frac{1}{2}}(-1)^{S_2}]
\end{aligned} \tag{2.132}
$$

式中

$$
S_1 = \sum_{i=0}^{n-1} b_i(j)b_i(u)
$$

$$
S_2 = \sum_{i=0}^{n-1} b_i(k)b_i(v)
$$

这样,就能够逐次应用任何计算一维哈达玛变换的算法求二维变换对。

2.4.2 哈达玛变换矩阵

令 $\boldsymbol{H}_N$ 代表 N 阶矩阵(矩阵元素个数 $= N \times N$),最低阶哈达玛矩阵为

$$
\boldsymbol{H}_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{2.133}
$$

再令 $\boldsymbol{H}_{2N}$表示 $2N$ 阶矩阵(矩阵元素个数为 $2N \times 2N$),$2N$ 阶哈达玛矩阵 $\boldsymbol{H}_{2N}$与 N 阶哈达玛矩阵 $\boldsymbol{H}_N$ 之间的递推关系为

$$
\boldsymbol{H}_{2N} = \begin{bmatrix} \boldsymbol{H}_N & \boldsymbol{H}_N \\ \boldsymbol{H}_N & \boldsymbol{H}_N \end{bmatrix} \tag{2.134}
$$

为了方便变换表达式的书写,我们利用阶的方根将对应的哈达玛矩阵规格化,这时,在 $N \times N$ 的情况下,

$$
\boldsymbol{H}_2 = (2)^{-\frac{1}{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{2.135}
$$

$$
\boldsymbol{H}_4 = (4)^{-\frac{1}{2}} \left[\begin{array}{cc:cc} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ \hdashline 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{array}\right] \tag{2.136}
$$

$$
\boldsymbol{H}_8 = (8)^{-\frac{1}{2}} \left[\begin{array}{cccc:cccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ \hdashline 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{array}\right] \tag{2.137}
$$

(2.136)、(2.137)式中应用了递推性质。即

$$
\boldsymbol{H}_4 = \boldsymbol{H}_2 \otimes \boldsymbol{H}_2 = \begin{bmatrix} \boldsymbol{H}_2 & \boldsymbol{H}_2 \\ \boldsymbol{H}_2 & -\boldsymbol{H}_2 \end{bmatrix} \tag{2.138}
$$

$$\boldsymbol{H}_8 = \boldsymbol{H}_2 \otimes \boldsymbol{H}_2 \otimes \boldsymbol{H}_2 = \begin{bmatrix} \boldsymbol{H}_4 & \boldsymbol{H}_4 \\ \boldsymbol{H}_4 & -\boldsymbol{H}_4 \end{bmatrix} \tag{2.139}$$

在构造 N 阶哈达玛变换矩阵时，除系数项 $(N)^{-\frac{1}{2}}$ 外，矩阵元素 $h_{ij}(i,j$ 分别为 $0,1,\cdots,N-1)$还可由下列关系式确定：

$$h_{ij}(-1)^q, \quad q = \sum_{k=0}^{n-1} i_k j_k \tag{2.140}$$

式中 i,j 为二进制码表示的行、列坐标。即

$$i = i_{n-1}i_{n-2}\cdots i_1 i_0$$

$$j = j_{n-1}j_{n-2}\cdots j_1 j_0$$

q 的计算原则与(2.124)式的 P 和(2.131)式的 Q 一样。如 $N=4(n=2)$时，

$$q = \sum_{k=0}^{1} i_k j_k$$

$$\boldsymbol{H}_4 = \begin{bmatrix} h_{00} & h_{01} & h_{02} & h_{03} \\ h_{10} & h_{11} & h_{12} & h_{13} \\ h_{20} & h_{21} & h_{22} & h_{23} \\ h_{30} & h_{31} & h_{32} & h_{33} \end{bmatrix}$$

对 h_{12}，$i=01B$，$j=10B$，得

$$q = (1\times 0)\oplus(0\times 1) = 0$$

$$h_{12} = (-1)^0 = 1$$

对 h_{32}，$i=11B$，$j=10B$，有

$$q = (1\times 0)\oplus(1\times 1) = 1$$

$$h_{32} = (-1)^1 = -1$$

如此等等，其结果和采用 $\boldsymbol{H}_{2N}$递推关系一致。

2.4.3 向量表示的哈达玛变换

当变换核 $g(j,k,u,v)$和 $h(j,k,u,v)$存在下列形式时：

$$g(j,k,u,v) = g_1(j,u)g_1(k,v) \tag{2.141}$$

$$h(j,k,u,v) = h_1(j,u)h_1(k,v) \tag{2.142}$$

$g(j,k,u,v)$、$h(j,k,u,v)$被认为是可分离与对称的。对应的以向量表示的哈达玛正变换和反变换分别为

$$\boldsymbol{f}_H = \boldsymbol{AFA} \tag{2.143}$$

$$\boldsymbol{F} = \boldsymbol{Bf}_H\boldsymbol{B} \tag{2.144}$$

式中 $\boldsymbol{f}_H$ 为哈达玛变换域；

$\boldsymbol{F}$ 为原图像矩阵；

$\boldsymbol{A}$ 为正变换核；

$\boldsymbol{B}$ 为反变换核。

2.4.2 节已述，哈达玛变换核是具有递推关系的 $\boldsymbol{H}_N$ 矩阵。又知，哈达玛变换的正、反变

换核具有同样的形式(见(2.129)、(2.130)式)。所以,(2.143)、(2.144)式可改写为

$$f_H = H_N F H_N \tag{2.145}$$

$$F = H_N f_H H_N \tag{2.146}$$

2.4.4 定序哈达玛变换

在哈达玛矩阵中,沿某一列符号改变的次数通常称为这个列的列率。

表 2.3、表 2.4 中的列率与 u 变量间没有明确的升序关系。如果使哈达玛变换矩阵中的列率随 u 的增大而线性增加,这种哈达玛变换命名为定序哈达玛变换。

一维定序哈达玛变换对为

$$H(u) = \frac{1}{N}\sum_{j=0}^{N-1} f(j)(-1)^M \tag{2.147}$$

$$f(j) = \sum_{u=0}^{N-1} H(u)(-1)^M \tag{2.148}$$

$$M = \sum_{i=0}^{n-1} b_i(j)p_i(u) \tag{2.149}$$

除常数项 $\frac{1}{N}$ 外,定序哈达玛变换的正变换核与反变换核也是相同的,均为 $(-1)^M$,M 的求和计算同样遵循模 2 加原则。其中

$$p_0(u) = b_{n-1}(u)$$

$$p_1(u) = b_{n-1}(u) \oplus b_{n-2}(u)$$

$$p_2(u) = b_{n-2}(u) \oplus b_{n-3}(u)$$

$$\cdots\cdots$$

$$p_{n-1}(u) = b_1(u) \oplus b_0(u)$$

表 2.5 给出了 $n=3$ 时定序哈达玛变换核矩阵值(忽略了 $1/N$)及列率。表 2.6 是表 2.5 中 $j=6, n=3$ 时定序哈达玛变换核的计算过程,表中的 $b_i(j)$, $b_i(u)$ 为变量 j 和 u 的二进制形式,$p_i(u)$ 是定序后 u 的二进制形式。

表 2.5 定序哈达玛变换核的值及列率

u \ j	0	1	2	3	4	5	6	7	变号次数
0	1	1	1	1	1	1	1	1	0
1	1	1	1	1	-1	-1	-1	-1	1
2	1	1	-1	-1	-1	-1	1	1	2
3	1	1	-1	-1	1	1	-1	-1	3
4	1	-1	-1	1	1	-1	-1	1	4
5	1	-1	-1	1	-1	1	1	-1	5
6	1	-1	1	-1	-1	1	-1	1	6
7	1	-1	1	-1	1	-1	1	-1	7

表 2.6 定序哈达玛变换核的计算过程

u	$b_i(j)$	$b_i(u)$	$p_i(u)$	M	$(-1)^M$
0	110	000	000	0	1
1	110	001	100	1	-1
2	110	010	110	0	1
3	110	011	010	1	-1
4	110	100	011	1	-1
5	110	101	111	0	1
6	110	110	101	1	-1
7	110	111	001	0	1

二维离散定序哈达玛变换与二维离散非定序哈达玛变换具有同样的特性，即正、反变换核表示式一样，变换核可分离，对称且正交。因此，二维定序哈达玛变换对定义为

$$H(u,v) = \frac{1}{N}\sum_{j=0}^{N-1}\sum_{k=0}^{N-1}F(j,k)(-1)^T \tag{2.150}$$

$$F(j,k) = \frac{1}{N}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1}H(u,v)^T \tag{2.151}$$

$$T = \sum_{i=0}^{n-1}[b_i(j)p_i(u) + b_i(k)p_i(v)] \tag{2.152}$$

变换核

$$g(j,k,u,v) = [N^{-\frac{1}{2}}(-1)^{T_1}][N^{-\frac{1}{2}}(-1)^{T_2}] \tag{2.153}$$

式中

$$T_1 = \sum_{i=0}^{n-1}b_i(j)p_i(u)$$

$$T_2 = \sum_{i=0}^{n-1}b_i(k)p_i(v)$$

同样可用两步一维定序变换来实现二维定序变换。

3 图像增强

●单点增强　●多方式滤波　●边缘锐化
●彩色技术　●改善视觉　●利于处理

图像增强的主要目的有两个：一是改善图像的视觉效果，提高图像成分的清晰度；二是使图像变得更利于计算机的处理，如锐化处理可突出图像边缘轮廓线，编程控制计算机进行跟踪，便可作各种特征分析。

图像增强理论(方法)目前尚无统一的权威性的定义，因为还没有衡量图像增强质量的通用标准。从增强处理的作用域出发，图像增强可分为空间域法和频率法两大类。空间域处理时直接面对图像灰度级作运算。频率域处理是在图像的某种变换域内，对图像的变换系数值进行运算，即作某种修正，然后通过逆变换获得增强图像，这是一种间接增强的方法。图 3.1 列出了图像增强方法概况。

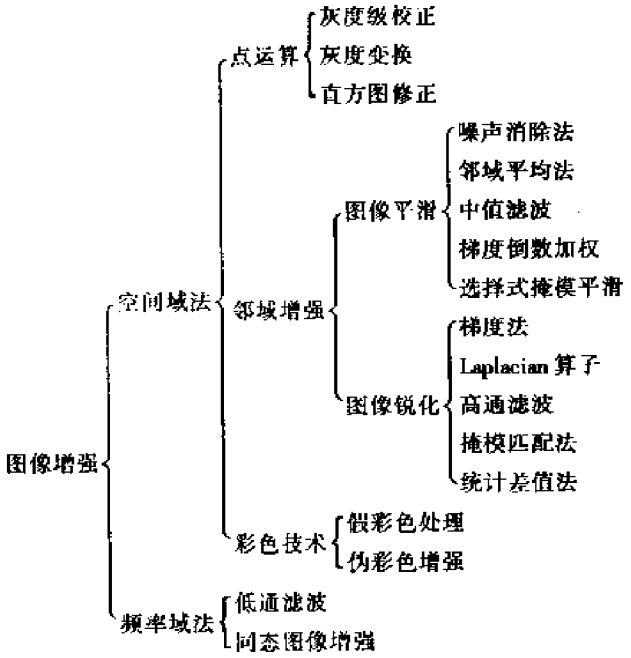


图 3.1　图像增强方法概况

3.1　空间域单点增强

这是在笛卡尔坐标平面内按帧扫描顺序以单个像素为对象所进行的简单、实用的图像增强技术。

3.1.1 灰度级校正

在成像过程中，扫描系统、光电转换系统中的诸多因素（如光照强弱，感光部件灵敏度，光学系统不均匀性，元器件电特性不稳定等）均可造成图像不均匀，即同样像源（如景物、照片、负片等）由于上述原因造成某些部分亮，某些部分暗。灰度级校正就是在图像采集系统级对图像像素作修正，使整幅图像成像均匀。

令理想输入系统输出的图像为 $F(j,k)$，实际获得的降质图像为 $G(j,k)$，有

$$G(j,k) = E(j,k)F(j,k) \tag{3.1}$$

$E(j,k)$为降质函数或观测系统的灰度失真系数。显然只要知道了 $E(j,k)$，就可求出不失真图像 $F(j,k)$。

如何确定降质函数呢？方法之一是用一个标准来标定系统的失真系数。即已知一幅图像的灰度级为均匀常数 C，而系统实际输出为 $G_C(j,k)$，有

$$G_C(j,k) = E(j,k)C$$

从而可得降质函数

$$E(j,k) = G_C(j,k)C^{-1} \tag{3.2}$$

将(3.2)式代入(3.1)式，就是由降质图像 $G(j,k)$经校正后所恢复的原始图像 $F(j,k)$。

$$F(j,k) = C\frac{G(j,k)}{G_C(j,k)} \tag{3.3}$$

请注意，经灰度级校正后的图像为连续图像，由于其乘了一个系数 $C/G_C(j,k)$，所以校正后的 $F(j,k)$有时可能出现“溢出”现象，即灰度级值可能超过某些记录器件或显示器的灰度输入范围，这时，需再作适当修正。对修正后的图像再进行量化。

3.1.2 灰度变换

灰度变换可使图像动态范围加大，图像对比度扩展，图像清晰，特征明显，是图像增强的重要手段。

1）全域线性变换

在曝光不足或过度的情况下，图像灰度可能会局限在一个很小的范围内。这时在显示器上看到的将是一个模糊不清、似乎没有灰度层次的图像。用一个线性单值函数，对帧内的每一个像素作线性扩展，将有效地改善图像视觉效果。

令原图像 $F(j,k)$的灰度范围为$[a,b]$，线性变换后图像 $F'(j,k)$的范围为$[a',b']$（图 3.2）。$F'(j,k)$与 $F(j,k)$之间存在以下关系：

$$F'(j,k) = a' + \frac{b'-a'}{b-a}(F-a) \tag{3.4}$$

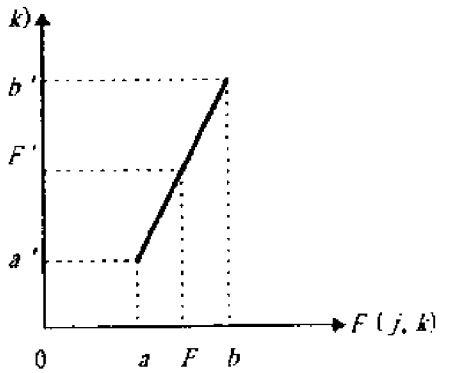


图 3.2 全域线性变换示意图

$[a,b]$可从直方图上获得，由于$|a'-b'|$总是大于$|a-b|$，所以对离散图像来说，尽管变换前后像素个数不变，但不同像素之间的灰度差变大，对比度加大，图像观看质量必然优于变换前。对于连续图像，如果背景与目标物的灰度

之差很小，在$[a,b]$区间内量化时可能进入同一灰度级内而不能分辨。如果线性变换时使

$$|a'-b'|=G|a-b|\quad,G\geqslant 2 \tag{3.5}$$

则量化时$F(j,k)$在$[a',b']$区间内就可取$2n$个以上不同的灰度值(n是$F(j,k)$在$[a,b]$区间内所取的灰度级个数)，人的眼睛原本不能检测的目标用增强手段就可突现出来。

另一种情况，图像中大部分像素的灰度级在$[a,b]$范围内，少部分像素分布在小于a和大于b的区间内。此时可用(3.6)式作变换(参见图3.3)：

$$F'(j,k)=\begin{cases}a' & ; \quad F(j,k)<a\\ a'+\dfrac{b'-a'}{b-a}(F-a); & a\leqslant F(j,k)<b\\ b' & ; \quad F(j,k)\geqslant b\end{cases} \tag{3.6}$$

应该清醒认识到，这种两端"截取式"的变换使小于灰度级a和大于等于灰度级b的像素强行压缩为a'和b'，将会造成一小部分信息丢失。不过有时为了某种应用，作这种"牺牲"是值得的，如利用遥感资料分析降水时，在预处理中去掉非气象信息，既可减少运算量，又可提高分析精度。

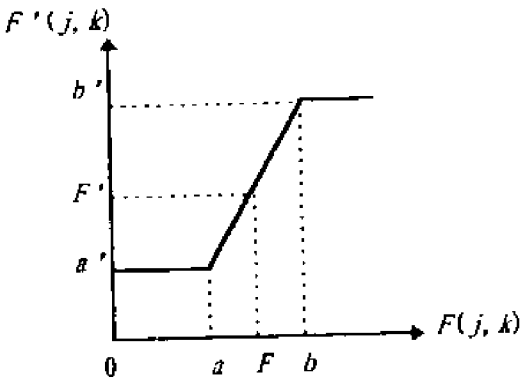


图3.3 截取式线性变示意图

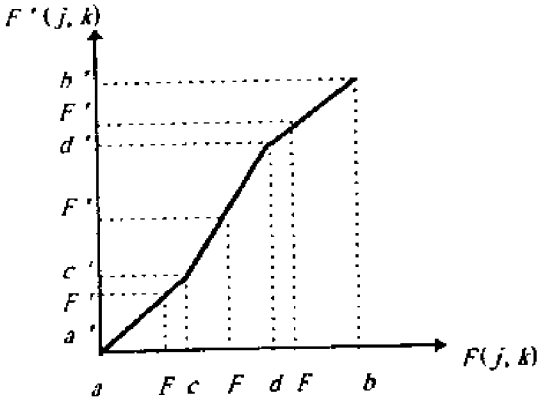


图3.4 分段线性变换示意图

2) 分段线性变换

将图像灰度区间分成两段乃至多段分别作线性变换称之为分段线性变换。图3.4是分三段作线性变换的示意图。分段线性变换的优点是可以根据用户的需要，拉伸特征物体的灰度细节，相对抑制不感兴趣的灰度级。图3.4中的a,a',b,b',c,c',d,d'等变换区间边界能通过键盘随时作交互式输入，因此，分段线性变换是非常灵活的。

分段线性变换遵循以下原则：

$$F'(j,k)=\begin{cases}\dfrac{c'-a'}{c-a}(F-a)+a'; & a\leqslant F(j,k)<c\\ \dfrac{d'-c'}{d-c}(F-c)+c'; & c\leqslant F(j,k)<d\\ \dfrac{b'-d'}{b-d}(F-d)+d'; & d\leqslant F(j,k)<b\end{cases} \tag{3.7}$$

如果$|a'-c'|>|a-c|$，$|c'-d'|<|c-d|$，$|d'-b'|=|d-b|$，则是扩展第一区间$[a,c]$，压缩第二区间$[c,d]$，维持第三区间$[d,b]$，得到低灰度级被增强的图像。

如果$|d'-b'|>|d-b|$，则可得高灰度级被增强的图像。

3.1.3 直方图修正

直方图是用来表达一帧图像灰度级分布情况的统计图表。直方图的横坐标是灰度，一般用 r 表示。纵坐标视图像类型而定，对连续图像，它是灰度值出现的概率密度 $P_r(r)$，对数字图像，它是灰度值为 r_i 的像素个数或出现这个灰度值的概率 $P_r(r_i)$。并知

$$P_r(r) = \lim_{\Delta r \to 0} \frac{A(r + \Delta r) - A(r)}{\Delta r A} \tag{3.8}$$

$$P_r(r_i) = \frac{\text{灰度为 } r_i \text{ 的像素个数}}{\text{一帧图像像素总数}} \tag{3.9}$$

$$\int_{r_{\min}}^{r_{\max}} P_r(r)\mathrm{d}r = 1 \tag{3.10}$$

$$\sum_{i=0}^{k-1} P_r(r_i) = 1 \tag{3.11}$$

式中 A 为图像面积；k 为一帧图像对应的灰度级数。

为讨论方便起见，我们明确在下面的讨论中，以 r 和 S 分别表示归一化了的原图像灰度和经直方图修正后的图像灰度。即

$$0 \leqslant r, \quad S \leqslant 1 \tag{3.12}$$

在[0,1]区间内的任一个 r 值，都可产生一个 S 值，且

$$S = T(r) \tag{3.13}$$

$T(r)$为变换函数。$T(r)$满足下列条件：

(1) 在 $0 \leqslant r \leqslant 1$ 区间内是单值单调增加函数；

(2) 在 $0 \leqslant r \leqslant 1$ 区间内，有 $0 \leqslant T(r) \leqslant 1$。

条件(1)保证灰度级从黑到白的次序，条件(2)确保映射后的像素灰度在允许的范围内。

反变换关系为

$$r = T^{-1}(S) \tag{3.14}$$

$T^{-1}(S)$对 S 同样满足上述条件。

由概率论知，若 $P(r)$和变换函数 $S = T(r)$已知，$T^{-1}(S)$是单调增长函数，则变换后的图像灰度级的概率密度函数 $P_s(S)$如下式所示：

$$P_s(S) = \left[P_r(r)\frac{\mathrm{d}r}{\mathrm{d}S}\right]_{r = T^{-1}(S)} \tag{3.15}$$

图像直方图反映了不同灰度值的像素的面积(连续图像)或像素个数(离散图像)在帧中的比例，表达了图像的某些信息。图 3.5(a)中，大部分像素集中在低灰度级区域，图像呈现暗的特性，而图 3.5(b)中的图像则相反，大部分像素集中在高灰度级，图像呈现亮的特性。直方图修正从本质上来说就是从 $T(r)$入手，调整图像灰度的概率密度函数，从而改变图像的特性。

1) 直方图均衡化

对于连续图像，变换函数 $T(r)$与原图像概率密度函数 $P_r(r)$之间的关系为

$$S = T(r) = \int_0^r P_r(r)\mathrm{d}r \tag{3.16}$$

因为由前面给定的归一化条件知

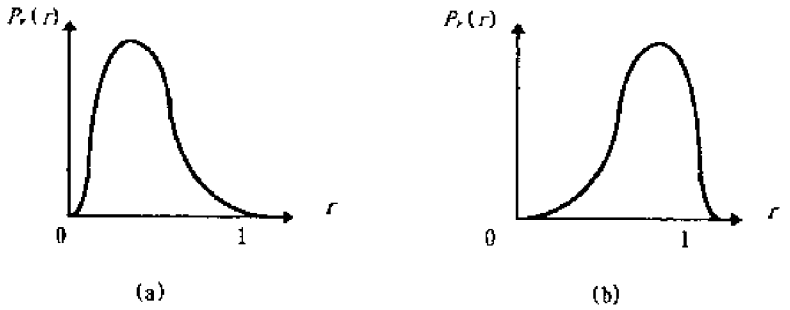


图 3.5　灰度概率密度函数

(a)"暗"特性图像；　(b)"亮"特性图像

$$P_s(s) = 1$$

于是根据(3.15)式有

$$\frac{dr}{dS} = \frac{1}{P_r(r)} \tag{3.17}$$

即　　$dS = P_r(r)dr$

将(3.17)式代入(3.15)式，变成

$$P_s(S) = \left[P_r(r)\frac{1}{P_r(r)}\right]_r = T^{-1}(S) = 1 \tag{3.18}$$

(3.18)式表明，在 S 的定义域内，$P_s(S)$是均匀密度。

对于离散图像，第 i 个灰度级 r_i 出现的频数用 n_i 表示，该灰度级像素对应的概率值 $P_r(r_i)$为

$$P_r(r_i) = \frac{n_i}{n} \tag{3.19}$$

n 是帧内像素总数，r_i 满足归一化条件。至此，可写出离散图像的变换函数表达式：

$$S_i = T(r_i) = \sum_{i=0}^{k-1} P_r(r_i) = \sum_{i=0}^{k-1} \frac{n_i}{n} \tag{3.20}$$

式中，k 为灰度级数。相应的反变换为

$$r_i = T^{-1}(S_i) \tag{3.21}$$

举一个例子。令图像大小为 32×32，8 个灰度级。各灰度级对应的像素个数及概率 $P_r(r_i)$列于表 3.1。根据(3.20)式，可以得到一组变换函数：

$$S_0 = P_r(r_0)$$
$$S_1 = P_r(r_0) + P_r(r_1)$$
$$S_2 = P_r(r_0) + P_r(r_1) + P_r(r_2)$$
$$S_3 = P_r(r_0) + P_r(r_1) + P_r(r_2) + P_r(r_3)$$
$$\cdots\cdots$$
$$S_{k-1} = P_r(r_0) + P_r(r_1) + P_r(r_2) + \cdots + P_r(r_{k-1})$$

本例计算结果是：

$S_0 = 0.19$　　$S_1 = 0.44$　　$S_2 = 0.65$

$S_3 = 0.81$　　$S_4 = 0.89$　　$S_5 = 0.95$

$S_6 = 0.98$　　$S_7 = 1.00$

因为原图像灰度限定为8层，所以必须把上述 S_i 值以 1/7 为量化单位进行舍入计算，得到

$$S'_0 = 1/7 \qquad S'_1 = 3/7 \qquad S'_2 = 5/7$$
$$S'_3 = 6/7 \qquad S'_4 = 6/7 \qquad S'_5 = 1$$
$$S'_6 = 1 \qquad S'_7 = 1$$

将相同值的 S'_i 归并起来，即为直方图均衡化修正后的灰度级变换函数，并仍以 S_i 表示。结果是

$$S_0 = 1/7 \qquad S_1 = 3/7 \qquad S_2 = 5/7$$
$$S_3 = 6/7 \qquad S_4 = 1$$

表 3.1　各灰度级对应的概率分布

灰度级	n_i	n_i/n
$r_0 = 0$	198	0.19
$r_1 = 1/7$	256	0.25
$r_2 = 2/7$	212	0.21
$r_3 = 3/7$	164	0.16
$r_4 = 4/7$	82	0.08
$r_5 = 5/7$	61	0.06
$r_6 = 6/7$	31	0.03
$r_7 = 1$	20	0.02

表 3.2　修正后各灰度级对应的概率分布

r_i	S'_i	S_i	n_{S_i}	$P_r(r_i)$
r_0	$S'_0 = 1/7$	S_0	198	0.19
r_1	$S'_1 = 3/7$	S_1	256	0.25
r_2	$S'_2 = 5/7$	S_2	212	0.21
r_3	$S'_3 = 6/7$	S_3	246	0.24
r_4	$S'_4 = 6/7$			
r_5	$S'_5 = 1$			
r_6	$S'_6 = 1$	S_4	112	0.11
r_7	$S'_7 = 1$			

上面的讨论表明，经直方图均衡化修正后，灰度级由原来的8个变为5个，各 S_i 对应的像素数及概率值均已发生了变化。具体见表 3.2.

图 3.6(a)、3.6(b)是本例修正前、后的直方图。不难看出，直方图均衡化修正后，图像直方图趋于平坦化，且灰度间隔被拉大，从而有利于图像的分析和识别。

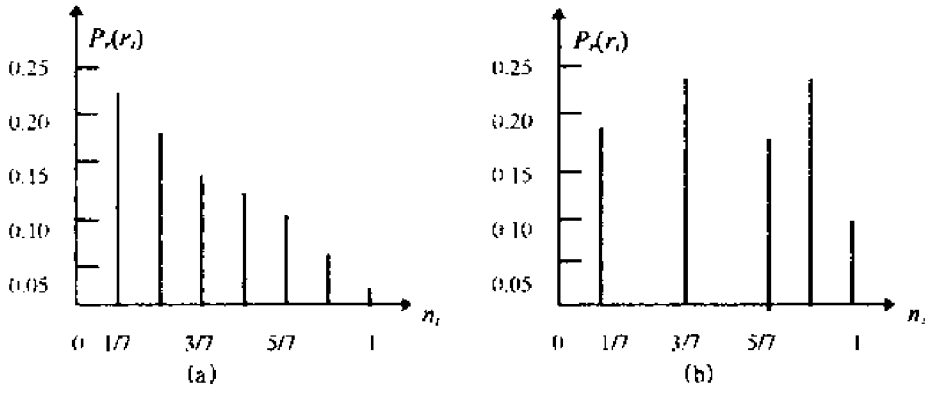


图 3.6　直方图均衡化示例

(a)原始直方图；　(b)均衡化后的直方图

从理论上说，直方图均衡化就是通过变换函数将原图的直方图调整为平坦的直方图，然

后用此均衡直方图校正图像。而实际上直方图均衡化修正后的图像直方图并不是十分均衡的，这是因为在操作过程中原直方图上频数较小的某些灰度级要并入一个或几个灰度级中。

2) 直方图规定化

直方图均衡化校正不具备交互作用特性。直方图规定化校正在运用均衡化原理的基础上，向人们提供了根据给定直方图作图像增强的手段。

令 $P_r(r)$和 $P_Z(Z)$分别为原始图像和期望图像的灰度概率密度函数。如果对原始图像和期望图像均作直方图均衡化处理，应有

$$S = T(r) = \int_0^r P_r(r)\mathrm{d}r \tag{3.22}$$

$$V = G(Z) = \int_0^Z P_Z(Z)\mathrm{d}Z \tag{3.23}$$

$$Z = G^{-1}(V) \tag{3.24}$$

由于都是进行均衡化处理，所以处理后的原图像概率密度数 $P_s(S)$及理想图像概率密度函数 $P_V(V)$是相等的。于是，我们可以用变换后的原始图像灰度级 S 代替(3.24)式中的 V。即

$$Z = G^{-1}(S) \tag{3.25}$$

这里的灰度级 Z 便是所希望的图像的灰度级。

此外，利用(3.22)式与(3.24)式还可得到组合变换函数

$$Z = G^{-1}[T(r)] \tag{3.26}$$

利用此式可从原始图像得到希望的图像灰度级。

对连续图像，重要的是给出逆变换解析式。对离散图像而言，类似于(3.19)、(3.20)式，有

$$P_Z(Z_i) = \frac{n_i}{n} \tag{3.27}$$

$$V_i = G(Z_i) = \sum_{i=0}^{k-1} P_Z(Z_i) \tag{3.28}$$

$$Z_i = G^{-1}(S_i) = G^{-1}[T(r_i)] \tag{3.29}$$

现在，我们仍以前面的 32 × 32、8 个灰度级图像为例，说明直方图规定化增强的过程。图 3.7(a)是原图像直方图，图 3.7(b)是期望图像的直方图。期望图像所对应的直方图的具体数值列于表 3.3。

首先由(3.20)式计算直方图均衡化后原始图像灰度 r_i 对应的变换函数S_i，为简单起见，直接引用表 3.2 的结果，抄录如下：

$S_0 = 1/7$　　$n_{S0} = 198$　　$P_S(S_0) = 0.19$

$S_1 = 3/7$　　$n_{S1} = 256$　　$P_S(S_1) = 0.25$

$S_2 = 5/7$　　$n_{S2} = 212$　　$P_S(S_2) = 0.21$

$S_3 = 6/7$　　$n_{S3} = 246$　　$P_S(S_3) = 0.24$

$S_4 = 1$　　$n_{S4} = 112$　　$P_S(S_4) = 0.11$

然后，求出给定直方图对应的灰度级 V_i(图 3.7(c))：

$V_0 = 0.00 = G(Z_0)$　　$V_1 = 0.00 = G(Z_1)$

$V_2 = 0.00 = G(Z_2)$　　$V_3 = 0.15 = G(Z_3)$

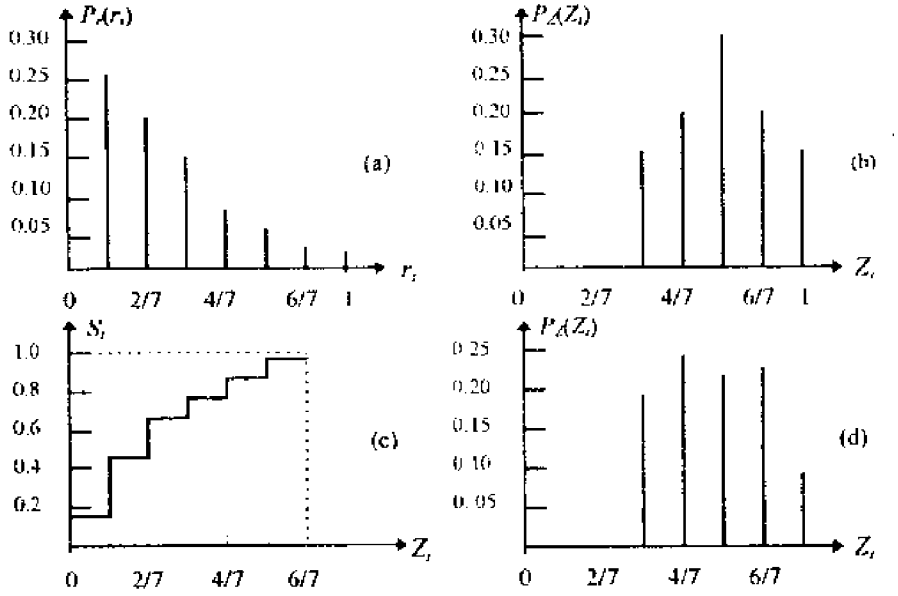


图 3.7　直方图规定化图例

(a)原始直方图；　(b)规定的直方图；　(c)变换函数；　(d)结果直方图

$$V_4 = 0.35 = G(Z_4) \qquad V_5 = 0.65 = G(Z_5)$$

$$V_6 = 0.85 = G(Z_5) \qquad V_7 = 1.00 = G(Z_7)$$

再根据(3.25)式要求，用 S_i 代替 V_i。由于是离散图像，所以采用“最靠近”原则，即用最靠近 V_i 的 S_i 来代替 V_i，并映射成灰度级 Z_i。结果是

$$S_0 = (1/7) \rightarrow V_3 \rightarrow G(Z_3) \rightarrow Z_3 = 3/7$$

$$S_1 = (3/7) \rightarrow V_4 \rightarrow G(Z_4) \rightarrow Z_4 = 4/7$$

$$S_2 = (5/7) \rightarrow V_5 \rightarrow G(Z_5) \rightarrow Z_5 = 5/7$$

$$S_3 = (6/7) \rightarrow V_6 \rightarrow G(Z_6) \rightarrow Z_6 = 6/7$$

$$S_4 = (1\quad) \rightarrow V_7 \rightarrow G(Z_7) \rightarrow Z_7 = 1$$

或者由(3.29)式反变换，即

$$G^{-1}(S_0) = Z_3 = 3/7 \qquad G^{-1}(S_1) = Z_4 = 4/7$$

$$G^{-1}(S_2) = Z_5 = 5/7 \qquad G^{-1}(S_3) = Z_6 = 6/7$$

$$G^{-1}(S_4) = Z_7 = 1$$

至于如何运用(3.24)式作映射，请读者自己分析。

最后应用上述映射重新分配像素，并用 32×32 去除，得到表 3.3 和图 3.7(d)。

综上所述，直方图规定化就是把直方图均衡化结果映射到设想的理想直方图上，使图像按人的意愿去变换。实现过程如下：

(1) 将原始图像作直方图均衡化处理，求出原图像中每一个灰度级 r_i 所对应的变换函数 S_i；

(2) 对给定直方图作类似计算，得到理想图像每一个灰度级 Z_i 对应的变换函数 V_i；

(3) 找出 $V_i \approx S_i$ 的点对，并映射到 Z_i；

(4) 求出 $P_Z(Z_i)$。

表 3.3 规定直方图和结果直方图

规定直方图		结果直方图	
Z_i	$P_Z(Z_i)$	n_i	$P_Z(Z_i)$
$z_0=0$	0.00	0	0.00
$z_1=1/7$	0.00	0	0.00
$z_2=2/7$	0.00	0	0.00
$z_3=3/7$	0.15	198	0.19
$z_4=4/7$	0.20	256	0.25
$z_5=5/7$	0.30	212	0.21
$z_6=6/7$	0.20	242	0.24
$z_7=1$	0.15	112	0.11

3.2 空间域图像平滑

任何一幅未经处理的原始图像，都存在着一定程度的噪声干扰。噪声恶化了图像质量，使图像模糊，甚至淹没特征，给分析带来困难。

消除图像噪声的工作称之为图像平滑或滤波。平滑的目的有两个：改善图像质量和抽出对象特征。

由于噪声源众多(如光栅扫描、底片颗粒、机械元件、信道传输等)，噪声种类复杂(如加性噪声、乘性噪声、量化噪声等)，所以平滑方法也多种多样。平滑可以在空间域进行，也可以在频率域进行。本节介绍空间域常用方法，频率域平滑将在 3.4 节中论述。

3.2.1 噪声消除法

由传感器或信道引起的噪声通常呈现孤立离散性分布，它们常常与邻域像素有比较明显的差异。噪声消除法平滑时，顺序检测每一个像素，如果某个像素的幅度大于其邻域像素的平均值，且达到一定水平，则判定该像素为噪声，继而用邻域像素均值取代这一像素。可见，噪声法平滑的关键是准确判断噪声像素。我们取一个 $N\times N$ 的窗口($N=3,5,7,\cdots$)，令窗口中心像素的灰度为 Z，窗口内其他像素的灰度的值为 Z_i，Z 与 Z_i 之差的绝对值为 ε_i。再定义一个统计变量 CNT 和门限值 T。将 (N^2-1) 个 ε_i 逐个与 T 作比较，如果 ε_i 大于等于 T，则给计数器 CNT 加 1(CNT 初始值为零)。处理完毕，若计数器 CNT 的值大于约定值 Y，就认为该像素为噪声，继而输出窗口均值，否则原值输出。

将窗口顺序移过全帧，即可完成噪声消除平滑。

噪声消除法简单易行，对单点噪声有较好的效果。需要注意的是，在操作中对门限 T 的选择要慎重，T 太小，噪声消除不干净；T 太大，易使图像模糊。

3.2.2 邻域平均法

令被讨论像素的灰度值为 $F(j,k)$，以其为中心，窗口像素组成的点集以 A 表示，集内像素数以 L 表示。经邻域平均法滤波后，像素 $F(j,k)$ 对应的输出为

$$G(j,k) = \frac{1}{L}\sum_{(x,y)\in A} F(x,y) \tag{3.30}$$

即用窗口像素的平均值取代 $F(j,k)$原来的灰度值。

邻域平均法平滑时，邻域的选取通常有两种方式：以单位距离为半径或单位距离的$\sqrt{2}$倍为半径。以3×3窗口为例(见图3.8)，单位距离为半径时，其邻域为

$$A_4 = \{(j,k-1),(j+1,k),(j,k+1),(j-1,k)\}$$

单位距离的$\sqrt{2}$倍为半径时，其邻域为

$$A_8 = \{(j-1,k-1),(j-1,k),(j-1,k+1),(j,k-1),(j,k+1),$$
$$(j+1,k-1),(j+1,k),(j+1,k+1)\}$$

前者称之为四点邻域，后者称之为八点邻域。四点邻域时，$L=4$。八点邻域时，$L=8$。

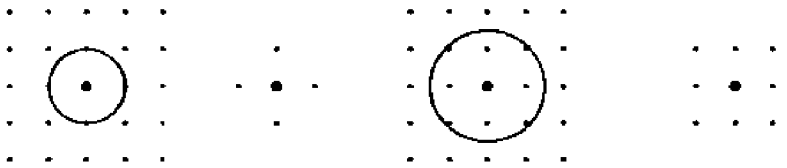


图3.8 (j,k)点的两种邻域

(a)四点邻域； (b)八点邻域

邻域平均法有力地抑制了噪声(可以证明，邻域平均法处理后，噪声功率将由原来的σ^2降到$N^{-1}\sigma^2$)，同时，也出现了因平均作用而引起的模糊(加宽)现象，模糊程度与邻域半径成正比。为了尽可能减小模糊失真，有人提出了“超限邻域平均法”，公式如下：

$$G(j,k) = \begin{cases} \dfrac{1}{L}\displaystyle\sum_{(x,y)\in A} F(x,y); & \left| F(j,k) - \dfrac{1}{L}\displaystyle\sum_{(x,y)\in A} F(j,k)\right| > T \\ F(j,k) \quad ; & \text{其他} \end{cases} \tag{3.31}$$

就是说当$F(j,k)$大于邻域平均值一定值后，作噪声处理，否则不改变。

图3.9是焊缝X射线底片图像经5次邻域平均法滤波后的情况，原来被淹没的缺陷明显表现了出来，而某些边界也随之变粗。

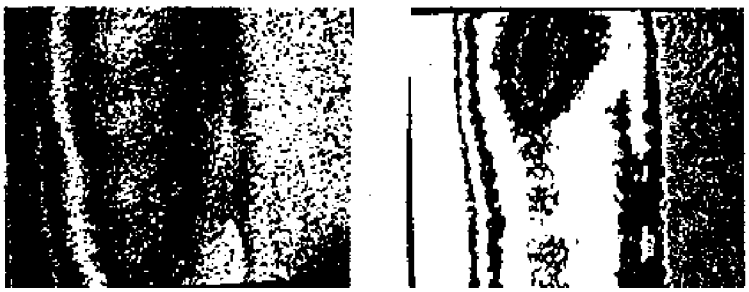


图3.9 邻域平均法平滑图例

(a)原始图像； (b)邻域平均法5次平滑后

3.2.3 中值滤波

中值滤波是一种对干扰脉冲和点状噪声有良好抑制作用，而对图像边缘能较好保持的非线性图像增强技术。在一维形式下，中值滤波器是一个含有奇数个像素的滑动窗口，经排序后，窗口像素序列为

$$\{F_{i-v},\cdots,F_{i-1},F_i,F_{i+1},\cdots,F_{i+v}\}$$

$v=(L-1)/2$，L 为窗口长度，F_i 即为窗口像素的中值滤波输出。记作

$$G_i=\mathrm{Med}\{F_{i-v},\cdots,F_i,\cdots,F_{i+v}\} \tag{3.32}$$

$\mathrm{Med}\{\cdot\}$表示取窗口中值。如一窗口长度为 5，像素灰度分别为{20,10,30,15,25}，则

$$G_i=\mathrm{Med}\{10,15,20,25,30\}=20$$

若灰度级为 30 的像素为椒盐噪声(表现为黑图像上的白点，白图像上的黑点)，在经过中值滤波后即被消除。

一维中值滤波的概念很容易推广到二维。这时取某种形式的二维窗口，将窗口内像素排序，生成单调二维数据序列$\{F_{jk}\}$。类似于一维，二维中值滤波输出 $G(j,k)$为

$$G(j,k)=\mathrm{Med}\{F_{jk}\} \tag{3.33}$$

一般来说，二维中值滤波器比一维滤波器更能抑制噪声。二维中值 滤波器的窗口形状可以有多种，如线状、方形、十字形、圆形、菱形等(见图 3.10)。不同形状的窗口产生不同的滤波效果，使用中必须根据图像的内容和不同的要求加以选择。从以往的经验看，方形或圆形窗口适宜于外廓线较长的物体图像，而十字形窗口优选于有尖顶角物体的图像。

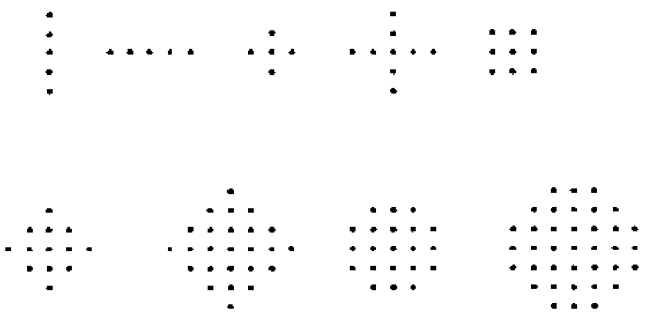


图 3.10 中值滤波器常用窗口

使用中值滤波器滤除噪声的方法有多种，且十分灵活。一种方法是先使用小尺度窗口，后逐渐加大窗口尺寸，直到中值滤波器的坏处多于好处为止。另一种方法是一维滤波器和二维滤波器交替使用。此外还有叠代操作，就是对输入 图像重复进行同样的中值滤波，直到输出不再有变化为止。

中值滤波具有许多重要特性，如：

(1) 对离散阶跃信号、斜升信号不产生影响，连续个数小于窗口长度一半的离散脉冲将被平滑，三角函数的顶部平坦化(见图 3.11)

(2) 令 C 为常数，则

$$\mathrm{Med}\{CF_{jk}\}=C\mathrm{Med}\{F_{jk}\}$$

$$\mathrm{Med}\{C+F_{jk}\}=C+\mathrm{Med}\{F_{jk}\}$$

$$\mathrm{Med}\{F_{jk}+f_{jk}\}\neq\mathrm{Med}\{F_{jk}\}+\mathrm{Med}\{f_{jk}\}$$

(3) 中值滤波后,信号频谱基本不变。

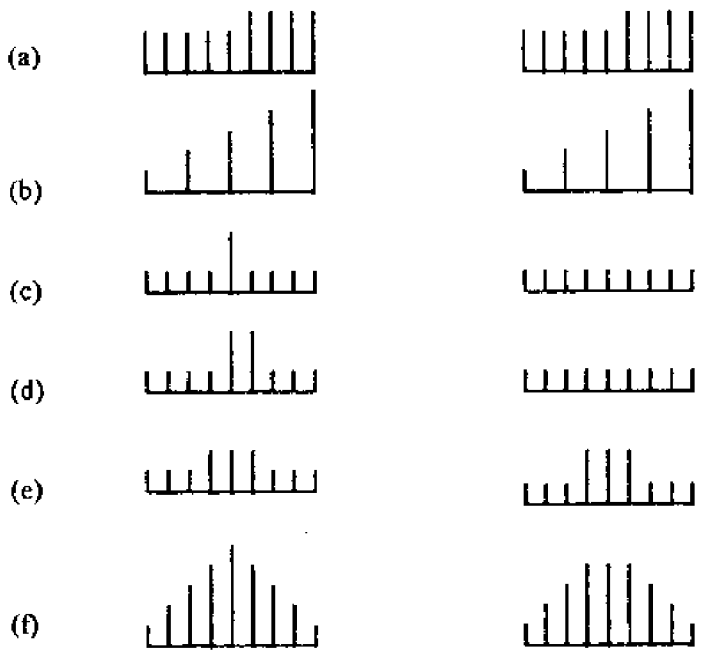


图 3.11 中值滤波示例

(a)阶跃; (b)斜坡; (c)单脉冲; (d)双脉冲; (e)三脉冲; (f)三角形

3.2.4 梯度倒数加权法

梯度倒数加权法平滑的产生起源于这样的考虑:在一帧离散图像中,相邻区域的变化大于区域内部的变化,在同一区域中中间像素的变化小于边沿像素的变化。梯度值正比于邻近像素灰度级差值,即在图像变化缓慢区域,梯度值小,反之则大。现在取梯度倒数,该倒数之大小正好与梯度相反,以梯度倒数作权重因子,则区域内部的邻点权重就大于边沿近旁或区域外的邻点。也就是说,这种平滑其贡献主要来自区域内部的像素,平滑后图像边沿和细节不会受到明显损害。

梯度倒数加权平滑的第一步是建立归一化的权重矩阵 W 作为平滑的掩模。对 3×3 窗口,其组成为

$$W=\begin{bmatrix} W(j-1,k-1) & W(j-1,k) & W(j-1,k+1) \\ W(j,k-1) & W(j,k) & W(j,k+1) \\ W(j+1,k-1) & W(j+1,k) & W(j+1,k+1) \end{bmatrix} \tag{3.34}$$

这里规定, $W(j,k)=\frac{1}{2}$,其余 8 个加权系数之和为 $\frac{1}{2}$ 。

定义除 $W(j,k)$外的其他权重矩阵元素为

$$W(j+m,k+n)=\frac{g(j,k;m,n)}{2\sum_{m=-1}^{1}\sum_{n=-1}^{1}g(j,k;m,n)} \tag{3.35}$$

$$g(j,k;m,n)=\frac{1}{|F(j+m,k+n)-F(j,k)|} \tag{3.36}$$

式中　m,n 分别为 $-1,0,1$。m,n 不能同时为零。

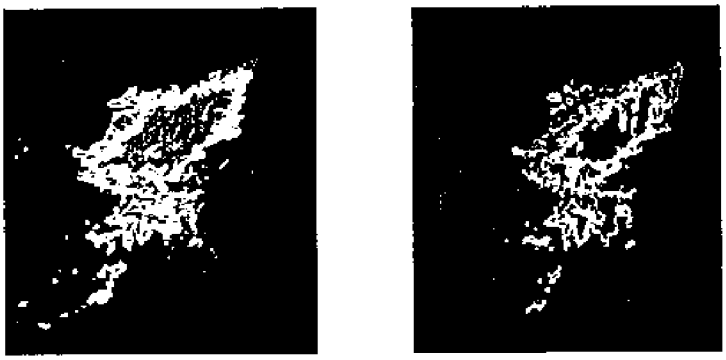
在此基础上，用窗口像素分别与相应位置上的权重因子相乘，积之和就是 $F(j,k)$的平滑输出 $G(j,k)$。

如有一窗口，其像素集为{3,5,7,3,9,4,4,3,6}(按行顺序展开)。根据(3.35)、(3.36)式计算得到

$$W=\begin{bmatrix}0.042 & 0.063 & 0.126\\ 0.042 & 0.50 & 0.051\\ 0.051 & 0.042 & 0.083\end{bmatrix}$$

$$G(j,k)=0.042\times3+0.063\times5+0.126\times7+0.042\times3+0.500\times9$$
$$+0.051\times4+0.051\times4+0.042\times3+0.083\times6\approx7$$

图 3.12(a)为初始图像，干扰甚多。图 3.12(b)为梯度倒数加权法平滑后的情景，其视觉效果大为改观。



(a)　　(b)

图 3.12　梯度倒数加权法平滑图例

(a)平滑前；　(b)经梯度倒数加权法平滑后

3.2.5　选择式掩模平滑

噪声消除法和邻域平均法在消除噪声的同时，都不可避免地带来平均化的缺憾，致使尖锐变化的边缘或线变得模糊。选择式掩模平滑旨在追求既完成滤波操作，又不破坏区域边界的细节。下面介绍一下基本思路。

取 5×5 窗口，在窗口内以中心像素 $F(j,k)$为基准点，制作 4 个五边形、4 个六边形、1 个边长为 3 的正方形共 9 个掩模(见图 3.13)。计算各掩模的均值(a_i)及方差(k_i)，

$$a_i=\left[\sum_{l=1}^{Q}F_l(j+m,k+n)\right]/Q \tag{3.37}$$

$$'k_i=\sum_{l=1}^{Q}\{[F(j+m,k+n)]^2-a_i^2\} \tag{3.38}$$

式中　$i=1,2,\cdots,9$;

Q 为各掩模对应的像素个数；

m、n 为掩模内像素相对于中心像素(j,k)的位移量。

在此基础上，对 k_i 排序，最小方差 $k_{i\min}$所对应的掩模的灰度级均值 a_i 作为 $F(j,k)$的平

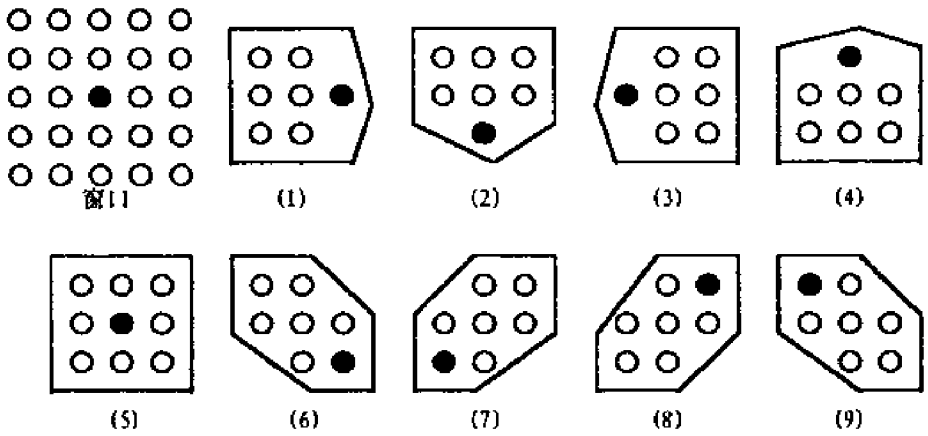


图 3.13　5×5 窗口及 9 个掩模(● 中心像素)

滑输出(凡含有尖锐边沿的区域,方差必定较平缓区域为大)$G(j,k)$。

用同样的方法作用于每一像素(因窗口是 5×5,边缘两行两列如不作延伸,将处理不到),即可完成全帧图像的平滑。

已知某窗口的灰度分布如下:

3	6	4	2	1
4	7	3	2	4
8	4	1	4	3
4	2	1	5	3
4	3	2	1	6

经计算,得

$$a_i = \{4,4,3,2,3,4,2,3,3\}$$
$$k_i = \{54,7,17,17,28,31,23,26,0\}$$
$$k_{i\min} = k_9$$
$$G(j,k) = 3$$

3.3　空间域图像锐化

锐化的目的是增强图像边缘,使人看起来惬意,以及便于对目标的识别和处理。

图像锐化有多种方法,如针对平均或积分运算使图像模糊,可逆其道而采取微分运算;使用高通滤波器优化高频分量,抑制低频分量,提高图像边界清晰度等。

3.3.1　梯度法

梯度是图像处理中最常用的一次微分方法。对图像函数 $F(j,k)$,其在点(j,k)上的梯度定义为矢量

$$G[F(j,k)] = \begin{bmatrix} \dfrac{\partial F}{\partial j} \\ \dfrac{\partial F}{\partial k} \end{bmatrix} \tag{3.39}$$

$G[F(j,k)]$的幅度

$$G[F(j,k)] = [(\frac{\partial F}{\partial j})^2 + (\frac{\partial F}{\partial k})^2]^{\frac{1}{2}} \tag{3.40}$$

在后面的讨论中,均以(3.40)式为基础。现在约定把 $G[F(j,k)]$叫做梯度,而不再说 $G[F(j,k)]$是 $G[F(j,k)]$的幅度。

对离散图像而言,可用差分近似表示(3.40)式,一种常用的近似关系式是:

$$G[F(j,k)] = \{[F(j,k) - F(j+1,k)]^2 + [F(j,k) - F(j,k+1)]^2\}^{\frac{1}{2}} \tag{3.41}$$

为便于编程和提高运算速度,在某些场合可进一步简化为

$$G[F(j,k)] \approx |F(j,k) - F(j+1,k)| + |F(j,k) - F(j,k+1)| \tag{3.42}$$

还有一种称为罗伯特梯度(Roberts Gradient)的算子,表达式为

$$G[F(j,k)] = \{[F(j,k) - F(j+1,k+1)]^2 + [F(j+1,k) - F(j,k+1)]^2\}^{\frac{1}{2}} \tag{3.43}$$

对应的绝对值表达式为

$$G[F(j,k)] = |F(j,k) - F(j+1,k+1| + |F(j+1,k) - F(j,k+1)| \tag{3.44}$$

(3.42)式和(3.44)式中各因子之间的关系分别如图 3.14(a)、3.14(b)所描述。对照此图,是很容易记住梯度差分表达式的。(3.42)式是沿水平轴和垂直轴两个方向的微分和,(3.44)式是取旋转 ±45°两个方向的微分的和。

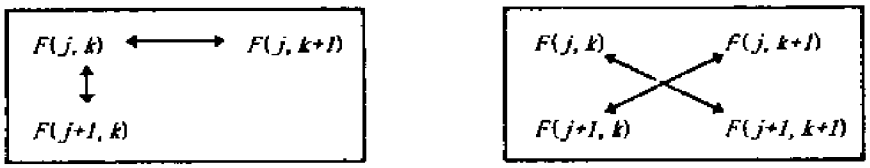


图 3.14　二维梯度的两种差分算法

知道了梯度 $G[F(j,k)]$后,如何确定锐化输出 $G(j,k)$呢? 提供几种思路,供大家选择。

(1) 直接以梯度值代替

$$G(j,k) = G[F(j,k)] \tag{3.45}$$

这种方法直截了当,简单。但在 $F(j,k)$均匀的区域,因梯度值 $G[F(j,k)]$很小(极端情况下甚至为零),会表现出暗的特性,这在有些场合是不适宜的。

(2) 辅以门限判断

$$G(j,k) = \begin{cases} G[F(j,k)]; & G[F(j,k)] \geqslant T \\ F(j,k) \quad ; & \text{其他} \end{cases} \tag{3.46}$$

该方法基本上不破坏图像背景,又可增强边缘。

(3) 给边缘规定一个特定的灰度级,即

$$G(j,k) = \begin{cases} L_a \quad ; & G[F(j,k)] \geqslant T \\ F(j,k) \; ; & \text{其他} \end{cases} \tag{3.47}$$

(4) 给背景规定特定的灰度级

$$G(j,k)=\begin{cases} G[F(j,k)]; & G[F(j,k)]\geqslant T \\ L_b \quad ; & \text{其他} \end{cases} \tag{3.48}$$

(5) 二值图像

$$G(j,k)=\begin{cases} L_a \quad ; & G[F(j,k)]\geqslant T \\ L_b \quad ; & \text{其他} \end{cases} \tag{3.49}$$

梯度是向量,各向同性。梯度方向对应于 $F(j,k)$最大变化率方向上,即

$$Q=\arctan\left[\frac{\partial F(j,k)}{\partial k}\Big/\frac{\partial F(j,k)}{\partial j}\right] \tag{3.50}$$

梯度幅度比例于相邻像素的灰度级差值。在灰度陡变区域,梯度值大;在灰度相似区域,梯度值小;在灰度级为常数的区域,梯度为零。熟悉梯度的这些性质,根据不同的目的选择不同的算法,就能获得满意的锐化效果。

3.3.2 Laplacian 算子

Laplacian 算子是线性二次微分算子,与梯度算子一样,具有旋转不变性,从而满足不同走向的图像边界的锐化要求。

对图像 $F(j,k)$,其 Laplacian 算子为

$$\nabla^2 F(j,k)=\frac{\partial^2 F(j,k)}{\partial j^2}+\frac{\partial^2 F(j,k)}{\partial k^2} \tag{3.51}$$

其离散形式为

$$\begin{aligned}\nabla^2 F(j,k)&=\{[F(j+1,k)-F(j,k)]-[F(j,k)-F(j-1,k)]\}\\&\quad+\{[F(j,k+1)-F(j,k)]-[F(j,k)-F(j,k-1)]\}\\&=F(j+1,k)+F(j-1,k)+F(j,k+1)+F(j,k-1)-4F(j,k)\end{aligned} \tag{3.52}$$

也可写成以下形式:

$$\begin{aligned}\nabla^2 F(j,k)=-5\{F(j,k)-\frac{1}{5}[&F(j+1,k)+F(j-1,k)\\&+F(j,k)+F(j,k+1)+F(j,k-1)]\}\end{aligned} \tag{3.53}$$

(3.53)式可解释为 Laplacian 算子等于 $F(j,k)$与其邻域均值之差。如果把包含 $F(j,k)$自身在内的邻域均值看成是 $F(j,k)$扩散形成的模糊的话,(3.53)式又可理解为 Laplacian 算子等于 $F(j,k)$与 $F(j,k)$模糊之差。

Laplacian 算子是标量,一般取正值或绝对值。

Laplacian 算子锐化时,其锐化输出为

$$G(j,k)=F(j,k)-\nabla^2 F(j,k) \tag{3.54}$$

已知 $1\times n$ 维数字图像集合

$$F(j)=\{\cdots,0,0,0,1,2,3,4,5,5,5,5,5,5,6,6,6,6,6,6,3,3,3,\cdots\}$$

参照(3.53)式,经计算

$$\nabla^2 F(j)=\{\cdots,0,0,1,0,0,0,0,-1,0,0,0,0,1,-1,0,0,0,0,-3,3,0,0,\cdots\}$$

锐化后的图像序列

$$\begin{aligned}G(j)&=F(j)-|\nabla^2 F(j)|\\&=\{\cdots,0,0,-1,1,2,3,4,6,5,5,5,5,4,7,6,6,6,6,9,0,3,3,\cdots\}\end{aligned}$$

结合图 3.15,可以发现一些有意义的现象:

(1) 在灰度均匀区间或灰度斜坡部分,$\nabla^2 F$ 为零。

(2) 在灰度级斜坡的顶部和底部均不为零,且底部形成灰度级"下冲"(如图 3.15 中 $j=3$ 的点),顶部形成灰度级"上冲"(如图 3.15 中 $j=8$ 的点)。

(3) 灰度级界线的低灰度级一侧形成灰度级"下冲"(图 3.15 中 $j=20$ 的点),而界线的高灰度级一侧则形成灰度级的"上冲"(图 3.15 中 $j=14, j=19$ 的点)。

(4) 正是这些"下冲"与"上冲"过程增加了灰度坡度的陡度,勾划出图像区域的外部廓线。

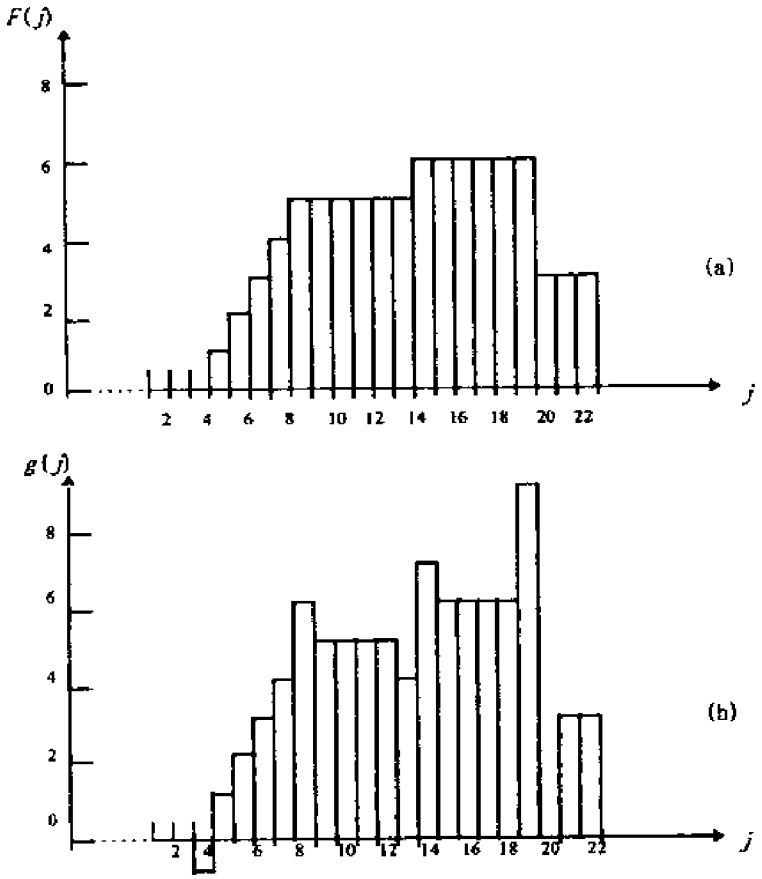


图 3.15 一维图像 Laplacian 算锐化示例

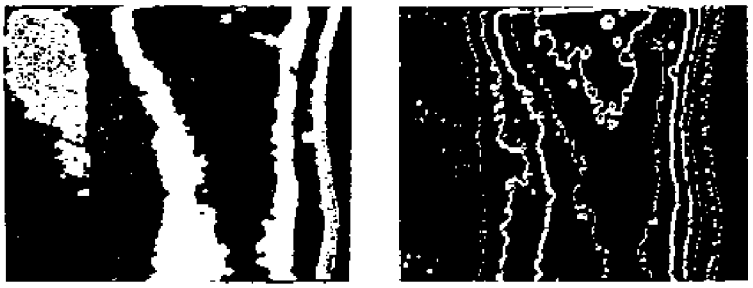
(a) 原始图像; (b) Laplacian 算子锐化后图像

二维离散图像 $F(j,k)$经 Laplacian 算子锐化后,具有一维图像 $F(j)$对 Laplacian 算子锐化所拥有的特性。因为由(3.52)、(3.54)式可得

$$G(j,k)=5F(j,k)-[F(j+1,k)+F(j-1,k)+F(j,k+1)+F(j,k-1)] \tag{3.55}$$

当 $F(j,k)$在斜坡中间或均匀区间时,$G(j,k)=F(j,k)$。当点(j,k)恰好在斜坡底,或者在灰度界线的低灰度级一侧时,而邻点的灰度高于或等于 $F(j,k)$,这时必然有这样的邻点,其$(F-\nabla^2 F)$小于 $F(j,k)$点的灰度,从而产生"下冲"。同样地,在斜坡顶部或灰度界线的

高灰度级一侧，没有灰 度级高于 $F(j,k)$，于是$(F-\nabla^2 F)$大于 $F(j,k)$的邻点就会产生“上冲”。和梯度一样，Laplacian 算子对图像界线有明显的增强作用(见图 3.16)。



(a)　　(b)

图 3.16　Laplacian 算子锐化图例

(a)锐化前图像；　(b)Laplacian 算子锐化处理图像

3.3.3　高通滤波

图像边缘与高频分量相对应。

高通滤波器让高频分量畅通无阻，而对低频分量则充分限制，从而达到图像锐化目的。

建立在离散卷积基础上的空间域高通滤波关系式：

$$g(m_1,m_2)=\sum_{n_1}\sum_{n_2}F(n_1,n_2)H(m_1-n_1+1,m_2-n_2+1) \tag{3.56}$$

式中　$g(m_1,m_2)$为锐化输出；

$F(n_1,n_2)$为输入图像；

$H(m_1-n_1+1,m_2-n_2+1)$为冲激响应阵列(亦叫卷积阵列)。

几种常用的归一化冲激阵列如下：

$$H_1=\begin{bmatrix}0&-1&0\\-1&5&-1\\0&-1&0\end{bmatrix} \tag{3.57}$$

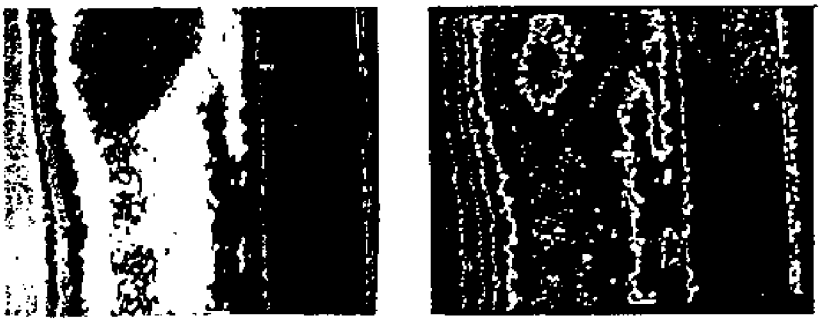
$$H_2=\begin{bmatrix}-1&-1&-1\\-1&9&-1\\-1&-1&-1\end{bmatrix} \tag{3.58}$$

$$H_3=\begin{bmatrix}1&-2&1\\-2&5&-2\\1&-2&1\end{bmatrix} \tag{3.59}$$

$$H_4=\frac{1}{7}\begin{bmatrix}-1&-2&-1\\-2&19&-2\\-1&-2&-1\end{bmatrix} \tag{3.60}$$

$$H_5=\frac{1}{2}\begin{bmatrix}-2&1&-2\\1&6&1\\-2&1&-2\end{bmatrix} \tag{3.61}$$

图 3.17(a)是原始图像，图 3.17(b)是运用(3.59)式作高通滤波后的情形。



(a)　　(b)

图 3.17　空间域高通滤波图例(H_3 冲激响应阵列)

(a)锐化前；　(b)空间域高通滤波后

3.3.4　掩模匹配法

所谓掩模匹配法，就是事先准备好 8 个方向(图 3.18)、大小为 3×3 的掩模，锐化时顺序作用于同一图像窗口，对每一个掩模，将窗口内各像素灰度值分别乘以该掩模相应的阵列元素，对积求累加和并以 $NUM_i(i=0,1,2,\cdots,7)$表示。将 NUM_i 排序，最大 NUM_i 即是窗口中心像素的锐化输出，NUM_i 所对应的模板的方向就是此窗口中心像素的方向。

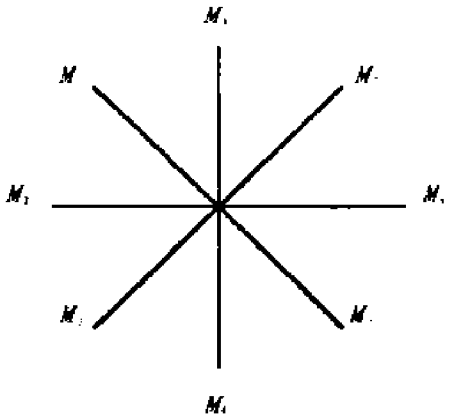


图 3.18　掩模方向定义图

典型空间域边缘检出掩模有三种，即罗比逊(Robison)掩模(图 3.19)、普雷外特(Prewitt)掩模(图 3.20)和柯赤(Kirsch)掩模(图 3.21)。同一图像，三种掩模锐化的结果并不完全一样。诚然，也难以划分各掩模的具体适用范围，使用中可通过人机交互方式予以选择比较，择优采用，即选择匹配最佳的掩模作为当前掩模。

与梯度和 Laplacian 算子相比较，掩模匹配除了能增强图像边缘外，还具有能平滑噪声的优点，所以从总体上要优于梯度和 Laplacian 算子。不足的是计算量大于前者。图 3.22 是罗比逊、普雷外特、柯赤三掩模作用于同一图像的情形。

M_0	M_1	M_2	M_3
1　2　1	2　1　0	1　0—1	0—1—2
0　0　0	1　0—1	2　0—2	1　0—1
—1—2—1	0—1—2	1　0—1	2　1　0
M_4	M_5	M_6	M_7
—1—2—1	—2—1　0	—1　0　1	0　1　2
0　0　0	—1　0—1	—2　0　2	—1　0　1
1　2　1	0　1　2	—1　0　1	—2—1　0

图 3.19　Robison 模板

M_0	M_1	M_2	M_3
1 1 1	1 1 1	1 1—1	1—1 1
1—2 1	1 -2—1	1—2—1	1—2—1
—1—1—1	1—1—1	1 1—1	1 1 1
M_4	M_5	M_6	M_7
—1—1—1	—1 -1 1	—1 1 1	1 1 1
1—2 1	—1—2 1	—1—2 1	—1—2 1
1 1 1	1 1 1	—1 1 1	—1—1 1

图 3.20 Prewitt 模板

M_0	M_1	M_2	M_3
5 5 5	5 5—3	5—3—3	—3—3—3
—3 0—3	5 0—3	5 0—3	5 0—3
—3—3—3	—3—3—3	5—3—3	5 5—3
M_4	M_5	M_6	M_7
—3—3—3	—3 -3—3	—3—3 5	—3 5 5
—3 0—3	—3 0 5	—3 0 5	—3 0 5
5 5 5	—3 5 5	—3—3 5	—3—3—3

图 3.21 Kirsch 模板



(a) (b) (c)

图 3.22 掩模匹配法锐化图例

(a)罗比逊掩模； (b)普雷外特掩模； (c)柯赤掩模

3.3.5 统计差值法

这是利用图像的局部统计特性(均值与方差)对图像边界作锐化的方法。基本思路是：

在二维离散图像 F 中，令点(j,k)处的灰度值为 $F(j,k)$，以(j,k)为中心，$(2r+1)(2s+1)$区域(图 3.23)内的灰度均值为 $E\{F(j,k)\}$，r、s 是大于零的整数。又令方差以 σ^2 表示。于是有

$$E\{F(j,k)\} = \frac{1}{(2r+1)(2s+1)}\sum_{m=j-r}^{j+r}\sum_{n=k-s}^{k+s}F(m,n) \tag{3.62}$$

$$\sigma^2(j,k) = \frac{1}{(2r+1)(2s+1)}\sum_{m=j-r}^{j+r}\sum_{n=k-s}^{k+s}[F(m,n) - E\{F(m,n)\}]^2 \tag{3.63}$$

$(m,n)\in S_{jk}$，S_{jk}为$(2r+1)$区域。

(j,k)点的锐化输出

$$G(j,k) = \frac{F(j,k)}{R(j,k)} \tag{3.64}$$

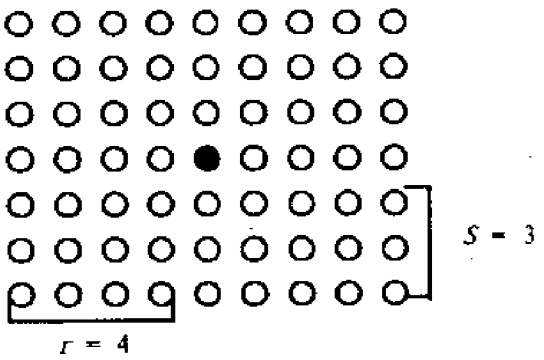


图 3.23　区域 S_R 示意图

沃利斯(Wallis)提出了统计差值法的另一种通用表达式：

$$G(j,k) = [F(j,k) - E\{F(j,k)\}][\frac{A\sigma_a}{A\sigma(j,k) + \sigma_d}] + [\alpha E_d\{F(j,k)\} + (1-\alpha)E\{F(j,k)\}] \tag{3.65}$$

式中　A 为增益系数；

σ_a 为期望标准差；

E_d 为理想局部均值；

α 为控制边缘与背景组成的比例系数。

作适当简化，沃氏统计差值算子变成

$$G(j,k) = E_d\{F(j,k)\} + \frac{\sigma_d}{\sigma(j,k)}[F(j,k) - E_d\{F(j,k)\}] \tag{3.66}$$

3.4　频率域增强处理

3.1 ~ 3.3 节较系统地介绍了在空间域内图像增强的目的、原理和方法。前面的讨论告诉我们，图像增强有两个方面的含义，一是消除图像生成或传输过程中引起的噪声，以改善图像的视觉形象；二是抽出图像廓线或突出图像边缘，以适应计算机处理的要求或作为图像识别的特征。针对上述含义，引出了空间域上的平滑操作和锐化处理。

类似于空间域，在频率域内同样可以进行滤波和边缘检出。前者采用低通滤波器和同态滤波器，后者一般借助高通滤波器实现。

3.4.1　低通滤波

在傅立叶变换域，变换系数反映了某些图像特征。如频谱的直流分量比例于图像的平均亮度，噪声对应于频率较高的区域，图像实体位于频率较低的区域等。变换域具有的这些内在特性常被用于图像增强，如构造一个低通滤波器，使低频分量顺利通过而有效地阻止高频分量，即可滤除频域噪声，再经反变换来取得平滑图像。

由卷积定理，低通滤波数学表达式是

$$G(u,v) = F(u,v)H(u,v) \tag{3.67}$$

式中　$F(u,v)$ 为含有噪声的原图像的傅立叶变换域；

$H(u,v)$ 为传递函数(又称转移函数)；

$G(u,v)$ 为经低通滤波后输出图像的傅立叶变换。

这里，假定噪声和信号成分在频率上可分离，且噪声表现为高频成分。H 滤波滤去了高频成分，而低频信息基本无损失地通过。滤波后的输出图像

$$\begin{aligned} G(j,k) &= \bar{f}^{-1}|F(u,v)H(u,v)| \\ &= \bar{f}^{-1}|G(u,v)| \end{aligned} \tag{3.68}$$

选择合适的传递函数 $H(u,v)$ 对频率域低通滤波关系重大。常用的频率域低通滤波器有：

(1) 理想圆形低通滤波器(Ideal Circular LowPass Filter　简写为 ICLPF)

它是一个在傅立叶平面上半径为 D_0 的圆形滤波器(图 3.24)，其传递函数

$$H(u,v) = \begin{cases} 1 & ;D(u,v) \leqslant D_0 \\ 0 & ;D(u,v) > D_0 \end{cases} \tag{3.69}$$

式中　D_0 为截止频率，大于零；$D(u,v)$ 是点(u,v)到傅立叶频率域原点的距离。

$$D(u,v) = (u_2 + v_2)^{\frac{1}{2}} \tag{3.70}$$

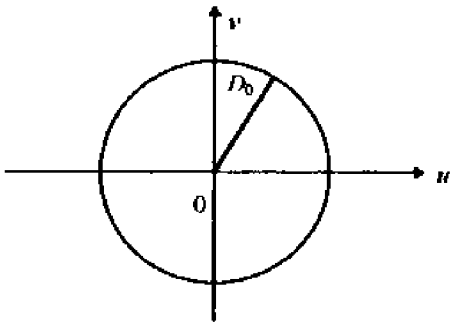


图 3.24　傅立叶平面上的 ICLPF

从理论上说，D_0 半径内的频率分量无损通过，而圆外的频率分量会被滤除。在 D_0 适当的情况下，ICLPF 不失为简单易行的平滑工具。若滤除的高频分量中含有大量的边缘信息，将会发生图像边缘模糊现象。

(2) 巴特沃思低通滤波器(Butterworth LowPass Filter　简写为 BLPF)

该滤波器的转移函数由下式决定：

$$H(u,v) = \frac{1}{1 + [D(u,v)/D_0]^{2n}} \tag{3.71}$$

或

$$H(u,v) = \frac{1}{1 + (\sqrt{2} - 1)[D(u,v)/D_0]^{2n}} \tag{3.72}$$

式中　正整数阶数 n 影响曲线的形状。在 $n = 1, D(u,v) = D_0$ 时，对(3.71)、(3.72)式来说，其 $H(u,v)$ 将分别为 2^{-1} 和 $(\sqrt{2})^{-1}$。图 3.25(a)是(3.71)式对应的 BLPF 特性曲线，特性曲线呈连续性衰减，而不像 ICLPF 曲线(图 3.25(b))那样有陡峭的截止区。因此，BLPF 滤波后，图像边缘的模糊程度会大大减轻(曲线的"尾部"含有许多与边界相关的高频分量)。

3.4.2　同态图像增强

图像信号 $F(j,k)$ 可以分解为入射光分量 $F_i(j,k)$ 和反射光分量 $F_r(j,k)$。同态图像增强就是在频率域内运用同一滤波器对这两个分量分别滤波，然后再合成、反变换。图 3.26 表示同态图像增强的过程框图。

图 3.26 表明，同态图像增强可通过下列步骤实现：

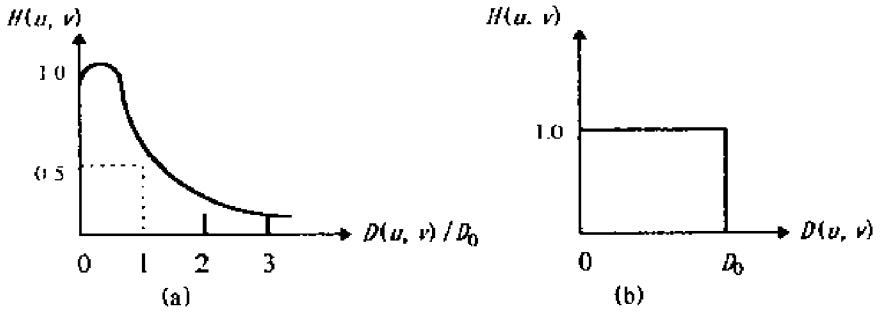


图 3.25　滤波器特性曲线

(a)BLPF 特性曲线；(b)ICLPF 特性曲线

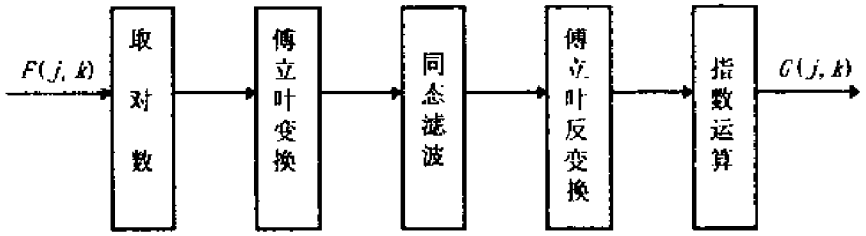


图 3.26　同态图像增强框图

(1)分解 $F(j,k)$,取对数

$$F(j,k)=F_i(j,k)\cdot F_r(j,k) \tag{3.73}$$

$$\ln F(j,k)=\ln F_i(j,k)+\ln F_r(j,k) \tag{3.74}$$

(2) 对(3.74) 式进行傅立叶变换

$$\begin{aligned}F(u,v)&=\tilde{f}\{\ln F(j,k)\}=\tilde{f}\{\ln F_i(j,k)\}+\tilde{f}\{F_r(j,k)\}\\&=F_i(u,v)+F_i(u,v)+F_r(u,v)\end{aligned} \tag{3.75}$$

(3) 用同一滤波器对频率域 $F_i(u,v)$ 和 $F_r(u,v)$ 进行滤波

$$G(u,v)=F_i(u,v)H(u,v)+F_r(u,v)H(u,v) \tag{3.76}$$

式中　$G(u,v)$ 为同态滤波输出图像的傅立叶变换域；$H(u,v)$ 为传递函数。

(4) 取傅立叶反变换

$$\begin{aligned}\ln F(j,k)&=\tilde{f}^{-1}\{G(u,v)\}\\&=\tilde{f}^{-1}\{F_i(u,v)H(u,v)\}+\tilde{f}^{-1}\{F_r(u,v)H(u,v)\}\end{aligned} \tag{3.77}$$

(5) 计算同态图像增强输出图像 $G(j,k)$

因为　$\ln F(j,k)=\tilde{f}^{-1}\{G(u,v)\}$

所以　$G(j,k)=\exp[\ln F(j,k)]$

$$=\exp[\tilde{f}^{-1}\{F_i(u,v)H(u,v)\}]\cdot\exp[\tilde{f}^{-1}\{F_r(u,v)H(u,v)\}] \tag{3.78}$$

同态图像增强的效果与滤波曲线的分布形状有关。我们知道，入射光分量是图像的基本成分，与傅立叶平面的低频分量相对应；反射光分量表示灰度的急剧变化部分，与傅立叶平面上的高频分量相关。如果令 H_h 和 H_l 分别表示高频分量频率场和低频分量频率场滤波特

性，则当取

$$\left.\begin{aligned} 0 \leqslant H_l < l \\ H_h > 1 \end{aligned}\right\} \tag{3.79}$$

时（见图 3.27(a)）入射光分量受到抑制，反射光分量得到增强，从而突出了图像边缘及线等结构。图 3.27(b) 则相反，读者自行分析。

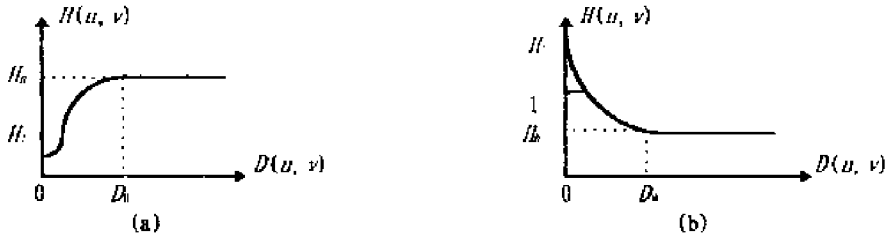


图 3.27　同态滤波器径向剖面图

(a) 增强高频分量；　(b) 增强低频分量

同态图像增强时，截止频率 D_0 要通过试验确定，它与 $F_i(u,v)$，$F_r(u,v)$ 场和频谱幅度比值有关。

3.4.3　高通滤波

高通滤波与低通滤波的作用相反，它使高频分量顺利通过，使低频分量受到削弱。频率域内常用的高通滤波器有四种，即理想高通滤波器、巴特沃斯高通滤波器、指数高通滤波器和梯形高通滤波器。参照 3.4.1 小节的低通滤波器，不难分析高通滤波器的工作原理，这里仅列出这四种高通滤波器的转移函数 $H(u,v)$。

(1) 理想高通滤波器(Ideal HighPass Filter　简写为 IHPF)

使特定频率区域的高频分量通过并保持不变，而其它频率区域的分量全部被抑制的滤波器。其转移函数为

$$H(u,v) = \begin{cases} 0 & ;D(u,v) \leqslant D_0 \\ 1 & ;D(u,v) > D_0 \end{cases} \tag{3.80}$$

(2) 巴特沃斯高通滤器(Butterworth HighPass Filter　简写为 BHPF)

$$H(u,v) = \frac{1}{1 + [D_0/D(u,v)]^{2n}} \tag{3.81}$$

或

$$H(u,v) = \frac{1}{1 + (\sqrt{2}-1)[D_0/D(u,v)]^{2n}} \tag{3.82}$$

为二维空间上的连续高通滤波器，n 为阶数，D_0 为指定常数，$D(u,v) = (u_2 + v_2)^{\frac{1}{2}}$。

(3) 指数高通滤波器(Exponential HighPass Filter　简写为 EHPF)

$$H(u,v) = e^{-[D_0/D(u,v)]^n} \tag{3.83}$$

式中，n 决定指数函数的衰减率。当 $D(u,v) = D_0$，$n = 1$ 时，$H(u,v) = e^{-1}$

(4) 梯形高通滤波器(Trapezoidal HighPass Filter　简写为 THPF)

$$H(u,v)=\begin{cases}0 & ; \quad D(u,v)<D_1\\ \dfrac{D(u,v)-D_1}{D_0-D_1} & ; \quad D_1\leqslant D(u,v)\leqslant D_0\\ 1 & ; \quad D(u,v)>D_0\end{cases} \tag{3.84}$$

这是滤波特性介于理想滤波器和完全平滑滤波器之间的一种高通滤波器。

图 3.28 给出上述频率域高通滤波器转移函数 $H(u,v)$ 的特性曲线。

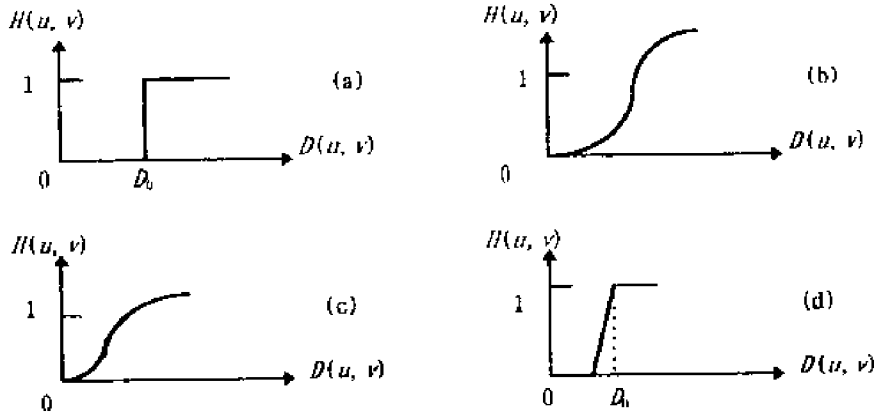


图 3.28　高通滤波器 $H(u,v)$ 特性曲线

(a)IHPF;　(b)BHPF;　(c)EHPF;　(d)THPF

3.5　彩色技术

彩色技术是从可视性角度实现图像增强的有效方法之一。众所周知，对一般观察者来说，眼睛能分辨的灰度级介于十几到二十几级之间，而对不同亮度和色调的彩色分辨能力却可达到灰度分辨能力的百倍以上。利用人类视觉系统的这一特性，将灰度图像变成彩色图像，或者改变已有彩色的分布，无疑都会改善图像的可视性。

3.5.1　假彩色处理

假彩色处理的对象是三基色描绘的自然图像或同一景物的多光谱图像。对自然图像，方法之一是将目标物(人们所关注的对象)映射为假彩色 —— 与原色不同的颜色。例如，绿色草原置成红色，蓝色海洋换成绿色等。这样做的目的是使目标物置于奇特的环境中以引起观察者的注意。方法之二是根据人的眼睛的色觉灵敏度，重新分配图像成分的颜色。如眼网膜中视锥体和视杆体对可见光区的绿色波长比较敏感，于是可将原来非绿色描述的图像细节或目标物经假彩色处理变成绿色，来达到提高目标分辨率的目的。自然图像的假彩色映射可定义为

$$\begin{bmatrix}R_F\\G_F\\B_F\end{bmatrix}=\begin{bmatrix}T_{11}&T_{12}&T_{13}\\T_{21}&T_{22}&T_{23}\\T_{31}&T_{32}&T_{33}\end{bmatrix}\begin{bmatrix}R_f\\G_f\\B_f\end{bmatrix} \tag{3.85}$$

式中　R_f、G_f、B_f 为原基色分量；

R_F、G_F、B_F 为假彩色三基色分量；

T_{ij} 为转移函数。

例如，如果按下列关系转化：

$$\begin{bmatrix} R_F \\ G_F \\ B_F \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} R_f \\ G_f \\ B_f \end{bmatrix}$$

则

$$\begin{aligned} R_F &= T_{11}R_f + T_{21}G_f + T_{31}B_f = B_f \\ G_F &= T_{12}R_f + T_{22}G_f + T_{32}B_f = R_f \\ B_F &= T_{13}R_f + T_{23}G_f + T_{33}B_f = G_f \end{aligned}$$

即原图像中的红色变成蓝色，绿色变成红色，蓝色变成绿色。下面讨论多光谱图像。

现在，多光谱图像在图像家族中已占有极重要 的位置。仅遥感图像而言，就有地质遥感图像、海洋遥感图像、气象遥感图像、森林遥感图像、农业遥感图像、军事遥感图像等。多光谱图像中，有一部分是非可见光波段，这些非可见光波段往往包含了可见光波段所不包含的信息。如红外波段，它具有夜间探测和全天候探测的能力。将红外波段与可见光波段感应的图像结合起来，可确保图像资料的连续性，丰富图像资料的内容。三基色与具有 n 个波段的多光谱图像 F_i 之间的关系为

$$\begin{cases} R_F = S_R\{F_i, i = 1,2,\cdots,n\} \\ G_F = S_G\{F_i, i = 1,2,\cdots,n\} \\ B_F = S_B\{F_i, i = 1,2,\cdots,n\} \end{cases} \tag{3.86}$$

式中　F_i 为第 i 波段图像；

$S_R\{\cdot\}$、$S_G\{\cdot\}$、$S_B\{\cdot\}$ 表示通用的函数运算。

3.5.2　伪彩色增强

伪彩色增强是针对灰度图像提出的，其宗旨是把离散黑白图像 $F(j,k)$ 的不同灰度级按照线性或者非线性映射成不同彩色，来提高图像内容的可辨识度。

1）强度分层

强度分层是伪彩色图像增强技术中原理最简单、操作最简便的一种。届时 $F(j,k)$ 看成是一个二维的强度函数，然后用 m 个平行于水平面的切割平面作切割，得到 m 个不同的灰度区间 $L_1, L_2\cdots, L_m$（见图 3.29）。这是一种均匀分层过程。还可以是非等间隔分层，即对关心的灰度级区间分得密一些，其他区段分得稀疏些。强度分层可通过硬件实现，也可由编程来实现。原则上说，强度分层技术的效果和分层密度成比例，层次越多，细节越丰富，彩色越柔和，但分层的层数受显示系统的硬件性能约束。

2）变换—合成法

将图像灰度 $F(j,k)$ 同时送入具有不同特性的红、绿、蓝三个变换器（见图 3.30），产生受变换函数调制的三基色分量 $R(j,k)$，$G(j,k)$，$B(j,k)$，用这三个分量分别控制彩色显示

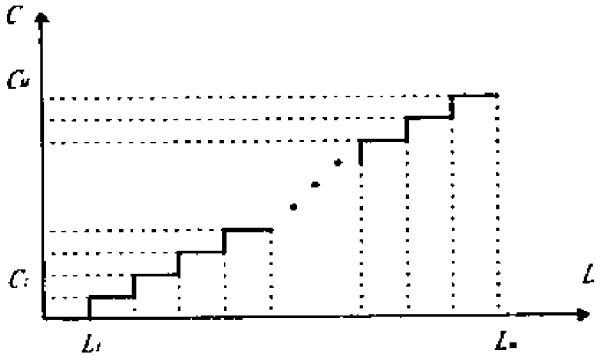


图 3.29　灰度区间与彩色代码

器的红、绿、蓝三个电子枪，从而合成与 $F(j,k)$ 幅度相对应的彩色。

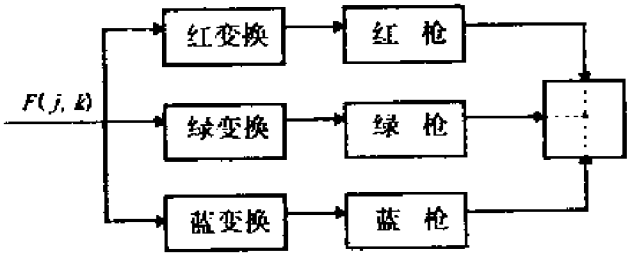


图 3.30　灰度图像转化成伪彩色图像原理图

变换—合成法产生的伪彩色是渐变的，色分量的形成受变换函数特性支配。图 3.31 是一组典型的变换函数特性曲线。其中，图 3.31(a)、(b)、(c)分别是红、绿、蓝三基色变换曲线，图 3.31(d)是前三种特性曲线的合成表示。纵观图 3.31 可以看到，$F(j,k)=0$ 时呈蓝色，$F(j,k)=L/2$ 时呈绿色，$F(j,k)=L$ 时呈红色。除此之外，将由三基色合成而产生不同的彩色。

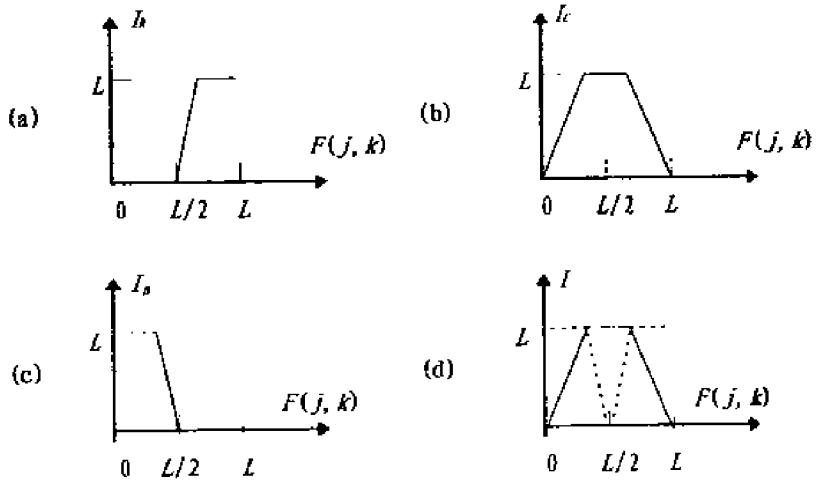


图 3.31　一组典型的彩色变换函数

(a)红基色；　(b)蓝基色；　(c)绿基色；　(d)合成表示

4 图像分割

●微分算子 ●曲面拟合 ●模板匹配
●区域生长 ●特征分类 ●纹理分析

从总体上说,图像分割就是把图像分成若干有意义的区域(部分)的处理技术。这里的"有意义"泛指"与目标对应"或"所研究的问题的函数"。例如,在医学图像处理中,如输入的是一幅细胞的照片,就需要分割出单个细胞。又如,如输入的是地形航空照片或地貌遥感图像,则就需要检析出山区、平原、水域、森林、城市、道路等。这些从图像域中分离出来的"目标"便是图像分割的对象。

图像分割的基础是像素间的相似性和跳变性。所谓"相似性"是指在某个区域内像素具有某种相似的特性,如灰度一样,纹理相同;所谓"跳变性"是指特性的不连续,如灰度值突变等。图像分割的方法有多种,如依据工作对象来分,可分为点相关分割和区域相关分割;按算法分类,可分为阈值法、界线检测法、匹配法、跟踪法等。本章采用基于像素灰度突变的边缘提取和基于特性相似性的区域生成法及纹理分析来研究图像的分割。

4.1 图像边缘提取

图像边缘对图像识别和计算机分析十分有用。边缘能勾划出目标物体,使观察者一目了然;边缘蕴含了丰富的内在信息(如方向、阶跃性质、形状等),是图像识别中抽取图像特征的重要属性。从本质上说,图像边缘是图像局部特性不连续性(灰度突变、颜色突变等)的反映,它标志着一个区域的终结和另一个区域的开始。

边缘提取首先检出图像局部特性的不连续性,然后再将这些不连续的边缘像素连成完备的边界。边缘的特性是沿边缘走向的像素变化平缓,而垂直于边缘方向的像素变化剧烈。所以,从这个意义上说,提取边缘的算法就是检出符合边缘特性的边缘像素的数学算子。目前,提取边缘常采用边缘算子法、曲面拟合法、模板匹配法、门限化等方法。下面分别予以讨论。

4.1.1 边缘算子法

1) 微分算子

图 4.1 是用梯度算子(见(4.3)式)提取边缘的示例。在图 4.1(a)中,在灰度级为 1 的背景上,含有两个灰度级为 2 - 3 的目标图像。图 4.1(b)是边缘提取后的情况,它十分清晰地勾勒出目标图像的轮廓(图中,最右侧一列和最下边一行为初始灰度级)。此例表明,微分算子的特性之一,是在图像灰度变化缓慢的区域其值较小;在图像灰度迅速变化的点处,其值

较大;在目标区域内部(假设灰度一致)其值为零。很自然,这类算子可以作为图像边缘检测器,它在边缘像素处的值就代表该点的"边缘强度"。

1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	3	2	1	1	1	2	1	1	1
1	1	1	3	3	3	3	1	2	2	2	1	1
1	2	3	3	3	3	3	1	1	2	1	1	1
1	1	2	3	3	3	2	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1

(a)

0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	2	1	0	0	0	1	0	0	1
0	0	0	4	1	2	2	0	2	2	1	0	1
0	1	4	0	0	0	2	1	2	2	2	0	1
1	2	1	0	0	0	3	0	1	2	0	0	1
0	1	2	2	2	3	2	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1

(b)

图 4.1 梯度算子边缘提取示例

(a)图像灰度分布; (b)梯度分布

(1) 梯度算子

在 3.3 节曾经讨论,在点 F(j,k)处,梯度 $G[F(j,k)]$的幅度为

$$G[F(j,k)] = [(\frac{\partial F}{\partial j})^2 + \frac{\partial F}{\partial k})^2]^{\frac{1}{2}} \tag{4.1}$$

对数字图像,(4.1)式可写成

$$G[F(j,k)] = \Delta_j F^2 + \Delta_k F^2)^{\frac{1}{2}} \tag{4.2}$$

有时为了简化计算,进一步取

$$G[F(j,k)] = |\Delta_j F| + |\Delta_k F| \tag{4.3}$$

或

$$G[F(j,k)] = \max(|\Delta_j F|, |\Delta_k F|) \tag{4.4}$$

式中(参见图 4.2(a))

$$\Delta_x F = F(j,k) - F(j+1,k)$$

$$\Delta_k F = F(j,k) - F(j,k+1)$$

在图像边缘提取中,还经常使用一种叫做"罗伯特梯度算子"的微分算子,

$$G[F(j,k)] = (\Delta'_j F^2 + \Delta'_k F^2)^{\frac{1}{2}} \tag{4.5}$$

或

$$G[F(j,k)] = |\Delta'_j F| + |\Delta'_k F| \tag{4.6}$$

或

$$G[F(j,k)] = \max(|\Delta'_j F|, |\Delta'_k F|) \tag{4.7}$$

其中(参见图 4.2(b))

$$\Delta'_j F = F(j,k) - F(j+1,k+1)$$

$$\Delta'_k F = F(j+1,k) - F(j,k+1)$$

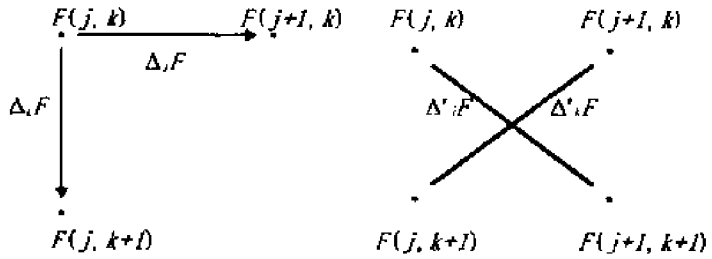


图 4.2 $\Delta_j F$、$\Delta_k F$、$\Delta'_j F$、$\Delta'_k F$ 示意图

(2) Laplacian 算子

二阶微分算子 Laplacian 算子也可用来提取图像的边缘。

在数字图像中,Laplacian 算子的一般形式为

$$\nabla^2 F(j,k) = \sum_{u,v \in S} [F(u,v) - F(j,k)] \tag{4.8}$$

式中,S 可以是 $F(j,k)$为中心上、下、左、右 4 邻点的集合(见图 4.3(a)),也可以是 8 邻点集合(见图 4.3(b)),或者是对角线 4 邻点的集合(图 4.3(c))。与之相对应的表达式分别为(4.9)~(4.11)式。

$$\nabla^2 F(j,k) = F(j+1,k) + F(j-1,k) + F(j,k+1) + F(j,k-1) - 4F(j,k) \tag{4.9}$$

$$\begin{aligned}\nabla^2 F(j,k) = {} & F(j-1,k-1) + F(j-1,k) + F(j-1,k+1) + F(j,k-1) \\ & + F(j,k+1) + F(j+1,k-1) + F(j+1,k) \\ & + F(j+1,k+1) - 8F(j,k)\end{aligned} \tag{4.10}$$

$$\begin{aligned}\nabla^2 F(j,k) = {} & F(j-1,k-1) + F(j-1,k+1) + F(j+1,k-1) \\ & + F(j+1,k+1) - 4F(j,k)\end{aligned} \tag{4.11}$$

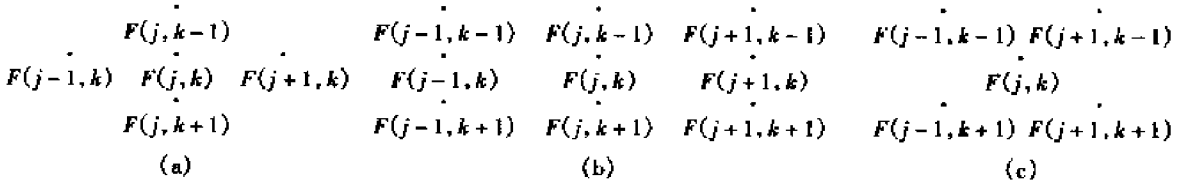


图 4.3 Laplacian 算子集合

下面以图 4.4 所示的图像边缘灰度分布,运用(4.9)式进行计算,以了解 Laplacian 算子用于边缘提取的特性。处理结果见图 4.5。在图 4.4、4.5 中,中心的点用作比较输入和输出图案的参考值,虚线框为处理范围。分析两图不难看出,孤立点(见图 4.4(a))的输出是一个略为扩大或略带模糊的点(见图 4.5(a)),其输出幅度是该点灰度值的 4 倍;对于线结构

(见图 4.4(b)),输出宽度加粗,外观仍呈线型,组成初始线的各位置上的 $\nabla^2 F$ 值是原来幅度的 2 倍(图 4.5(b));对阶跃边缘(图 4.4(c)),其检测结果是线(图 4.5(c))。

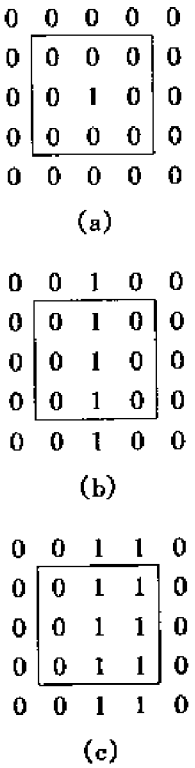


图 4.4 灰度分布

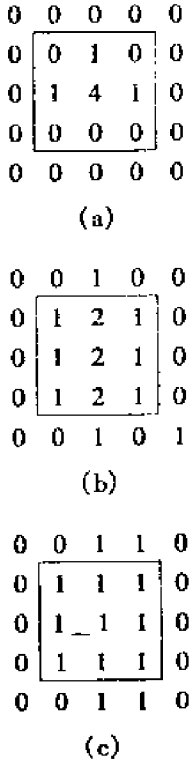


图 4.5 $\nabla^2 F$ 值

以上是直接以 $|\nabla^2 F(j,k)|$ 作为边缘像素灰度,实际工作中也可以把 $|\nabla^2 F(j,k)| > T$(门限)或 $|F(j,k) - \nabla^2 F(j,k)|$ 的像素作为边缘元。不同的定义方法其边缘检出后所获得的目标图像的形状与面积以及组成边缘的像素的灰度值会略有差异。不同的定义方法适用于不同的图像,在使用中可通过试验选择与实际图像相匹配的算法。

2) Laplacian – Gauss 算子

梯度算子和 Laplacian 算子对噪声比较敏感。对此,一方面可在运用这两种算子作边缘提取前,先用邻域平均法等作平滑处理,另一方面可"先用高斯形二维低通滤波器对图像 $F(j,k)$ 进行滤波,然后再对图像作 Laplacian 边缘提取"的所谓 Laplacian – Gauss 算子法。具体是,令 $g(j,k)$ 为高斯低通滤波后的图像,$\nabla^2 g(j,k)$ 表示边缘提取后的图像,则

$$\begin{aligned}\nabla^2 g(j,k) &= \frac{\partial^2 g}{\partial j^2} + \frac{\partial^2 g}{\partial k^2} \\ &= \frac{g(j,k)}{\sigma^2 \exp\left(-\frac{j^2+k^2}{2\sigma^2}\right)}\left[\frac{j^2+k^2}{\sigma^2} - 2\right] \cdot \exp\left(-\frac{j^2+k^2}{2\sigma^2}\right)\end{aligned} \tag{4.12}$$

式中

$$\sigma = \frac{W}{2\sqrt{2}} \tag{4.13}$$

参数 W 为正瓣宽度(见图 4.6)。

图 4.6 描述了函数 $\nabla^2 g(j,k)$ 的截面情况。它以坐标原点为中心全方位对称，并带有与主瓣相反的拖尾。也可用 Laplacian – Gauss 算子对 $F(j,k)$ 作卷积计算，所用窗口不宜太小，窗口尺寸与主瓣宽度有关，窗口模板内的系数之和为零。边缘与卷积计算后的零交叉点对应。

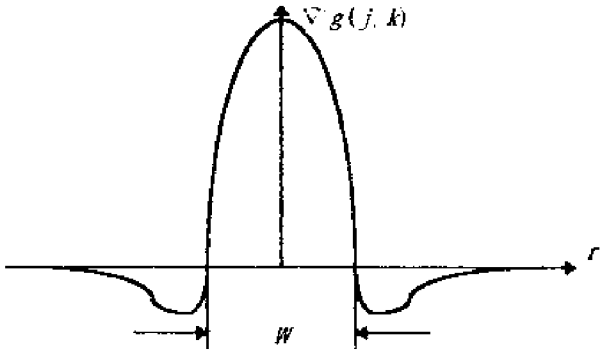


图 4.6 $\nabla^2 g(j,k)$ 算子的截面图

4.1.2 曲面拟合法

如果说用微分算子进行边缘提取存在“提升噪声”缺陷的话，曲面拟合法则可以在完成边缘检出的同时，能较好地抑制噪声的干扰。

曲面拟合法的基本思路是用一个平面或曲面去逼近一个图像面积元，然后用这个平面或曲面的梯度代替点的梯度，从而实现边缘检测。

1) 一次平面拟合

令图像面积元由 $F(j,k+1)$、$F(j+1,k+1)$、$F(j,k)$、$F(j+1,k)$ 4 个相邻像素组成(图 4.7(a))。现用一次平面 $ax+by+c$ 去拟合该面积元 Δs 上 4 个相邻像素，即用

$$f(x,y)=ax+by+c \tag{4.14}$$

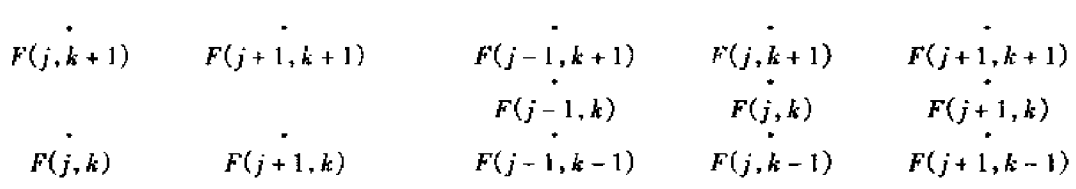


图 4.7 面积元示意图

(a)4 点面积元； (b)9 点面积元

去逼近 $F(x,y)$。已知 $f(x,y)$ 与 $F(x,y)$ 之间的均方误差

$$\begin{aligned}\varepsilon &= \sum_{x,y\subset\Delta s}[f(x,y)-F(x,y)]^2\\ &=[aj+bk+c-F(j,k)]^2+[a(j+1)+bk+c-F(j+1,k)]^2\\ &\quad+[aj+b(k+1)+c-F(j,k+1)]^2+[a(j+1)+b(k+1)\\ &\quad+c-F(j+1,k+1)]^2\end{aligned} \tag{4.15}$$

为达到最佳吻合，应使均方误差最小。为此，将(4.15)式对 a、b、c 求导并令其等于零，即

$$j[aj+bk+c-F(j,k)]+(j+1)[a(j+1)+bk+c-F(j+1,k)]$$

$$+ j[aj + b(k+1) + c - F(j,k+1)] + (j+1)[a(j+1) + b(k+1) + c - F(j+1,k+1)] = 0$$

$$k[aj + bk + c - F(j,k)] + k[a(j+1) + bk + c - F(j+1,k)] + (k+1)[aj + b(k+1) + c - F(j,k+1)] + (k+1)[a(j+1) + b(k+1) + c - F(j+1,k+1)] = 0$$

$$[aj + bk + c - F(j,k)] + [a(j+1) + bk + c - F(j+1,k)] + [aj + b(k+1) + c - F(j,k+1)] + [a(j+1) + b(k+1) + c - F(j+1,k+1)] = 0$$

对 a、b 和 c 求解上述方程式，实际解为

$$a = \frac{F(j+1,k) + F(j+1,k+1)}{2} - \frac{F(j,k) + F(j,k+1)}{2} \tag{4.16}$$

$$b = \frac{F(j,k+1) + F(j+1,k+1)}{2} - \frac{F(j,k) + F(j+1,k)}{2} \tag{4.17}$$

$$c = \frac{1}{4}[3F(j,k) + F(j+1,k) + F(j,k+1) - F(j+1,k+1) - ja - kb] \tag{4.18}$$

根据梯度定义，平面 $ax + by + c$ 上的梯度幅度

$$G[f(x,y)] = [(\frac{\partial f}{\partial x})^2 + (\frac{\partial f}{\partial y})^2]^{\frac{1}{2}} = (a^2 + b^2)^{\frac{1}{2}} \tag{4.19}$$

也可进一步表示为

$$G[f(x,y)] = |a| + |b| \tag{4.20}$$

$$G[f(x,y)] = \max(|a|, |b|) \tag{4.21}$$

由(4.16)、(4.17)式可知，a 是两列的平均值的差分，b 是两行的平均值的差分。由于这里的差分是建立在平滑基础上，所以对噪声就不像 4.1.1 小节中直接使用微分算子那样敏感。

因为这个平面是对已知 2×2 邻域内的图像灰度级的最好近似，故可合理地把平面的梯度看作是在该邻域的中心 $(j + \frac{1}{2}, k + \frac{1}{2})$ 点处图像梯度的近似值。当然，这种近似值是一种似是而非的“数字梯度”，因为它由水平差分和垂直差分组成。

2) 二次曲面拟合

设被检测像素所在的面积元(图 4.7(b))

$$\Delta s = \{F(j-1,k+1), F(j,k+1), F(j+1,k+1), F(j-1,k), F(j,k), F(j+1,k), F(j-1,k-1), F(j,k-1), F(j+1,k-1)\}$$

如同一次曲面拟合那样，用二次曲面

$$f(x,y) = ax^2 + bxy + cy^2 + dx + ey + g \tag{4.22}$$

去拟合面积元 Δs，并产生均方误差

$$\varepsilon = \sum_{x,y \in \Delta s} [(ax^2 + bxy + cy^2 + dx + ey + g) - F(x,y)]^2 \tag{4.23}$$

同样认为 $f(x,y)$、$F(x,y)$ 之间的均方误差最小，即用 ε 分别对 a、b、c、d、e、g 求偏导，并令

$$\frac{\partial \varepsilon}{\partial a} = 0 \quad ; \quad \frac{\partial \varepsilon}{\partial b} = 0 \quad ; \quad \frac{\partial \varepsilon}{\partial c} = 0$$

$$\frac{\partial \varepsilon}{\partial d} = 0 \quad ; \quad \frac{\partial \varepsilon}{\partial e} = 0 \quad ; \quad \frac{\partial \varepsilon}{\partial g} = 0$$

从而解得系数 a、b、c、d、e、g。再根据式

$$G[f(x,y)] = [(\frac{\partial f}{\partial x})^2 + (\frac{\partial f}{\partial y})^2]^{\frac{1}{2}} \tag{4.24}$$

求得曲面梯度幅度。

4.1.3 模板匹配法

模板(又称掩模)是为了检测某些图像区域特征而设计的阵列。模板元素和窗口像素之间的对应关系(以 3×3 窗口为例)定义如下:

模板

$$[M] = \begin{bmatrix} M_{-1,-1} & M_{-1,0} & M_{-1,1} \\ M_{0,-1} & M_{0,0} & M_{0,1} \\ M_{1,-1} & M_{1,0} & M_{1,1} \end{bmatrix} \tag{4.25}$$

窗口灰度

$$[F] = \begin{bmatrix} F(j-1,k-1) & F(j-1,k) & F(j-1,k+1) \\ F(j,k-1) & F(j,k) & F(j,k+1) \\ F(j+1,k-1) & F(j+1,k) & F(j+1,k+1) \end{bmatrix} \tag{4.26}$$

可分别用$[M] = M_{m,n}$和$[F] = F(j+m,k+n]$表示,m、$n = -1,0,1$。此外,也可以把上述模板阵列和图像窗口阵列堆叠成 9 维向量形式:

$$\boldsymbol{M} = \begin{bmatrix} M_{-1,-1} \\ M_{-1,0} \\ \cdots \\ M_{1,1} \end{bmatrix} \tag{4.27}$$

和

$$\boldsymbol{F} = \begin{bmatrix} F(j-1,k-1) \\ F(j-1,k) \\ \cdots \\ F(j+1,k+1) \end{bmatrix} \tag{4.28}$$

当窗口扩大到 $N \times N$ 时,就要处理 N^2 维的矢量。

模板匹配过程就是下式求乘积和的过程:

$$f(j,k) = \sum_{m=-l}^{l} \sum_{n=-l}^{l} F(j+m,k+n) M_{m,n} \tag{4.29}$$

式中 $l = L/2$,L 为窗口宽度,对 3×3 窗口,$l = -1,0,1$;

$f(j,k)$为模板匹配法边缘检测输出。

$\boldsymbol{M}$ 和 $\boldsymbol{F}$ 的内积为

$$\begin{aligned} \boldsymbol{M}^T\boldsymbol{F} = {} & M_{-l,-l}F(j-l,k-l) + \cdots + M_{-l,0}F(j-l,k) + \cdots \\ & + M_{l,l}F(j+l,k+l) \end{aligned} \tag{4.30}$$

这和上面讨论的乘积的和是一致的。

1) 点模板

点模板(图 4.8(a)、图 4.8(b)为线模板)一般用于背景强度恒定、目标图像灰度相同或基本相同的图像(如二值图像)。点模板检测图像区域时,通常是拖动模板在图像域移动,横

向移动间隔取1个像素，纵向移动间隔取1个扫描行。在每一个位置上，将模板元素分别与相应的图像灰度级相乘并求和(当和小于零时，可作两种处理，一是取绝对值，二是作零对待)。当$f(j,k)=0$时，说明模板位于均衡背景或目标区域内部，$F(j+m,k+n)$相同；当$f(j,k)\neq 0$，表明当前窗口内既有背景也有目标；当模板中心正好是目标和背景的交界处时，$f(j,k)$值最大；模板中心离开目标和背景的交界处时，$f(j,k)$值减小。图4.9(a)是背景灰度为1、目标灰度为4的二值图像，图4.9(b)是点模板中间计算结果，图4.9(c)是点模板最后匹配输出，结果与上述结论吻合。

			M_1			M_2			M_3			M_4		
-1	-1	-1	-1	-1	-1	-1	-1	2	-1	2	-1	2	-1	-1
-1	8	-1	2	2	2	-1	2	-1	-1	2	-1	-1	2	-1
-1	-1	-1	-1	-1	-1	2	-1	-1	-1	2	-1	-1	-1	2

(a) (b)

图4.8 匹配滤波器模板

(a)点模板； (b)线模板

2) 线模板

线模板能有效地检出线型类图像结构。

图4.8(b)$M_1 \sim M_4$分别是0°、45°、90°、-45°四个方向的线模板。当$M_1 \sim M_4$在图像域内自上而下、自左向右移动时，在背景灰度级不变、线宽度为1个像素的情况下，M_1模板对水平线响应最佳；M_2模板对45°方向的线匹配最好；M_3、M_4模板则对垂直线和斜线(\)响应最好。

1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	4	1	1	1	1	1	1
1	1	1	4	4	4	4	4	1	1	1	1
1	1	1	4	4	4	4	4	4	1	1	1
1	1	1	1	4	4	4	4	4	1	1	1
1	1	1	1	4	4	4	4	1	1	1	1
1	1	1	1	1	4	4	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	0	-3	-6	-12	15	-12	-6	-3	0	0	1
1	0	-6	15	6	6	6	12	-9	-3	0	1
1	0	-6	12	3	0	0	3	12	-6	0	1
1	0	-3	-12	6	0	0	3	12	-6	0	1
1	0	0	-6	12	3	3	9	-9	-3	0	1
1	0	0	-3	-9	12	12	-9	-3	0	0	1
1	1	1	1	1	1	1	1	1	1	1	1

(b)

图4.9 点模板应用示例

1	1	1	1	1	1	1	1	1	1	1	1
1	0	0	0	0	15	0	0	0	0	0	1
1	0	0	15	6	6	6	12	0	0	0	1
1	0	0	12	3	0	0	3	12	0	0	1
1	0	0	0	6	0	0	3	12	0	0	1
1	0	0	0	12	3	3	9	0	0	0	1
1	0	0	0	0	12	12	0	0	0	0	1
1	1	1	1	1	1	1	1	1	1	1	1

(c)

续图 4.9　点模板应用示例

(a)初始灰度级分布；　(b)点模板中间计算结果；　(c)点模板匹配输出

利用线模板对图像作线检测的过程是：对某一给定的窗口，分别计算模板 M_1、M_2、M_3、M_4 的匹配输出 $f_m(j,k)$，$m=1,2,3,4$，窗口输出

$$f(j,k)=\max\{f_m(j,k)\} \tag{4.31}$$

式中

$$f_m(j,k)=\sum_{x=-l}^{l}\sum_{y=-l}^{l}h_{x,y}F(j+x,k+y) \tag{4.32}$$

$h_{x,y}$是线模板系数。对 3×3 窗口 $l=-1,0,1$。

图 4.10(a)、(b)、(c)为初始图像灰度分布，其中分别嵌有垂直、倾斜、水平三种线型图像结构(虚线框内)。现用图 4.8(b)中的线模板对其作匹配检测，过程列于表 4.1，结果见图 4.10(d)、(e)、(f)，可以看到线结构获得了明显加强。

1	1	1	1	1
1	1	2	1	1
1	1	2	1	1
1	1	2	1	1
1	1	1	1	1

(a)

1	1	1	1	1
1	1	1	2	1
1	1	2	1	1
1	2	1	1	1
1	1	1	1	1

(b)

1	1	1	1	1
1	1	1	1	1
1	2	2	2	1
1	1	1	1	1
1	1	1	1	1

(c)

1	4	1
0	6	0
1	4	1

(d)

1	1	4
1	6	1
4	1	2

(e)

1	0	1
4	6	4
1	0	1

(f)

图 4.10 线模板示例

表 4.1

灰　　度	$f_1(j,k)$	$f_2(j,k)$	$f_3(j,k)$	$f_4(j,k)$	$f_5(j,k)$
$F(j-1,k-1)$	1	-2	-2	1	1
$F(j-1,k)$	1	1	4	1	4
$F(j-1,k+1)$	1	1	-2	-2	1
$F(j,k-1)$	0	0	-3	0	0
$F(j,k)$	0	0	6	0	6
$F(j,k+1)$	0	0	-3	0	0
$F(j+1,k-1)$	1	1	-3	-2	1

续表 4.1

灰　度	$f_1(j,k)$	$f_2(j,k)$	$f_3(j,k)$	$f_4(j,k)$	$f_5(j,k)$
$F(j+1,k)$	1	1	4	0	4
$F(j+1,k+1)$	1	-2	-2	1	1
$F(j-1,k-1)$	-1	-1	-1	1	1
$F(j-1,k)$	1	-2	1	-2	1
$F(j-1,k+1)$	1	4	1	1	4
$F(j,k-1)$	1	-2	1	-2	1
$F(j,k)$	0	6	0	0	6
$F(j,k+1)$	1	-2	1	-1	1
$F(j+1,k-1)$	1	4	1	1	4
$F(j+1,k)$	1	-2	-1	-2	1
$F(j+1,k+1)$	-1	-1	-1	2	2
$F(j-1,k-1)$	-2	-2	1	1	1
$F(j-1,k)$	-3	0	0	0	0
$F(j-1,k+1)$	-2	1	1	-2	1
$F(j,k-1)$	4	1	1	1	4
$F(j,k)$	6	0	0	0	6
$F(j,k+1)$	4	1	1	1	4
$F(j+1,k-1)$	-2	1	1	-2	1
$F(j+1,k)$	-3	0	0	0	0
$F(j+1,k+1)$	-2	-2	1	1	1

引进矢量概念，令 M_1、M_2、M_3、M_4 是图 4.8(b)所示的 4 个模板对应的 9 维矢量，对某一像素(j,k)，线模板的各个响应为 M_i^TF（F 为模板匹配的图像区域），$i=1,2,3,4$。对所有的 c 值，除 $c=i$ 外，有

$$M_i^TF > M_c^TF \tag{4.33}$$

就说 F 和第 i 个模板最相近。例如，若 M_1^TF 大于 M_cF，$c=2,3,4$，就可判定 F 有水平方向的性质。

3) 边缘模板

边缘与区域间变化相对应。提取边缘经常采用的方法之一是使用某种形式的算子，如 4.1.1 小节中介绍的基于 3×3 样本的 Laplacian 算子和基于 2×2 样本的梯度。现在把梯度的概念扩展到 3×3 子集，推出方便简捷的 Sobel 梯度模板(图 4.11)。

$$\begin{matrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{matrix} \qquad \begin{matrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{matrix}$$

(a)　　(b)

图 4.11　Sobel 梯度模板

(a)j 方向模板($h^1_{m,n}$)；(b)k 方向模板($h^2_{m,n}$)

参照图 4.11，有

$$G_j = \sum_{m=-1}^{1}\sum_{n=-1}^{1} h^1_{m,n}F(j+m,k+n)$$

$$= [F(j-1,k+1) + 2F(j,k+1) + F(j+1,k+1)] - [F(j-1,k-1+2F(j,k-1) + F(j+1,k-1)] \tag{4.34}$$

$$G_k = \sum_{m=-1}^{1}\sum_{n=-1}^{1} h^2_{m,n}F(j+m,k+n) = [F(j-1,k-1) + 2F(j-1,k) + F(j-1,k+1)] - [F(j+1,k-1) + 2F(j+1,k) + F(j+1,k+1)] \tag{4.35}$$

像点(j,k)处的边缘提取输出

$$g(j,k) = [G_j^2 + G_k^2]^{\frac{1}{2}} \tag{4.36}$$

或

$$g(j,k) = |G_j| + |G_k| \tag{4.37}$$

或

$$g(j,k) = \max(|G_j|,|G_k| \tag{4.38}$$

将图 4.11 和(4.34)、(4.35)式作比较，可以发现 G_j 为第一列和第三列的差，G_k 为第一行和第三行的差，其中，像点(j,k)的 4 邻点，即$(j,k+1)$、$(j,k-1)$、$(j-1,k)$、$(j+1,k)$，其位置上的权重等于角隅上像点的 2 倍。当 $G_j > G_k$ 时，预示着在(j,k)处在垂直方向的边缘通过，反之，则有水平方向的边缘通过。G_j，G_k 是 j，k 方向上微分算子的估计值。

上述过程也可用矢量表示，具体为

$$\boldsymbol{G}_j = \boldsymbol{M}_1^T\boldsymbol{F} \tag{4.39}$$

$$\boldsymbol{G}_k = \boldsymbol{M}_2^T\boldsymbol{F} \tag{4.40}$$

$$\boldsymbol{g} = [(\boldsymbol{M}_1^T\boldsymbol{F})^2 + (\boldsymbol{M}_2^T\boldsymbol{F})^2]^{\frac{1}{2}} \tag{4.41}$$

式中 $\boldsymbol{M}_1$、$\boldsymbol{M}_2$ 为图 4.11 中的两个模板矢量。

4.1.4 门限化

门限化就是将灰度根据主观愿望分成两个或多个等间隔或不等间隔灰度区间，用门限确定其区域或得到区域的边界。

最常见的形式是将灰度设置成二值状态，对图像 $F(j,k)$，其门限化输出

$$g(j,k) = \begin{cases} 1 & ; \quad F(j,k) \geqslant T \\ 0 & ; \quad \text{其他} \end{cases} \tag{4.42}$$

或

$$g(j,k) = \begin{cases} 0 & ; \quad F(j,k) \geqslant T \\ 1 & ; \quad \text{其他} \end{cases} \tag{4.43}$$

或

$$g(j,k) = \begin{cases} 1 & ; \quad T_1 \leqslant F(j,k) \leqslant T_2 \\ 0 & ; \quad \text{其他} \end{cases} \tag{4.44}$$

式中 T 和 T_i 为门限。

借助直方图可方便地确定物体图像与背景的灰度突变位置，并把门限选择在此处。图 4.12 是一幅灰度层次较丰富的物体(A_2 灰度区)叠加在暗背景(A_1 灰度区)上的图像的直方图，门限选择在背景与物体图像灰度交界处的物体图像灰度一侧。第一次扫描时，通过对图像中每一扫描行各列像素 $F(j,k)$ 与相邻行像素 $F(j,k-1)$ 的比较，按下式产生中间图像 $g_1(j,k)$ 中的一个相应的行：

$$g_1(j,k)=\begin{cases}L_e & ; \quad F(j,k)\text{与}F(j,k-1)\text{不同在}A_1\text{或}A_2\text{灰度区间内}\\ L_b & ; \quad \text{其他}\end{cases} \tag{4.45}$$

式中 L_e 和 L_b 分别为指定的边界和背景的灰度级。第二次扫描，对图像 $F(j,k)$ 的每一列，即 $\{(j,0),(j,1),(j,2),\cdots,(j,N-1)\}$，运用(4.46)式产生中间图像 $g_2(j,k)$ 中的一个相应的列：

$$g_2(j,k)=\begin{cases}L_e & ; \quad F(j,k)\text{与}F(j-1,k)\text{不同在}A_1\text{或}A_2\text{灰度区间内}\\ L_b & ; \quad \text{其他}\end{cases} \tag{4.46}$$

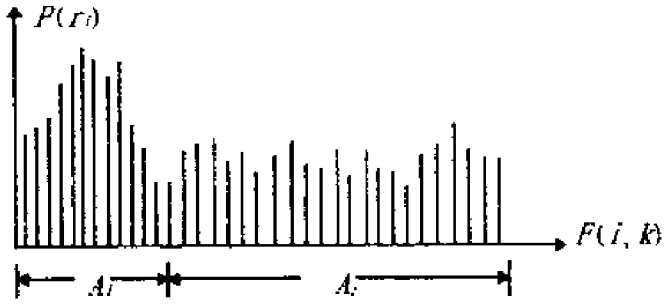


图 4.12 直方图及门限

最后输出

$$g(j,k)=\begin{cases}L_e & ; \quad g_1(j,k)=L_e\text{或}\\ & \quad g_2(j,k)=L_e\\ L_b & ; \quad \text{其他}\end{cases} \tag{4.47}$$

图 4.13(a)是一幅 12×16、背景灰度为 1、2，图像灰度为 3～7 的离散图像。现令灰度 1～2 为 A_1 区，灰度 3～7 为 A_2 区，$L_e=8$，$L_b=0$。L_e 为门限化后的边界灰度，L_b 为门限化后的背景。图 4.13(b)、(c)、(d)分别是第一次扫描、第二次扫描和合成后的门限化输出。由图 4.13(d)可以看出，图像边界被十分完整地检测出来。

1	1	1	1	1	2	2	1	1	1	1	1
1	1	1	1	1	1	2	1	1	1	1	1
1	1	1	1	2	2	3	3	1	1	1	1
2	2	1	1	1	3	3	3	3	2	2	1
2	2	1	2	3	4	5	3	4	3	1	2
1	2	2	4	4	5	6	5	6	5	3	1
1	1	5	5	6	7	6	5	3	4	4	4
1	2	4	4	5	5	5	4	5	5	4	3
1	2	2	3	4	5	6	7	7	6	3	2
1	1	2	4	4	4	3	6	7	6	1	2
2	1	1	1	5	5	6	6	6	2	1	1
1	1	2	2	1	3	4	4	3	2	2	1
1	1	1	1	1	1	3	5	1	1	1	1
1	1	1	2	2	1	1	4	1	1	1	1
1	1	1	1	2	2	1	1	1	1	1	1
1	1	1	1	1	2	2	1	1	1	1	1

(a)

1	1	1	1	1	2	2	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	8	8	0	0	0	0
0	0	0	0	0	8	0	0	8	0	0	0
0	0	0	0	8	0	0	0	0	8	0	0
0	0	0	8	0	0	0	0	0	0	8	0
0	0	8	0	0	0	0	0	0	0	0	8
0	0	8	0	0	0	0	0	0	0	0	0
0	0	8	0	0	0	0	0	0	0	0	8
0	0	0	0	0	0	0	0	0	0	8	0
0	0	0	8	0	0	0	0	0	8	0	0
0	0	0	0	8	0	0	0	0	0	0	0
0	0	0	0	0	8	0	0	8	0	0	0
0	0	0	0	0	0	8	0	0	0	0	0
0	0	0	0	0	0	0	8	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

(b)

图 4.13 门限化图像分割示例

1	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	8	0	8	0	0	0		0	0	0	0	0	0	8	8	8	0	0	0
2	0	0	0	0	8	0	0	0	8	0	0		0	0	0	0	0	8	0	0	8	8	0	0
2	0	0	0	8	0	0	0	0	0	8	0		0	0	0	0	8	0	0	0	0	8	8	0
1	0	0	8	0	0	0	0	0	0	0	8		0	0	0	8	0	0	0	0	0	0	8	8
1	0	8	0	0	0	0	0	0	0	0	0		0	0	8	0	0	0	0	0	0	0	0	8
1	0	8	0	0	0	0	0	0	0	0	0		0	0	8	0	0	0	0	0	0	0	0	0
1	0	0	0	8	0	0	0	0	0	0	8		0	0	8	8	0	0	0	0	0	0	0	8
1	0	0	8	0	0	0	0	0	0	8	0		0	0	0	8	0	0	0	0	0	0	8	0
2	0	0	0	8	0	0	0	0	8	0	0		0	0	0	8	8	0	0	0	0	8	0	0
1	0	0	0	0	8	0	0	0	8	0	0		0	0	0	0	8	8	0	0	0	8	0	0
1	0	0	0	0	0	8	0	8	0	0	0		0	0	0	0	0	8	8	8	8	0	0	0
1	0	0	0	0	0	0	8	8	0	0	0		0	0	0	0	0	0	8	8	8	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	8	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0

(c) (d)

续图 4.13 门限化图像分割示例

(a)原始图像； (b)第一次扫描； (c)第二次扫描； (d)合成输出

上述方法可以方便地推广到两个以上门限的场合。在门限适当的前提下，能够分割出更丰富的图像结构，表现出更强的边缘提取能力。

选择最佳门限是门限化图像分割的首要工作。利用直方图选择门限是常用手段之一。当图像灰度直方图峰型分布明显时，常以谷底作为门限候选值。

令图像由物体和背景两部分组成，背景与物体所对应的灰度级概率分布密度分别为 $P_1(F)$、$P_2(F)$，面积比分别为 P_1、$P_2(P_1+P_2=1)$。其混合概率分布密度函数

$$P(F)=P_1P_1(F)+P_2P_2(F) \tag{4.48}$$

在高斯情况下

$$P(F)=\frac{P_1}{\sqrt{2\pi}\sigma_1}\exp\left[-\frac{(F-\mu_1)^2}{2\sigma_1^2}\right]+\frac{P_2}{\sqrt{2\pi}\sigma_2}\exp\left[-\frac{(F-\mu_2)^2}{2\sigma_2^2}\right] \tag{4.49}$$

式中 μ_1,μ_2 分别是背景和物体的均值； σ_1,σ_2 为图绕均值的标准偏差。

假定暗区相当于背景，亮区相当于物体。在这种情况下，$\mu_1<\mu_2$，凡灰度级低于门限 T 的像素均作为背景来对待，凡灰度级高于门限 T 的像素均视作构成物体的点。至此，把物体上的点错分类成为背景上的点的概率为

$$E_1(T)=\int_{-\infty}^{T}P_2(F)\mathrm{d}F \tag{4.50}$$

同样，把背景上的点错划为物体上的点的概率为

$$E_2(T)=\int_{T}^{\infty}P_1(F)\mathrm{d}F \tag{4.51}$$

总的误分概率

$$E(T)=P_2E_1(T)+P_1E_2(T) \tag{4.52}$$

为了获得最佳门限，将(4.52)式中的 $E(T)$ 对 T 求微分，并使其结果为零。从而有

$$P_1P_1(T)=P_2P_2(T) \tag{4.53}$$

A_1，产生新的核 A_2。重复上述过程，直到没有可并入的点时，生长过程结束。这里的接收准则可以是一维的，如灰度幅度特性，可设置邻近像素与 A_i 灰度均值之差作为门限 T，凡灰度小于该门限的邻点均认为是区域附属点而并入。接收准则也可以是二维的。

图 4.14 是一个简单连接区域生长的例子。连接核为 9(图 4.14(a)中有下划线的点)，接收准则是可并入的点的灰度与连接核灰度均值的差 $\Delta F < 2$。第一次区域生长有 $F(j-1,k)$、$F(j,k-1)$、$F(j,k+1)$3 个邻点被连接核吸纳，形成连接核 A_1(图 4.14(b)中有下划线的区域)，其灰度均值 $F_1 = 8.25$。第二次向外生长，A_1 俘获到邻点 $F(j+1,k)$、$F(j,k+2)$，组成新的连接核 A_2(图 4.14(c)中有下划线的区域)，其灰度均值 $F_2 = 7.83$。第三次生长，捕捉到点 $F(j+1,k+1)$，形成连接核 A_3(图 4.14(d)中有下划线的区域)，灰度均值 $F_3 = 6.29$。再一次生长时，不存在满足接收准则的点，生长过程终止。

5	5	8	5	4	5	5	8	5	4	5	5	8	5	4	5	5	8	5	4
4	8	9	7	4	4	8	9	7	4	4	8	9	7	4	4	8	9	7	4
2	2	8	6	4	2	2	8	6	4	2	2	8	6	4	2	2	8	6	4
3	3	7	4	4	3	3	7	4	4	3	3	7	4	4	3	3	7	4	4
4	2	4	4	3	4	2	4	4	3	4	2	4	4	3	4	2	4	4	3
		(a)					(b)					(c)					(d)		

图 4.14　简单连接区域生长示例

2) 子区合并法

子区合并法以小区域内的灰度分布的相似性来进行区域合并。其过程是：

(1) 将图像分割成 $M \times N$ 个、大小为 $L_x \times L_y$ 的小区域(简称子区)。

(2) 从左上角第一个子区开始，分别计算子区和相邻子区的灰度统计量(直方图)，然后作相似性判定。若两者的灰度分布相似且符合接收准则，相邻子区并入当前子区，形成下一轮判定合并时的当前子区(因有相邻子区并入，所以当前子区的面积是不断增大的)。如果某个相邻子区不符合接收准则，就打上“未分割标记”。

继续新的一轮判定，使当前子区不断“生长”，直到没有可归并的子区为止，一个区域分割完毕。

(3) 搜索图像全域，对凡具有“未分割标记”的子区重复上述操作。

步骤(2)中进行相似性判定的常用判据是：

$$\left.\begin{aligned} N &= \max \mid h_1(F) - h_2(F) \mid \\ N &< T_1 \end{aligned}\right\} \tag{4.59}$$

$$\left.\begin{aligned} N &= \sum \mid h_1(F) - h_2(F) \mid \\ N &< T_2 \end{aligned}\right\} \tag{4.60}$$

(4.59)、(4.60)式分别称为凯蒙高罗夫 - 史密诺夫(Kolmogorov - Sminov)判定和平滑差(Smoothed difference)判定。当 $N < T_1$ 或 $N < T_2$ 时，相邻子区被并入。式中，$h_1(F)$和$h_2(F)$分别是当前子区和相邻子区的灰度分布直方图，T_1、T_2 为门限。

4.2.2　特征空间分类法

将二维图像射映到特征空间，根据某种分类方法(如群聚法)分析特征空间的像素并分

类，再对一个类像素群中的各像素冠以同一编号，最后再反变换到二维平面，这就是特征空间分类的基本思路。图 4.15 是这方面的一个示例，8×8 图像域，像素具有 x_i, y_i, z_i 三个特征，映射到特征空间后，被分成 3 类，标号 1,2,3。反变换到图像平面后，被分割成 3 个区域。

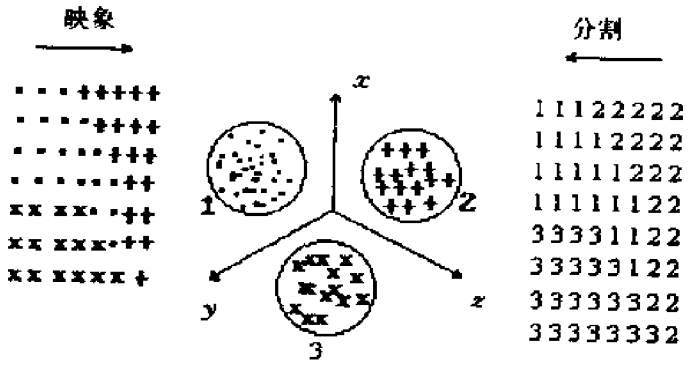


图 4.15　特征空间分类法分割图像示例

5 图像编码与压缩

●预测编码 ●正交变换编码 ●统计编码
●轮廓编码 ●二值图像编码 ●多媒体图像编码

近年来,随着计算机与数字通信技术的迅速发展,特别是网络和多媒体技术的兴起,图像编码与压缩已受到了人们越来越多的关注。

图像编码与压缩从本质上来说就是对要处理的图像源数据按一定的规则进行变换和组合,从而达到以尽可能少的代码(符号)来表示尽可能多的数据信息。压缩通过编码来实现,或者说编码带来压缩的效果。所以,一般把此项处理称之为压缩编码。

5.1 概述

这里主要对图像编码压缩的必要性、可能性及分类等作一简要讨论。

5.1.1 图像编码压缩的必要性

图像数据的特点之一是信息量大。一幅 512×512、灰度等级为 8 比特的图像,其数据量为 256K 字节。以四通道卫星遥感探测器为例,以此视窗,一个时次即达 1M 字节。实际上,卫星遥感探测时,视窗远大于此,通道远多于此。如地球同步卫星,探测范围达全球面积的三分之一。试想,要保存这些图像数据,该需要多大的存储空间。

在多媒体中,海量图像数据的存储和处理是难点之一。根据计算,一张 600M 字节的光盘,仅能存放 20 秒左右的 640×480 像素的图像画面信息。无疑,如不进行编码压缩处理,那在多媒体信息保存工作中所遇到的困难和成本之高是可想而知的。

在现代通信中,图像传输已成为重要内容。在工作中除要求设备可靠、图像保真度高以外,实时性将是重要技术指标之一。很显然,在信道带宽、通信链路容量一定的前提下,采用编码压缩技术,减少传输数据量,是提高通信速度的重要手段。

可以这样认为,没有编码压缩技术的发展,大容量图像信息的存储与传输是难以实现的,多媒体等新技术在实际中的应用也会碰到困难。

5.1.2 图像编码压缩的可能性

数据压缩技术最初始于信息论。80 年代前,研究的主要内容是有关信息熵、编码方法及数据压缩比,即通过某种算法,对源数据进行编码,使编码后的数据流长度小于源数据流长度,从而达到减少存储空间和提高信息传输速度的目的。其中,对图像压缩的研究还只限于静止图像。进入 80 年代后,计算机科学、数字信号处理、数字通信、多媒体技术迅猛发展,

图像编码的研究由静止图像扩展到运动图像。有关国际组织相继定义了一些算法和压缩标准，如由国际标准化组织(International Standards Organization，简写为 ISO)和国际电报电话协商委员会(Consultative Committee for International Telegraph and Telephone，简写为 CCITT)组织的联合影像专家小组，就提出了面向连续色调静止图像的 DCT(Descrete Cosine Transform)有失真压缩算法和 DPCM(Differential Pulse Code Modulation)无失真编码算法，进一步推动了图像数据处理和编码压缩工作。

众所周知，组成图像的各像素之间，无论是在行方向还是在列方向，都存在着一定的相关性。例如，背景常取同样的灰度，某种特征中像素灰度相同或相近。应用某种编码方法提取或减少这些相关性，便可达到压缩数据的目的。例如，30 个灰度级均为 128 的连续像素，压缩前需要 30 个字节的存储空间。若采用"起始位置，连续个数，灰度级"格式的编码方法，并假定像素坐标、连续像素个数变量占 3 个字节，则压缩后数据流长度为 33 比特，压缩比达到 86.3%。

从信息论观点看，描述图像信源的数据是由有效信息量和冗余量两部分组成的，去除冗余量能够节省存储和传输中的开销，同时又不损害图像信源的有效信息量。

在有些场合，一定限度的失真是允许的。如人的眼睛对图像灰度分辨的局限性，监视器显示分辨率的限制，都可以对图像信源作一定程度有时甚至是很大程度的压缩。

5.1.3 图像编码压缩的分类

图像编码压缩的方法目前已有多种，其分类方法视出发点不同而有差异。常见的分类情况是：

(1) 冗余度压缩法。该方法的核心是基于统计模型，减少或完全去除源数据流中的冗余，同时保持信息不变。如把图像数据中出现概率大的灰度级以短的代码表示，概率小的灰度级用相对长的代码表示，处理的平均码长必然短于未编码压缩前的平均码长。在解码过程中，可以根据相对应的规则或算法，将冗余量插入到图像数据中去，严格恢复原 图像，实现编码与解码的互逆。因此，冗余编码压缩又称之为无损压缩或无失真压缩。著名的哈夫曼(Huffman)编码、香农(Shannon)编码就属于这一类。

(2) 熵压缩法。这是一种以牺牲部分信息量为代价而换取缩短平均码长的编码压缩方法。由于其在压缩过程中在允许的前提下丢失了一部分信息，所以图像还原后与压缩前不会完全一致，故人们将这种压缩谓之有损压缩。

以上分类是以信息保真为出发点的。若以具体编码技术来考虑，又可分为预测编码、变换编码、统计编码、轮廓编码、模型编码等。

5.1.4 图像编码压缩术语简介

1) 图像熵与平均码字长度

令图像像素灰度级集合为$\{d_1, d_2, \cdots, d_m\}$，其对应的概率分别为 $p(d_1), p(d_2), \cdots, p(d_m)$，图像熵定义为

$$H(d) = -\sum_{i=1}^{m} p(d_i)\log_2 p(d_i) \tag{5.1}$$

单位为比特/字符。图像熵表示图像灰度级集合的比特数均值，或者说描述了图像信源的平均信息量。

如果令 $m=2^L$，当灰度级集合 $\{d_1, d_2, \cdots, d_m\}$ 中 d_i 出现的概率相等（均为 2^{-L}）时，熵 $H(d)$ 最大，等于 L 比特。只有当 d_i 不等时，$H(d)$ 才会小于 L。

借助于熵的概念可以定义量度任何特定码的性能的准则：平均码长度 $R(d)$，

$$R(d) = \sum_{i=1}^{m} \beta_i p(d_i) \tag{5.2}$$

式中，β_i 为灰度级 d_i 所对应的码字的长度。显然，$R(d)$ 的单位也是比特/字符。

2) 编码效率

编码效率常用下式表示：

$$\eta = \frac{H(d)}{R(d)}(\%) \tag{5.3}$$

如果 $R(d)$ 与 $H(d)$ 相等，编码效果最佳；$R(d)$ 接近 $H(d)$，编码效果为佳；$R(d)$ 远大于 $H(d)$，编码效果差。

3) 压缩比

压缩比是衡量数据压缩程度的指标之一。(5.4)式是目前常用的压缩比定义式：

$$P_r = \frac{L_s - L_d}{L_s} \times 100\% \tag{5.4}$$

式中 L_s 为源代码长度；

L_d 为压缩后代码长度；

P_r 为压缩比（百分数）。

压缩比的物理意义是被压缩掉的数据占源数据的百分比。当压缩比 P_r 接近 100% 时，压缩效果最理想。

5.2 预测编码

预测编码就是根据过去时刻的样本序列，运用一种模型，预测当前的样本值。

预测编码通常不直接对信号编码，而是对预测误差编码。当预测比较准确，误差较小时，即可达到编码压缩的目的。这种编码称之为差分脉冲编码调制（DPCM），图 5.1 是原理

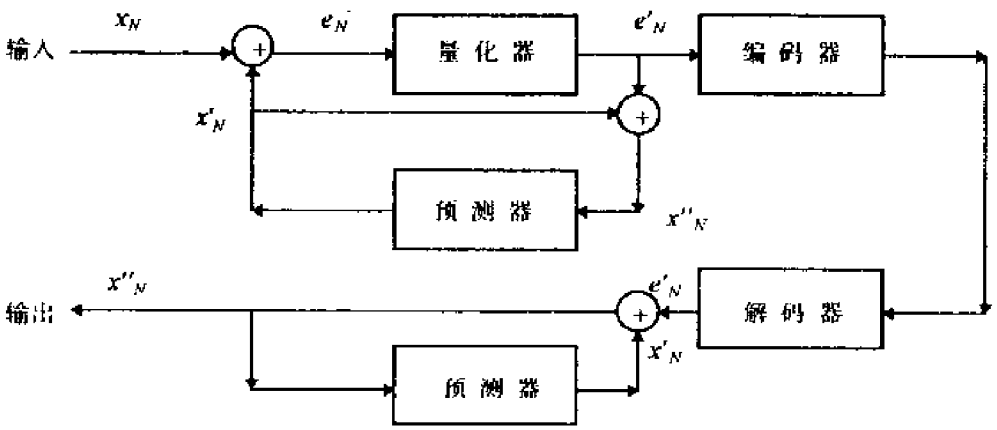


图 5.1 DPCM 原理框图

框图。在该系统中，x_N 为 t_N 时刻的亮度取样值。预测器根据 t_N 时刻之前的样本值 x_1、x_2，

$\cdots, x_{N-1}$对 x_N 作预测，得到预测值 x'_N。x_N 与 x'_N 之间的误差为

$$e_N = x_N - x'_N \tag{5.5}$$

量化器对 e_N 进行量化得到 e'_N。编码器对 e'_N 进行编码发送。收端解码时的预测过程与发端相同，所用预测器亦相同。收端恢复的输出信号 x''_N 是 x_N 的近似值，两者的误差是

$$\Delta x_N = x_N - (x'_N - e'_N) = x_N - x''_N = e_N - e'_N \tag{5.6}$$

当 Δx_N 足够小时，输入信号 x_N 和 DPCM 系统的输出信号 x''_N 几乎一致。预测编码分为线性预测和非线性预测两类，下面分别予以介绍。

5.2.1 线性预测编码

在排序后的图像序列$\{x_i\}(i=1,2,\cdots,N-1)$中，据 $x_1, x_2, \cdots, x_{N-1}$对 x_N 作预测。令 x_N 的预测估计值为 x'_N，假如 x'_N 是$x_1, x_2, \cdots, x_{N-1}$的线性组合，则称对 x_N 的预测为线性预测。

假定二维图像信号 $x(t)$是一个均值为零、方差为 σ^2 的平稳随机过程，$x(t)$在 $t_1, t_2, \cdots, t_{N-1}$时刻的抽样值集合为

$$\{x\} = x_1, x_2, \cdots, x_{N-1} \tag{5.7}$$

由上式可得到 t_N 时刻抽样值的预测值

$$x'_N = \sum_{i=1}^{N-1} a_i x_i = a_1 x_1 + a_2 x_2 + \cdots + a_{N-1} x_{N-1} \tag{5.8}$$

式中 a_i 为预测系数。

根据线性预测定义，x'_N 必须十分逼近 x_N，这就要求 $a_1, a_2, \cdots, a_{N-1}$为最佳系数。采用均方误差最小的准则，可得到最佳的 a_i。

设 x_N 的均方误差为

$$\begin{aligned} E\{[e_N]^2\} &= E\{[x_N - x'_N]^2\} \\ &= E\{[x_N - (a_1 x_1 + a_2 x_2 + \cdots + a_{N-1} x_{N-1})]^2\} \end{aligned} \tag{5.9}$$

为使 $E\{[e_N]^2\}$最小，在(5.9)式中对 a_i 求微分，即

$$\begin{aligned} \frac{\partial}{\partial a_i} E\{[e_N]^2\} &= \frac{\partial}{\partial a_i} E\{[x_N - (a_1 x_1 + a_2 x_2 + \cdots + a_{N-1} x_{N-1})]^2\} \\ &= -2E\{[x_N - (a_1 x_1 + a_2 x_2 + \cdots + a_{N-1} x_{N-1})] x_i\} \\ & \qquad i = 1, 2, \cdots, N-1 \end{aligned}$$

根据极值定义，得到$(N-1)$个方程组成的方程组：

$$\left.\begin{aligned} &E\{[x_N - (a_1 x_1 + a_2 x_2 + \cdots + a_{N-1} x_{N-1})] x_1\} = 0 \\ &E\{[x_N - (a_1 x_1 + a_2 x_2 + \cdots + a_{N-1} x_{N-1})] x_2\} = 0 \\ &\qquad \cdots \\ &E\{[x_N - (a_1 x_1 + a_2 x_2 + \cdots + a_{N-1} x_{N-1})] x_{N-1}\} = 0 \end{aligned}\right\} \tag{5.10}$$

简记为

$$\begin{aligned} E\{[x_N - (a_1 x_1 + a_2 x_2 + \cdots + a_{N-1} x_{N-1})] x_i\} &= 0 \\ i &= 1, 2, \cdots, N-1 \end{aligned} \tag{5.11}$$

假设 x_i 和 x_j 的协方差为

$$R_{ij} = E\{x_i, x_j\}; \quad i, j = 1, 2, \cdots, N-1 \tag{5.12}$$

则(5.11)式可表示成

$$R_{Ni} = a_1 R_{1i} + a_2 R_{2i} + \cdots + a_{N-1} R_{(N-1)i}$$
$$i = 1,2,\cdots,N-1 \tag{5.13}$$

若所有的协方差 R_{ij} 已知，则在特定的算法下，$(N-1)$ 个预测系数 a_i 即可解得。

在线性预测编码中，若只采用 x_{N-1} 对 x_N 进行预测，称为前值预测。若采用同一行中 x_N 前的若干抽样值来对 x_N 预测，称为一维预测。若采用几行内的抽样值来预测 x_N，称为二维预测。

根据前面的定义，这里讨论的图像是平稳的随机过程，因此可以用自相关系数代替 R_{ij}，以求得预测系数。

DPCM 预测是一种近似处理系统，因为不同图像的自相关系数是不尽相同的，这一幅图像适用的模型和系数，对另一幅图像不一定适用。在实际工作中，常采用针对不同情况，采用相对固定的 a_i 系数的方法。如对于隔行扫描的电视图像，一般采用以下简化预测公式：

$$F(j,k) = \frac{1}{2}F(j,k-1) + \frac{1}{4}F(j-1,k) + \frac{1}{8}F(j-1,k-1) + \frac{1}{8}F(j-1,k+1) \tag{5.14}$$

这是常用的二维预测模型。由于所使用的样本像素相对于预测像素之间的距离为

$$S[F(j-1,k-1)]、S[F(j-1,k+1)] > S[F(j-1,k)] > S[F(j,k-1)]$$

所以预测系数分别选用 $\frac{1}{2}$、$\frac{1}{4}$、$\frac{1}{8}$、$\frac{1}{8}$。

5.2.2 非线性预测编码

线性预测编码的基础是假设图像全域为平稳的随机过程，自相关系数与像素在域中的位置无关。实际上，图像的起伏始终是存在的，被描述像素和周围像素之间，含有多种多样的关系。线性预测系数 a_i 是一种近似条件下的常数，其忽略了像素的个性特性。

非线性预测充分考虑了图像的统计特性和个别变化，尽量使预测系数与图像所处的局部特性相匹配，即预测系数随预测环境而变，从而获得更高的压缩比。

现在讨论非线性预测的基本原理和过程。

设图像信号为 m 阶马尔可夫(Markov)序列(见图 5.2)，当前像素 s_0 的邻近像素 $s_i(i=1,2,\cdots,n)$按其与 s_0 之间的距离排列。由前面对非线性预测的解释可知，s_0 的预测值 s'_0 与 s_i 有关，即像素 s_0 的预测值 s'_0 可用其邻近像素 s_i 的联合概率密度来表示：

$$s'_0 = P(s_1,s_2,\cdots,s_n) \tag{5.15}$$

$$f(x'_0) = \int (s_0 - s'_0)^2 p(s_0/s_1,s_2,\cdots,s_n)\mathrm{d}s_0 \tag{5.16}$$

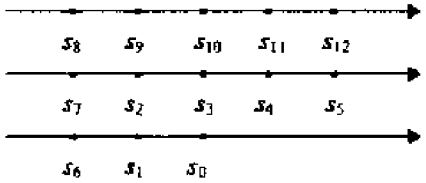


图 5.2 s_0 及其邻近像素

$P(s_0/s_1,s_2,\cdots,s_n)$为 s_0 在$(s_1,s_2,\cdots,s_n)$状态下发生的条件概率密度，则全帧图像非线性预

测误差$(s_0 - s'_0)$的均方差为

$$E\{e^2\} = \iint \cdots \int f(x'_0) p(s_0, s_1, s_2, \cdots, s_n) \mathrm{d}s_0 \, \mathrm{d}s_1 \, \mathrm{d}s_2 \cdots \mathrm{d}s_n \tag{5.17}$$

令

$$\frac{\partial f(s'_0)}{\partial s'_0} = 0$$

$$s'_0 = \int s_0 p(s_0/s_1, s_2, \cdots, s_n) \mathrm{d}s_0 \tag{5.18}$$

这里，s'_0 是 $p(s_0/s_1, s_2, \cdots, s_n)$条件概率下 s_0 的均值。在此条件下可获得最佳预测值。

改写(5.18)式，有

$$s'_0 = \sum_{i=1}^{m} (s_i + d_i)\lambda_i \tag{5.19}$$

式中　$d_i = s_0 - s_i$；　λ_i 为权重(预测)系数。

对(5.19)式稍作分析，d_i 越小，权重系数 λ_i 取值应越大。反之亦然。可见，非线性预测体现了相邻像素的制约作用，因为 d_i, λ_i 都是 $s_1, s_2, \cdots, s_n$ 的函数。图像的局部特性达到了很好的匹配。

知道了 d_i, λ_i，就不难得到预测值 s'_0。目前常用的方法是：

(1) 确定 d_i

假定 $m = 5$，则 d_i 的理论值

$$d_1 = s_0 - s_1 \qquad d_2 = s_0 - s_2 \qquad d_3 = s_0 - s_3$$

$$d_4 = s_0 - s_4 \qquad d_5 = s_0 - s_5$$

d_i 对应的一阶估测值为

$$d_1 = \alpha(s_3 - s_2) \qquad d_2 = \alpha(s_1 - s_7) \qquad d_3 = \alpha(s_1 - s_2)$$

$$d_4 = \alpha(s_1 - s_3) \qquad d_5 = \alpha(s_1 - s_4)$$

α 为估测系数，一般采用 $\alpha = 0.5$。对照图 5.2 可以看出，d_i 被定义为最近平行的像素之差和 α 系数的积。

(2) 选择权重系数 λ_i

λ_i 与 d_i 密切相关。d_i 越小，s_i 与 s_0 越接近，相关性越强，λ_i 越大。λ_i 的选择要视 d_i 的分布方可进行。定义

$$\lambda_i = \frac{1}{Q} \cdot \frac{1}{D_i} \tag{5.20}$$

$$Q = \sum_{i=1}^{m} \frac{1}{D_i} \tag{5.21}$$

$$D_i = \sum_{k=1}^{N} | s_{ik} - s_{0k} | \tag{5.22}$$

以上三式中　$i = 1, 2, \cdots, m$；

Q 为归一化函数；

N 为预测阶数；

$|s_{ik} - s_{0k}|$为与$|s_i - s_0|$最近平行的像素之差的绝对值。

请注意，在一阶情况下，D_i 用 d_i 代替。在阶数 $k \geqslant 2$ 时，按(5.22)式计算 D_i。如 $k = 2$，$m = 5$时，对照图 5.2，得

$$D_1 = |s_1 - s_6| + |s_3 - s_2| \qquad D_2 = |s_1 - s_7| + |s_3 - s_9|$$
$$D_3 = |s_1 - s_2| + |s_3 - s_{10}| \qquad D_4 = |s_1 - s_3| + |s_4 - s_{12}|$$
$$D_5 = |s_1 - s_4| + |s_4 - s_{12}|$$

(3) 计算 s'_0

$$s'_0 = \sum_{i=1}^{m}(s_i + d_i)\lambda_i \tag{5.23}$$

在计算机模拟中,经常按下式作非线性预测:

$$s'_0 = (s_1 + \frac{s_3 - s_2}{2})\lambda_1 + (s_2 + \frac{s_1 - s_7}{2})\lambda_2 + (s_3 + \frac{s_1 - s_2}{2})\lambda_3 + (s_4 + \frac{s_1 - s_3}{2})\lambda_4 + s_1\lambda_5 + \frac{s_1 + s_4}{2}\lambda_6 + 0.5 \tag{5.24}$$

式中

$$\lambda_1 = (|s_1 - s_6| + |s_3 - s_2| + 0.1)^{-1}$$
$$\lambda_2 = (|s_1 - s_7| + |s_3 - s_9| + 0.1)^{-1}$$
$$\lambda_3 = (|s_1 - s_2| + |s_3 - s_{10}| + 0.1)^{-1}$$
$$\lambda_4 = (|s_1 - s_3| + |s_4 - s_{12}| + 0.1)^{-1}$$
$$\lambda_5 = (|s_1 - s_3| + |s_1 - s_2| + 0.1)^{-1}$$
$$\lambda_6 = (2|s_1 - s_4| + 0.1)^{-1}$$

5.3 正交变换编码

关于正交变换,在本篇第2章中已有比较深入的讨论。从理论上说,正交变换本身不能对图像数据进行有效的压缩。但正交变换改变了图像数据的表现域或表现形式,为编码压缩提供了可能。如傅立叶变换,将图像数据从空间域变换到了频域,原来彼此相关的图像阵列变成了基本独立的变换系数矩阵,利用频率域中低频分量与图像基本成分相对应的关系,即可作编码压缩。

5.3.1 变换编码的基本原理

变换编码就是对正交变换得到的系数矩阵进行量化编码。

图5.3是变换编码系统的组成框图。整个变换系统由三部分组成,即图像输入与变换、编码、逆变换。在变换阶段,系统将原图分成若干子块,对每个子块进行某种正交变换。通过变换,降低或消除相邻像素之间或相邻扫描行之间的相关性,提供用于编码压缩的变换系数矩阵。编码过程实现图像信息的压缩。统计表明,在变换域中,图像信号的绝大部分能量集中在低频部分,编码中如果略去那些能量很小的高频分量,或者给这些高频分量分配合适的比特数,就可明显减少图像传输或存储的

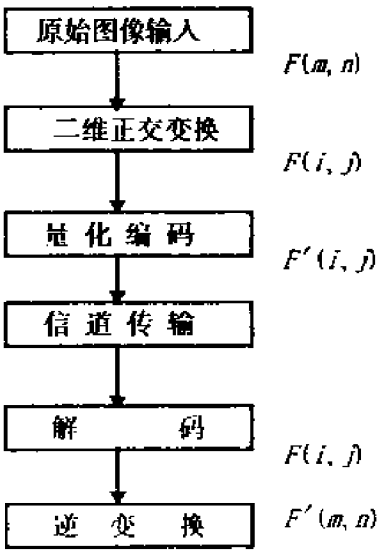


图5.3 正交变换编码系统框图

数据量。

5.3.2 正交变换编码特性评价

在具体讨论变换编码方法之前，先介绍几种评价变换编码特性的准则。

1) 残余相关准则

变换域内变换系数具有的相关性称之为残余相关性，它代表经过正交变换后帧内图像相关性被削弱的程度。显然，如果变换方法及有关参数选择恰当，则变换系数矩阵所对应的相关系数会很小，原图像的相关性得到了充分的消除，冗余度得到了很好的压缩。

在正交变换后，通过计算变换系数的相关系数，可从一个方面评价变换编码的特性。

2) 均方差准则

图像 $F(m,n)$经过变换后，编码单元对变换系数编码，接收端对 $F'(i,j)$作逆变换，获得复原图像 $F'(m,n)$。前面已经谈到，正交变换编码是一种有失真的编码技术，$F'(m,n)$与 $F(m,n)$之间有误差。我们规定以均方误差作为准则，在允许的失真度下，实现有效的编码压缩。

3) 主观评价准则

人眼对图像失真的感受和均方误差并不完全一致。所谓主观评价，就是以人的眼睛感觉出来的优劣作为评价指标。

实践表明，人眼对正交变换编码后的复原图像的失真感觉比较灵敏，特别是对把图像分割成若干子域进行变换编码后形成的"方块效应"更是敏感。

5.3.3 变换编码

从系统的角度看，图像变换编码的一般过程是：

(1) 确定图像矩阵阶数；

(2) 选择正交变换方法；

(3) 正交变换；

(4) 对变换系数矩阵编码；

(5) 信道传输；

(6) 逆变换，图像复原。

步骤(1)～(3)产生图像变换域，其中，选择哪种变换方法，将视存储容量、图像质量、硬件条件等因素综合考虑。步骤(4)实现图像压缩。

在正交变换中，往往要将一帧图像划分成若干正方形的图像子块来进行。子块越小，计算量越小，实现时硬件装置的规模也越小。不利的是均方误差较大，在同样的允许失真度下，压缩比小。因此，从改善图像质量考虑，适当加大图像子块是明智的举措。但这并不意味子块可以任意地大(极端情况子块等于图像全帧)，因为硬件的复杂程度与规模和子块大小成正比。同时，图像像素 $F(m,n)$与周围像素之间的相关性存在一定距离之内。也就是说，当子模块足够大以后，再加大子块，则加进来的像素与中心像素之间的相关性甚小，甚至不相关，而计算的复杂性将显著加大。目前，变换中图像子块一般取 8×8 或 16×16。

对变换系数矩阵编码时，比特数的分配可采用等长码与不等长码等方式。所谓等长码，就是先对变换系数作归一化处理，然后用统一的编码器对归一化变换系数编码，分配相同的

比特。不等长编码就是对不同变换系数赋予不同的比特。

下面介绍三种常用的正交变换编码方法。

1) 区域编码

图 5.4(a)是 16 × 16、6 比特图像数据阵列，图 5.4(b)是原点在中心、与图 5.4(a)相对应的傅立叶变换域矩阵。由该图可见，在傅立叶变换域中，大部分区域谱能量为零，谱能量主要集中在中心附近的少数列率谱上。

62	62	62	61	60	55	54	50	41	35	26	20	16	14	12	14
52	62	62	61	60	55	54	50	41	35	26	20	16	14	12	14
63	63	62	61	59	56	54	40	30	22	19	15	14	13	12	11
62	62	60	55	51	45	35	27	22	19	16	15	12	12	13	12
39	37	34	30	22	17	15	13	12	12	12	13	16	23	30	39
56	46	42	31	21	16	14	13	13	13	13	15	20	26	36	41
57	50	40	32	20	15	13	13	13	15	18	24	32	40	42	44
55	50	36	24	16	14	12	13	15	17	22	27	37	42	43	44
55	50	35	25	16	14	12	13	15	20	25	33	38	42	44	45
46	35	24	16	15	13	13	16	21	30	36	40	42	44	45	45
34	25	19	16	15	14	14	20	29	37	40	42	43	43	43	42
25	23	20	19	16	15	17	25	36	41	42	42	42	42	42	41
30	28	26	25	23	16	16	25	35	41	42	42	42	42	42	42
41	40	32	30	26	20	18	21	33	40	42	42	42	42	42	42
40	36	33	30	26	18	20	28	38	40	42	42	42	42	42	41
52	48	36	33	28	20	23	34	40	41	41	41	41	41	41	41
56	50	40	35	33	23	26	35	39	40	40	40	40	40	40	41

(a)

0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	2	1	1	0	0	0	0	0	0
0	0	0	0	0	1	0	2	2	2	0	0	0	0	0	0
0	0	0	1	0	1	1	6	3	2	0	0	0	0	0	0
0	0	0	1	1	1	2	4	32	4	2	1	1	1	0	0
0	0	0	0	1	1	0	2	3	6	1	1	0	1	0	0
0	0	0	0	0	0	0	2	2	2	0	0	0	0	0	0
0	0	0	0	0	0	0	1	1	2	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	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	0	0	0	0	1	0	1	0	0	0	0	0	0

(b)

图 5.4　二维傅立叶变换频率谱分布

(a)原始图像数据；　(b)傅立叶变换频谱分布

针对变换系数矩阵的这一特点，人们自然提出了如下设想：选择能量集中的区域进行编码，舍弃能量为零和零星能量区域，从而达到数据压缩的目的，这就是区域编码的含义。

2) 阈值编码

区域编码是对设定区域内的变换系数进行编码，而略去区域外的部分。该方法的压缩效率与编码区域的选定有一定关联。在原始图像事先经过滤波、滤波器的频率特性已知的情况下，运用此方法还是比较适宜的。如果对变换域内的能量分布不十分清楚，或者说希望对变换 域内的所有变换系数作某种规则的编码，则区域编码就显得力不从心。于是有人提出了不按区域，而是按变换系数的幅度进行编码的方法，这就是阈值编码。

阈值编码将变换系数与门限值相比较，大于门限的给予编码，否则舍弃。

图 5.5(a)为 8×8 图像子块的灰度分布，经沃尔什变换后，变换系数分布如图 5.5(b)示。假定表示像素位置的行号、列号均以 4 比特表示，阈值大于零，变换系数统一以 7 比特编码，则对于图 5.5(b)来说，编码输出总码长为 45 比特，具体是 0000 0000 1000000 0010 0010 0100000 0111 0111 0100000B。

1	0	1	0	1	0	1	0
0	1	0	1	0	1	0	1
1	0	3	2	3	2	1	0
0	1	2	3	2	3	0	1
1	0	3	2	3	2	1	0
0	0	2	3	2	3	0	1
1	0	1	0	1	0	1	0
0	1	0	1	0	1	0	1

(a)

64	0	−32	0	0	0	0	0
0	0	0	0	0	0	0	0
−32	0	32	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	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

(b)

图 5.5 阈值编码示例

(a)原始图像灰度分布；(b)沃尔什变换域

3) 混合编码

如前所述，预测编码利用邻近像素的灰度相关性对 t_N 时刻的像素 x_N 进行预测，并对其预测误差进行编码。

预测编码实现简单，但抗误码能力较差。变换编码利用正交变换 域中能量分布比较集中这一特性，在约定规则下对部分变换系数进行编码。变换编码抗误码能力强，但实现起来比较复杂。

为了适应图像信息千变万化这一现实和充分利用不同编码方法的优点以进一步提高编码可靠性及编码效率，哈比比(Habibi)在 1974 年提出了混合编码的问题。混合编码就是集预测编码和变换编码于一个系统中。以一帧图像为例，可以先对图像在行的方向上作一维变换编码，然后将变换后的系数阵列在列方向上作预测编码。图 5.6 是其原理框图。令原图像阵列为 $F(m,n)$，其中 $m,n=1,2,\cdots,N$。在行方向对某行作一维变换编码后，该行序列输出为 $f(m,1),f(m,2),\cdots,f(m,N)$。接着对 $f(m,n)$在列方向进行 DPCM 编码，某列 DPCM 输出为 $e(1,n),e(2,n)\cdots,e(N,n)$。略去信道传输的影响，解码器从接收信号中检出 $f(m,n)$，再经逆变换，获得复原图像 $F'(m,n)$。

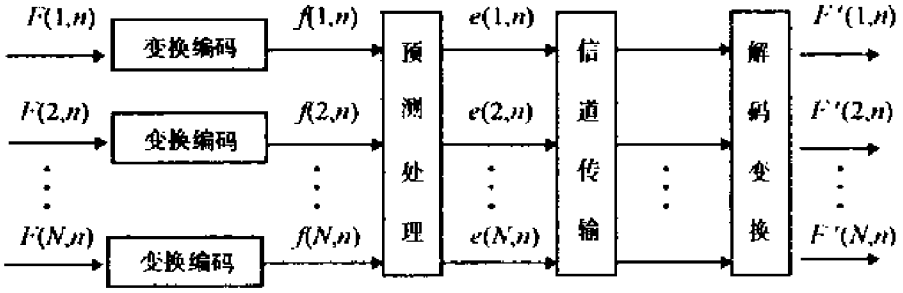


图 5.6 帧内混合编码原理框图

5.4 统计编码

前面介绍的预测编码和变换编码，成功地运用了削弱图像信源相关性和最小误差原理，减少存储和传输的比特数，从而达到数据压缩的目的。

统计编码则是根据信源的概率分布特性，分配可变长码字(其具有唯一可译性)，降低平均码字长度，以提高传输速度，节省存储空间。

5.4.1 哈夫曼编码(Huffman Coding)

哈夫曼在 1952 年根据可变长最佳编码定理，提出了依据信源集中各信号出现的概率分配不同长度即时码的哈夫曼算法。具体是，在信号概率已知的基础上，概率大的信号配置的码字短，概率小的信号对应的码字长。这些码字与被描述对象之间有固定对应关系。接收端在得到哈夫曼编码后，“即刻”解码得到与输入一致的图像信号。哈夫曼编码是一种无失真编码。

1) 即时码

即时码必须是唯一可译码(又称单义可译码)。对一组即时码来说，其中的任意一个码字都只与一种信号存在对应关系，而且任意一个码字都不能是其他码字的前缀。例如，有一维图像序列 $f(j)=\{f(1),f(2),f(3),f(4)\}$，设定的码字集合 $c(j)=\{0,10,110,111\}$。编码系统解码时，只要一遇到“0”，便知道对应的是 $f(1)$。若接收到的是“1”，则等待下一个比特，若下一个比特为“0”。即确定是 $f(2)$，若下一个比特是“1”，则等待第三个比特。第三个

比特为"1",判定为 $f(4)$,否则为 $f(3)$。

即时码的产生常采用根为 1 的树型结构法。图 5.7(a)对应左边是"0"、右边是"1"分支,产生 10、1101、1110、11110、11111 即时码组。图 5.7(b)分支情况则相反,得到与图 5.7(a)不同的一组即时码(11、1011、1010、1001、10001、10000)。

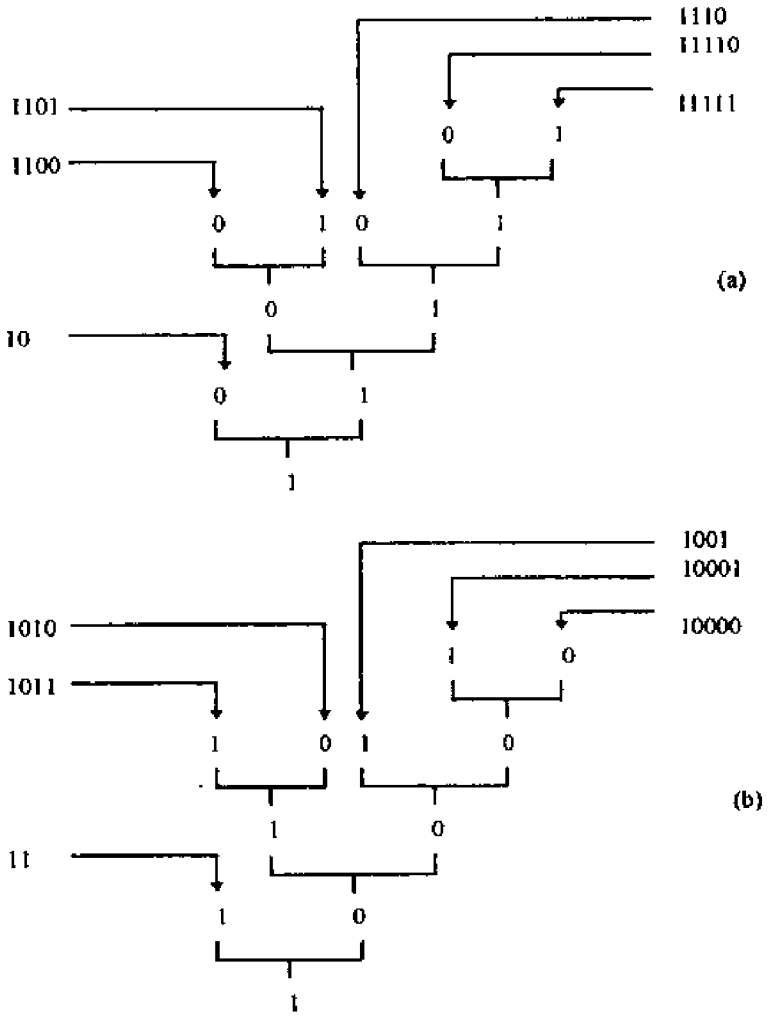


图 5.7 即时码产生示例

(a)左边是"0",右边是"1"分支; (b)右边是"0",左边是"1"分支

2) 哈夫曼编码

让我们借助一个例子来说明哈夫曼编码方法。

设有一图像序列,含有 8 个灰度级 $x_1, x_2, \cdots, x_8$。他们的概率分别为 $P_1 = 0.40, P_2 = 0.18, P_3 = 0.010, P_4 = 0.10, P_5 = 0.07, P_6 = 0.06, P_7 = 0.05, P_8 = 0.04$,按照哈夫曼编码算法,过程是:

(1) 将灰度等级按概率大小进行排序,大的在前,小的在后。

(2) 取两个最小概率相加,之和作为一个新的概率集合(新的概率集合的概率个数总是比执行此步骤前的概率集合少一个)。新的概率在新的概率集合中仍遵循降幂规则确定所处的位置。

被相加的两个最小概率所对应的灰度级成为哈夫曼树的一个叶结点，这两个结点构造一个双亲结点。

重复进行此步骤，直到只有两个概率(其和一定是 1)为止。这时，哈夫曼树到达根结点(整个过程见图 5.8)。

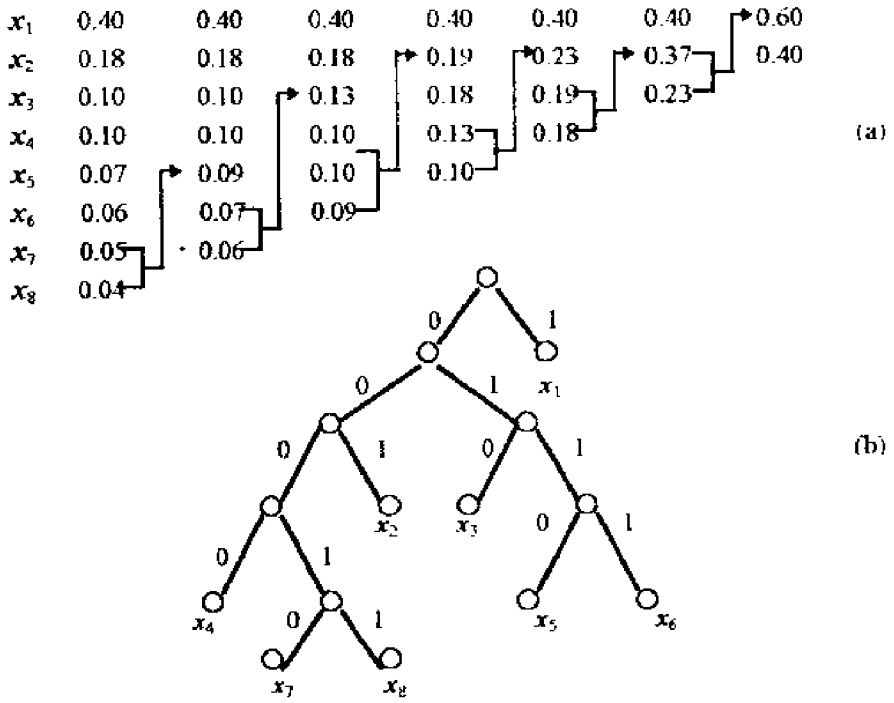


图 5.8 哈夫曼编码示例

(a)编码过程； (b)哈夫曼树型图

(3) 设所有结点的左后代为“0”，右后代为“1”(也可左为“1”，右为“0”)。那么，从根开始，经各中间结点到叶结点的路径代码即是该叶结点的哈夫曼码。在本例中，最终所得各灰度级的哈夫曼代码如下：

x_1:1 x_2:001 x_3:011 x_4:0000

x_5:0100 x_6:0101 x_7:00010 x_8:00011

3) 哈夫曼编码效率及其他

哈夫曼编码效率优于其他编码方法，是一种最佳变长码。本例中，编码效率

$$\eta = -(0.40\log_2 0.4 + 0.18\log_2 0.18 + 0.1\log_2 0.1 + 0.1\log_2 0.1 + 0.07\log_2 0.07 + 0.06\log_2 0.06 + 0.05\log_2 0.05 + 0.04\log_2 0.04)/(0.40\times 1 + 0.18\times 3 + 0.10\times 3 + 0.10\times 4 + 0.07\times 4 + 0.06\times 4 + 0.05\times 5 + 0.04\times 5)$$

$$= 0.978$$

哈夫曼编码时，对同一源图像序列，哈夫曼码并不是唯一的。在图 5.8(b)中，如果结点后代取值改为左“1”右“0”，则相应的哈夫曼码字变成

x_1:0 x_2:110 x_3:100 x_4:1111

x_5:1011 x_6:1010 x_7:11101 x_8:11100

对照两组哈夫曼码不难看出，尽管它们的组成不同，但两者的平均码长是一致的，均为

2.61 比特/字符。

实现哈夫曼编码的基础是统计源数据集中各信号的概率分布。当源数据成分复杂时，哈夫曼码表较大，码表生成难度增加，编译码速度变慢，从而限制了哈夫曼编码的实际应用。

4) 哈夫曼编码系统

图 5.9 给出了哈夫曼编码的基本过程。统计各灰度级出现的频度是哈夫曼编码的第一步。可建立一个整型数组，以存放各灰度级出现的次数。构造哈夫曼树是哈夫曼编码中的一个难点，其思路可以这样予以概述：将权重值非零的结点定义为活动结点，在参与构造哈夫曼树的 n 个活动结点中，找出两个权重值最小的活动结点(定义为 x_l, x_m)并将这两个结点的权重求和，形成一个新的活动结点，同时将 x_l, x_m 置零，表示这两个结点变成非活动结点，不再参与构造哈夫曼树的后续操作。新的活动结点与原来剩余的活动结点组成新一轮活动结点集。重复上述操作，直到只有一个活动结点(根结点)为止。

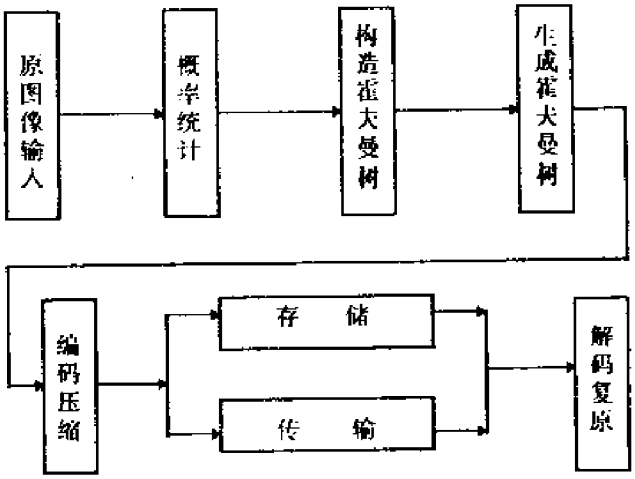


图 5.9 基本哈夫曼编码系统框图

由建成的哈夫曼树，便可生成各灰度级(信号)对应的哈夫曼代码。方法是依次由树根出发，前进到每个叶片，所经路径上对应的"0"或"1"的集合便是各信号对应的哈夫曼码(图 5·8(b))。编码压缩模块的功能是将源图像文件转换成哈夫曼代码形式存储到另一个文件中或用于信道传输。压缩后的图像文件是一些二进制的"0"、"1"串，且没有分界符。译码模块负责将这些"0"、"1"串参照哈夫曼树恢复为原图像信号。

5.4.2 香农编码(Shannon Coding)

香农编码的过程与哈夫曼编码有些相似。其编码步骤简述如下：

(1) 图像灰度级 x_i 按概率递减顺序排序。

(2) 将 x_i 分成两组，每组的概率和相同或相似。对第一组分配代码"0"，第二组分配代码"1"。

(3) 执行步骤(2)后，若每组还是由两个或两个以上的灰度级组成，就重复上述步骤，直到每组只有一个灰度级(具体见图 5.10)。

至此，可获得表 5.1 所示的香农编码(为比较起见，将哈夫曼编码也附于其内)。由表 5.1 可计算出香农编码的平均码字长度。

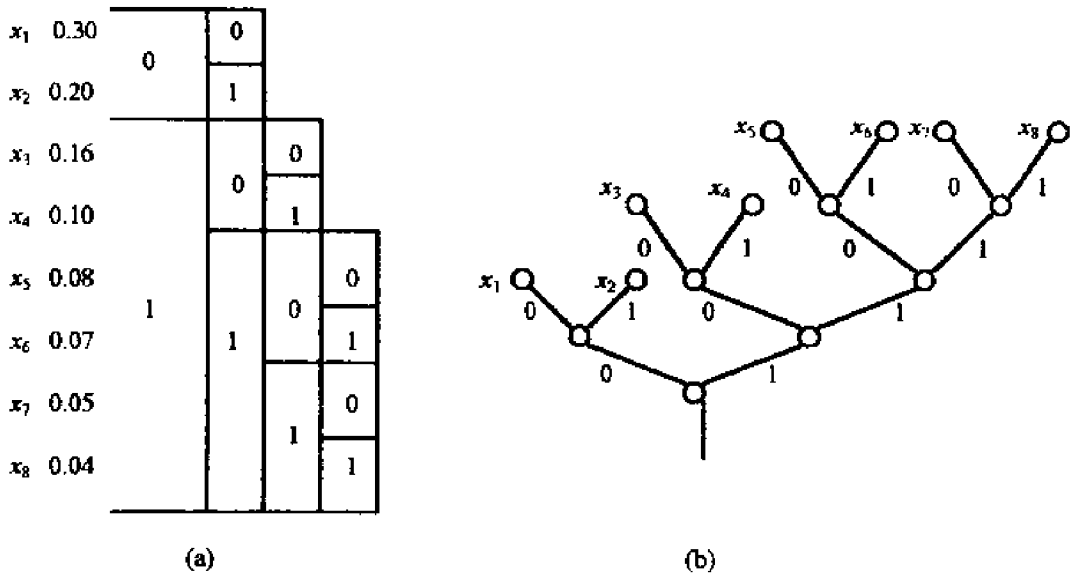


图 5.10　香农编码

(a)香农编码过程；　(b)香农树型图

表 5.1　香农编码与哈夫曼编码

香农编码			哈夫曼编码		
灰度级	码　字	码长	灰度级	码　字	码长
x_1	00	2	x_1	01	2
x_2	01	2	x_2	10	2
x_3	100	3	x_3	000	3
x_4	101	3	x_4	110	3
x_5	1100	3	x_5	0010	4
x_6	1101	4	x_6	0011	4
x_7	1110	4	x_7	1110	4
x_8	1111	4	x_8	1111	4

$$R(d) = \sum_{i=1}^{8} \beta_i p(d_i)$$

$$= 0.3 \times 2 + 0.2 \times 2 + 0.16 \times 3 + 0.10 \times 3 + 0.08 \times 4 + 0.07 \times 4 + 0.05 \times 4 + 0.04 \times 4$$

$$= 2.74 \text{ 比特/字符}$$

可见，运用本小节介绍的方法进行香农编码与哈夫曼编码相比较，其编码效果(平均码字长度或压缩比)没有明显差异。本例两者的码字长度一致，这是一种巧合。绝大部分情况下，两者之间是有差异的，从总的效果看，香农编码效率次于哈夫曼编码。

顺便指出，哈夫曼编码与香农编码最终都可得到编码树，但建立的方法是不一样的，哈

夫曼编码是由上向下的方式，而香农编码则相反。

5.4.3 算术编码(Arithmetics Coding)

先看一个实例。

令源数据集为{"*bcabd*"}，其出现的概率依次为 $P(a)=0.2, P(b)=0.4, P(c)=0.2, P(d)=0.2$。

各数据符号在半封闭实数区间[0,1)内的赋值范围设定为：

$$a=[0.0,0.2) \qquad b=[0.2,0.4)$$

$$c=[0.4,0.8) \qquad d=[0.8,1.0)$$

为讨论方便起见，再给出一组关系式：

$$N_area_s = F_area_s + C_flag_l \times L \tag{5.23}$$

$$N_area_e = F_area_s + C_flag_r \times L \tag{5.24}$$

式中 N_area_s 为新子区间的起始位置；

N_area_e 为新子区间的结束位置；

F_area_s 为前子区间的起始位置；

C_area_l = 当前符号的区间左端；

C_flag_r = 当前符号的区间右端；

L 为前子区间的长度。

根据以上假定，本例编码过程大致为

第一个被压缩的符号为"*b*"，代码的取值范围为[0.2,0.4)。

第二个被压缩的符号为"*c*"，由于前面的符号"*b*"已将取值区间限制在[0.2,0.4)范围内，所以"*c*"的取值范围应在前符号区间[0.2,0.4)的[0.4,0.8)之内。由(5.23)、(5.24)式，

$$N_area_s = 0.2 + 0.4 \times 0.2 = 0.28$$

$$N_area_e = 0.2 + 0.8 \times 0.2 = 0.36$$

即"*c*"的实际编码区间在[0.28,0.36)之间。

第三个被压缩符号是"*a*"，编码取值应在[0.28,0.36)之中的[0.0,0.2)之间，

$$N_area_s = 0.28 + 0.0 \times 0.28 = 0.28$$

$$N_area_e = 0.28 + 0.2 \times 0.8 = 0.296$$

依此类推，第四个符号"*b*"的编码取值在[0.28,0.296)之中的[0.2,0.4)区间内，第五个符号"*d*"的编码取值在第四个符号编码取值范围之内的[0.8,1.0)区间内。取终结果是：

数据流	编码区间	区间长度
b	[0.2,0.4)	0.2
c	[0.28,0.36)	0.08
a	[0.28,0.296)	0.016
b	[0.2864,0.2928)	0.016
d	[0.29152,0.2928)	0.00128

至此，数据串"*bcabd*"已被描述为一个实数区间[0.29152,0.2928)，或者说在 0.29152～0.2928 范围内的任一实数值都唯一对应串"*bcabd*"。这样，我们就可以用一实数表示一个数

据串。

区间[0.29152,0.2928)的二进制形式为[0.010010101010,0.0100101011)。二进制代码0.010010101位于区间内并最短,故其作为串"*bcabd*"的编码输出。考虑到算术编码中任一数据串的编码都含有"0.",所以本例的算术编码可进一步简化为010010101共9个比特,平均码长1.8比特/字符。

解码是编码的逆过程。根据编码时的概率分配表和压缩后数据代码所在的范围,确定代码所对应的每一个数据符号。

5.5 轮廓编码(Contour Coding)

任何一幅图像,总存在若干具有相同灰度等级的图像区域。如果我们运用一定的规则将它们的边界迹线记录下来,就等于记录了这些区域,而所用的比特数则远少于逐个像素描述法。轮廓编码就是一种基于上述想法的利用像素的空间坐标及其灰度值来压缩图像信息的编码技术。

先介绍轮廓编码中所用到的T算法与IP算法。

5.5.1 T(Trace)算法

定义具有同一灰度级的图像区域的边界线为等值线(落在等值线外面而又邻近等值线的任何一个像素与等值线上的像素都不会具有相同灰度级)。T算法用于跟踪图像等值线,并给出等值线的起始点位置、灰度级以及等值线移动的方向序列,为轮廓编码奠定基础。

如何跟踪等值线呢?T算法遵循"最先左看"的规则,即相对于进入方向,对于当前像素,最先观察其左边像素。如果左边像素与当前像素的灰度级相同,则将左边像素作为当前像素,否则扫描当前像素上边的像素。如果上边的像素的灰度级与当前像素一样,则将上边的像素作为当前像素,否则观察当前像素右边的像素。假定右边像素与当前像素没有差异,则以右边像素作为当前像素,否则考察当前像素下边的像素。如果下边像素与当前像素相同,则下边像素变为当前像素,否则当前像素位置不改变(参见图5.11,图中O为当前像素,·为4邻点像素,二进制代码00、01、10、11为方向码)。

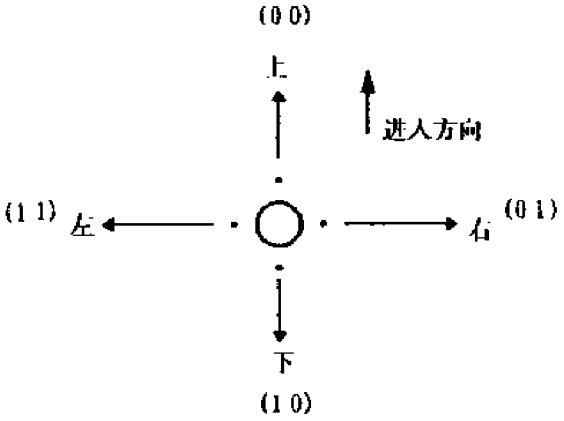


图5.11 T算法"最先左看"规则

T算法在跟踪等值线的过程中,将给组成等值线的每一个像素赋以方向标志,从而组成轮廓方向序列。方向标志的确定原则如表5.2所示。由表可见,分配给某一像素的方向标志取决于移入和移出该像素的方向。如果遇到某像素被通过两次,则先按表5.2得到两个标记,后再由表5.3确定最终标志符。

表 5.2　方向标志规则表

进入像素的方向	移出像素的方向 ↑或←	↓或→
↑或←	A	R
↓或→	R	D

表 5.3　二次通过方向标志规则表

标志符配置（第一次通过，第二次通过）	(D,A) (A,D) (R,R)	(D,R) (R,D) (D,D)	A,R (R,A) (A,A)
最终标志符	R	D	A

举一个例子。

有一幅 8×8、灰度级 0,1,2,3 的图像阵列(分布见图 5.12,图中符号·,+,o,× 分别表示灰度级 0,1,2,3)。运用 T 算法获得 4 条等值线。4 条等值线的起始点分别用 ip_1, ip_2, ip_3, ip_4 表示(图 5.12(a)。由于方向序列要求从等值线的起始点开始,历经等值线的第一点,第二点,直到最后一点,再回到起始点为止,所以等值线 ip_3, ip_4 都有某些点被通过两次。

5.5.2　IP (Initial Point)算法

IP 算法又称初始点算法,主要用于确定一条等值线(轮廓)的起始点。

先介绍比较点表 CPL (Comparison Point List)的概念。构成 CPL 的规则如下:

(1) 在扫描某一行前,此表是空的。

(2) 对某一行从左向右,逐点扫描。若某个像素点标志符为 A,则将该像素的灰度值送入表的底部;如果标志符为 D,则将表中标志符为 A 的像素灰度值删除;如果遇到标志符为 I 或 R,则表维持不变。

(3) 细心的读者从图 5.12(b)可以发现,等值线经过某一行时,向下的点数总是等于向上的点数,或者说 A 点数一定等于 D 点数,所以在某扫描行的端点,CPL 内总是空的。

知道了 CPL,就可以讨论 IP 算法判断等值线起始点准则。如果某个像素满足下列两个条件,该像素为轮廓的起始点:

(1) 它的标志符为 I。

(2) 它的灰度级值不等于 CPL 中最左边的标记为 A 的像素灰度值(一幅图像的第一行、第一列的像素除外)。

5.5.3　轮廓编码

一幅数字图像轮廓编码的过程是:

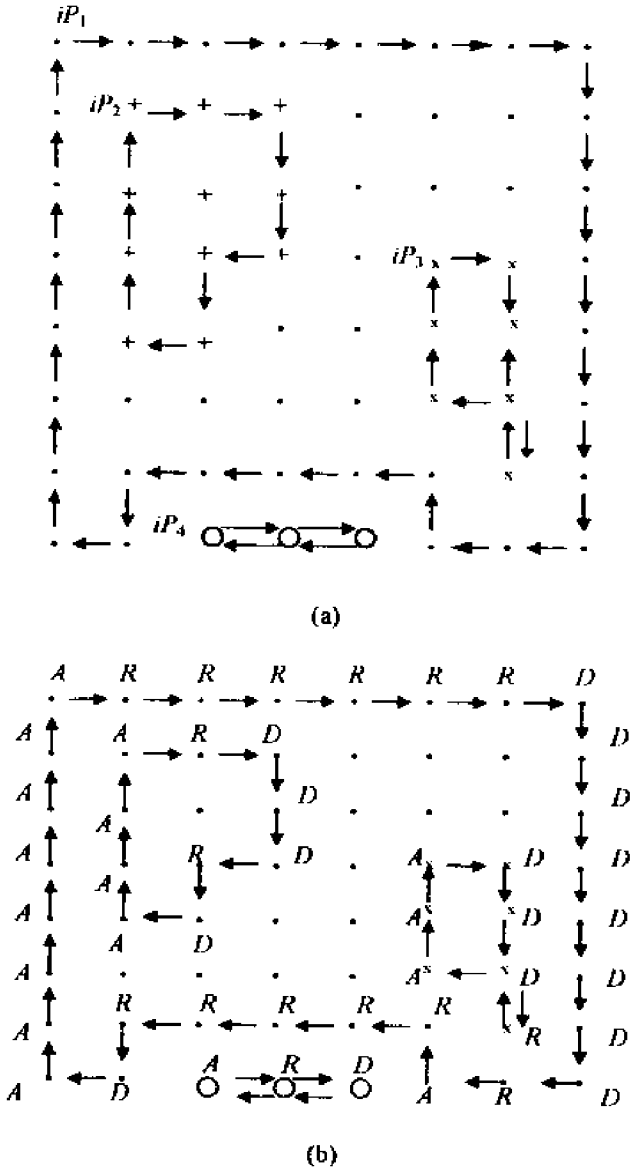


图 5.12　轮廓编码图例

(a)等值线跟踪；　(b)标志符配置

(1) 将图像阵列中所有像素都标志为 I。

(2) 进入图像的第一个像素定义为 ip_1，运用 T 算法找到第一条等值线，给组成该等值线的各像素按表 5.2、表 5.3 设置标志符。

(3) 由 IP 算法找到第二条，条三条，…，第 n 条等值线的起始点 $ip_2, ip_3, \cdots, ip_n$。用 T 算法完成所有等值线的跟踪和标志符的设置。

(4) 对等值线的三个属性(灰度级、IP 位置、方向序列)进行编码。

表 5.4 图 5.12 例编码约定

等值线		灰度级		行/列		序列方向	
编号	码字	自然码	码字	自然码	码字	方向	码字
#1	00	0	00	1	000	↑	00
#2	01	1	01	2	001	→	01
#3	10	2	10	3	010	↓	10
#4	11	3	11	4	011	←	11
				5	100		
				6	101		
				7	110		
				8	111		

表 5.4 是对图 5.12 例进行轮廓编码的约定。对于每条等值线，其编码输出将由等值线编号码字、灰度级码字、起始点空间坐标码字、方向序列码字四部分组成。这样，在图 5.12 中各条等值线的编码分别为：

#1:00,00,000,000,01,01,01,01,01,01,10,10,10,10,10,10,10,00,11,11,11,11,10,11,00,00,00,00,00,00 共 66 比特(逗号不在编码内，下同)。

#2:01,01,001,001,01,10,10,11,10,11,00,00,00 共 28 个比特。

#3:10,10,011,101,01,10,10,10,00,11,00,00 共 26 个比特。

#4:11,11,111,010,01,01,11,11 共 18 个比特。

这样，经过轮廓编码后该图像共需 138 比特的存储空间，而如果不作这样的处理，则需要 512 比特的存储空间，压缩比

$$P_r = (512 - 138)/138 = 73\%$$

平均码字长度为 2.16 比特。应该说压缩比还是不错的。

遵循轮廓编码的逆过程，即按照等值线编号、起始点空间位置、灰度值以及跟踪方向码的顺序，可以不失真的恢复原图像。轮廓编码一般适用于灰度级较少的图像。

5.6 二值图像编码

只有"0"和"1"两个灰度级的图像称之为二值图像，如文件、表格、工程图纸等。由于二值图像具有两个灰度级，这就决定了它有许多与灰度图像(风景、人物、遥感资料等)不同的统计特性，并由此产生了与前述不同的图像压缩方法。

5.6.1 直接编码

用一位二进制码(0 或 1)描述二值图像一个像素的编码方法称之为直接编码。直接编码时二值图像的像素数即是编码文件的毕特数。

二值图像的分辨率通常以每毫米达到的点数来表示。一般情况下，中文文件的分辨率为 5 点/毫米，英文打字文件的分辨率约为 4 点/毫米，报纸文字部分的分辨率是 16 点/毫米

左右，而某些晚报的分辨率可达到 20 点/毫米上下。

从视觉效果考虑，人们总是希望图像的分辨率高一些，但伴随而来的是信息量的迅速增加，从而使存储量加大，处理时间延长，对硬设备的要求提高。

5.6.2 跳跃空白编码

将图像的每一条扫描线分成若干等长的段，每段有 m 个像素。这些扫描线段不外乎下列三种情况之一：全是"0"像素；全是"1"像素；由"0"、"1"像素混合组成。我们定义全是"0"像素的段为"空白"。编码时"空白"用码字"0"表示，其余线段用"1"加直接编码表示。如假定扫描线由 160 个像素组成，令 $m = 10$，则有

线段号	线段组成	编码
#1	0000000000	0
#2	0111010011	10111010011
#3	1111111111	11111111111
…	…	…
#16	1000000000	11000000000

在二值图像中，"空白块"总是占有一定的比例，所以尽管在对非"空白块"编码时比直接编码多了一个分割码"1"，但从总的编码情况看，跳跃空白编码优于直接编码，特别是对背景比例较大的二值图像，其效果更佳。

"空白块"编码的平均码字长度

$$R(d_m) = p_m \times 1 + (1 - p_m) \times (m + 1) = m(1 - p_m) + 1 \tag{5.25}$$

压缩比为

$$P_r = (1 + \frac{1}{m} - p_m)^{-1} \tag{5.26}$$

式中，p_m 为白色块出现的概率。m 一般取 8 ~ 12。

跳跃空白编码的概念也可推广到二维，思路是：将图像全域划分成若干个 $m \times n$ 个像素的子块，当子块为全"0"像素时，编码为"0"；否则，子块编码首位为"1"，其余 $m \times n$ 个比特采用直接编码。

5.6.3 游程长度编码

对由信号 $x_1, x_2, \cdots, x_k$ 组成的二值图像扫描行，可以看成是若干个白像素段（简称白长）和黑像素段（简称黑长）之和。又知，在二值图像扫描行上，黑、白像素段总是交替产生的（全白或全黑扫描行除外）。游程编码就是充分考虑二值图像扫描行的这些特点，将黑长和白长以游程分别分配码字的分配方法。如有一条扫描线自左向右线段组成如下所示：

线段序号	线段性质	线段组成	线段长度
#1	黑长	0000000000	10
#2	白长	111111111	9
#3	黑长	0000	4
#4	白长	111111	6
#5	黑长	000	3

游程编码只要写成 10 个 0,9,4,6,3（含意为 10 个黑，9 个白，4 个黑，6 个白，3 个黑）即可，因

为第一个线段的性质指明以后，后面线段的性质就可以判断出来。此外，由于图像的行、列数已知，故各扫描行间就不必用分割符。

下面讨论黑长、白长分别编码对应的最少比特率。

令　白长的熵 $H(b)$ 为

$$H(b) = -\sum_{i=-1}^{s} p_i(b)\log_2 p_i(b) \tag{5.27}$$

式中　q 为白长或黑长拥有的像素个数，$1 \leqslant q \leqslant N$，$N$ 为扫描行的像素数；$p_i(b)$ 为游程长度为 i 的白长出现的概率。根据

$$H(b) \leqslant \bar{l}(b) < H(b) + 1 \tag{5.28}$$

可得到

$$E(b) = \sum_{i=1}^{q} ip(b) \tag{5.29}$$

于是

$$\frac{H(b)}{E(b)} \leqslant \frac{\bar{l}(b)}{E(b)} < \frac{H(b)+1}{E(b)} \tag{5.30}$$

式中　$\bar{l}(b)$ 为各白长的平均比特数(码字的平均长度)；

$E(b)$ 为白长的平均象数个数。

令

$$h(b) = \frac{H(b)}{E(b)}$$

$$\bar{l}(b) = \frac{\bar{l}(b)}{E(b)}$$

从而有

$$h(b) \leqslant \bar{l}(b) < h(b) + \frac{1}{E(b)} \tag{5.31}$$

对黑长进行同样的编码，得

$$h(h) \leqslant \bar{l}(h) < h(h) + \frac{1}{E(h)} \tag{5.32}$$

将白长和黑长的编码进行平均，生成

$$\bar{h}(hb) = p(b)h(b) + p(h)h(h) \tag{5.33}$$

$$\bar{l}(hb) = p(b)\bar{l}(b) + p(h)\bar{l}(h) \tag{5.34}$$

式中　$\bar{h}(hb)$ 为平均后每个像素的熵值；

$\bar{l}(hb)$ 为平均后每个像素的编码长度(比特数)。

进一步整理，可得

$$\bar{h}(hb) \leqslant \bar{l}(hb) < \bar{h}(hb) + \frac{p(b)}{E(b)} + \frac{p(h)}{E(h)} \tag{5.35}$$

一般情况下，黑长、白长分别编码比黑长、白长混合编码效果好。

游程编码从本质上说就是计算图像信号出现的行程长度，然后将行程长度转换为代码。游程编码有多种编码方式，除上述白长、黑长分别编码外，还有准最佳可变长度编码等。国际电报电话委员会(International Telegraph Consultative Committee，简写为 CCITT)第 15 研究组为适应综合业务数字通信网需要而主持制定的传真压缩标准 G_3、G_4 中的一维压缩方法，就

采用了可变长码行程编码方法。全组码字由结束代码、组成代码和附加组成码三部分组成(分别见表 5.5、表 5.6、表 5.7)。当行程在 0 ~ 63 范围内时,行程用相应的结束代码字表示;当行程在 63 ~ 2623(2560 + 63)区间时,首先由最接近实际行程长度的组成代码编码,后再跟结束代码字;当行程大于 2624 时,首先按上述方法由组成代码 2560 编码,如果行程的剩余部分大于 2560,则按表 5.7 产生的附加的组成代码,直到行程的剩余部分小于 2560,再按前述办法编码。

表 5.5 结束代码表

白行程长度	码 字	黑行程长度	码 字
0	00110101	0	0000110111
1	000111	1	010
2	0111	2	11
3	1000	3	10
4	1011	4	011
5	1100	5	0011
6	1110	6	0010
7	1111	7	00011
8	10011	8	000101
9	10100	9	000100
10	00111	10	0000100
11	01000	11	0000101
12	001000	12	0000111
13	000011	13	00000100
14	110100	14	00000111
15	110101	15	000011000
16	101010	16	0000010111
17	101011	17	0000011000
18	0100111	18	0000001000
19	0001100	19	00001100111
20	0001000	20	00001101000
21	0010111	21	00001101100
22	0000011	22	00000110111
23	0000100	23	00000101000
24	0101000	24	00000010111
25	0101011	25	00000011000
26	0010011	26	000011001010
27	0100100	27	000011001011
28	0011000	28	000011001100
29	00000010	29	000011001101

续表 5.5

白行程长度	码　字	黑行程长度	码　字
30	00000011	30	000001101000
31	00011010	31	000001101001
32	00011011	32	000001101010
33	00010010	33	000001101011
34	00010011	34	000011010010
35	00010100	35	000011010011
36	00010101	36	000011010100
37	00010110	37	000011010101
38	00010111	38	000011010110
39	00101000	39	000011010111
40	00101001	40	00001101100
41	00101010	41	00001101101
42	00101011	42	000011011010
43	00101100	43	000011011011
44	00101101	44	000001010100
45	00000100	45	000001010101
46	00000101	46	000001010110
47	00001010	47	000001010111
48	00001011	48	000001100100
49	01010010	49	000001100101
50	01010011	50	000001010010
51	01010100	51	000001010011
52	01010101	52	000000100100
53	00100100	53	000000110111
54	00100101	54	000000111000
55	01011000	55	000000100111
56	01011001	56	000000101000
57	01011010	57	000001011000
58	01011011	58	000001011001
59	01001010	59	000000101011
60	01001011	60	000000101100
61	00110010	61	000001011010
62	00110011	62	000001100110
63	00110100	63	000001100111

表 5.6 组成代码表

白行程长度	码 字	黑行程长度	码 字
64	11011	64	0000001111
128	10010	128	000011001000
192	010111	192	000011001001
256	0110111	256	000001011011
320	00110110	320	000000110011
384	00110111	384	000000110100
448	01100100	448	000000110101
512	01100101	512	0000001101100
576	01101000	576	0000001101101
640	01100111	640	0000001001010
704	011001100	704	0000001001011
768	011001101	768	0000001001100
832	011010010	832	0000001001101
896	011010011	896	0000001110010
960	011010100	960	0000001110011
1024	011010101	1024	0000001110100
1088	011010110	1088	0000001110101
1152	011010111	1152	0000001110110
1216	011011000	1216	0000001110111
1280	011011001	1280	0000001010010
1344	011011010	1344	0000001010011
1408	011011011	1408	0000001010100
1472	010011000	1472	0000001010101
1536	010011001	1536	0000001011010
1600	010011010	1600	00000001011011
1664	011000	1664	0000001100100
1728	010011011	1728	0000001100101
EOL	000000000001	EOL	000000000001

表 5.7 附加组成代码表

码字	白或黑行程长度
1792	00000001000
1856	00000001100
1920	00000001101
1984	000000010010
2048	000000010011
2112	000000010100
2176	000000010101
2240	000000010110
2304	000000010111
2368	000000011100
2432	000000011101
2496	000000011110
2560	000000011111

5.7 多媒体图像编码压缩技术

多媒体技术是集文本、声音、图像、动画等诸多媒体信息于计算机系统，经综合处理后于一体的技术。

在多媒体技术中，首当其冲的难题是庞大数据的采集、存储、处理和传输。例如，一幅 640×350，彩色深度为 24 比特的彩色图像，其数据量为 672 Kb，达到动画效果每秒至少显示 30 帧图像，这就要求信道传输速度在 20 M/s 以上。如果该尺度图像达到真彩色，则信道传输速度要大于 161 M/s。显然，这样的传输速度目前在硬件上实现还是有困难的。此外，从存储角度看，一张 1 Gb 的光盘，也只能存放 186 张上述低分辨真彩色图像。可见，数据压缩已成为多媒体最重要的技术之一。

5.7.1 经典编码压缩技术

表 5.8 给出了多媒体图像编码压缩中所用到的主要经典编码技术。这些编码方法理论完善，技术成熟，编码效率较高，在现行的视频压缩国际标准 CCITT 的 H.261，ISO 的 JPEG 及 MPEG 系列中普遍采用。关于这些编码压缩技术的原理及方法前面已有详细讨论，作为回顾，此处作一级扼要的表述。

1) 冗余压缩

减少或完全去除行内相邻像素间、帧内相邻像素行间以及帧间对应像素间的相关性(冗余)，以减少存储、处理及传输的数据量，这一过程称之为冗余压缩。

冗余压缩是可逆过程,利用压缩后的数据可以重新构造获得图像源数据。冗余压缩是无损压缩。

(a) 哈夫曼编码。依赖于数据概率特性的可编长编码,从理论上说,它是一种最佳编码方法。

(b) 算术编码。用某个实数区间来表示若干被编码的信息,用该区间的一个实数对应的二进制码作为被讨论像素的编码输出。

(c) 游程编码。将图像信号中灰度值相同的连续像素点串用(样本值,样本数)对来描述,并分别赋以不同长度的码字。

(d) 香农编码。把图像灰度序列按概率递减的顺序排列起来,然后根据概率相等或相近原则,将灰度序列一分为二,上半子集赋以"0",下半子集赋以"1"。重复这一操作,直到任一子集都只有一个灰度。在此基础上,顺序读出各灰度级路径中的"0"、"1"二进制串,即是分配的码字。

(e) 轮廓编码。跟踪图像区域边界廓线(等值线),以起始点、灰度值、方向序列三要素描述廓线并予以编码。

表 5.8 多媒体图像压缩经典技术

编码类型	编码方法	编码基础
冗余压缩	哈夫曼编码	概率分布特性
	算术编码	随源数据符号不断缩小的实数区间
	游程编码	游程长度
	香农编码	概率分布特性
	轮廓编码	区域边界迹线及其属性
熵压缩	预测编码	相邻像素相关性
	变换编码	变换域
	混合编码	像素相关性与变换域有机结合

2) 熵压缩

利用人的视觉心里特性,在允许的失真度下,减少信息熵。熵压缩后恢复的图像与原图像有一定的差别,熵压缩属于有损压缩。

(a) 预测编码。利用相邻像素间的相关性,用已传送(或已知的)的像素的灰度值预测下一像素,对预测误差进行编码。

(b) 变换编码。经正交变换(DFT, DCT ,KLT 等)获得图像变换域,再对变换系数作抽样编码。

(c) 混合编码。对图像数据同时以两种或两种以上编码方法进行编码。如行内变换编码/行间 DPCM,帧内余弦变换/帧间 DPCM 或帧内傅立叶变换/帧间 DPCM 等。

5.7.2 分形编码

经典图像编码方法对任何图像都是以纯数据的方式去对待,利用信号理论和信息论,采

用信号序列或信号矩阵形式，从微观角度去编码。经典编码技术比较成熟，处于实用阶段。但经典编码方法也有其局限性，如压缩比有时不是很高，没有考虑图像自身固有的特点及人的眼睛对图像的视觉特性等。如一幅图像其画面由一幢房子、三棵树、两只鸡所组成，经典编码方法不考虑其实际内容，直接以数据逐点处理。而宏观方法则只对房子、树和鸡这三种对象作某种方式的编码，这种基于模型方式描述图像的编码效率显然优于经典编码方法。

分形编码属模型编码的范畴。

分形的概念源于数学家曼德布劳(70年代初提出)。其含义是一类无规则、混乱而复杂的宿体，其局部与整体有相似的形。如崇山峻岭，无论你是从1千米高空还是10千米高空去俯视它，其都有相似的外貌。又如委婉绵延的海岸线，1千米长和100千米长其轮廓都是相仿的。这种局部与整体的自相似性体系，为构造一种新的编码方法提供了可能。

分形编码的步骤是：

(a) 把图像划分成互不重叠、任意大小、任意形状的 domain 分区，所有 domain 分区拼起来应恰为原图。

(b) 对每个 domain 分区再划分一个 range 分区，range 分区必须大于相应的 domain 分区，range 分区之间可以重叠，所有 range 分区拼起来无须恰好覆盖原图全图。range 分区在三维仿射变换以后应尽可能与 domain 分区中的图像接近。

(c) 生成 FIF(Fractal Imange Format)文件。每个 domain 分区都可通过仿射变换(n 维空间函数图像经过旋转、平移、拉伸等多种变换的综合过程，仿射变换能使图像构成更好的组形)恢复。仿射变换系数决定了仿射变换过程。FIF 文件的开头规定如何划分 domain 分区，接着依次记载各 domain 分区对应的仿射变换系数。

(d) 对 FIF 文件编码。由于每个 domain 分区对应的仿射变换系数只有为数不多的几个参数(如给出方程 $y = ax + b$ 中的 a、b 值，就可画出一个任意分辨率的直线图形)，所以分形编码可获得很高的压缩比。

分形编码的解码过程很简单，首先从 FIF 文件中读取 domain 分区信息和 domain 分区对应的仿射变换系数等数据。然后建立两个同样大小的缓冲区，并分别取名 D 缓冲区和 R 缓冲区。D 缓冲区用于存放 domain 分区图像，R 缓冲区用于存放 range 分区图像。接着按 FIF 文件的规定，将 D 缓冲区划分成 domain 分区，R 缓冲区划分成 range 分区，将指针顺序指向 domain 分区，根据仿射变换系将相应的 range 分区作仿射变换并用变换后的系数覆盖 domain 分区的原有内容。一轮完毕，D 缓冲区内形成一幅新的 domain 图像。重复进行迭代操作，直到 R、D 中的图像相似(很难看出差别)，D 中的图像即为复原图像。

综上所述，分形编码是一种构思新颖、完全不同于经典方法的编码技术。其主要特点是高压缩比(比经典编码高 1～2 个量级)；运算速度与提高图像分辨率关系不大；选择恰当的分形模型完全可以构造清晰的边缘细节；压缩时计算量大，时间长，而解压时则很快。分形编码的这些特点使其成为正在大力开发的图像编码方案之一。

分形编码目前还未完全进入实用化，其主要困难是如何恰到好处地分割图像？如何构造迭代函数系统？这两条直接关系到编码压缩比和复原图像的质量。domain 分区大，domain 分区的总数就少，FIF 文件就小，编码效率提高。但 range 分区迭代的图像与 domain 分区原来的图像的吻合程度降低。

5.7.3 自适应网格编码

在多媒体通信中，有一类这样的图像，其画面简洁，帧间变化小，数据量大。如可视电话，图像内容基本上是人的上半身像，变化部分主要是人的眼睛。为了降低成本和便于普及，人们希望采用 PSTN (Public Switch Telephone Network)那样的普通通信条件。这时，高压缩比成了编码工作追求的首要目标，而信息保真度退到了次要位置(有时只要能认识是谁就可以了)。在这样的前景下，开发"超低比特率"的编码压缩方法成为研究热点。

1991 年，Terzopoulos 和 Vasilescu 提出了一种能自动跟踪图像特征的自适应网格编码方法(Self－Adaptive－Mesh Coding)。所谓网格，是指由纵、横线条组成的多边形集合。如果划分一幅图像的网格均由正方形或长方形组成，称之为正规网格。如果划分一幅图像的网格因受图像局部特性的影响而变成不规则多边形，则称之为自适应网格。组成同一形状的网格的边数是固定的，多边形的顶点称之为结点。

自适应网格编码从原理上说就是设想把一张由结点组成的正规网格覆盖在图像表面上，图像上重要性不同的特征对结点产生与其相适应的吸引力，正规网格上的结点就会按图像特征的重要性大小，不同程度地聚集在特征周围。形成能跟踪特征的自适应网格。在此基础上，对自适应网格的结点的位置、值以及重建图像所需的系数进行编码。

目前正在研究的多媒体图像编码压缩技术远不止这些，如小波变换图像压缩编码、三维模型压缩编码(物体基编码、知识基编码、语义基编码)等都已取得良好的研究成果，不久可望付诸应用。

6 图像恢复与重建

●图像退化　●反向滤波　●约束还原
●维纳滤波　●点冲激函数　●几何畸变

6.1 图像恢复的基本概念

在景物成像的过程中，受多种因素的影响，图像质量都会有所下降，这种图像质量下降的过程称为图像的退化。例如，成像目标物体的运动，在摄像后所形成的运动模糊。当人们拍摄照片时，由于手持照像机的抖动，结果像片上的景物是一个模糊的图像。由于成像系统的光散射现象而导致图像的模糊。又如传感器特性的非线性，光学系统的像差，以致在成像后与原来景物发生了不一致的现象，称之为畸变。再加上多种环境因素，在成像后造成噪声干扰。人类的视觉系统对于噪声的敏感程度要高于听觉系统，在声音传播中的噪声虽然降低了质量，但时常是感觉不到的。但景物图像的噪声，如显示器上的图像或像纸上的图像噪声，即使很小，都逃不出敏锐的视觉系统。图像恢复的过程则为了还其图像的本来面目，即由退化了的图像恢复到能够真实反映景物的图像。

图像恢复的过程从某种意义上来说，也是为了改善退化了的图像质量，这似乎与图像增强是一回事了。其实不然，这两者之间是有明显的区别的。图像增强的过程基本上是一个探索的过程，利用人的心理状态和视觉系统去控制图像质量，人们应用多种增强的方法去改进图像质量，直到人们的视觉系统满足为至。例如，对比度展宽是一个图像增强技术，因为它可以给视觉系统较原图像更清晰的图像，这是一个视觉系统的心理感受过程。

图像恢复则是试图利用退化现象的某种先验知识，建立退化现象的数学模型，再根据模型进行反向的推演运算，以恢复原来的景物图像。因而，图像恢复可以理解为图像降质过程的反向过程。建立图像恢复的反向过程的数学模型，就是图像恢复的主要任务，而图像增强并不存在这个任务。经过反向过程的数学模型的运算，要想恢复全真的景物图像是件很困难的事情。所以，图像恢复本身往往需要有一个质量标准，即衡量接近全真景物图像的程度，或者说，对原图像的估计是否到达最佳的程度。为了描述图像退化的过程所建立的数学模型往往多种多样，而恢复的质量标准也往往存在差异性，所以图像恢复是一个烦杂的数学过程，并且方法有多种。本章先讨论图像退化的一般模型，主要运用线性代数的矩阵方法去求解基于线性的、空间不变的退化模型。根据评价的准则，也会涉及到非线性的恢复问题。在大多数情况下，不考虑传感器数字变换器的退化问题，着重讨论退化的数字图像的灰度级恢复问题。对于成像系统的非线性所引起的图像像素位置发生位移的图像恢复，本章则介绍几何畸变的恢复。

6.2 图像退化模型

原始图像 $f(x,y)$经过一个退化算子或退化系统 $H(x,y)$的作用，并且和噪声 $n(x,y)$进行叠加，形成退化后的图像 $g(x,y)$，就得到退化后的图像。图 6.1 示出退化过程的输入和输出的关系。图中 $H(x,y)$概括了退化系统的物理过程，就是所要寻找的退化数学模型。

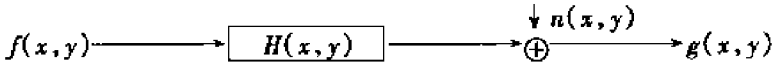


图 6.1 图像退化过程的模型化

数字图像的图像恢复问题可看为，根据退化了的图像 $g(x,y)$和退化算子的 $H(x,y)$的形式，沿着反向过程去求解原始图像 $f(x,y)$，或者说是逆向地寻找原始图像的最佳近似估计。图像退化的模型化过程可以用数学表达式写成如下的形式：

$$g(x,y) = H[f(x,y)] + n(x,y) \tag{6.1}$$

式中 $H[f(x,y)]$可理解为综合所有退化因素的函数。如果 $H[\cdot]$是线性算子，则需满足：

$$H[k_1 f_1(x,y) + k_2 f_2(x,y)] = k_1 H[f_1(x,y)] + k_2 H[f_2(x,y)] \tag{6.2}$$

式中 k_1 和 k_2 为常数；$f_1(x,y)$和 $f_2(x,y)$为任何两个输入图像。

若 $k_1 = k_2 = 1$，则 $H[\cdot]$退化算子的线性表达为

$$H[f_1(x,y) + f_2(x,y)] = H[f_1(x,y)] + H[f_2(x,y)] \tag{6.3}$$

若忽略了噪声的作用，则

$$g(x,y) = H[f(x,y)] \tag{6.4}$$

对于任一个 $f(x,y)$和任一个 α 和 β，若存在如下关系：

$$Hf(x-\alpha, y-\beta) = g(x-\alpha, y-\beta) \tag{6.5}$$

则说明 $H[\cdot]$退化算子不因位置而变化，即空间是不变的。这个关系说明图像中某一像素的退化结果只取决于在该点的输入值，而不取决于该点的位置。

$$\begin{aligned} g(x,y) = H[f(x,y)] &= H\left[\int_{-\infty}^{\infty}\int f(\alpha,\beta)\delta(x-\alpha, y-\beta)\mathrm{d}\alpha\mathrm{d}\beta\right] \\ &= \int_{-\infty}^{\infty}\int H[f(\alpha,\beta)\delta(x-\alpha, y-\beta)]\mathrm{d}\alpha\mathrm{d}\beta \\ &= \int_{-\infty}^{\infty}\int f(\alpha,\beta)H[\delta(x-\alpha, y-\beta)]\mathrm{d}\alpha\mathrm{d}\beta \\ &= \int_{-\infty}^{\infty}\int f(\alpha,\beta)h(x,y;\alpha,\beta)\mathrm{d}\alpha\mathrm{d}\beta \end{aligned} \tag{6.6}$$

式中　$h(x,y;\alpha,\beta) = H\delta(x-\alpha,y-\beta)$

它称为 H 的冲激响应。在光学中，冲激为一个光点。也就是说，若 $n(x,y)=0$，则 $h(x,y;\alpha,\beta)$ 是 H 对在坐标为(α,β) 处的冲激响应，也称为点扩展函数(PSF)。本式的重要性在于，如果已知 H 对冲激的响应，那末在 $g(x,y)$ 已知的条件下，$f(\alpha,\beta)$ 的响应便可求出。这就是说，线性系统 H 完全可以由其冲激响应表征出来。利用卷积定义，则退化图像 $g(x,y)$ 可表示为

$$g(x,y) = f(x,y) * h(x,y) \tag{6.7}$$

加上噪声的影响，退化图像 $g(x,y)$ 则表示为

$$g(x,y) = f(x,y) * h(x,y) + n(x,y) \tag{6.8}$$

从数学的表达式可以清楚地看出，图像恢复的过程就可以认为已知 $g(x,y)$ 和有关 $h(x,y)$，$n(x,y)$ 的一些先验知识，求出 $f(x,y)$。这就是经典的图像恢复过程的退化模型。

将连续函数离散化，退化模型可写成为

$$\begin{aligned} &g = Hf + n \qquad\qquad (6.9)\\ &g = V_{ec}[g_e(x,y)] \qquad (x=0,1,\cdots,M-1;y=0,1,\cdots,N-1)\\ &f = V_{ec}[f_e(x,y)] \qquad (x=0,1,\cdots,M-1;y=0,1,\cdots,N-1)\\ &n = V_{ec}[n_e(x,y)] \qquad (x=0,1,\cdots,M-1;y=0,1,\cdots,N-1) \end{aligned}$$

V_{ec} 为线性代数中将矩阵转换为向量的算子，式中：

$$g_e(x,y) = f_e(x,y) * h_e(x,y) + n_e(x,y) \tag{6.10}$$

式中的 $f_e(x,y)$、$h_e(x,y)$ 和 $n_e(x,y)$ 分别为 $f(x,y)$、$h(x,y)$ 和 $n(x,y)$ 的延伸函数。离散的退化模型在二维情况下的讨论这样来进行：分别给出大小 $A\times B$ 和 $C\times D$ 的两个数字图像 $f(x,y)$ 和 $h(x,y)$，可以做成大小为 $M\times N$ 的延伸图像，则令

$$f_e(x,y) = \begin{cases} f(x,y) & \text{当 } 0\leqslant x\leqslant A-1 \text{ 和 } 0\leqslant y\leqslant B-1 \\ 0 & \text{当 } A\leqslant x\leqslant M-1 \text{ 和 } B\leqslant y\leqslant N-1 \end{cases} \tag{6.11}$$

$$h_e(x,y) = \begin{cases} h(x,y) & \text{当 } 0\leqslant x\leqslant C-1 \text{ 和 } 0\leqslant y\leqslant D-1 \\ 0 & \text{当 } C\leqslant x\leqslant M-1 \text{ 和 } D\leqslant y\leqslant N-1 \end{cases} \tag{6.12}$$

把延伸函数 $f_e(x,y)$ 和 $h_e(x,y)$ 看作二维函数，也就是说，在 x 和 y 方向上的变化周期为 M 和 N，则

$$g_e(x,y) = \sum_{m=0}^{M-1}\sum_{n=1}^{N-1} f_e(x,y)h_e(x-m,y-n) + n_e(x,y) \tag{6.13}$$

式中　$x = 0,1,2,\cdots,M-1;$

$y = 0,1,2,\cdots,N-1$。

卷积函数 $g_e(x,y)$ 为周期函数，其周期 $f_e(x,y)$ 和 $h_e(x,y)$ 的周期一样。当 $M\geqslant A+C-1$，和 $N\geqslant B+D-1$ 时可避免卷积周期的交叠。

将 $M \times N$ 维的函数矩阵 $f_e(x,y)$、$g_e(x,y)$ 和 $n_e(x,y)$ 的各行卷积，即形成 f、g 和 n 代表的 MN 维列向量。或者说成 f、g 和 n 是 $(MN) \times 1$ 维列向量，那末 H 即为 $MN \times MN$ 维矩阵。这一矩阵包括 M^2 个部分，每个部分的大小为 $N \times N$，排成下列顺序：

$$H = \begin{bmatrix} H_0 & H_{M-1} & H_{M-2} & \cdots & H_1 \\ H_1 & H_0 & H_{M-1} & \cdots & H_2 \\ H_2 & H_1 & H_0 & \cdots & H_3 \\ \cdots & \cdots & \cdots & \cdots & \\ H_{M-1} & H_{M-2} & H_{M-3} & \cdots & H_0 \end{bmatrix} \tag{6.14}$$

每个部分 H_j 是由延伸函数 $H_e(x,y)$ 的第 j 行构成的，构成方法为

$$H_j = \begin{bmatrix} h_e(j,0) & h_e(j,N-1) & h_e(j,N-2) & \cdots & h_e(j,1) \\ h_e(j,1) & h_e(j,0) & h_e(j,N-1) & \cdots & h_e(j,2) \\ h_e(j,2) & h_e(j,1) & h_e(j,0) & \cdots & h_e(j,3) \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ h_e(j,N-1) & h_e(j,N-2) & h_e(j,N-3) & \cdots & h_e(j,0) \end{bmatrix} \tag{6.15}$$

$h_e(x,y)$ 具有周期性；H_j 是一个循环矩阵，即矩阵的第一行的尾与第二行的头相同，矩阵的末行的尾与矩阵的首行的头相同。H 矩阵也具有同样的性质，称为分块循环矩阵。图像恢复的任务则是根据给定的 g，和在掌握了 H 和 n 的函数以后估计出 f 来。但是，直接估计出 f 是一项非常烦杂的过程。若 $M = N = 512$，那末，$H = (MN) \times (MN) = 262144 \times 262144$。直接得到 f 需要求解 262144 个联立方程组，这个过程非常烦杂，利用 H 矩阵的分块循环的特点可以进行简化。

根据矩阵的基本理论，引入对角线矩阵。如果 H 是循环矩阵，可得到如下关系：

$$H = WDW^{-1} \tag{6.16}$$

或者表示为

$$D = W^{-1}HW \tag{6.17}$$

对于上式有两点需要进行讨论。其一，矩阵 W 称为矩阵 H 的本征向量。它的大小为 $MN \times MN$。假设利用 H 的 M 个本征向量作为列，组成 $M \times M$ 矩阵 W，那末

$$W = [W(0), W(1), W(2), \cdots, W(M-1)] \tag{6.18}$$

本征向量 W 的第 k 行、i 列个元素，用 $W(k,i)$ 来表示，则定义为

$$W(k,i) = \exp[j\frac{2\pi}{M}ki]W_N \tag{6.19}$$

式中 k, i 分别为 $0,1,2,\cdots,M-1$。W_N 为 $N \times N$ 矩阵：

$$W_N(k,n) = \exp[j\frac{2\pi}{N}kn]$$

式中 k, n 分别为 $0,1,\cdots,N-1$。根据矩阵理论可知，对任意的分块循环矩阵，例如 H 矩阵，W 都可使其对角线化。

其二，D 是对角线矩阵，称为 H 矩阵的本征值，可表示为

$$D(k,k) = \sum_{i=0}^{M-1} h_e(i)\exp[-j\frac{2\pi}{M}ki] \tag{6.20}$$

式中 $k=0,1,\cdots,2,M-1$。右端则为 $MH(k)$，$H(k)$是延伸函数 $h_e(x)$的离散傅立叶变换。可以写成

$$D(k,k) = MH(k) \tag{6.21}$$

D 是一个对角线矩阵，D 与任何一个向量之积就是用 D 的一个单独的对角线元素去乘该向量的每一个相应的元素。

$$g = Hf + n = WDW^{-1}f + n \tag{6.22}$$

用 W^{-1}乘上式两端，利用 $H = WDW^{-1}$，$D = W^{-1}HW$ 和 $W \cdot W^{-1} = W^{-1} \cdot W = I$。其中，$W^{-1}$为一个 $MN \times MN$ 矩阵。D 为 $MN \times MN$ 对角线矩阵，H 为分块循环矩阵，f 和 g 为 MN 维向量。

对角线矩阵 D 各个元素与延伸函数 $h_e(x,y)$的傅立叶变换有关。可写成

$$H(u,v) = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} h_e(x,y)\exp[-j2\pi(ux/M + vy/N)] \tag{6.23}$$

式中 $u=0,1,2,\cdots,M-1$

$v=0,1,2,\cdots,N-1$

对角线阵 D 包括 MN 个元素，是这样组成的：第一组 N 个元素为 $H(0,0)$，$H(0,1)\cdots$，$H(0,N-1)$；第二组 $H(1,0)$，$H(1,1)$，$\cdots$，$H(1,N-1)$；按此推理，第 M 组的 N 个元素为 $H(M-1,0)$，$H(M-1,1)$，$\cdots$，$H(M-1,N-1)$。这样组成的整个矩阵再乘以 MN，可得到 D，用数学表达为

$$D(k,i) = \begin{cases} MNH([k/N], k_{\mathrm{MOD}\,N}) & \text{如果 } i = k \\ 0 & i \neq k \end{cases} \tag{6.24}$$

式中 $[k/N]$和$(k_{\mathrm{MOD}\,N})$表示 D 矩阵的行号与列号，可形成一个对角线阵 D。

这样，可以得到(6.25)式：

$$G(u,v) = MNH(u,v)F(u,v) + N(u,v) \tag{6.25}$$

式中 $u=0,1,2,\cdots,M-1$；

$v=0,1,2,\cdots,N-1$；

将此式与 $g = Hf + n$ 比较，可以看出，原来给定模型的烦杂的方程组，可化简为计算大小为 $M \times N$ 的少量的离散傅立叶变换。如果 $M = N = 512$，运用傅立叶变换，就会将一个烦杂的运算变成一个简单的运算。

6.3 图像恢复方法

6.3.1 反向滤波器法

图像退化的过程是原图像 $f(x,y)$通过物理特性为 $h(x,y)$的成像系统，形成了退化的图像 $g(x,y)$。因此，可以把 $h(x,y)$理解为一个滤波函数，用傅立叶变换的形式可表示为

$$G(u,v) = H(u,v) * F(u,v) \tag{6.26}$$

就是说，原函数 $F(u,v)$乘以 $H(u,v)$则得到该退化的图像 $g(x,y)$的傅立叶变换。所以图像恢复过程也就是已知的退化图像 $G(u,v)$除以 $H(u,v)$，构成了一个反向的滤波过

程。用傅立叶变换的形式通过反向滤波的物理过程，对原图像的“估计”应该是 $G(u,v)/H(u,v)$的反变换。

$$\hat{f}(x,y) = \hat{f}^{-1}[G(u,v)/H(u,v)] = \hat{f}^{-1}[\hat{F}(u,v)] \tag{6.27}$$

式中　x,y 分别为 $0,1,2,\cdots,N-1$。

考虑噪声的存在，对原图像的估计应加上噪声的影响，可表达为

$$\hat{F}(u,v) = (u,v) + \frac{N(u,v)}{H(u,v)} \tag{6.28}$$

如果 $H(u,v)$变得很小或为零时，则噪声项 $N(u,v)/H(u,v)$对复原结果起很大的影响，甚至恢复了图像被噪声淹没而面目全非。实际的计算表明：$H(u,v)$在离开 u,v 平面的原点时，下降速度很快，这导致噪声项的影响急速上升。

退化模型为：$g = Hf + n$。那末，噪声项 $n = g - Hf$。从此表达式可以清楚地看出，若 n 越小，所求得的原始图像的估计 $\hat{f}$ 越能满足 $H\hat{f} = g$。当达到 $n = 0$，则 $H\hat{f} = g$。所以，若使 $J(\hat{f}) = \| g - H\hat{f} \| = \| n \|^2$ 最小，就可以寻找到最近似于 g 的 $H\hat{f}$。式中 $\| n \|^2$ 项称为噪声项的范数。

根据范数的定义　$\| n \|^2 = n^T n$

$$\| g - H\hat{f} \|^2 = (g - H\hat{f})^T (g - H\hat{f})$$

可以得到一个准则函数　$J(\hat{f}) = \| g - H\hat{f} \|^2$

求出对于 $\hat{f}$，使 $J(\hat{f})$ 为最小的一类问题、除了要求 $J(\hat{f})$ 为最小外，$\hat{f}$ 不以任何方式受到约束。使 $J(\hat{f})$ 为最小的问题可以用微分的办法来进行，就是将 $J(\hat{f})$ 对 $\hat{f}$ 求微分，并使其结果为零：

$$\frac{\partial J(\hat{f})}{\partial \hat{f}} = 0$$

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

得到　$\hat{f} = (H^T H)^{-1} H^T g$

若退化函数 H 为一方阵，并且 $M = N$，假设 H^{-1}的存在，则该式变为

$$\hat{f} = H^{-1}(H^T)^{-1} H^T g = H^{-1} g \tag{6.29}$$

根据上节的讨论得知：

$$\hat{f} = (WDW^{-1})^{-1} g = WD^{-1} W^{-1} g \tag{6.30}$$

$$W^{-1}\hat{f} = D^{-1} W^{-1} g$$

又根据 W 的性质，可以得到

$$\hat{F}(u,v) = \frac{G(u,v)}{N^2 H(u,v)} \tag{6.31}$$

从此可以看出，若已知 $g(x,y)$和 $h(x,y)$，也即知道了 $G(u,v)$和 $H(u,v)$，由此可以得到 $\hat{F}(u,v)$，再经过反变换，就能求出 $\hat{f}(x,y)$。这种图像恢复方法称为反向滤波。这种恢复的方法有一个特点，即对恢复的图像不做任何约束和规定。所以反向滤波方法又称为非约

束还原。

6.3.2 约束还原

恢复原始图像 $\hat{f}$ 的最终目标是要满足视觉效果，恢复其真面目。但如何通过求解达到这一目标，必须在数学表达上首先对 $\hat{f}$ 进行规定，使其某个指标达到一个极限值。

图像退化模型为：$g = Hf + n$

在对噪声项的精确求解目前还不能实现的情况下，就是要寻找一个 $\hat{f}$，使得 $H\hat{f}$ 最接近于 g，范数 $\| n \|^2$ 的最小化得到了成立。

$$\| n \|^2 = \| g - H\hat{f} \|^2$$

在最小化的问题中，可以利用拉格朗日乘数法进行。其做法是，首先把约束条件表达为 $\partial(\| g - H\hat{f} \|^2 - \| n \|^2)$，然后加上函数 $\| Q\hat{f} \|^2$，Q 为线性算子，用数学方法寻找准则函数为最小。

$$J(\hat{f}) = \| Q\hat{f} \|^2 + \partial(\| g - H\hat{f} \|^2 - \| n \|^2) \tag{6.31}$$

∂ 为一常数，称为拉格朗日乘子。加一个上述的约束条件，就可用微分方法进行最小化。

$$\frac{\partial J(\hat{f})}{\partial \hat{f}} = 0$$

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

求解 $\hat{f}$ 得到：$\hat{f} = (H^TH + \gamma^TQ)^{-1}H^Tg$ (6.32)

式中 $\gamma = 1/\alpha$，这个常数必须调整到约束被满足为止。

1) 约束最小平方滤波

在进行图像恢复计算时，由于退化算子矩阵 $H[\cdot]$ 的某些性质，多数在零点附近数值起伏过大，使得复原后的图像产生了多余的噪声和边缘。通过选择合理的 Q，并优化 $\| Qf \|^2$，从而可以去掉恢复的图像的这种尖锐的部分，即增加图像的平滑性。

从图像增强的拉普拉斯算子 $\nabla^2 f = (\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2})f$ 的计算过程可知，它具有突出边缘的作用。然而 $\iint \nabla^2 f \mathrm{d}x\mathrm{d}y$ 则恢复了图像的平滑性。所以，在作图像恢复时可将其作为约束。现在的问题是如何将其表达为 $\| Qf \|^2$ 的形式，从而可使用 $\hat{f} = (H^TH + \gamma Q^TQ)^{-1}H^Tg$。

如前所述，拉普拉斯算子 ∇^2 可用下面的差分运算实现：

$$\begin{aligned}\frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2} &= f(x+1,y) - 2f(x,y) + f(x-1,y) + f(x,y+1) \\ &\quad - 2f(x,y) + f(x,y-1) \\ &= f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) \\ &\quad - 4f(x,y)\end{aligned} \tag{6.33}$$

这个差分运算也可用 $f(x,y)$ 与下面的模板算子进行卷积来实现：

$$p(x,y) = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \tag{6.34}$$

在离散卷积的过程中，可利用延伸 $f(x,y)$ 和 $p(x,y)$ 来避免交叠误差。延伸后的函数为 $p_e(x,y)$。建立分块循环矩阵，将平滑准则表示为矩阵形式：

$$C=\begin{bmatrix} C_0 & C_{M-1} & C_{M-2} & \cdots & C_1 \\ C_1 & C_0 & C_{M-1} & \cdots & C_2 \\ C_2 & C_1 & C_0 & \cdots & C_3 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ C_{M-1} & C_{M-2} & C_{M-3} & \cdots & C_0 \end{bmatrix} \tag{6.35}$$

(6.35)式中每个子矩阵 C_j 是 $p_e(x,y)$ 的第 j 行组成的 $N\times N$ 循环矩阵。即 C_j 如下式表示：

$$C_j=\begin{bmatrix} p_e(j,0) & p_e(j,N-1) & \cdots & p_e(j,1) \\ p_e(j,1) & p_e(j,0) & \cdots & p_e(j,2) \\ \cdots & \cdots & \cdots & \cdots \\ p_e(j,N-1) & p_e(j,N-2) & \cdots & p_e(j,0) \end{bmatrix} \tag{6.36}$$

因为 C 是分块循环矩阵，它可用前述的矩阵 W 进行对角化，也就是说：

$$E=W^{-1}CW \tag{6.37}$$

式中，E 为对角线矩阵，其元素为

$$E(k,i)=\begin{cases} p([\frac{k}{N}]), \quad k_{\text{MOD }N} & \text{，若 } i=k \\ 0, & \text{若 } i\neq k \end{cases} \tag{6.38}$$

在此情况下，$P(u,v)$ 是 $p_e(x,y)$ 的二维傅立叶变换。并且，可以将 $\iint \nabla^2 f\mathrm{d}x\mathrm{d}y$ 写成 f^TC^TCf，定义 $Q=C$，则 $f^TC^TCf=\|Qf\|^2$。

实际上，如果要求约束条件 $\|g-Hf\|^2=\|n\|^2$ 得到满足，在 $Q=C$ 时，有

$$\begin{aligned} \hat{f} &= (H^TH+\gamma C^TC)^{-1}H^Tg \\ &= (WD^*DW^{-1}+\alpha WE^*EW^{-1})^{-1}WD^*W^{-1}g \end{aligned} \tag{6.39}$$

$$\begin{aligned} W^{-1}\hat{f} &= W^{-1}(WD^*DW^{-1}+\alpha WE^*EW^{-1})^{-1}WD^*W^{-1}g \\ &= (WD^*D+\gamma WE^*E)^{-1}WD^*W^{-1}g \\ &= (D^*D+\gamma E^*E)^{-1}D^*W^{-1}g \end{aligned} \tag{6.40}$$

式中 D^* 为 D 的共轭矩阵。

所以

$$\begin{aligned} \hat{F}(u,v) &= \frac{N^2H^*(u,v)}{N^4|H(u,v)|^2+\gamma N^2|P(u,v)|^2}G(u,v) \\ &= \frac{H^*(u,v)}{N^2|H(u,v)|^2+\gamma N^2|p(u,v)|^2}G(u,v) \end{aligned} \tag{6.41}$$

式中 u,v 分别为 $0,1,\cdots,2,N-1$，而且 $|H(u,v)|^2=H^*(u,v)H(u,v)$。本滤波器也称为最小平方滤波器。

2）维纳滤波

在进行图像分析时，可以将原始图像 f 和对原始图像的估计 $\hat{f}$ 看作为随机变量，因而该图像就可以视为随机过程。在一般情况下，可以认为它是一个平稳的随机过程。假设 R_f 和 R_n

为 f 和 n 的自相关矩阵，其定义为

$$R_f = E\{ff^T\} \tag{6.42}$$
$$R_n = E\{nn^T\}$$

式中 $E\{\cdot\}$ 代表数学期望运算。R_f 和 R_n 均为实对称矩阵，对大多数图像函数来说，f 和 n 的元素之间的相关不会延伸到20到30个像素的距离之外。因此，一般来说，自相关矩阵在主对角线附近有一个非零元素带，而在右上角和左下角的区域内将为零值。如果像素之间的相关是像素之间距离的函数，而不是它们位置的函数，数学分析表明 R_f 和 R_n 可以近似为分块循环矩阵。因而，可以用前述的 W 矩阵进行对角线化，用 A 和 B 代表矩阵，则有

$$R_f = WAW^{-1} \tag{6.43}$$
$$R_n = WBW^{-1}$$

可以证明，A 和 B 的元素分别为 R_f 和 f_n 中的自相关元素的傅立叶变换。这些自相关的傅立叶变换被分别定义为 $f_e(x,y)$ 和 $n_e(x,y)$ 的谱密度 $S_f(u,v)$ 和 $S_n(u,v)$。

定义 $Q^TQ = R_f^{-1}R_n$，将此式代入 $\hat{f} = (H^TH + \gamma Q^TQ)^{-1}H^Tg$，得到

$$\hat{f} = (H^TH + \gamma R_f^{-1}R_n)^{-1}H^Tg \tag{6.44}$$

进一步可推导出：

$$\hat{f} = (WD^*DW^{-1} + \gamma WA^{-1}BW^{-1})^{-1}WD^*W^{-1}g \tag{6.45}$$

经过矩阵变换可得：

$$W^{-1}\hat{f} = (D^*D + \gamma A^{-1}B)^{-1}D^*W^{-1}g \tag{6.46}$$

根据 A，B 的元素的意义，括号内的矩阵均为对角线矩阵。

假设 $M = N$，则

$$\hat{F}(u,v) = \left[\frac{H^*(u,0)}{|H(u,v)|^2 + \gamma[S_n(u,v)/S_f(u,v)]}\right]G(u,v)$$
$$= \left[\frac{1}{H(u,v)} \cdot \frac{|H(u,v)|^2}{|H(u,v)|^2 + \gamma[S_n(u,v)/S_f(u,v)]}\right]G(u,v) \tag{6.47}$$

对于 u,v 分别为 $0,1,\cdots,2,N-1$，式中 $|H(u,v)|^2 = H^*(u,v)H(u,v)$。当 $\gamma = 1$，称之为维纳滤波器。若 γ 为变数，此式为参变维纳滤波器。当无噪声影响时，即 $S_n(u,v) = 0$，称之为理想的反向滤波器。γ 必须调正到约束条件 $\|g - Hf\|^2 = \|n\|^2$。显然 $\gamma = 1$，得到的解是最佳的，意思是它使数学期望 $E\{[f(x,y) - \hat{f}(x,y)]^2\}$ 最小。

维纳滤波器对噪声放大有自动抑制作用。如果 $H(u,v)$ 在某处为零，由于有 $\frac{S_n(u,v)}{S_f(u,v)}$，就不会出现被零除的情形。在某一频谱区，若信噪比很高，即 $S_n(u,v) \ll S_f(u,v)$，滤波器的效果趋向于反向滤波。若信噪比很小，即 $S_n(u,v) \gg S_f(u,v)$，滤波器表现不敏感。这说明维纳滤波器在反向滤波的过程中，避免了对噪声的放大作用。

在使用参变维纳滤波法时，$H(u,v)$ 由点扩展函数确定，而当噪声是白噪声时，$S_n(u,v)$ 为常数，可通过计算一幅噪声图像的功率谱求解。

由于 $S_g(u,v) = |H(u,v)|^2 S_f(u,v) + S_n(u,v)$，所以 $S_f(u,v)$ 可通过本式求得。

6.4 点冲激函数的退化模型

像素点是图像的基本要素，根据前述的抽样定理已知，所谓点冲激函数就是在二维视场

中，设 x 方向间隔为 Δx 个单位，以及设 y 方向间隔为 Δy 个单位的点脉冲。由于成像设备的非线性或记录载体与景物的相对位移，结果成像后就不再是一个点了，可能是一个模糊的像团，也可能是类似带尾的慧星云团等。如何将其原始图像的点函数恢复出来，对于整个由点像素集组成的二维图像的恢复是极为重要的。同时，退化了的点像的恢复也为退化过程准备了先验知识。

二维冲激函数在坐标为(x_0, y_0)的表达式为

$$\iint_{-\infty}^{\infty} f(x,y)\delta(x - x_0, y - y_0)\mathrm{d}x\mathrm{d}y = f(x_0, y_0) \tag{6.48}$$

如果点光源是一个单位冲激函数的近似，那么可得

$$G(u,v) = H(u,v)F(u,v) \approx H(u,v) \tag{6.49}$$

这是因为 $f[\delta(x - x_0, y - y_0)] = 1$。此式的物理意义在于，退化函数$H(u,v)$可由点光源图像的傅立叶变换来近似。在实际应用中，可将一点光源图像加以模糊，便可得到$H(u,v)$的近似。

假设有一个点光源产生了运动，在成像系统中，胶片上的总曝光量是在快门打开时间 T 内对瞬时曝光进行积分得到的。若光学成像过程是正常的，在曝光时间 T 内，由于运动而得到的模糊图像为

$$g(x,y) = \int_0^T f[x - x_0(t), y - y_0(t)]\mathrm{d}t \tag{6.50}$$

其傅立叶变换为

$$\begin{aligned} G(u,v) &= \iint_{-\infty}^{\infty} g(x,y)\exp[-j2\pi(ux + vy)]\mathrm{d}x\mathrm{d}y \\ &= \iint_{-\infty}^{\infty} \left[\int_0^T f[x - x_0(t), y - y_0(t)]\mathrm{d}t\right]\exp[-j2\pi(ux + vy)]\mathrm{d}x\mathrm{d}y \\ &= \int_0^T \left[\iint_{-\infty}^{\infty} [f(x - x_0(t), y - y_0(t))\mathrm{d}t]\exp[-j2\pi(ux + vy)]\mathrm{d}x\mathrm{d}y\right]\mathrm{d}t \\ &= \int_0^T F(u,v)\exp[-j2\pi(ux_0(t) + vy_0(t))]\mathrm{d}t \end{aligned} \tag{6.51}$$

由于 $G(u,v) = H(u,v)F(u,v)$

所以可以得出退化函数为

$$H(u,v) = \int_0^T \exp[-j2\pi(ux_0(t) + vy_0(t))]\mathrm{d}t \tag{6.52}$$

设点光源在 x 方向上作匀速直线运动，其速率为 $x_0(t) = at/T$，当 $t = T$ 时，光源移动的总距离为 a，因为 $y_0(t) = 0$，所以它的退化函数 $H(u,v)$为

$$\begin{aligned} H(u,v) &= \int_0^T \exp[-j2\pi ux_0(t)]\mathrm{d}t \\ &= \int_0^T \exp[-\mathrm{j}2\pi uat/T]\mathrm{d}t \\ &= \frac{T}{\pi ua}\sin(\pi ua)\mathrm{e}^{-\mathrm{j}\pi a} \end{aligned} \tag{6.53}$$

本式也称为点扩展函数 PSF。

光学系统的聚焦不好，会造成一个模糊的图像，这是众所周知的。由于没有景物与光学

系统间的相对运动，因而光学系统散焦的点扩展函数是圆函数，经过推导可得表达式为

$$H(u,v) = J_1(\pi d\rho)/\pi d\rho \tag{6.54}$$

式中 $\rho=\sqrt{u^2+v^2}$

在光学系统成像的过程中，由于大气湍流等介质的影响，也会造成图像的模糊，这个影响显然与介质的湍流过程有关，其点扩展函数可表示为

$$H(u,v) = \exp[-c(u^2+v^2)^{5/6}] \tag{6.55}$$

式中 c 是与湍流性质有关的常数。

6.5 几何畸变校正

由于成像系统的非线性，成像后的图像与原景物图像相比，产生比例失调，被描述的景物产生扭曲，这类图像退化现象称为几何畸变，解决这类失真的问题称为几何畸变校正。

设原图像为$[f(x,y)]$，畸变后的图像为$[f(x',y')]$，要将$[f(x',y')]$恢复为$[f(x,y)]$，就是要找到(x',y')的坐标与(x,y)的坐标的转换关系，这个转换关系称为坐标变换，用 Ta 表示如下式：$(x,y) = Ta(x',y')$

景物在成像过程中产生的扭曲，会使图像比例失调。对此，可用仿射变换来校正各种畸变，即找出坐标转换的系数。可以写成

$$\begin{aligned} x &= Ax' + By' + c \\ y &= dx' + ey' + D \\ f(x,y) &= f[(Ax' + By' + C),(Dx' + Ey' + D] \end{aligned} \tag{6.56}$$

所以，几何畸变的校正归根结底为坐标转换系数 A,B,C,D,E,F 的求解。为了解决成像系统的非线性的补偿因子，可以在畸变了的景物图像中找出三个可识别的特殊点的坐标(x_1',y_1'),(x'_2,y'_2),(x'_3,y'_3)，它们对应于原景物图像中的三个特殊点的坐标为(x_1,y_1),(x_2,y_2),(x_3,y_3)。它们之间的关系必然为

$$\left.\begin{aligned} x_1 &= Ax'_1 + By'_1 + C \\ y_1 &= Dx'_1 + Ey'_1 + F \end{aligned}\right\} \tag{6.57}$$

$$\left.\begin{aligned} x_2 &= Ax'_2 + By'_2 + C \\ y_2 &= Dx'_2 + Ey'_2 + F \end{aligned}\right\} \tag{6.58}$$

$$\left.\begin{aligned} x_3 &= Ax'_3 + By'_3 + C \\ y_3 &= Dx'_3 + Ey'_3 + F \end{aligned}\right\} \tag{6.59}$$

由这六个方程组可以求解成像系统的非线性的补偿因子 A,B,C,D,E,F。由线性代数可得出下列变换矩阵：

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} x'_1 & y'_1 & 1 \\ x'_2 & y'_2 & 1 \\ x'_3 & y'_3 & 1 \end{bmatrix} \begin{bmatrix} A \\ B \\ C \end{bmatrix} \tag{6.60}$$

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} x'_1 & y'_1 & 1 \\ x'_2 & y'_2 & 1 \\ x'_3 & y'_3 & 1 \end{bmatrix} \begin{bmatrix} D \\ E \\ F \end{bmatrix} \tag{6.61}$$

又写如下：$H=\begin{bmatrix} x'_1 & y'_1 & 1 \\ x'_2 & y'_2 & 1 \\ x'_3 & y'_3 & 1 \end{bmatrix}$

由此得到 $A—F$ 的最小二乘方估计 $\hat{A}—\hat{F}$：

$$\begin{bmatrix} \hat{A} \\ \hat{B} \\ \hat{C} \end{bmatrix} = (H^T H)^{-1} H^T \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \tag{6.62}$$

$$\begin{bmatrix} \hat{D} \\ \hat{E} \\ \hat{F} \end{bmatrix} = (H^T H)^{-1} H^T \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} \tag{6.63}$$

所以，几何畸变的校正后的坐标表示为

$$\left.\begin{aligned} x &= [\hat{A}\,x + \hat{B}\,y' + \hat{C}] \\ y &= [\hat{D}\,x' + \hat{E}\,y' + \hat{F}] \end{aligned}\right\} \tag{6.64}$$

由此得到的图像即为恢复后的图像 $f(x,y)$。

6.6 图像重建的基本概念

一幅图像进行量化处理后，得到一个二维数组，这个过程是从图像到数据，而这个数据描述了图像。有了这个描述图像的数据，就可以进行各种处理，从而进入图像处理的领域。然而，人们如果有一组与图像有关的物理数据，能否反过来建立图像呢？这就是图像重建所要解决的问题了，因而可以将图像重建理解为上述过程的逆过程。自从英国科学家 Housfield 在 60 年代发明了断层扫描重建图像以后，图像重建这一领域迅速地发展起来。Housfield 将物体横剖面的一组投影数据，经过他的算法的处理，得到物体横剖面的二维图像。在放射学、核医学、非破坏性工业测试、数据压缩等领域，图像重建技术得到了广泛的应用，显示出了它的重要的价值。这组物体内部断层数据是由能量流穿过物体而获得的，这种能量流通常有 X 射线束、电子束、超声波。关于这些能量流的产生与使用不属本章的讨论范围，本章着重讨论图像重建的实现算法。断层扫描重建图像在医学上应用特别广泛，称为 CT 摄影。

6.7 图像重建的方法

6.7.1 平行投影

图 6.2 表示了图像重建的原理。假设有一图像区域 $f(x,y)$，图像重建的任务则是从可测量的数据得到 $f(x,y)$ 的区域图像。图中 (x,y) 坐标系为图像坐标系。若将 (x,y) 图像坐标系旋转 θ 角，就可以得到投影坐标系 (t,s)。因而，两个坐标系的转换关系可以如式

(6.65)的结果：

$$\begin{bmatrix} t \\ s \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{6.65}$$

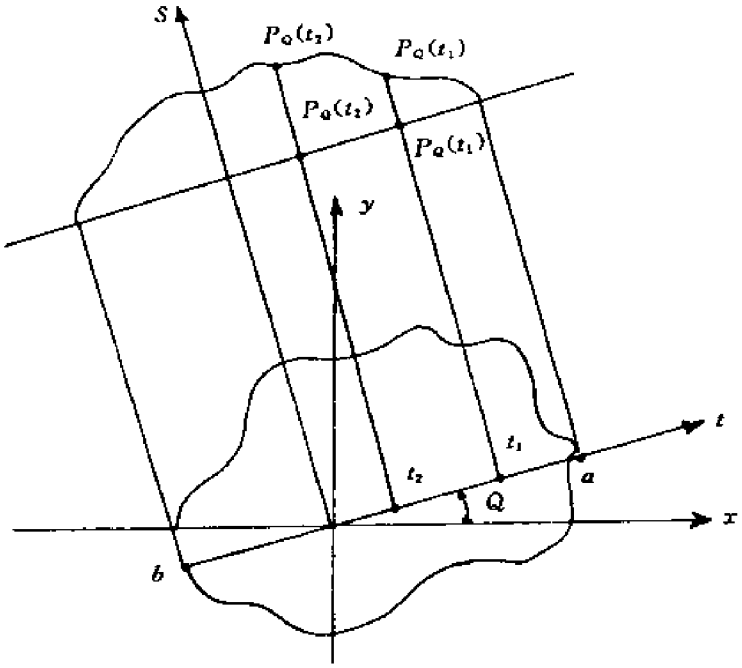


图 6.2　图像投影的原理

在(t,s)坐标系中，s方向称为射线方向，沿射线方向将图像$f(x,y)$进行积分，也就是说，s是积分的途径，见下式：

$$P_Q(t) = \int_s f(x,y)\mathrm{d}s \tag{6.66}$$

这样，在(t,s)坐标系中，垂直于t轴的方向上取t_1、t_2，所以到s射线方向的积分为

$$\left.\begin{aligned} P_Q(t_1) &= \int_s f(x,y)\mathrm{d}s \\ P_Q(t_2) &= \int_s f(x,y)\mathrm{d}s \end{aligned}\right\} \tag{6.67}$$

沿着一组垂直于t的射线方向，取$f(x,y)$的积分组成了一束平行的投影束，称之为平行投影。所以，当Q为某常数时，$P_Q(t)$就是t的函数。换句话说，$P_Q(t)$为$f(x,y)$在θ角位置的平行投影，人们也称之为$f(x,y)$的雷顿变换。之所以称为雷顿变换，是因为奥地利数学家雷顿(Radon)在1917年提出了一个假设，他认为，二维或三维的物体可以通过无限多个投影来进行重建。

平行投影的结果，在t轴上，从$f(x,y)$的包络点a积分到点b，就可以得到$P_Q(t)$曲线。$P_Q(t)$曲线反映了射线穿过$f(x,y)$区域断面的性质。很明显，如果人们用物理实测的方法获得类似$P_Q(t)$的曲线，那就是说，人们可以得到当θ角为一常数时，区域$f(x,y)$断面图像的性质。若θ围绕原点环绕一周作360°的旋转变化，就可以使人们运用数学方法在环绕一周的方向上获得区域$f(x,y)$的断面图像组成信息，如图6.3所示。图中对应于Q_1的平行投影为$P_{Q1}(t)$和$P_{Q2}(t)$两条曲线。

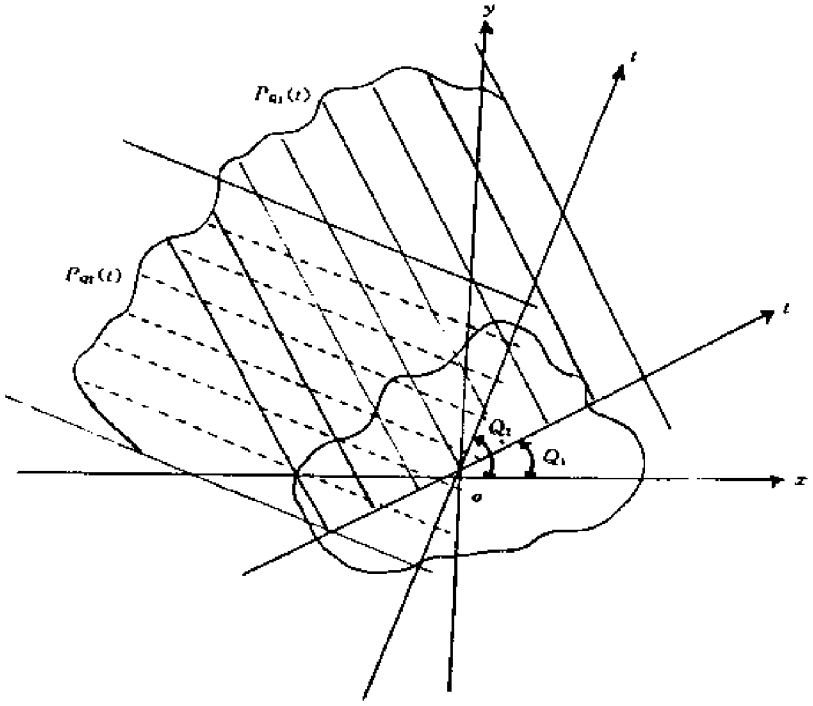


图 6.3 当 Q 变化时的平行投影

6.7.2 获得平行投影的物理方法

为了获得 $P_Q(t)$平行投影曲线，人们自然想到具有穿透力的 X 射线，或其它光子、电子射线。当物理射线穿过物体时，由于物体内部组织的吸收、散射和反射等作用，会使射线的强度衰减，射线的衰减规律表示为

$$N_{out} = N_{in} \exp[-\int_s u(x,y)ds] \tag{6.68}$$

式中 s 为射线方向，表示沿 s 方向的积分。

$u(x,y)$为射线穿过物体时的衰减系数，它正好反映了物体内部的组成性质，如物体内部的密度变化情况等。这样，$u(x,y)$表征了物体断面的图像 $f(x,y)$的组成信息。然而 N_{in} 为入射的光子数，N_{out}为穿透物体后收集到的光子数。这两个物理量可以通过物理测量得到，那么表示断面图像的平行投影也可从(6.69)式得到

$$P_Q = \int_s f(x,y)ds = \int_s u(x,y)ds = \ln(N_{in}/N_{out}) \tag{6.69}$$

在医学计算机数据图像摄影中使用的不是平行射线束，而是点源出发的辐射线束，如图 6.4 所示，其原理同平行射线束，主要是为了获得 $P_Q(t)$平行投影曲线。图中点源为 s_1、s_2，其中轴线夹角与 y 轴夹角分别为 Q_1、Q_2，可以用物理方法测得，并计算得到 $P_{Q1}(t)$和 $P_{Q2}(t)$曲线。

6.7.3 图像重建的傅立叶变换法

假设已知二维图像 $f(x,y)$的傅立叶变换，那么其表达式为

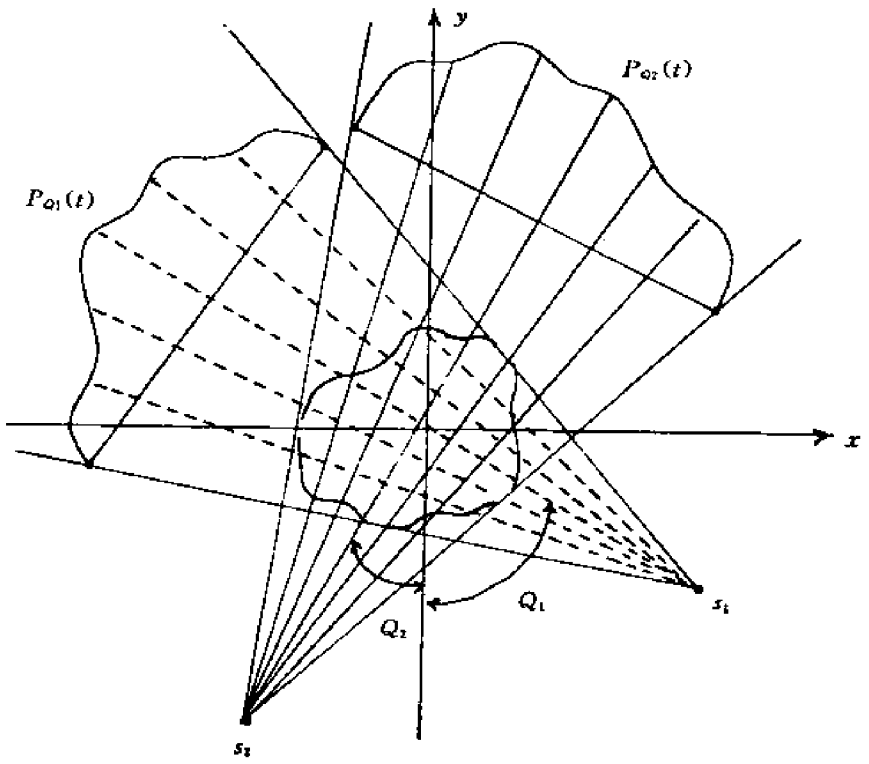


图 6.4　点源辐射线投影原理

$$F(u,v) = \iint_{-\infty}^{\infty} f(x,y)\exp[-j2\pi(ux+vy)]\mathrm{d}x\mathrm{d}y \tag{6.70}$$

平行投影 $P_Q(t)$的傅立叶变换为

$$S_Q(w) = \int_{-\infty}^{\infty} P_Q(t)\exp[-j2\pi wt]\mathrm{d}t \tag{6.71}$$

在 $f(x,y)$傅立叶变换 $F(u,v)$的频域(u,v)中,取 $v=0$,则其变换为

$$\begin{aligned} F(u,0) &= \iint_{-\infty}^{\infty} f(x,y)\exp(-j2\pi ux)\mathrm{d}x\mathrm{d}y \\ &= \int_{-\infty}^{\infty}\left[\int_{-\infty}^{\infty} f(x,y)\mathrm{d}y\right]\exp[-j2\pi ux]\mathrm{d}x \end{aligned} \tag{6.72}$$

再看平行投影 $P_Q(t)$,若 $Q=0$,则(t,s)坐标系与(x,y)坐标系重合,$t=x$,$s=y$,那末平行投影 $P_Q(t)$变为 $P_0(t)$,即为

$$P_0(t) = \int_{-\infty}^{\infty} f(x,y)\mathrm{d}y \tag{6.73}$$

这样,变换 $F(u,0)$可写为

$$F(u,0) = \int_{-\infty}^{\infty} P_0(t)\exp(-j2\pi ut)\mathrm{d}x \tag{6.74}$$

式(6.74)的右边为 $f(x,y)$在垂直于 x 轴方向上的一维傅立叶变换,它等于 $f(x,y)$的傅立叶变换当频域 $v=0$ 时的径线上的值 $F(u,0)$。由此可以推理,$F(\omega,\theta)$是变换 $F(u,v)$沿 Q 方向径线上的值。$S_Q(\omega)$则为 $P_Q(t)$投影的一维傅立叶变换。那么得到

$$F(\omega,\theta) = S_Q(\omega) \tag{6.75}$$

在(6.75)式中，右边 $S_Q(\omega)$可以通过物理方法测量射线的衰减得到。那么，$F(\omega,\theta)$所对应的在 $\theta_1,\theta_2,\cdots,\theta_k$ 下的各个投影均可得到，即 $P_Q(t)$的值。当平行投影的数目很大时，就能得到 $F(u,v)$在频域(u,v)平面的一组数值，再用反变换，由(6.76)式得到其图像值 $f(x,y)$为

$$f(x,y)=\iint_{-\infty}^{\infty}F(u,v)\exp[\mathrm{j}2\pi(ux+vy)]\mathrm{d}u\mathrm{d}v \tag{6.76}$$

6.7.4 滤波逆投影法

在傅立叶变换的频域(u,v)，若用(ω,θ)一组极坐标代替，那末其极坐标的表达为

$$\left.\begin{aligned}u&=\omega\cos\theta\\v&=\omega\sin\theta\end{aligned}\right\} \tag{6.77}$$

若将极坐标形式代入傅立叶变换，可得

$$\begin{aligned}f(x,y)&=\int_0^{2\pi}\int_0^{\infty}F(\omega,\theta)\exp[\mathrm{j}2\pi\omega(x\cos\theta+y\sin\theta)]\omega\mathrm{d}\omega\mathrm{d}\theta\\&=\int_0^{\pi}\int_0^{\infty}F(\omega,\theta)[\mathrm{j}2\pi\omega(x\cos\theta+y\sin\theta)]\omega\mathrm{d}\omega\mathrm{d}\theta\\&\quad+\int_0^{\pi}\int_0^{\infty}F(\omega,\theta+\pi)\exp[\mathrm{j}2\pi\omega(x\cos\theta+y\sin(\theta+\pi)]\omega\mathrm{d}\omega\mathrm{d}\theta\end{aligned} \tag{6.78}$$

根据傅立叶变换的共轭对称特性，可以得到 $F(\omega,\theta+\pi)=F(-\omega,\theta)$，将此式代入(6.78)式，可得

$$\begin{aligned}f(x,y)&=\int_0^{\pi}\left[\int_{-\infty}^{\infty}F(\omega,\theta)\mid\omega\mid\exp(\mathrm{j}2\pi\omega t)\mathrm{d}\omega\right]\mathrm{d}\theta\\&=\int_0^{\pi}\left[\int_{-\infty}^{\infty}S_\theta(\omega)\mid\omega\mid\exp(\mathrm{j}2\pi\omega t)\mathrm{d}\omega\right]\mathrm{d}\theta\end{aligned} \tag{6.79}$$

若令 $Q_\theta(T)=\int_{-\infty}^{\infty}S_\theta(\omega)\mid\omega\mid\exp(\mathrm{j}2\pi\omega t)\mathrm{d}\omega$，得到

$$f(x,y)=\int_0^{\pi}Q_\theta(t)\mathrm{d}\theta=\int_0^{\pi}Q_\theta(x\cos\theta+y\sin\theta)\mathrm{d}\theta \tag{6.80}$$

式中 $t=x\cos\theta+y\sin\theta$

根据物理射线测量得到 $\ln(N_{in}/N_{out})$，即为 $P_\theta(t)$。由前节知道 $S_\theta(\omega)\sim P_\theta(t)$的关系，所以 $Q_\theta(t)$可以求出。那么，由此可重建二维图像 $f(x,y)$。因为 $S_\theta(\omega)$相当于一个滤波器，所以这里所讨论的图像重建方法也称为滤波—逆投影法。

7 图像特征与分析

●纹理特征 ●灰度共生矩阵 ●形状特征
●区域边界 ●矩阵方法与投影 ●视差与光流场

图像处理的众多方法，如前面的章节所分析讨论的那样，其最终目的在于增强视觉效果，使得原来看不清的图像看得清了，难以辨认的图像能辨认得清了。但是，人们的视觉系统认识的图像如何能让计算机系统也能认识呢？必须寻找出算法，分析图像的特征，然后将其特征用数学的办法表示出来并教会计算机也能懂得这些特征，这样，计算机也就有了认识或者叫识别图像的本领了。这称为机器学习，也叫模式识别。要使计算机具有识别的本领，首先要分析这些特征，称为图像分析。所以，图像分析是图像处理后的进一步的工作。

图像分析工作的结果给出了某一具体的图像中与其它图像相区别的特征。例如，描述物体表面灰度变化的纹理特征，描述物体外形的，如弯曲、大小的形状特征等。这类分析的结果需要一定的表达方式，要让计算机懂得，这就是本章的任务。

7.1 纹理特征与分析

7.1.1 纹理分析的基本概念

讲到纹理，人们自然会立刻想到木制家俱上的木纹，又会立即想到人的衣着花布上的花纹，木纹为天然纹理，花纹为人工纹理，它们反映了物体表面颜色和灰度的某种变化，而这些变化又与物体本身的属性相关。比如同一个树种的木材有相同或相近似的纹理，人们通过识别木纹的种类识别木材的树种和材质。

图像的纹理分析已在许多学科得到了广泛的应用。气象云图多是纹理型的，在红外云图上，各种云类呈现的纹理特征完全不同，所以几种不同纹理特征的云类，如卷云、积雨云、积云和层云的机器识别就可以用纹理作为一大特征。

卫星遥感地表图像，相当于人们站在宇宙空间来看地球，地表的山脉、草地、沙漠、大片森林、城市建筑群等均表现了不同的纹理特征。分析卫星遥感图像的纹理特征，可以进行区域识别，国土整治，森林利用，城市发展，土地荒漠化等在国民经济中各方面很有价值的宏观研究及应用。

在显微图像中，如细胞图像、金相图像、催化剂表面图像等均具有明显的纹理特征，对于它们的纹理结构的分析，可以得到细胞性质的鉴别的信息，金相结构物理信息和催化剂的活性信息。

图像的纹理从一般意义上来说是图像中灰度和颜色的变化，反复出现的纹理基元和它

的排列规则。什么叫纹理基元？众所周知，图像中最基本的单元是像素，而由像素组成的具有一定形状和大小的集合，如条状、丝状、圆斑、块状等，则称为纹理基元。所以，纹理分析的任务，则首先从像素出发，检测出纹理基元，并找出纹理基元排列的信息，建立纹理基元模型。如果已知纹理基元，需要描述基元的排列规则，称为纹理的结构分析。

纹理分析方法，大致分为统计方法和结构方法。例如前述遥感图像中的森林、山脉、草地的纹理细而无规则，一般采用统计方法。衣着花布图案等较规则的图像，一般采用结构方法。统计方法又分空间域方法和变换域方法。方法很多，本章着重介绍其中最经典、最常用的几种。

7.1.2 纹理分析中的直方图方法

1) 窗口直方图法

直方图是图像窗口中，多种不同灰度的像素分布的概率统计。视觉系统所观察到的图像窗口中的纹理基元必然对应于一定的概率分布的直方图，其间存在着一定的对应关系。根据这个特点，就可以让计算机来进行两个适当大小的图像窗口的纹理基元的计算和分析。若已知两个图像窗口中的一个窗口里的纹理基元，且两个窗口的直方图相同或相似，则说明第二个窗口中可能具有类似第一个窗口的纹理基元。若将连续的图像窗口的直方图的相似性进行比较，就可以发现及鉴别纹理基元排列的周期性及紧密性等。

需要指出的是，相同的纹理基元具有相同的直方图，但相同的直方图可能会有不同的纹理基元相对应，所以在运用直方图进行纹理基元的分析和比较时，还要加上基元的其它特征。

2) 边缘方向直方图

所谓图像边缘指的是图像中感兴趣的景物或区域与其余部分的分界。因而，图像边缘往往包含有大量的二维信息。取沿着边缘走向的像素的邻域，分析其直方图，若在直方图上的某一个灰度范围内具有尖峰，就可以说明在这个灰度范围内，纹理所具有的方向性。利用这样的办法，就可以识别某些纹理。

3) 图像灰度梯度方向矩阵

如果在一个小的图像区域内确定其方向，并将若干个小区域的方向加以综合，就可以找出该区域的纹理基元或纹理走向。如果选择 16 个图像像素为一个图像区域，那么每 4 个像素则组成一个小区域，如图 7.1 所示。

下面的问题是计算小区域的灰度梯度，找出其方向。图中 abcd 四个像素组成的小区域，计算出 8 个可能的方向的灰度梯度的差分值为

$$\left.\begin{aligned} G_0 &= f(\mathrm{a}) - f(\mathrm{b}) - [f(\mathrm{c}) - f(\mathrm{d})] \\ G_1 &= \sqrt{2}[f(\mathrm{b}) - f(\mathrm{c})] \\ G_2 &= f(\mathrm{b}) + f(\mathrm{d}) - [f(\mathrm{a}) + f(\mathrm{c})] \\ G_3 &= \sqrt{2}[f(\mathrm{d}) - f(\mathrm{a})] \\ G_4 &= -G_0 \\ G_5 &= -G_1 \\ G_6 &= -G_2 \\ G_7 &= -G_3 \end{aligned}\right\} \tag{7.1}$$

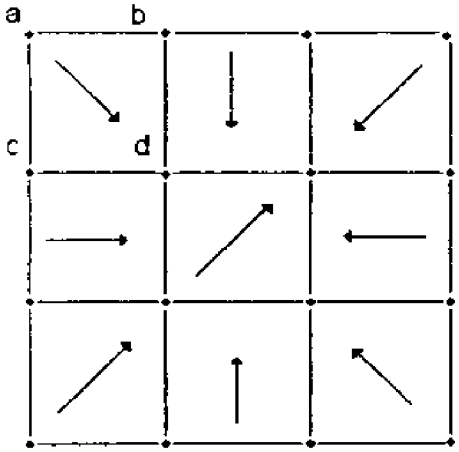


图 7.1　图像灰度区域

若 $G_0>0$,则 G_0 指向上方;而若 $G_0<0$,则 G_0 指向下方。同理,若 $G_1>0$,则 G_1 指向斜上方,与横轴沿顺时针方向夹角 45°;而若 $G_1<0$,则 G_1 指向斜下方,与横轴沿逆时针方向夹角 45°。同样的方法可以推出 $G_2, G_3, G_4, G_5, G_6, G_7$ 的方向。对于这 8 个小方向,取其中最大值的方向作为该小区域的梯度方向。在计算完这 9 个小区域的方向后,就可计算这个 16 像素的图像区域中所有不同梯度方向的数目,可取最大数目的梯度方向为该图像区域的方向。用此 16 像素的图像区域为模板,在整个图像上平移计算,并进行分类,就可得出整个图像的灰度梯度方向,就是整个图像的纹理信息,包括纹理走向、纹理形状、纹理疏密等。

7.1.3　纹理分析的自相关函数方法

从遥感航片或卫片上,观察地球表面,地块的纹理特征非常突出,不同类型的地块具有不同的纹理特征。有一位名叫凯泽的遥感科学家将北极航空照片中取出七类不同的地表覆盖物的图像进行自相关函数的纹理分析,利用计算机识别,与目测比较,正确率达到 99%。

什么样的自相关函数呢? 若有一个图像 $f(i,j)$, i,j 分别为 $0,1,2,3,\cdots,N-1$,其自相关函数定义为

$$p(x,y)=\frac{\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}f(i,j)f(i+x,j+y)}{\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}f(i,j)} \tag{7.2}$$

$p(x,y)$也可以看作为一幅图像,其大小为 $N\times N$。若 $i+x>N-1$ 或 $j+y>N-1$,那末,$f(i+x,j+y)=0$,也就是说,图像之外的图像值为零。

自相关函数 $p(x,y)$随 x,y 大小而变化,与图像中纹理粗细的变化有着对应的关系。定义 $d=\sqrt{x^2+y^2}$,若 $x=0,y=0$ 时,从自相关函数定义可以看出,$p(x,y)=1$,为最大值。凯泽从计算 7 种不同的地表覆盖物的图像自相关函数发现,随着 d 的增加,$p(d)$呈下降状态。但对于纹理较粗的地物和纹理较细的地物的效果是不一样的:当纹理较粗时,$p(d)$随 d 的增加下降速度较慢;而若纹理较细时,$p(d)$随着 d 的增加下降较快。随着 d 的继续增加,$p(d)$则会呈现某种周期性的变化,其周期大小则描述了纹理基元的分布的稠密和稀疏程度。凯泽从北极航空照片中取出 7 类地表覆盖物的纹理图像,计算 $p(d)$与 d 的变化关系

曲线如图 7.2 所示。

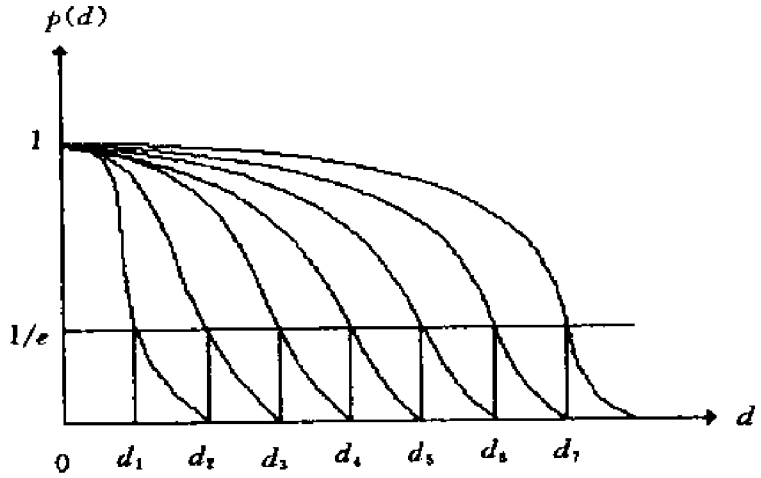


图 7.2　凯泽试验的 7 种纹理的自相关纹理分析曲线

从图 7.2 示的曲线，可以看出凯泽试验的效果是，$p(d)$下降速度快，对应较细的纹理；$p(d)$下降速度慢，对应于较粗的纹理。所以从第 1 个样本到第 7 个样本，是由细纹理到粗纹理，对应于 $p(d)$的同一个值 $1/e$，7 条曲线的值分别为 $d_1, d_2, d_3, d_4, d_5, d_6, d_7$。凯泽请来 20 个观测者，按纹理粗细进行目视判别，将 7 张纹理图像按纹理粗细进行排列，发现结果与计算基本一致。

7.1.4　纹理的灰度分布统计特征分析

1）灰度共生矩阵的纹理分析

任何图像都可以看作为三维空间中的一个曲面，直方图是研究单个像素在这个三维空间中的统计分布规律，但不能很好地反映像素之间的灰度级空间相关性的规律。在三维空间中，相邻某一间隔长度的 2 个像素，它们具有相同的灰度级，或者具有不同的灰度级，若能找出这样 2 个像素的联合分布的统计形式，对于图像的纹理分析将是很有意义的。人们找到描述这个关系的基础，从灰度为 i 的像素点出发，距离为(Dx, Dy)的另一个像素点的同时发生的灰度为 j，定义这 2 个灰度在整个图像中发生的概率，或称为频度，用数学表示则为

$$P(i,j,\delta,\theta) = 集合\{(x,y) \mid f(x,y)=i, f(x+Dx, y+Dy)=j; x,y=0,1,2,\cdots,N-1\} \tag{7.3}$$

式中　$i,j=0,1,2,\cdots,L-1$；

x,y 是图像中的像素坐标，L 为灰度级的数目。

这样，2 个像素灰度级同时发生的概率，就将(x,y)的空间坐标转换为(i,j)的“灰度对”的描述，它们形成的矩阵称为灰度共生矩阵。灰度共生矩阵可以理解为像素对或灰度级对的直方图。这里所说的像素对和灰度级对是有特定含义的，一是像素对的距离不变，二是像素灰度差不变。距离 δ 由(Dx, Dy)构成，如图 7.3 所示。

很明显，若 $Dx=1, Dy=0$，则 $\theta=0°$；$Dx=1, Dy=-1$，则 $\theta=45°$；$Dx=0, Dy=-1$，则 $\theta=90°$；若 $Dx=-1, Dy=-1$，则 $\theta=135°$。根据上述定义，所构成的灰度共生矩阵是一个集合，集合中的一个元素$[p(i,j,\delta,\theta)]$为第 i 行、第 j 列矩阵元素，表示所有在 θ 方向上、相邻间隔为 δ 的像素，一个为灰度 i 值，另一个为灰度 j 值的相邻点对数。这里所说的 θ 方向，

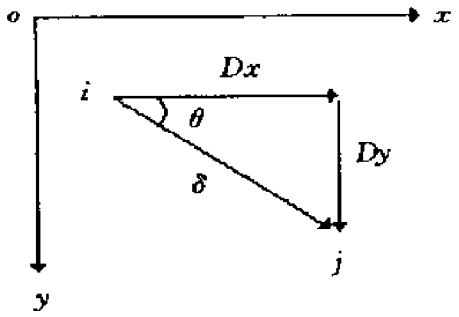


图 7.3　共生矩阵的像素对

一般取值 0°,45°,90°和 135°等 4 个方向。可以看出,灰度共生矩阵反映了图像灰度关于方向、相邻间隔、变化辐度的综合信息,它确实可以作为分析图像基元和排列结构的信息。目前一辐图像的灰度级数目一般是 256,这样计算出来的灰度共生矩阵太大。为了解决这一问题,常常在求灰度共生矩阵之前,根据直方图提供的方法,变换为 16 级的灰度图像。如图 7.4 所示的 2 幅灰度图像,从直觉观察可知,它们的纹理具有不同的模式。

图 7.4 示出 0°,45°,90°和 135°等 4 个方向的共生矩阵。从 2 幅图像的 8 个共生矩阵可以看出一个重要的特点,所有 8 个矩阵均是以主对角线为对称轴,两边对称。从共生矩阵 $P_A(0°)$可以看出,其主对角线上元素全部为 0,这说明水平方向上灰度变化的频度高,纹理较细。然而,共生矩阵 $P_B(0°)$则是主对角线上的元素值很大,表明水平方向上灰度变化的频度较低,说明纹理粗糙。再看 $P_A(135°)$的共生矩阵,主对角线上的元素值很大,其余元素为 0,说明该图像沿 135°方向无灰度变化。但是图像 B 沿 135°方向,偏离主对角线的元素值较大,说明纹理较粗。

为了利用灰度共生矩阵所提供的图像灰度方向、间隔和变化幅度的信息,则在共生矩阵的基础上抽取出其纹理特征,称为二次统计量。为使二次统计量的表达式清晰简单,取 $Dx=1,\theta=0°$,并作正规化处理如下:

$$\hat{p}(i,j)=p(i,j)/R \tag{7.4}$$

式中　R 为正规化常数。

R 含义是相邻点对的组合数,若 $Dx=0,\theta=0°$,每行邻对的可能的组合数为 $2(N-1)$,图像共 N 行,则 $p(i,j)$的元素之和为 $2N(N-1)$。正规化处理后,其元素之和为 1。经过正规化处理的灰度共生矩阵即是抽取二次统计量纹理特征系数的基础。这些纹理特征系数作为纹理识别,分类是非常有用的。在具体使用这些纹理系数时,可以应用到整幅图像,也可以用到图像中的局部窗口,或称为图像中的某个区域。

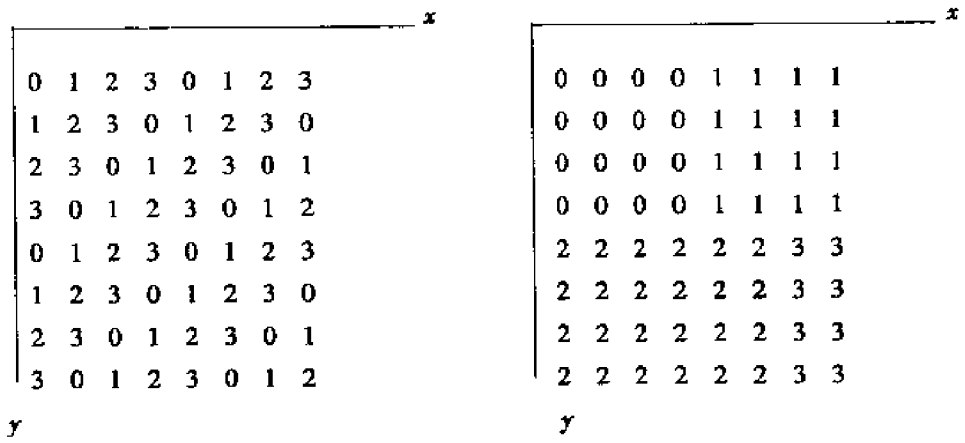
从灰度共生矩阵抽取出的纹理特征系数有以下几种:

(1) 角二阶矩

$$f_1=\sum_{i=0}^{L-1}\sum_{j=0}^{L-1}\hat{p}^2(i,j) \tag{7.5}$$

角二阶矩是图像灰度分布均匀性的度量。当灰度共生矩阵中的元素分布较集中于主对角线时,说明从局部区域观察图像的灰度分布是较均匀的,从图像整体来观察,纹理较粗,此时角二阶矩值 f_1 则较大,反过来则角二阶矩值 f_1 较小。角二阶矩是灰度共生矩阵像素值平

方的和，所以，它也称为能量。粗纹理角二阶矩值 f_1 较大，可以理解为粗纹理含有较多的能量。细纹理 f_1 较小，也即它含有较少的能量。



$Dx=\pm1, Dy=0$ $\theta=0°(\leftrightarrow)$　　$Dx=\pm1, Dy=\pm1$ $\theta=45°(\searrow)$　　$Dx=0, Dy=\pm1$ $\theta=90°(\downarrow)$　　$Dx=\pm1, Dy=\pm1$ $\theta=135°(\swarrow)$

$$P_A(0°)=\begin{bmatrix}0&14&0&14\\14&0&14&0\\0&14&0&14\\14&0&14&0\end{bmatrix}\qquad P_A(45°)=\begin{bmatrix}0&0&24&0\\0&0&0&24\\24&0&0&0\\0&24&0&0\end{bmatrix}$$

$$P_A(90°)=\begin{bmatrix}0&14&0&14\\14&0&14&0\\0&14&0&14\\14&0&14&0\end{bmatrix}\qquad P_A(135°)=\begin{bmatrix}24&0&0&0\\0&24&0&0\\0&0&24&0\\0&0&0&26\end{bmatrix}$$

$$P_B(0°)=\begin{bmatrix}24&4&0&0\\4&24&0&0\\0&0&40&4\\0&0&4&8\end{bmatrix}\qquad P_B(45°)=\begin{bmatrix}18&3&4&0\\3&18&1&2\\4&1&30&3\\0&2&3&6\end{bmatrix}$$

$$P_B(90°)=\begin{bmatrix}24&0&4&0\\0&24&2&2\\4&2&36&0\\0&2&0&12\end{bmatrix}\qquad P_B(135°)=\begin{bmatrix}18&3&3&0\\3&18&3&1\\3&3&30&3\\0&1&3&6\end{bmatrix}$$

图 7.4　2 幅灰度图像及其共生矩阵

(2) 对比度

$$f_2=\sum_{n=0}^{L-1}n^2\left\{\sum_{i=0}^{L-1}\sum_{j=0}^{L-1}\hat{P}(i,j)\right\}\qquad(7.6)$$

式中　$|i-j|=n$

图像的对比度可以理解为图像的清晰度，即纹理清晰程度。在图像中，纹理的沟纹越

深，则其对比度 f_2 越大，图像的视觉效果越是清晰。

(3) 相关

$$f_3 = \frac{\sum_{i=0}^{L-1}\sum_{j=0}^{L-1} ij\hat{p}(i,j) - u_1 u_2}{\sigma_1^2 \sigma_2^2} \tag{7.7}$$

式中 $u_1, u_2, \sigma_1, \sigma_2$ 分别定义为

$$u_1 = \sum_{i=0}^{L-1}\sum_{j=0}^{L-1} \hat{p}(i,j) \qquad u_2 = \sum_{i=0}^{L-1} j \sum_{j=0}^{L-1} \hat{p}(i,j)$$

$$\sigma_1^2 = \sum_{i=0}^{L-1} (i-u_1)^2 \sum_{j=0}^{L-1} \hat{p}(i,j)$$

$$\sigma_2^2 = \sum_{i=0}^{L-1} (j-u_2)^2 \sum_{j=0}^{L-1} \hat{p}(i,j)$$

相关是用来衡量灰度共生矩阵的元素在行的方向或列的方向的相似程度。例如，某图像具有水平方向的纹理，则图像在 $\theta = 0°$ 的灰度共生矩阵的相关值 f_3 往往大于 $\theta = 45°$，$\theta = 135°$ 或 $\theta = 90°$ 的灰度共生矩阵的相关值 f_3。

(4) 熵

$$f_4 = -\sum_{i=0}^{L-1}\sum_{j=0}^{L-1} \hat{p}(i,j)\log\hat{p}(i,j) \tag{7.8}$$

熵值是图像所具有的信息量的度量，纹理信息也属图像的信息。若图像没有任何纹理，则灰度共生矩阵几乎为零阵，则熵值 f_4 接近为零。若图像充满着细纹理，则 $\hat{p}(i,j)$ 的数值近似相等，则该图像的熵值 f_4 最大。若图像中分布着较少的纹理，$\hat{p}(i,j)$ 的数值差别较大，则该图像的熵值 f_4 较小。

上述 4 个统计参数为应用灰度共生矩阵进行纹理分析的主要参数，可以组合起来，成为纹理分析的特征参数使用。还有一些从灰度共生矩阵引导出来的参数可供纹理分析使用，其物理意义可自行理解，现列举如下：

(5) 方差

$$f_5 = \sum_{i=0}^{L-1}\sum_{j=0}^{L-1} (i-u)^2 \hat{p}(i,j) \tag{7.9}$$

式中 u 为 $\hat{p}(i,j)$ 的均值。

(6) 逆差矩

$$f_6 = \sum_{i=0}^{L-1}\sum_{j=0}^{L-1} \frac{\hat{p}(i,j)}{1+(i-j)^2} \tag{7.10}$$

(7) 和平均

$$f_7 = \sum_{k=0}^{2L-2}\sum_{i=0}^{L-1}\sum_{j=0}^{L-1} \hat{p}(i,j) \tag{7.11}$$

式中 $k = i + j$

(8) 和方差

$$f_8 = \sum_{k=0}^{2L-2}\sum_{i=0}^{L-1}\sum_{j=0}^{L-1} (i-f_7)\hat{p}(i,j) \tag{7.12}$$

式中 $k = i + j$

(9) 和熵

$$f_9 = -\sum_{k=0}^{2L-1}\sum_{i=0}^{L-1}\sum_{j=0}^{L-1}\hat{p}(i,j)\log\sum_{i=0}^{L-1}\sum_{j=0}^{L-1}\hat{p}(i,j) \tag{7.13}$$

式中 $k=i+j$

(10) 差平均

$$f_{10} = \sum_{k=0}^{L-1}\sum_{i=0}^{L-1}\sum_{j=0}^{L-1}\hat{p}(i,j) \tag{7.14}$$

式中 $k=|i-j|$

(11) 差方差

$$f_{11} = \sum_{k=0}^{L-1}(k-f_{10})\sum_{i=0}^{L-1}\sum_{j=0}^{L-1}\hat{p}(i,j) \tag{7.15}$$

式中 $k=|i-j|$

(12) 差熵

$$f_{12} = \sum_{k=0}^{L-1}\sum_{i=0}^{L-1}\sum_{j=0}^{L-1}\hat{p}(i,j)\sum_{j=0}^{L-1}\hat{p}(i,j) \tag{7.16}$$

式中 $k=|i-j|$。

2) 灰度—梯度共生矩阵法

图像的灰度直方图是一图像灰度在图像中分布的最基本的统计信息。图像的梯度信息的获得是通过使用各种微分算子，它检出了图像中灰度跳变的部分，如图像中景物的边缘、纹沟及其它尖锐的部分。若将图像的梯度信息加进灰度共生矩阵，则使得共生矩阵更能包含图像的纹理基元及其排列的信息。

若有一幅图像 $f(j,j)$，$i=1,2,\cdots,L_x$；$y=1,2,\cdots,L_y$

为避免太多的灰度级所带来的巨大的计算量，将其灰度进行正规化处理如下：

$$F(i,j) = \frac{f(i,j)N_g}{f_{\max}} + 1 \tag{7.17}$$

式中 N_g 为规定的最高灰度；

$f_{\max}$为图像的最高灰度。

正规化处理后，该图像的最高灰度从 $f_{\max}$变到 N_g。

一幅图像用前述的梯度算子，可得到它的梯度图像 $g(i,j)$，其中：$i=1,2,\cdots,L_x$；$j=1,2,\cdots,L_y$。同样，可将梯度图像进行正规化处理，以使其梯度值更加离散，其最大值为 L_s。因而梯度图像为

$$G(i,j) = [g(i,j)L_s/g_{\max}] + 1 \tag{7.18}$$

式中，$g_{\max}$为图像中的最大梯度值。

经过正规化处理，即可得到 2 个正规化矩阵：

灰度矩阵 $F(i,j)$，$i=1,2,\cdots,L_x$；$j=1,2,\cdots,L_y$，取值范围为 $1\sim L_g$；

梯度矩阵 $G(i,j)$，$i=1,2,\cdots,L_x$；$j=1,2,\cdots,L_y$，取值范围为 $1\sim L_s$；

灰度—梯度共生矩阵定义如下：

$$H(i,j),\quad i=1,2,\cdots,L_g;\quad j=1,2,\cdots,L_s \tag{7.19}$$

$H(i,j)$为表示正规化的图像，灰度为 i(在矩阵中的位置为第 i 行)，梯度为 j(在矩阵中的位置为第 j 行)的总像素点数。举例来证 $L_g=L_s=16$，$H(7,10)=14$，即表示一幅图像的

灰度取值范围和其梯度取值范围均为 16,在灰度—梯度共生矩阵中,灰度为 7($i=7$),梯度为 10,($j=10$)的总像素点数为 14。为避免过大的计算量,取灰度—梯度矩阵式的正规化处理,得到

$$\hat{H}(i,j)=\frac{H(i,j)}{L_x\cdot L_y} \tag{7.20}$$

式中 $i=1,2,\cdots,L_g$; $j=1,2,\cdots,L_s$

综上述,灰度—梯度共生矩阵提供了灰度直方图到梯度信息的合二为一的工具,因而可以从该矩阵中抽取图像的纹理统计特征系数:

(1) 小梯度优势

$$T_1=\frac{\sum_{j=0}^{L_s-1}\sum_{i=0}^{L_g-1}\hat{H}(i,j)/(j+1)^2}{\sum_{i=0}^{L_g-1}\sum_{j=0}^{L_s-1}\hat{H}(i,j)} \tag{7.21}$$

(2) 大梯度优势

$$T_2=\frac{\sum_{j=0}^{L_g-1}\sum_{i=0}^{L_s-1}\hat{H}(i,j)\cdot j^2}{\sum_{i=0}^{L_g-1}\sum_{j=0}^{L_s-1}\hat{H}(i,j)} \tag{7.22}$$

(3) 灰度分布不均匀表征

$$T_3=\frac{\sum_{i=0}^{L_g-1}[\sum_{j=0}^{L_s-1}\hat{H}(i,j)]^2}{\sum_{i=0}^{L_g-1}\sum_{j=0}^{L_s-1}\hat{H}(i,j)} \tag{7.23}$$

(4) 梯度分布不均匀表征

$$T_4=\frac{\sum_{j=0}^{L_s-1}[\sum_{i=0}^{L_g-1}\hat{H}(i,j)]^2}{\sum_{i=0}^{L_g-1}\sum_{j=0}^{L_s-1}\hat{H}(i,j)} \tag{7.24}$$

(5) 能量

$$T_5=\sum_{i=0}^{L_g-1}\sum_{j=0}^{L_s-1}\hat{H}(i,j)^2 \tag{7.25}$$

(6) 灰度平均

$$T_6=\sum_{j=0}^{L_g-1}i[\sum_{i=0}^{L_s-1}\hat{H}(i,j)] \tag{7.26}$$

(7) 梯度平均

$$T_7=\sum_{j=0}^{L_s-1}j[\sum_{i=0}^{L_g-1}\hat{H}(i,j)] \tag{7.27}$$

(8) 灰度均方差

$$T_8=\{\sum_{i=0}^{L_g-1}(i-T_6)^2[\sum_{j=0}^{L_s-1}\hat{H}(i,j)]\}^{\frac{1}{2}} \tag{7.28}$$

(9) 梯度均方差

$$T_9 = \{\sum_{i=0}^{L_R-1}(i - T_7)^2[\sum_{j=0}^{L_C-1}\hat{H}(i,j)]\}^{\frac{1}{2}} \tag{7.29}$$

(10) 相关

$$T_{10} = \sum_{i=0}^{L_R-1}\sum_{j=0}^{L_C-1}(i - T_6)(j - T_7)\hat{H}(i,j) \tag{7.30}$$

(11) 灰度熵

$$T_{11} = \sum_{i=0}^{L_R-1}\sum_{j=0}^{L_C-1}\hat{H}(i,j)\log\sum_{j=0}^{L_C-1}\hat{H}(i,j) \tag{7.31}$$

(12) 梯度熵

$$T_{12} = \sum_{j=0}^{L_C-1}\sum_{i=0}^{L_R-1}\hat{H}(i,j)\log\sum_{i=0}^{L_R-1}\sum_{j=0}^{L_C-1}\hat{H}(i,j) \tag{7.32}$$

(13) 混合熵

$$T_{13} = -\sum_{i=0}^{L_R-1}\sum_{j=0}^{L_C-1}\hat{H}(i,j)\log\hat{H}(i,j) \tag{7.33}$$

(14) 惯性

$$T_{14} = \sum_{i=0}^{L_R-1}\sum_{j=0}^{L_C-1}(i-j)^2\hat{H}(i,j) \tag{7.34}$$

(15) 逆差矩

$$T_{15} = \sum_{i=0}^{L_R-1}\sum_{j=0}^{L_C-1}\frac{\hat{H}(i,j)}{1+(i-j)^2} \tag{7.35}$$

7.2 形状特征与分析

人们的视觉系统对于景物认识的初级阶段则是其形状。图像经过边缘抽取和图像分割等操作，就会得到景物的边缘和区域，也就获取了景物的形状。任何一个景物形状特征均可由其几何属性(如长短、面积、距离、凸凹等)，统计属性(如投影)和拓扑属性(如连通、欧拉数)来进行描述。现在提出的问题是用计算机提取其形状特征，并用计算机所能提供的数字与符号将它表示出来，这就可以进行形状识别和理解。

能够给出景物的形状信息的概念有三种方式，一是图像经过分割处理后的区域，二是图像经过边缘抽取后的边界；三是区域的骨架。在图像上，区域或骨架提供了形状信息，人们的视觉系统关心的是其形状，其灰度信息则让位为次要地位。往往把区域内部或边界的像素赋予"1"值，而背景和其它不感兴趣的目标像素赋予"0"值，形成二值图像。二值图像给出清晰的形状概念。

区域所具有的形状特征可以通过对区域内部或外形通过各种变换来提取，也可以用图像层次型数据结构来表达。骨架是形状特征描述的重要方法，对于某些特殊的图像区域，如文字，它提供了极为重要的形状概念。所以，对于区域骨架的抽取及其形状特征识别是很重要的工作。

7.2.1 区域边界与骨架的提取

从图像分割可以得到一块块分割好的区域，再经边缘抽取，即可得到区域的边界。为了得到清晰的边界，可将图像转化为二值图像。若将边界点标以"1"值，图像的其余像素标以

"0"值,边界的数学表示则为

$$E(i,j)=\begin{cases}1 & (i,j)\in \text{边界点}\\ 0 & \text{其余像素}\end{cases} \tag{7.36}$$

图像的量化、离散化、获取、存储、传输的过程,往往会给边界位置带来一些噪声,这些噪声使得产生的边界不规则,不整齐,出现边界局部断裂现象等。为了进行边界的形状分析,必须消除噪声对边界的影响。有一个常用的方法是用小的直线段对提取的边界作多边形逼近,从而得到完整的边界。

对于某些分割后的区域,例如文字、数字和长度远远大于宽度的细窄长条状的区域,其骨架往往体现了区域的形状的本质特征。很明显,有些对称的区域图形,往往就可以抽取其中轴,由中轴的形状来表征该区域的形状,中轴为线段组成,比较容易进行形状分析。

7.2.2 区域边界的形状特征描述

1) 傅立叶描述器

经过正规化处理的边界为一封闭的曲线。封闭曲线的连续的点集的位置信息是特别重要的,可写成 $y=f(x)$,其中 x 为横坐标,y 为纵坐标。若以 $y=f(x)$直接进行傅立叶变换,则变换的结果依赖于具体的坐标 x 和 y 的值,不能满足平移和旋转的不变性要求。为了解决这个问题,引入封闭曲线弧长为自变量的参数表示形式,它的表现形式为

$$Z(l)=(x(l),y(l)) \tag{7.37}$$

式中 l 是曲线的弧长。

若封闭曲线的全长为 L,则 $L\geqslant l\geqslant 0$。若曲线的起始点为 $l=0$,则$\theta(l)$是曲线上弧长的切线方向。曲线在某点的切线方向可以理解为曲线在该点的走向,如果将所有点的走向都搞清楚,则其形状特征也就清楚了。现定义曲线的走向的变化规律:

$$\varphi(l)=\theta(l)-\theta(0) \tag{7.38}$$

$\varphi(l)$说明了曲线从起始点到弧长为 l 的点的曲线的旋转角度。$\varphi(l)$随弧长 l 而变化,显然它是平移和旋转不变的。把 $\varphi(l)$化为$[0,2\pi]$域的周期函数,用傅立叶级数展开,取其变换后的系数来描述区域边界的形状特征。$\varphi(l)$的变化规律可以用来描述封闭曲线 r 的形状。

引入另一个变量 t,弧长 l 为

$$l=\frac{L(t)}{2\pi}$$

很明显的是 $l\in[0,L]$, $t\in[0,2\pi]$,现可定义

$$\varphi^*(t)=\varphi\left(\frac{Lt}{2\pi}\right)+t \tag{7.39}$$

那末,$\varphi^*(t)$为$[0,2\pi]$上的周期函数,也就是如(7.40)式:

$$\varphi^*(0)=\varphi^*(2\pi)=0 \tag{7.40}$$

$\varphi^*(t)$在封闭曲线 r 平移和旋转条件下,均为不变。并且,$\varphi^*(t)$与 r 封闭曲线是一一对应的关系,如果反推则 $\varphi^*(t)\rightarrow z$,可得出一组相似的封闭曲线。由于 $\varphi^*(t)$为周期函数,可用傅立叶系数对它进行描述,在$[0,2\pi]$上展成傅立叶级数为

$$\Phi^*(t) = a_0 + \sum_{k=1}^{\infty}(a_k\cos kt + b_k\sin kt) \tag{7.41}$$

式中 $a_0 = \frac{1}{2\pi}\int_0^{2\pi}\varphi^*(t)\mathrm{d}t$

$$a_n = \frac{1}{\pi}\int_0^{2\pi}\varphi^*(t)\cos nt\ \mathrm{d}t$$

$$b_n = \frac{1}{\pi}\int_0^{2\pi}\varphi^*(t)\sin\ \mathrm{d}t$$

其中 $n = 1,2,\cdots$

在 7.2.1 节区域边界与骨架的提取中已经说明，曲线 r 是由多边形折线的逼近构成的，假设逼近曲线 r 的折线有 m 个顶点 $v_0, v_1, v_2, \cdots, v_{m-1}$，且该多边形的边长 $v_{i-1}v_i$ 的长度为 $\Delta l_i (i = 1,2,\cdots,m)$. 所以，其周长 $L = \sum_{i=1}^{m}\Delta l_i$. 那么在多边形的情况下，$\lambda = \frac{Lt}{2\pi}$，则傅立叶级数的系数分别为

$$a_0 = \frac{1}{2\pi}\int_0^{2\pi}\varphi^*(t)\mathrm{d}t = \frac{1}{L}\int_0^{L}\varphi(\lambda)\mathrm{d}\lambda + \pi = -\pi - \frac{1}{L}\sum_{k=1}^{m} l_k(\varphi_k - \varphi_{k-1}) \tag{7.42}$$

$$a_n = \frac{2}{L}\int_0^{L}[\varphi(\lambda) + \frac{2\pi\lambda}{L}]\cos\frac{2\pi n\lambda}{L}\mathrm{d}\lambda$$

$$= \frac{2}{L}\sum_{k=0}^{m-1}\int_{l_k}^{l_{k+1}}[\varphi(\lambda) + \frac{2\pi\lambda}{L}]\cos\frac{2\pi\lambda n}{L}$$

$$= -\frac{1}{n\pi}\sum_{k=1}^{m}(\varphi_k - \varphi_{k-1})\sin\frac{2\pi n l_k}{L}$$

$$b_n = \frac{1}{n\pi}\sum_{k=1}^{m}(\varphi_k - \varphi_{k-1})\cos\frac{2\pi n l_k}{L}$$

式中 $n = 1,2,\cdots$。曲线 r 就可用序列傅立叶级数来描述，故称为傅立叶描述器。傅立叶描述器是区域外形边界变换的一种经典方法，在二维和三维的形状分析中起着重要的作用，并且在手写体数字和机械零件的形状识别中获得成功。

2) 链码

通过边界的搜索等算法的处理，所获得的输出最直接的方式是各边界点像素的坐标，也可以用一组被称为链码的代码来表示，这种链码组合的表示既利于有关形状特征的计算，也利于节省存储空间。

对于离散的数字图像，区域的边界轮廓可理解为相邻边界象点之间的单元连线逐段相连而成。对于图像中的象点而言，它必然有 8 个方向的邻域，如图 7.5 所示。对于每一个方向赋于一种代码表示，如图(a)中 8 个方向分别对应 0,1,2,3,4,5,6,7，这种代码称为方向码。若水平与垂直方向的码段长度定义为 1，那么，对角线的码段长度则为 $\sqrt{2}$。图 7.5(b)所示的一个封闭曲线，若以上面 S 为出发点，按反时针的方向进行，所构成的边界链码应为 5565707001223333。其中，偶数链码段为垂直或水平方向的代码段，奇数链码段为对角线段。诚然，也可以按顺时针方向进行，所构成的边界链码完全不同反时针方向行进的情况。这样，边界链码具有行进的方向性，在具体应用时必须加以注意。有了这一链码的方向链，再加上一些标识码，即可构成链码。常用的标识方法有两种：一种是加上特殊专用的链码结束标志，如图 7.5(b)链码应为 5565707001223331，"!"就是结束标志。另外一种办法则为标上

起始点的坐标，如图 7.5(b)的链码为 556570700122333*XYZ*。*XYZ* 为起始点 *S* 的坐标，用三位 8 进制数表示。

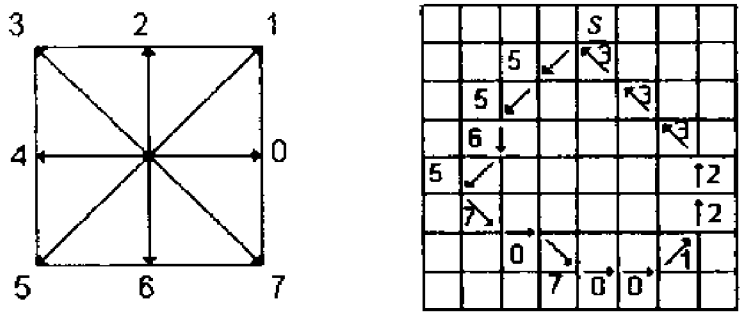


图 7.5　八链码原理图

从链码可以得到一序列的区域几何形状特征。

(1) 区域边界的周长

假设区域的边界链码为 $a_1, a_2, \cdots, a_i, \cdots, a_n$，每个码段 a_i 所表示的线段长度为 Δl_i，那末该区域边界的周长为

$$P = \sum_{i=1}^{n} \Delta l_i = n_e + (n - n_e)\sqrt{2} \tag{7.43}$$

式中　n_e 为链码序列中偶数码段数；

n 为链码序列中码段总数。

(2) 区域的面积

将该图像进行扫描，并转化为二值图像，计算灰度为 1 象点总数来表示区域的面积为

$$A = \sum_{x \in R} \sum_{y \in R} f(x, y) \tag{7.44}$$

式中　$f(x,y) = \begin{cases} 1 & (x,y) \in R, f(x,y) \geqslant T(\text{门限}) \\ 0 & (x,y) \in R, \text{其他} \end{cases}$

(3) 区域的圆度

定义区域的圆度为

$$C = \frac{P^2}{4\pi A} = \frac{(\text{周长})^2}{4\pi \cdot \text{面积}} \tag{7.45}$$

在相同面积的条件下，区域边界光滑且为圆形，则周长最短，其圆度为 $C = 1$。若区域边界凹凸变化程度增加，则相应周长也增加，C 值也随之增大，那末区域的形状越偏离圆形。可以用 C 来衡量区域的形状是否最接近于圆形或远离圆形，故称为圆度。

(4) 像素点的距离

假若两个像素点可用链码连接，则该两像素点之间的距离为

$$d = \sqrt{\left(\sum_{i=1}^{n} a_{ix}\right)^2 + \left(\sum_{i=1}^{n} a_{iy}\right)^2} \tag{7.46}$$

以上所得的区域形状特征用链码来进行计算，优点是直观性强，计算又很简单。但是在描述形状时，信息并不完全，这些几何数值与具体形状之间并不一一对应，而是一对多的对应。通过这些几何数值并不能唯一地得到原来的图形。所以，这类数值可用于形状的描述，

而不能用于形状的分类识别，但可作为它的补充。

7.2.3 区域的变换域的形状分析

1) 矩方法

当获得一个区域的二值图像或抑背景后的区域图像，那末，也就是说获得了区域内部的点的描述信息。有一个称为"矩"的区域描述器是由数学变换得到的，它对大小、旋转或平移的变化，均是不变的。

一幅二值图像可用二维连续函数 $f(x,y)$ 来描述，它的 $(p+q)$ 阶的矩定义为

$$M_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^p y^q f(x,y)\mathrm{d}x\mathrm{d}y \tag{7.47}$$

$$p、q = 0,1,2,\cdots$$

矩之所以可用来描述一幅二维图像，是因为它具有唯一性的特征。如果 $f(x,y)$ 是分段连续的，并且在 (x,y) 平面的有限部分中有非零值，则所有各阶矩均存在，并且矩序列 M_{pq} 唯一地被 $f(x,y)$ 所确定。反之，M_{pq} 也唯一地确定了 $f(x,y)$。

一幅二值化图像的 $(p+q)$ 阶矩的离散形式为

$$M_{pq} = \sum_x \sum_y f(x,y) \cdot x^p \cdot y^q \tag{7.48}$$

用矩方法可以得到一批用于区域形状识别的数学特征：

(1) 区域形心位置

$$\left.\begin{aligned} \bar{x} &= \frac{M_{10}}{M_{00}} \\ \bar{y} &= \frac{M_{01}}{M_{00}} \end{aligned}\right\} \tag{7.49}$$

低阶矩中的零阶矩 $M_{00} = \sum_x \sum_y f(x,y)$ 是区域密度的总和，可以理解为厚度为 1 的物体的质量。所以，低阶矩中的一阶矩 $M_{10} = \sum_x \sum_y xf(x,y)$ 和 $M_{01} = \sum_x \sum_y yf(x,y)$ 分别除以零阶矩 M_{00}，所得到的 $(\bar{x},\bar{y})$ 便是物体质量中心的坐标，或者说是区域灰度重心的坐标，故也称为质心。

(2) 中心矩

$$M_{pq} = \sum_x \sum_y f(x,y) \cdot (x-\bar{x})^p (y-\bar{y})^q \tag{7.50}$$

中心矩 M_{pq} 反映了区域 R 中的灰度相对于灰度重心是如何分布的度量。例如 M_{20} 和 M_{02} 分别表示围绕通过灰度重心的垂直和水平轴线的惯性矩。假如 $M_{20} > M_{02}$，则可能所计算的区域为一个水平方向拉长的区域。M_{30} 和 M_{03} 的幅值可以度量所分析的区域等于垂直和水平轴线的不对称性。如果某区域为垂直和水平对称，则 M_{30} 和 M_{03} 之值为零。

(3) Hu 矩组

在 1962 年，计算机工作者 Hu.M.K 证明了有七个矩构成的矩组，对于平移、旋转和大小尺度变化均为不变的，这个矩组对于区域形状识别是很有用的，后称为 Hu 矩组。

先定义一个规格化中心矩 η_{pq}，它的表达式为

$$\eta_{pq} = \frac{M_{pq}}{M_{00}^{r}} \tag{7.51}$$

式中 $r = \dfrac{p+q}{2}$ $(p+q=2,3,4\cdots)$

利用二阶和三阶规格中心矩可以导出 x 个不变矩组，就是 Hu 矩组：

$$\begin{aligned}
M_1 &= \eta_{20} + \eta_{02} \\
M_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \\
M_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} + \eta_{03})^2 \\
M_4 &= (\eta_{30} + \eta_{12}) + (\eta_{21} + \eta_{03})^2 \\
M_5 &= (\eta_{30} + 3\eta_{12})(\eta_{30} - \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\
&\quad + (\eta 21 - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\
M_6 &= (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta 21 + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\gamma_{21} + \gamma_{03}) \\
M_7 &= (3\eta_{21} - \eta_{30})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\
&\quad + (\eta_{21} - 3\eta_{03})(\eta_{21} - \eta_{03})[3(\eta_{03} + \eta_{12})^2 - (\eta_{12} + \eta_{03})^2]
\end{aligned} \tag{7.52}$$

(4) 面积

区域的面积定义为区域中的像素点数，计算如下式：

$$S = \frac{M_{00}}{G_{\max}} \tag{7.53}$$

式中 G_{max}为区域图像的灰度级。

(5) 扁度

扁度定义为长短轴之比，如下式：

$$e = \frac{M_{20} + M_{02}\sqrt{(M_{20} + M_{02})^2 - 4M_{20}M_{02} + 4M_{11}^2}}{M_{20} + M_{02} - \sqrt{(M_{20} + M_{02})^2 - 4M_{20}M_{02} + 4M_{11}^2}} \tag{7.54}$$

根据矩的定义可知，需要一个数量相当大的矩的序列才可以定义唯一的 $f(x,y)$，在实际应用中，通常只取其低阶矩，分析和识别的效果还是很理想的。

2) 投影

图像函数为 $f(x,y)$(见图 7.6)，P 为其投影方向，t 为其垂直方向，t 与 x 轴的夹角为 θ，则 $f(x,y)$沿着 P 投影变换如下式：

$$P(t,\theta) = \int_{-\infty}^{\infty} f(t\cos\theta - P\sin\theta, t\sin\theta + P\cos\theta)\mathrm{d}p \tag{7.55}$$

当 θ 固定时，$P(t,\theta)$为 t 的函数，它是一个一维的波形。当 θ 从 0～2π 变化时，可得到不同方向上的 $f(x,y)$的投影。根据投影定理，若已知 θ 从 0～2π 的全部方向上的 $p(t,\theta)$，就可以重新恢复出 $f(x,y)$，称为图像重建。为了分析图像的形状特征，在实际的应用中，取特定方向上的投影作为 $f(x,y)$的形状特征。在 x 轴和 y 轴的投影，即 $\theta = 0°$或 $\theta = \dfrac{\pi}{2}$ 时投影如下式所示：

$$\left.\begin{aligned}
P_x &= P(t,0) = \int_{-\infty}^{\infty} f(t,P)\mathrm{d}P = \int_{-\infty}^{\infty} f(x,y)\mathrm{d}y \\
P_y &= P(t,\frac{\pi}{2}) = \int_{-\infty}^{\infty} f(t,-P)\mathrm{d}P = \int_{-\infty}^{\infty} f(x,y)\mathrm{d}x
\end{aligned}\right\} \tag{7.56}$$

上述 x 轴和 y 轴上的投影均为波形曲线。也就是说，应用投影定理，把二维图像的分析

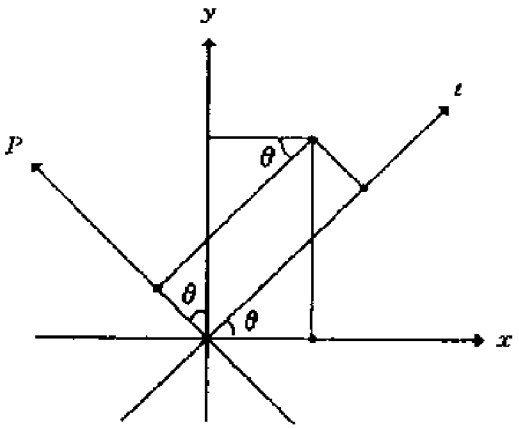


图 7.6 $f(x,y)$投影原理图

问题转变成一维的曲线波形的分析问题。

7.3 时间序列特征分析

前述的图像特征，无论是纹理特征或形状特征，主要是从静止图像上获取，或者可以理解为针对运动着的物体，在某一时刻瞬间所获得的图像。因为，从广义上理解，任何一个物体均处于运动状态。例如，在生产车间的自动流水线上，安装视觉系统，对零部件的观察，它所得到的是一个在流水线上运动着的零部件的图像。在战场上，安装的视觉系统，它所观察到的图像是运动着的坦克、飞机或其他移动目标，同样也是一个时间序列的图像。在医学图像中，医师为了掌握病人的心脏疾病情况，也需要一个时间序列的心脏收缩与扩张的动态图像。

从运动着的物体所获得的一组时间序列的二维图像，可以进行两样很重要的工作。一是提取运动物体本身的平移或旋转运动参数，从而获取物体的运动规律；二是可以从时间序列图像获取三维物体的深度、表面主方向等图像信息。

由不同时刻获取的图像序列的特征分析创造了另一个重要的领域，在工业、军事、生物医学，遥感等方面有着重要的应用，因而是图像特征与分析的一个重要方面。

7.3.1 视差、光流场

对于一个视觉系统，在相邻两时刻 t_{k-1}和 t_k 对同一个三维景物进行观察，所得到的图像为 $f(x,y,t_{k-1})$和 $f(x,y,t_k)$，被观察的三维景物的对应的特征点在两幅图像上的相对位置是不同的，这种位置的差称为视差。这个视差是运动着的三维景物表现在二维图像上的位移向量，它具有大小和方向。视差除以时间间隔 $\Delta t_k=(t_k-t_{k-1})$所得到的速度称为瞬时位置速度。

视觉系统对三维景物的某些特征点，如物体的边角点、图像上的最高灰度点的观察得到了一系列的特征点的视差，并由此计算出其瞬时位置速度的向量序列的分布，称为瞬时位置速度场，也称为光流场。下面的问题则是将光流场量化。

如图 7.7 所示，在三维空间 $Oxyz$ 中有一个运动物体 P，在 K 时刻，其位置为 $P_k(x_k,y_k,z_k)$。在 Δt 时刻之前，它的位置为 $P_{k-1}(x_{k-1},y_{k-1},z_{k-1})$。视觉系统从 $Oxyz$ 三维空间的顶

点向运动着的物体P进行观察。若ON为焦距,那么Nxy就是图像平面,那么成像平面上的两个坐标分别为(x_k, y_k)和(x_{k-1}, y_{k-1})。也就是说,从(x_{k-1}, y_{k-1})移动到(x_k, y_k)时间为Δt_k。

$$\Delta t_k = (t_k - t_{k-1})$$
$$\Delta x_k = (x_k - x_{k-1})$$
$$\Delta y_k = (y_k - y_{k-1})$$

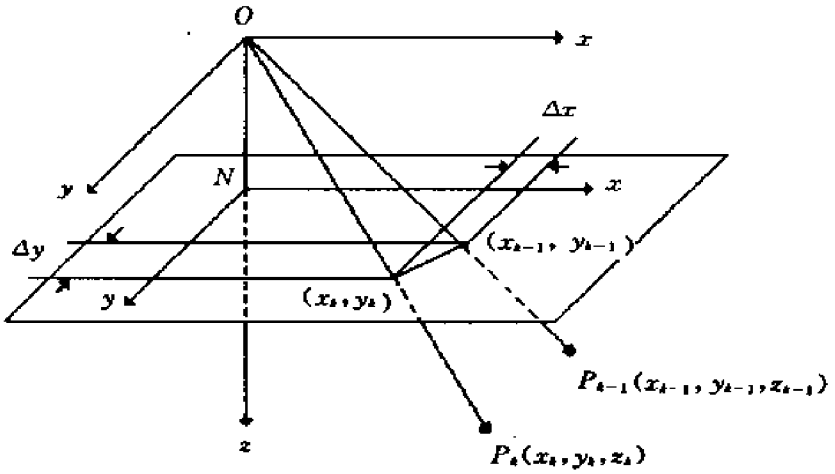


图7.7 视差原理图

Δx_k和Δy_k称为在成像平面上的x方向和y方向的视差。在图像平面上,物体图像相应位置的速度记为V_k,其大小及方向为

$$\left.\begin{aligned} |v_k| &= [(\Delta x_k)^2 + (\Delta y_k)^2]^{\frac{1}{2}}/\Delta t_k \\ \theta_k &= \operatorname{arctg}(\Delta x_k/\Delta y_k) \end{aligned}\right\} \tag{7.57}$$

V_k称为观察物体P运动到P_k点位置的瞬时位置速度,也称为光流。同样物体图像中若干特征点,如图像边角点,或灰度极大值点等,可称为瞬时位置速度场,也称为光流场。

7.3.2 序列图像中的特征点选择与匹配

每一帧图像均具有大量的信息,以512像元×512像元的小图像而言,它具有262K的信息,因而在不同时刻的图像之间比较的计算量太大。为了减少这个计算量,寻求图像的视差,就需要在图像中找出一组在序列图像中具有不变性质的结构:特征点。这类特征点如图像中局部景物的边缘和拐角点,图像中灰度局部极大值点。将不同帧图像的同种特征点进行匹配,从而求得视差。特征点的精心选择,使得在照度改变和景物发生某些变形时,并不影响视差计算的精度,而数据的计算量则大大地减少。

选择特征点时往往应该考虑到,在时间序列图像中,这些特征点一般都位于图像中相同的位置。而这些特征点应该位于灰度变化的区域的中心,例如某些孤立点、拐角点。要求以特征点为中心,在其周边方向上灰度差应当较大。这样,在特征点匹配时比较容易搜寻。如果在一条直线上,选择了若干特征点,因为直线上特征点的灰度差为零,在下一帧的图像的特征点的寻找就会发生混淆,以至找不到对应的特征点。

在某一图像$\{f(x_i, y_i, t_{k-1}), i, j = 0, 1, \cdots, N-1\}$中,对于每个像素$f(x_i, y_i, t_k)$取其周

边方向：水平、垂直、左斜角方向和右斜角方向，找出所有方向的灰度方差应为较大，先得求出在每个方向上与相邻像素灰度差的平方和。

水平方向灰度差的平方和为

$$G_h(i,j)=[f(x_i,y_i,t_{k-1})-f(x_{i-1},y_i,t_{k-1})]^2+[f(x_i,y_i,t_{k-1})-f(x_{i+1},y_i,t_{k-1})]^2 \tag{7.58}$$

垂直方向灰度差的平方和为

$$G_v(i,j)=[f(x_i,y_i,t_{k-1})-f(x_i,y_{i-1},t_{k-1})]^2+[f(x_i,y_i,t_{k-1})-f(x_i,y_{i+1},t_{k-1})]^2 \tag{7.59}$$

左斜角方向灰度差的平方和为

$$G_{ld}(i,j)=[f(x_i,y_i,t_{k-1})-f(x_{i-1},y_{i-1},t_{k-1})]^2+[f(x_i,y_i,t_{k-1})-f(x_{i+1},y_{i+1},t_{k-1})]^2 \tag{7.60}$$

右斜角方向灰度差的平方和为

$$G_{rd}(i,j)=[f(x_i,y_i,t_{k-1})-f(x_{i+1},y_{i-1},t_{k-1})]^2+[f(x_i,y_i,t_{k-1})-f(x_{i-1},y_{i+1},t_{k-1})]^2 \tag{7.61}$$

然后，取中间变量

$$M(i,j)=\min\{G_h(i,j),G_v(i,j),G_{ld}(i,j),G_{rd}(i,j)\} \tag{7.62}$$

在图像的范围内，求坐标(i^*,j^*)，使得

$$M(i^*,j^*)=\max\{M(i,j)\} \tag{7.63}$$

这样，就可求得坐标(x_i^*,y_j^*)的对应的点，就是这样的特征点，与它周边的像素的灰度方差为最大。

7.3.3 时间序列图像的运动估计和分析

运用时间序列图像估计运动参数，可以采用傅立叶变换，也可以采用不同时刻图像运动的差图像，还可采用不同时刻图像关于时间和空间灰度的变化来估计瞬时位置速度场，即光流场。空间灰度变化的时空梯度技术，物理概念清晰，易于理解。

被观察的物体点在时刻 t，位置(x,y)的灰度值为 $f(x,y,t)$。当物体点发生位移时，在$(t+\Delta t)$时刻，图像上的物体点从(x,y)位置移到$(x+\Delta x,y+\Delta y)$。由于 Δt 很小，可视为物体点的灰度在运动前后保持不变，则可得到下式：

$$f(x,y,t)=f(x+\Delta x,y+\Delta y,t+\Delta t) \tag{7.64}$$

将上式右边在(x,y,t)作泰勒展开，得到

$$f(x+\Delta x,y+\Delta y,t+\Delta t)=f(x,y,t)+\frac{\partial f}{\partial x}\Delta x+\frac{\partial f}{\partial y}\Delta y+\frac{\partial f}{\partial t}\Delta t \tag{7.65}$$

从上二式可得到

$$f(x,y,t)=f(x,y,t)+\frac{\partial f}{\partial x}\Delta x+\frac{\partial f}{\partial y}\Delta y+\frac{\partial f}{\partial t}\Delta t+0(\Delta x,\Delta y,\Delta t) \tag{7.66}$$

上等式两边消去 $f(x,y,t)$，且除以 Δt，可以得到

$$\frac{\partial f}{\partial x}\frac{\Delta x}{\Delta t}+\frac{\partial f}{\partial y}\frac{\Delta y}{\Delta t}+\frac{\partial f}{\partial t}+0=0 \tag{7.67}$$

令 $\Delta t\to 0$ 得到

$$\frac{\partial f}{\partial x}\frac{\mathrm{d}x}{\mathrm{d}t}+\frac{\partial f}{\partial y}\frac{\mathrm{d}y}{\mathrm{d}t}+\frac{\partial f}{\partial t}=0$$

若 $\frac{\partial f}{\partial x} = f_x$，$\frac{\partial f}{\partial y} = f_y$，$\frac{\partial f}{\partial t} = f_t$，$\frac{dx}{dt} = u$，$\frac{dy}{dt} = v$

$$f_x u + f_y v + f_t = 0$$

所以
$$f_t = -(f_x u + f_y v) \tag{7.68}$$

上式中，f_x、f_y、f_t 可以从图像序列样本进行估计，在最小的时间间隔 $\Delta t = (t_k - t_{k-1})$的两幅图像为

$$f(x_i, y_j, t_{k-1}), \qquad i,j = 0,1,\cdots,N-1$$
$$f(x_i, y_j, t_k), \qquad i,j = 0,1,\cdots,N-1$$

如图 7.8 所示，在四个相邻位置(x_i, y_j)、(x, y_{j+1})、(x_{i+1}, y_j)、(x_{i+1}, y_{j+1})关于 x 方向、y 方向和 t 方向的一阶差分平均值估计 f_x、f_y 和 f_t，那么

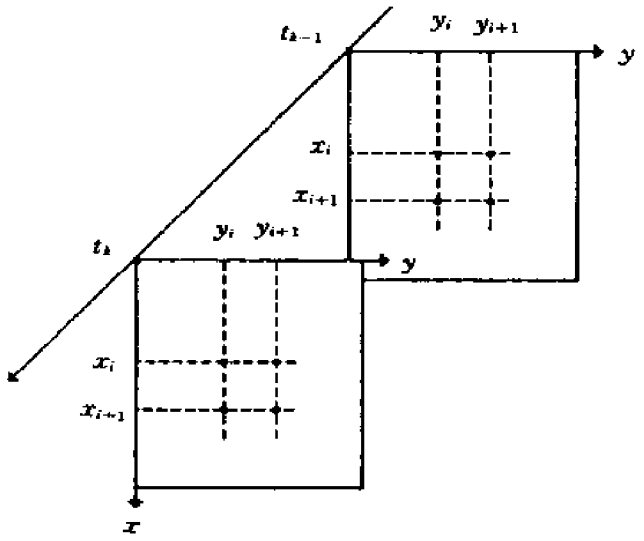


图 7.8 时间序列图像的运动分析原理

$$\begin{aligned}\hat{f}_x = \frac{1}{4}\{&[f(x_{i+1}, y_j, t_{k-1}) - f(x_i, y_j, t_{k-1})] \\ &+ [f(x_{i+1}, y_{j+1}, t_{k-1}) - f(x_i, y_{j+1}, t_{k-1})] \\ &+ [f(x_{i+1}, y_j, t_k) - f(x_i, y_j, t_k)] \\ &+ [f(x_{i+1}, y_{j+1}, t_k) - f(x_i, y_{j+1}, t_k)]\}\end{aligned} \tag{7.69}$$

$$\begin{aligned}\hat{f}_y = \frac{1}{4}\{&[f(x_i, y_{j+1}, t_{k-1}) - f(x_i, y_j, t_{k-1})] \\ &+ [f(x_{i+1}, y_{j+1}, t_{k-1}) - f(x_{i+1}, y_{j+1}, t_{k-1})] \\ &+ [f(x_i, y_{j+1}, t_k) - f(x_i, y_j, t_k)] \\ &+ [f(x_{i+1}, y_{j+1}, t_k) - f(x_{i+1}, y_j, t_k)]\}\end{aligned} \tag{7.70}$$

$$\begin{aligned}\hat{f}_t = \frac{1}{4}\{&[f(x_i, y_j, t_k) - f(x_i, y_j, t_{k-1})] \\ &+ [f(x_{i+1}, y_j, t_k - 1) - f(x_{i+1}, y_j t_{k-1})] \\ &+ [f(x_i, y_{j+1}, t_k) - f(x_i, y_{j+1}, t_{k-1})] \\ &+ [f(x_{i+1}, y_{j+1}, t_k) - f(x_{i+1}, y_{j+1}, t_{k-1})]\}\end{aligned} \tag{7.71}$$

将上三式代入前式，得到

$$\hat{f}_t = -(\hat{f}_x u + \hat{f}_y v) + \varepsilon \tag{7.72}$$

式中 ε 表示误差。

现在的目标是希望从本式得到瞬时位置速度在 x 和 y 方向的分量$[u,v]$的估计。在时间间隔 Δt 的两幅图像f_{k-1}和 f_k 中，取具有相同$[u,v]$的同一运动物体上的 n 个不同位置像素位置上 f_x, f_y, f_t 的估计$(k=1,2,3,\cdots,n)$，可以得到

$$\begin{bmatrix} -\hat{f}_t(1) \\ -\hat{f}_t(2) \\ \cdots \\ -\hat{f}_t(n) \end{bmatrix} = \begin{bmatrix} \hat{f}_x(1) & \hat{f}_y(1) \\ \hat{f}_x(2) & \hat{f}_y(2) \\ \cdots & \cdots \\ \hat{f}_x(n) & \hat{f}_y(n) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \cdots \\ \varepsilon_n \end{bmatrix} \tag{7.73}$$

若替代 $f_t = \begin{bmatrix} \hat{f}_t(1) \\ \hat{f}_t(2) \\ \cdots \\ \hat{f}_t(n) \end{bmatrix}$ $F_{xy} = \begin{bmatrix} \hat{f}_x(1) & \hat{f}_y(1) \\ \hat{f}_x(2) & \hat{f}_y(2) \\ \cdots & \cdots \\ \hat{f}_x(n) & \hat{f}_y(n) \end{bmatrix}$

那么瞬时位置速度的最小二乘平方估计如下：

$$\begin{bmatrix} \hat{u} \\ \hat{v} \end{bmatrix} = (F_{xy}^T F_{xy})^{-1} F_{xy}^T f_t \tag{7.74}$$

式中 $\hat{u}, \hat{v}$ 表示在(x_i, y_i)位置的瞬时位置速度在 x 和 y 方向的分量。

如果在一幅图像中有若干个物体作不同的运动，可以将图像分割成包含每个运动物体的小区域，对每个小区域进行上述估计，就可以得到各个不同运动的物体的瞬时位置速度。

7.4 二维图像特征提取与识别

7.4.1 二维图像特征的作用

自然界的景物一般皆为三维描述。但是从视觉系统，或称之为传感器的系统，所获得的图像一般是二维的。人类的眼睛在看了一张照片(二维图像)之后，能够很快地综合提取出三维信息，从而理解三维景物。但是，这些看来很简单的工作要让计算机做就很困难。人们可以想的办法则是利用序列图像，这包括时间序列图像或空间序列图像去进行理解。例如，对于心脏疾病的观测诊治，需要多帧时间序列图像的综合分析。再如，气象遥感云图的辅助气象预报，也一样要用多帧时间序列的图像。也就是说，多帧时间序列的二维图像的分析和研究，可以用计算机的手段去实现对三维景物的理解。一个立体场景的理解与分析可以用多角度的二维图像为工具去实现。因此，二维图像的处理与分析是基础，对二维图像数据录入后，进行增强，边缘提取，分割，纹理与形状特征提取等，提供最基本的信息，让视觉系统去理解有关三维的信息。

7.4.2 二维图像识别的目的

人们希望解决这样一个问题，能否利用计算机去理解和识别二维图像，若能圆满做到，那么计算机就可以替代部分人的工作，实现自动化，这在军事、国民经济的许多领域以及科学研究方面将会得到很重要的应用。对二维图像进行识别可以完成下述的工作，或者说达到以下的目的。

(1) 图像分割：将二维图像中一些有意义的场景目标从图像中分割出来，有些分割出来的景物可供人们直接理解。例如文字识别，需要能自动地将文字从背景中分割出来；细胞识别则需要将孤立的细胞从背景中分割出来。当这些景物能自动分割出来之后，人们就不难理解其含义了。

(2) 图像匹配：将已知的景物做成模板，然后与二维图像中的景物进行匹配。例如，具有某种形状的物体，如机器零件，或已知形状的字符。若匹配成功，则意味着理解了该类景物。

(3) 图像描述：利用计算机的一个数据结构，去填入一组特征值，从而实现了对自然界里的三维景物的描述。

(4) 图像分类：根据图像上的三维景物的特征值，可将该三维景物归类到已知的景物中去，即完成了自动分类识别的工作。

(5) 图像理解：三维景物的各个部分存在着一定的连续性或关系，根据这特定的关系，将三维景物各个部分联系起来，构成一个整体，达到理解该图像的目的。

8 图像文件格式

●图像数据　●位图数据　●编码压缩
●文件头　●图像格式　●图像生成

进行图像处理、分析和识别，就会接触大量的图像文件，图像文件又称为位图文件。这里需要说明的是，位图又称为光栅图和像素图。之所以有这样的名称，是因为在这类文件中，数据的存储是以像素为单位，而一幅图像好像是一张网格，每一个网格点就形成了图像中的一位。图像文件有许多不同的格式，并且还会有新的格式出现，这些不同的图像格式都有不同的软件来支持。所以，要很好地进行图像处理、分析和识别等工作，首先要很好地了解图像文件的一个重要的属性，就是构成图像文件的格式。因此，了解与掌握图像文件格式犹如了解与掌握一种工具。

每一个图像文件均有一个文件头，在文件头之后才是图像数据。文件头的内容由制作该图像文件的公司决定，一般包括文件类型、文件制作者、制作时间、版本号、文件大小等内容。各个图像文件的制作还涉及到图像文件的压缩方式，涉及到存储效率。与某一种压缩方式相对应的则是解压软件的类型等。本章将着重介绍几种比较流行的图像格式。

8.1 MacPaint 图像文件

在欧美广泛流行和使用的 Apple 系列的计算机，尤其是功能强大的 Apple Macintosh，所使用的图像文件格式均为 MacPaint。在我国，Apple Macintosh 系列机越来越多了，特别是在办公自动化、CAD、动画及广告制作等领域，由于其功能完美而显得很突出。与其它图像格式相比，MacPaint 比较简单。一幅 MacPaint 图像的水平方向总是 576 个像素，垂直方向总是 720 个像素。它的支持软件多数为 Macintosh 的应用程序，也支持不少的 PC 机应用程序。它的图像数据的压缩形式类似于 TIFF 图像文件的 PackBits 压缩数据。总体评价是，MacPaint 文件用于存储和传输时相当紧凑。由于这些原因，PC 机也成为 MacPaint 图像文件的使用平台了。

简单说来，MacPaint 图像文件包括三个部分：文件头（称为 Mac Binary）、图案数据和经压缩存储的 MacPaint 图像数据。

8.1.1 文件头（Mac Binary）

MacPaint 文件头的长度为 128 个字节，在该头文件中，偏移量为 00 和 74 的两个单元总为 0，用判断该两单元为 0 来判断文件头存在与否。在操作一个图像文件时，首先要确认一个文件的格式，还要知道其大小。MacPaint 文件具有统一的大小，处理起来就比较方便。

MamPaint 文件头中，文件名占 63 个字符，用 ASCII 码表示，在头文件中，偏移量为 02。文件类型码占 4 个字节，在该头文件中，偏移量为 65。其文件类型标记 4 个字符 PNTG，用 ASCII 码标识。

MacPaint 文件头中注明了文件的创建者，用 4 个字符标记，如果该图像文件来源于 MacPaint，则标上 MPNT。Macintosh 的应用程序也能产生 MacPaint 文件，所以也能有其它的创建者标识。

除此之外，还有文件创建日期和修改日期信息，各占 4 个字节，在偏移量 91，95 处。

8.1.2 图案数据

在 Mac Binary 文件头结束后的 4 个字节处，即偏移量为第 132 字节处为图案数据。每个图案占 8 个字节，总共有 38 种图案，没有压缩，也没有分界字节，它共占有 304 字节。该图案数据，对于绘图造型的应用编程提供了极好的手段。

8.1.3 位图数据

在图案数据之后，则为图像文件了。一个 MacPaint 图像的原始数据要占有 50K 字节。

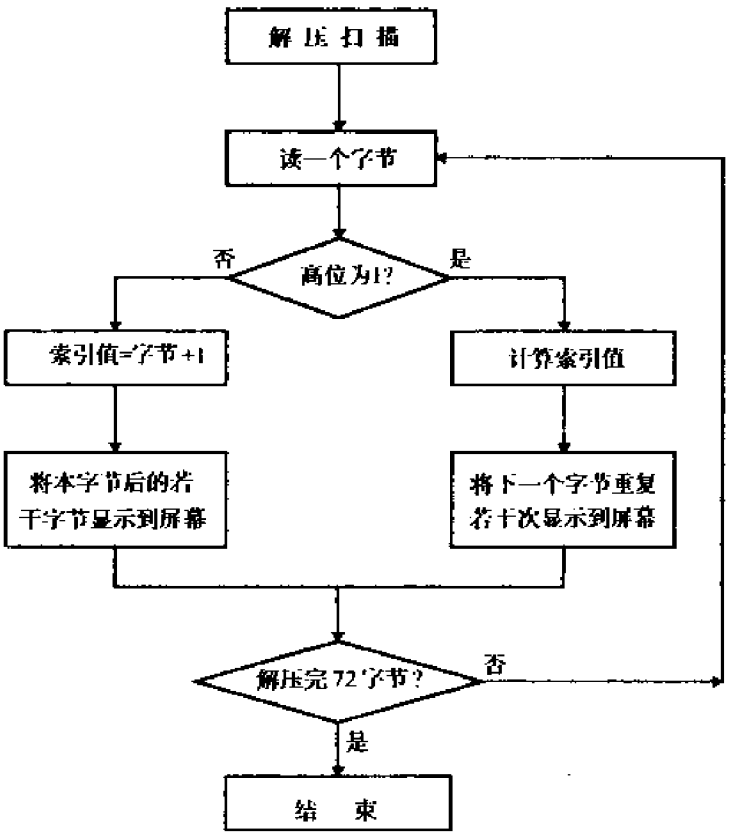


图 8.1 MacPaint 图像解压缩流程图

采用编码压缩技术将使图像空间大大减小。在 MacPaint 文件中，每行为 72 字节，共 720 行，每个字节 8 个像素，白 = 0，黑 = 1。由于每个扫描行只包括 72 字节，所以控制字节中只用 7 位就可以表达。实际的图像数据长度，标为索引值。最高位可用来表示压缩方式，以区别采

用的是游程编码压缩，还是字节串编码压缩。

图 8.1 示出了解压缩 MacPaint 图像数据一个扫描行的流程图。

8.1.4 创建 MacPaint 图像文件

创建一个 MacPaint 图像文件，首先写入文件头，就是将文件类型、文件大小、文件创建者等信息和图案集用 ASCII 码形式写入。再就是用与图 8.1 相反的编码过程，将图像数据压缩后写入，就构成 MacPaint 图像文件了。

8.2 IMG 图像文件

IMG 图像格式支持任意大小的图像。它的文件组成包括了文件头和图像数据两个部分。它的文件仅有 16 个字节，是几乎所有图像格式中文件头最短的。在图像数据的压缩方面也具有特色，它具有更高的压缩效率，且系统容易解压。它是 Digital Research 公司创建的。GEM 是该公司软件产品的标准操作环境，所以，绝大多数图形应用程序支持这种格式，它特别适用于应用程序之间的图像存储和传输。图像可以是单色图像、灰度图像或彩色图像。

8.2.1 IMG 的文件头

IMG 的文件头仅 16 个字节，是几乎所有图像中文件头最短的。它的文件头的结构如下：

```
WORD   0001H
WORD   0008H
WORD   0001H
WORD   PATTERN_LENGTH
WORD   PIXEL_WIDTH
WORD   PIXEL_DEPTH
WORD   LINE_WIDTH
WORD   IMAGE_DEPTH
```

很明显，根据这样的结构，其前三个字节是识别一个图像文件是否是 IMG 文件的标志。实际操作中，经常出现一个文件的后缀是 IMG，但该文件并不一定是 IMG 图像格式的文件，要进行鉴别，就去看这三个字节。

文件头的下一个域是模板长度，它通常为 1。在 IMG 文件解压缩的过程中，要被重复的字节可能是多个字节，模板长度就是要被重复的字节数，因此在这个字节中放的是被重复的字节数。紧接着，两个单元 PIXEL_WIDTH 和 PIXEL_DEPTH 定义的是水平和垂直方向像素的大小，每英寸 300 点，与普通激光打印机的一个点差不多大小。所以，这两个单元的数值为 55。下一个字单元 LINE_WIDTH 定义了一个扫描行的宽度，它以像素为单位，而不是字节。文件头的最后一个系数定义了图像的高度，就是图像水平扫描线的线数。这些就是 IMG 图像头文件 16 个整数字节的结构。

8.2.2 IMG 图像数据文件

IMG 图像文件按扫描行序列存储。扫描行的宽度字节数,由每行的像素数除以 8 得到,若有余数,则为其商(字节数)加 1。IMG 文件被压缩成不同的数据包,每种数据包都包含有标志,以指示是什么类型的数据包,并具有控制字节,比如重复的行数或字节数。每个扫描行的第一个数据块为重复数据块,它的结构为 16 进制,表达如下:

00H,00H,FFH,NN

该结构的 00H,00H,FFH 为重复数据块标志,它告诉解压程序将当前扫描行进行解压后,再重复 NN 次。若在一个扫描行的开始没有碰到行重复数据包,而系统为它设置了 NN 的缺省值为 1,也就是该行在解压之后不用复制,不需重复。

模板重复数据块则让解压程序知道将一个或多个字节重复一次或多次。需重复的字节的个数由文件头中的模板长度定之,其结构如下:

00H,NN

该结构的 00H 为模板重复数据块的标志,NN 则为重复的次数。在 NN 后面则跟随着模板字节。典型情况下它是两个字节。举例来看,假设模板长度为 2,则块 00H,02H,12,34,表示它是模板重复模块(标志为 00),重复次数为 2 次,其结果则是字节 12 和 34 需重复两遍。

字节串数据块的结构为 80H,NN。其标志为 80H,NN 为后面跟随 NN 个文字或数字的字节。还有一种数据块为实字节数据块。如果解压程序在数据块的第一个字节碰到了 00H 和 80H 之外的字节,就可以知道该数据块为实字节数据块。该数据块表示全为 1 或全为 0 像素的一个或多个字节。如果该块的高位为 1,则它是 1 的系列。例如:某字节为 8FH,表示有 15 个字节为 1。如果高位为 0,则是 0 的系列。如该块为 0CH 则表示有 12 个字节为 0。实际上许多 IMG 文件,只使用一个字节串数据块,存储完全不压缩的数据块。

举例来看,若有一块 IMG 格式的压缩了的图像文件为 16 进制表示如下:

00 00 FF 06 80 02 12 34 05 83 00 03 AB CD 80 01 98 01

其中 00 00 FF 06 为重复数据块,表示后面的数据要重复成 6 个同样的扫描行。而 80 02 表示后面紧跟的是字节串数据块,就是跟着 2 个数据 12 34。05 83 表示为 0 和 1 的数据字节,分别重复 5 次和 3 次。就是说,下面的数据为 00 00 00 00 00, FF FF FF。

接着为 00 03 为模板重复数据块的标志,也就是后面的 AB CD 需重复 3 次。得到 AB CD AB CD AB CD。再看下面,80 01 为字节串模块,表示后面为 1 个字节串,即 98 01 为实数据模块,表示该 1 个字节为 0。所以,经过解压后,这个 IMG 图像共有 6 个扫描行,每个扫描行均包括:

12 34 00 00 00 00 00 FF FF FF AB CD AB CD AB CD 98 00

8.3 GIF 图像文件

GIF 图像文件格式能存储单色图像，但最好是 16 色以上的彩色图像，它带有色彩表，因此正和 EGA/VGA 的色彩机制一致。尤其最适合 VGA 卡，能显示 2^{18}，就是 262144 色彩中的 256 色，也正是 GIF 的 256 色图像。所以，大量的优美的彩色图像以 GIF 图像文件而存在。

GIF 图像文件以它采取的复杂的图像压缩算法 LZW 而著称。利于图像的存储，存储效率高。GIF 的特点还在于它支持多幅图像定序或覆盖，交错多屏幕绘图以及文本覆盖。

8.3.1 GIF 的文件头

GIF 图像文件头的结构随着所存放的文件的不同而有所变化，而存放同一幅图像时，也可以有不同的结构。

在 GIF 文件中首先碰到的则是 GIF 文件标志，它是 3 个字节的串 GIF，再加上版本号，最常见的版本号为“87a”。

紧接着版本号的一个字节为全局标志，它包括了两方面的信息。该标志的低 3 位说明即将碰到的图像是多少种色彩的。将这 3 位加 1，就可得到图像中的一个像素所包含的位数。若全局标志的低 3 位为“111”，就是 7，它说明图像中的一个像素需 8 位表示，就是说图像为 256 色。因而，GIF 文件图像最多 256 色，与 VGA 所显示最大色彩数一样。全局标志所表达的第 2 个信息是它的最高位所表示的信息。若最高位为 1，则存在一个全局的色彩表，否则表示没有全局色彩表。在一个 GIF 文件中存在多幅图像时，每幅图像均有它的局部色彩表，就不需要有全局色彩表。但大多数情况下，GIF 只有一幅图像，只要一个全局色彩表就够了。

接在全局标志后面的是背景色标志，它指向色彩表中的对应的一项。因而 GIF 允许在实际显示图像之前，先把背景色置成适当的标志。在图像文件解码时，使用该标志可以在显示时先给出背景色。在大多数情况下背景色为 0。

在这之后，应是图像宽度和图像高度，后面跟的一个固定的字节为 0。若有全局色彩表需要定义，则全局色彩表跟着这个结构存放，色彩表的长度取决于前述的全局标志，若全局标志的低 3 位为“111”，一共有 256 种色彩。每一种色彩由 3 个字节决定，分别表示红、绿、蓝三种原色的强度，那么色彩表的长度为 256 * 3，就是 768 个字节。这样，GIF 图像的文件头就结束了。

8.3.2 GIF 的图像数据

GIF 的文件头的后面则是图像数据块本身。一个图像数据块包含 3 个部分：图像数据、扩展和文件结束符。扩展是用来增强 GIF 文件的描述功能而设置的。在大多数 GIF 图像中没有这部分，程序中可以不去考虑。逗号“,”开始表示一幅图像数据，感叹号“!”表示为一个扩展，分号“;”则表示为一个文件结束符。

图像数据块以“,”开始，后两个字节放的是图像显示在屏幕上的起始位置，因为 GIF 图像可以编程显示在屏幕的任意位置，所以，这两个字节并非很重要。下面两个字节为图像的宽度和高度。接着的一个字节则是局部标志，它的最高位指明是否存在一个局部色彩表。

其最后3位类似全局标志，说明色彩个数。若有局部色彩表，其位置是在局部标志之后，在局部色彩表后面放的则是图像数据了。

8.3.3 GIF的图像编码压缩算法：LZW

GIF的图像压缩效率较高，其压缩算法称为LZW，它来自三位发明此项算法的科学家的名字：Lempel，Ziv和Welch。LZW算法在压缩过程中主要处理3种数据流：输入字符流，就是图像像素数据；输出流，就是压缩生成的代码流；还有一张字符串表。LZW压缩算法的目的则是压缩后的代码串要远比原来的原始图像数据短得多。

LZW压缩算法的特点是在于它能动态地标记数据流中出现的重复串。它把压缩过程中遇到的字符串记录在这张字符串表中，在下次又碰到这一字符串时，就用一代码表示它，这样用短代码来表示较长的字符串就达到了压缩数据量的目的。

8.4 TIFF的图像文件

在运行的操作系统环境MS-DOS、UNIX和OS/2中，如果看到文件的扩展名为“.TIF”，那么它就是TIFF图像文件。大多数据扫描仪输出的图像文件均为TIFF文件格式。所以，了解TIFF图像文件的结构与特点是有意义的。TIFF是英文Tagged Image Format File的缩写，可译为“标志图像格式文件”。

TIFF图像格式支持任意大小的图像，从单色的二值图像到24位的真彩色图像，它广泛地应用于单面排版系统及与之相关的应用程序，而且与计算机的结构，操作系统和图形硬件无关。事实上，只要系统支持类似文件的8位字符流的存储介质，就可以使用TIFF图像格式。

8.4.1 TIFF的文件头

TIFF图像文件头结构比较简单，由8个字节组成，如图8.2(a)所示。文件头的第1项为“MM”或“II”，它指出了生成文件时微机的处理器类型。若为“II”，则为Intel系列的PC机，若为“MM”，则为Motorola的机器。对于这两种不同的机器类型，其数据的权重顺序是不同的。若为“II”，总是先存放低位字节，后存放高位字节。然而标记“MM”，则表示先存放高位字节，后存放低位字节。

第2项为常数，即002AH，或十进制42，它是TIFF文件标志。一旦系统识别该常数，则知道该图像文件为TIFF格式，这两字段的长度均为2个字节。

文件头的第3项为4个字节长的“图像文件目录块”(IFD)指针，它指向第一个图像文件目录块。图像文件目录项指针表见图8.2(b)所示。头2个字节为目录块的个数，下面则是多个目录块的指针，各占2个字节。若有n个目录块，则图像文件目录块指针表的长度为(2+n*12+4)。每一个目录块，也称为域，它是TIFF文件存放图像信息的基本单位。每个块长为12个字节长。它包括域的标志、类型、长度和指向数据的指针。每个域都有一个标志，该标志说明该域中存放什么信息，最常用的系数为图像宽度(100H)和图像长度(101H)。域的类型为一个短整数，共有五种类型：8位单字节整数，单字节字符，16位双字节整数，32位4字节整数据和8字节实数(4个字节分子、4个字节分母)。域的长度说明所含有的指定类

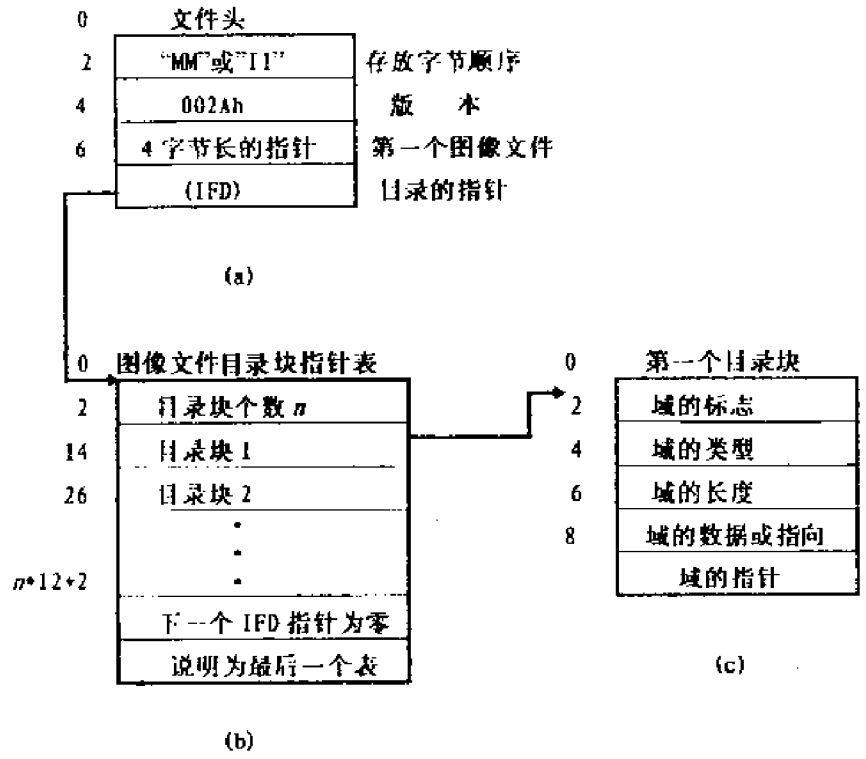


图 8.2 TIFF 图像文件的文件头

型的个数。域的数据字节长度为域的长度乘上数据类型单位字节长度。

第 4 项是域的数据或指向数据的指针。TIFF 图像文件目录块中的域是多功能的。

8.4.2 TIFF 图像文件数据

TIFF 图像将位图数据组织成图带存放。图带是若干条连续的扫描行所组成图像的一个带状的部分，TIFF 将数个图像划分为若干个固定扫描行组成的图带。图像文件的压缩方式在 TIFF 图像中有多种，有游程编码压缩，LZW 编码压缩，紧缩位法编码压缩等。也有的 TIFF 图像文件存放在物理介质上，并没有进行压缩编码，但存放时能尽可能地节省空间，比如扫描行后，紧跟下一行的数据，中间没有多余字节。

TIFF 图像文件优点很多，允许用户针对扫描仪、监视器或打印机的性能进行调整，所以应用范围很宽。

8.5 其他图像数据文件(PCX 和 BMP)

8.5.1 PCX 图像文件

PCX 图像文件特别适用于现在已广泛普及的 PC 机上的绘画程序，是在 PC 机上使用时间最久的一种位图格式，因而也是受到支持最广泛的一种。它支持单色、16 色，最高支持 256 色的图像。图像大小可达到 64K×64K。

PCX 图像文件由三个部分组成：文件头、图像数据和 256 色调色板。PCX 的文件头有

128 个字节，它包括版本号，被打印或扫描的图像的分辨率(单位为每英寸的点数)，大小(单位为像素)，每扫描行的字节数，每像素包含的位数据和彩色平面数。

PCX 的图像数据采用游程编码压缩方法，压缩的速度快，但效率不高。

PCX 的文件头中没有足够的空间容纳 256 色的调色板，这是因为 PCX 图像格式的设计者最初没有考虑需要这么多的调色板空间，于是调色板就放到 PCX 文件的最后面了。显示卡 VGA 能显示 256 色，所以，PCX 图像文件可以与 VGA 插件卡联手运用。

8.5.2 BMP 图像文件

众所周知，Microsoft 公司的 Microsoft Windows 窗口系统成为 PC 机环境下公认的最佳操作平台，而 BMP 图像文件格式是 Windows 窗口系统交换图形、图像数据的一种标准，因此在 Windows 环境中运行的图形、图像软件都支持 BMP 图像文件。

BMP 图像文件有以下三部分组成：文件头、位图信息、图像数据。

BMP 图像文件的文件头包括：文件类型；位图文件大小，以字节为单位；位图阵列的起始位置，也就是位图文件的位移量。它的位图信息包括图像宽度的像素值，图像长度的像素值，图像平面数，每个像素的位数，压缩类型，压缩图像的大小，水平和垂直分辨率(以每米的像素数表示)，使用的色彩数，色彩对照表等。有了这些信息，一幅图像的基本组成就有了。当每个像素具有 4 位和 8 位时，必然有一个色彩对照表，表项大小为 2，16 和 256。由于显示设备在某种情况下，可能不具备所需的色彩，在位图信息中有字段控制可选择的重要色彩数目。

BMP 的图像数据有压缩和非压缩两种。BMP 的图像文件使用一种 RLE 压缩算法，它由一系列的组构成。有三种类型的组：重复组、文字组和特殊组。重复组由两个字节构成，第一个字节是像素重复的次数，第二个字节为要被重复的像素。

例如：16 进制字节 05　63

表示像素系列 63636

文字组由一个零字节、一个像素计数值和文字像素字节组成。

例如：16 进制字节 00　05　12　34　50　00

表示像素系列　1　2　3　4　5

特殊组 00　00 表示一行的结束，而特殊组 00　01 表示图像文件结束。

在非压缩的 BMP 图像文件中，图像中的每一个点的像素值对应于位图阵列中的若干位，而位图阵列的大小由图像的高度、宽度和图像的色彩数决定。

8.6 图像文件格式转换

从前面的讲述知道，各种图像文件的格式是不一样的，图像数据的存储压缩方式也是不同的。往往会碰到这样的问题，某些图像处理软件要求图像具有某种格式，而当格式不合要求时，则需要进行图像格式转换。现在的 PC 机上的显示卡插件多为 EGA/VGA，SVGA，TVGA 等。它们的位图格式也要求多种图像文件能够适应，不适应就要转换。掌握了图像文件格式、数据压缩方式，转换则是应用编程问题了。

在进行图像格式转换时，会碰到一些具体问题，如内存大小问题、转换速度问题。现举

一例来说明图像格式的转换。前面已经说明，从扫描仪输出的图像文件，一般为 TIFF 格式。常规的 VGA 有两种文件格式，即顺序字节格式和位平面格式。如图 8.3 所示。若要将一幅扫描后的图像在屏幕上显示，必须将 TIFF 文件转换为 VGA 的文件格式，这就需要转换。

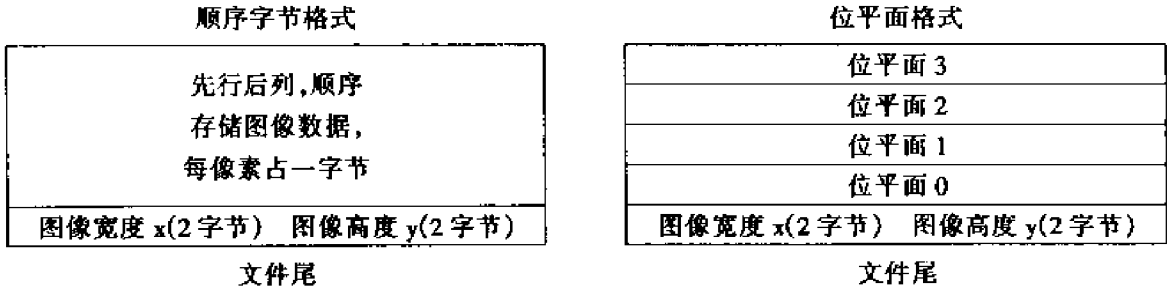


图 8.3　VGA 两种文件格式

上述的 VGA 顺序字节格式的优点是易于显示，易于装入内存进行逐点处理，但占用空间大，显示速度较慢。而图中的位平面格式与常规的 VGA 位平面显示方式的图像缓冲区的数据格式基本相同，不同之处仅在于位平面的高、宽是可变的。由于显示时可以直接将数据写入 VGA 卡上的缓冲区，故此格式的优点是显示速度极快。

为了实现从 TIFF 到 VGA 的转换，我们在配有 VGA 显示卡的微机上，用 Turbo C 2.0 编写了一个程序，实现了对上述两种 VGA 方式的转换，且使用方便。

启动程序后，按系统要求键入 TIFF 文件名，并按屏幕揭示选择 Watch 或 Convert。选择 Watch 后，若图像大小超过一屏，提示键入图像块左上角坐标，便可看到图像，看毕图像任击一键返回主菜单。选择 Convert 后，显示子菜单，选择要转换的图像区域。其中，Image out 提供交互裁剪功能，使用 ↑，↓，→，←键可控制"X"形光标，用回车键确定裁剪区域，用 ESC 键取消，转换完成后，程序自动读出转换后的图像并显示，任击一键结束程序。下面附上该程序清单及注释，供读者参考。

```
/* 在 VGA 环境下显示 TIFF 图像格式的数据文件 */

#include <io.h>
#include <fcntl.h>
#include <graphics.h>
#include <stdlib.h>
unsigned char buf[1000];

main (int argc,char * argv[])                                    /* 主程序 */
{char * filename[40];
printf("Tiff to vl6 converter. \n \n");
if (argc < 2){Printf("please input filename:");scanf("%s"filename);
    tifmain(filename);
```

```
        }
          else tifmain(argv[1]);
}
initscreen()                                                    /*初始化屏幕*/
{int g1, g2, i;
struct palettetype soc;
g1 = VGA;g2 = VGAHI; initgraph (&g1,&g2,"");
getpalette(&soc);
for(i=0;i<soc.size;i++)
    setrgbpalette(soc.colors[i],i*4,i*4,i*4);
}tifmain(filename;
char *filename;
{int i ,handle, *p,j,k,x,y,offset,x0,y0;
unsigned char c,c0;
                                                    /*打开 TIFF 文件,读文件头。*/
handle = open (filename,O_RDONLY|O_BINARY);|
if(hand=1){cputs("can't find file. \n");exit(0);}
read(handle,buf,8);
if(buf[2]!  =42){printf("%s isn't a tiff file. \n",filename);teturn;}
lseek(handle,buf[4],SEEK_SET);
read(handle,buf,2);
read(handle,buf+2,buf[0]*12);
                                                    /*读参数块*/
  for(i=0;i<((int*)buf)[0];i++)
    {switch(((int*)buf)[1+i*6])
    {case 256;x=((int*)buf)[5+i*6];break;
      case 257;y=((int*)buf)[5+i*6];break;
      case 258;if(((int*)buf)[5+i*6]!  =4
        {printf("%s isn't a 16 color/grey level image. \n".filename);
      return;
        }
        break;
  case 273;offset=((int*)buf)[5+i*6];break;
}
}
lseed(handle,offset,SEEK_SET);
printf("File:%s \n",filename);
printf("width=%d point,hight=%d point. \n",x,y);
x0=0;y0=0;
```

```
                                                        /* 主菜单 */
mainmanu:
  printf("please select: \n1.Watch image. \n2.Convert to order bytes. \n");
  printf("3.Convert to bit plane pattern. \n0ther key to exit. \n");
      if((c=getch())==='1')
        {printf("Input the left_top coordinate of the mage_block to watch. \n");
        if(x>640){printf("x=(0-%d)",x-640);scanf("%d",&x0);}
      else x0=0;
        if(y>480){printf("y=0-%d)",y-480);scanf("%d",&y0);}
      else y0=0;
        x0=x0/2*2;
                                                        /* 读出像素点并显示图像 */
          initscreen();
          iseed(handle,(long)(x+1)/2*y0,SEEK_CUR);
          if((y-y0)>480)y=480+y0;
          for(j=0;j<y-y0;j++)
{if(x0>0)lseek(handle,x0/2,SEEK_CUR);
    for (i=0;i<min(320,(x-x0+1/2);i++)
        {read(handle,&c,1);
        putpixel(i*2+1,j,c&15);putpixel(i*2,j,(c>>4)&15);
}
if((x-x0)>640)lseek(handle,(x-x0-639)/2,SEEK_CUR);
}
          getch();restorecrtmode();
      goto mainmanu;
          }
if(c='2'||c='3')convert(handle,c.x0,y0,x,y,offset);
close(handle);
}
                                                        /* 画剪裁框 */
draw(x1,y1,x2,y2,state)
int x1,y1,x2,y2,state;
{line (x2-5,y2-5,x2+5,y2+5);line(x2+5,y2-5,x2-5,y2+5);
if(state)
  {moveto(x2,y2);lineto(x2,y1);lineto(x1,y1);lineto(x1,y2);lineto(x2,y2);}
}
                                                        /* 转换图像文件格式 */
convert(handle,mode,x0,y0,x,y,offset);
int handle,x0,y0,x,y,offset;
```

```
{char c,s[40],buf,buf1[4];
int i,j,k,l,h,x1,y1,x2,y2,xc,yc,p0t,quit=0,state=0,count=0;
char far * p0;
printf(" Input the traget filename:');
scanf("%s",s);
h=open(s,0_WRONLY|0_BINARY|0CREAT);
if(h==-1){cputs("can't creat file. \n");return;}
ptintf{"select convert range: \n1. All image. \n2.All screen. \n3,User input. \n4.Imagecut. \
n");
c=getch();
switch(c)
{case'1';x0=0,y0=0,x1=x,y1=y;break;
case'2';x1=640;y1=480;break;
case'3';printf("printf("Input the coordinate of left top point of the image_block to convert. \n");
   printf("x=0-%d)",x);scanf("%d",&x0);
   printf("y=0-%d)",x);scanf("%d",&x0);
   printf('width=(0-%d)",x-x0);scanf("%d",&x1);
   printf("width=(0-%d)",y-y0);scanf("%d",&y1);
   break;
case'4':initscreen();
   setwritemode(1);
   lseek(handle,offset,SEEK_SET);
   lseek(handle,(long)(x+1)72*y0,SEEK_CUR);
   if((y-y0)>480)y=480+y0;
   for(j-0;j<y-y0;j++)
   {if(x0>0)lseek(handle,x0/2,SEEK_CUR;)
for(i=0;i<min(320,(x-x0+1)/2);i++)
   {read(handle,&c,1);
   putpixel(i*2,j,c&15);
   putpixel(i*2+1,j(C>>4)&15);
}
if((x-x0)>640)lseek(handle,(x-x0-640)/2,SEEK)CUR);
}
xc=0,yc=0;
draw(0,0,xc,yc,state);
while(quit==0)
{c=getch();
switch(c)
{case 13;if(state==0){x2=xc);y2=yc;
```

```
    xc = min(x - x0,639); yc = min(y - y0,479);
    state = 1;
    draw(x2,y2,xc,yc,state);}
else{x1 = xc - x2 + 1;y1 = yc - y2 + 1;quit = 1;}
    break;
case 27:if(state = = 1){draw(x2,y2,xc,yc,state);
    xc = x2;yc = y2;
    state = 0;
}
    break;
    case 0:c = getch();
    draw(x2,y2,xc,yc,state);
    switch(c)
    {case72:if(yc > = 8)yc - = 8;
    break;
case 80:if(yc < = min(y - y0,479) - 8))yc + = 8;
    break;
case 75:if(xc > = 8)xc - = 8;
    break;
case 77:if(xc < = (min(x - x0,639) - 8))xc + = 8;
    break;
}
      draw(x2,y2,xc,yc,state);
    }
}
    if(x1 < 0){x0 = x0 + x2 + x1;x1 = - x1;}
        else y0 + = x2;
    if(y1 < 0){y0 = y0 + y2 + y1;y1 = - y1;}
        else y0 + = y2;
          restorecrtmode();
    break;
}
printf("Converting")
lseek(handle,offset,SEEK_SET);
x0 = x0/2 * 2;
if(x1 > (x - x0))x1 = x - x0;
if(y1 > (y - y0))y1 = y - y0;
x1 = x1/2 * 2;
lseek(handle,(long)(x + 1)/2 * y0,SEEK_CUR);
```

```
if(mode = = '2')                                    /* 转换为顺序字节格式 */
  {for(j = 0;j < y1;j + + )
    if(x0 > 0)lseek(handle,x0/2,SEEK-CUR);
for(i = 0;i < x1/2;i + + )
  {read(handle,&C,1);
    buf = (C > > 4)&15;write(h,&buf,1);
    buf = C&15,write(h,&buf,1);
  }
if((x - x0) > x1)lseek(handle,(x + 1 - x0 - x1)/2,SEEK-CUR);
if( + + count > = 10){printf(".");count - = 10;}
      }
  }
else                                                /* 转换为位平面格式 */
      {for(k = 0;k < 4;k + + )
{for(j = 0;j < y1;j + + )
  {if(x0 > 0)lseek(handle.x0/2.SEEK-CUR);
i = 0;
while(i < x1/2)
  {read(handle,buf1.4);
    i + = 4;
    if(i > x1/2){for(i = i - x1/2;i < 4;i + + )buf1[i] = 0;
    lseek(handle,x1/2 - i,SEEK-CUR);
}
    buf = 0;
    for(i = 0;i < 4;i + + ){buf = (buf < < 1) + (buf1[i] > > (7 - k)&1);
      buf = (buf( < < 1) + buf1[1] > > (3 - k))&1);
      }
    write(h,&buf,1)
    }
  if((x - x0) > x1)lseek(handle,(x + 1 - x0 - x1)/2,SEEK_CUR);
if( + + count > = 40){printf("·");count - = 40;}
  }
lseek(handle,offset,SEEK_SET);
lseek(handle,(long)(x + 1)72 * y0,SEEK_CUR);
      }
    }
                                                    /* 写入图像的宽度、高度 */
write(h,&x1,2);write(h,&y1,2);
close(h)
```

```
                                                    /* 显示已转换的图像 */
h = open(s,O_RDONLY|O_BINARY);
lseek(h, -4,_SEEK_END);read(h,&x1,2);read(h,&1,2);
lseek(h,0,SEEK_SET);
initscreen();
if(mode = = '2')
      for(j = 0;j < y1;j + + )
for(i = 0,i < x1;i + + ){read(h,&c,1);putpixel(i,j,c);}
else
      {c = 8;
      for(k = 0;k < 4;k + +4)
{p0t = 0;outportb(0x3c4,2);outportb(0x3c5,c);c > > = 1;
for(j = 0;j < y1;j + + )
         p0 = MK_FP(0xa000,p0t);read(h,p0,(x1 - 1)/8 + 1);p0t + = 80;}
   }
      }
close(h);getch();restorecrtmode();
}
```

9 PostScript 文件

●矢量图形　●灰度位图　●彩色位图
●页面描述语言　●硬拷贝输出　●堆栈顶

基本的 PostScript 是页面描述语言，它很好地描述了矢量图和位图。最初用于打印机和其它输出设备，现在也用于图像的存储和交换。它的使用平台主要是 Macintosh 机和它的外部设备，现在也用于 PC 机和 UNIX 工作站。几乎所有的 Macintosh 应用程序，许多 PC 机文字处理和单面排版程序，一些高性能的 PC 机图形和 CAD 应用程序，均支持 PostScript 图文件格式。

PostScript 有 4 个变体：Level、Leve2、封装的 PostScript 和显示的 PostScript。Level 是原始语言集，主要用于单色图像和 8 位深的灰度图像，也能提供 R、G、B 彩色图像。Leve 2 的版本要比 Leve 1 版本高，有不少改进的地方，各种色彩空间的使用，以及一位象素具有多达 32 位深的彩色位图。封装的 PostScript 是一个单页描述。显示的 PostScript 是实时的，与设备无关的计算机监视器接口界面。它支持 PostScript 程序的多任务执行，表示名字的二进制编码、窗口系统和位图字模的使用。

PostScript 是一种编码语言，它象常规的算法语言那样，处理语法及语义的正确性。并且，PostScript 编译器有一个文档结构约定(DSC：Document Structure Convention)，这个约定使假脱机打印程序、后处理程序和其他管理程序正确地处理 PostScript 文件。PostScript 语言是一种常驻于任何 PostScript 输出设备上的程序语言，只要将需打印的内容用 PostScript 语言编写的程序描述出来，由主机传送到打印机的 PostScript 编译器解释执行。

9.1 矢量类型图形

PostScript 语言支持矢量类型的图形编程。若要用 PostScript 去绘制一个正方形矢量图，可用下面的语句实现：

```
2.0  2.0  moveto
4.0  2.0  lineto
4.0  4.0  lineto
2.0  4.0  lineto
2.0  2.0  lineto
stroke
```

x y moveto 语句中的 x 和 y 是实数型数据表达的用户坐标，它确立了一个开始点，不划线，执行该语句则确立了划图的起点。x y lineto 语句从当前点划线到 x，y 坐标位置。

此外,语句 x y r angle 1 angle 2 Arc 是一条划弧线的指令。x,y 是圆心坐标,r 为半径,系统将按用户坐标系,从 x 坐标的正方向作逆时针旋转,angle 1 和 angle 2 是弧的开始和结束点。

语句 colse path 用一条线段连接当前点和起始点,使划线的线条成为一个闭合图形。

语句 newpath 删除当前划线路径,取消定义当前点。

最后一个语句保存 Stroke 沿着当前的路径画线,颜色为当前的颜色。

从上述可见 PostScript 的语句集对绘制矢量线段构成的图形的功能是很强的。

9.2 灰度位图图像

PostScript 支持灰度位图图像。对于 Leve 1 PostScript 灰度图像文件,支持每个象素 1,2,4,8 位的灰度图像。而 Leve 2 PostScript 除此之外还支持每象素 12 位的灰度图像。

对这两类 Leve 的图像输出灰度位图像的 PostScript 语言的书写格式为

width height bits/sample matrix datasource image

其中 width 和 height 分别给出原始象素平面的行和列的数目。bits/sample 给出了每个象素的位数,分别为 1,2,4,8 和 12。matrix 是一个特殊的 6 元素的图像矩阵。那些从左到右,从上到下的扫描图像,matrix 为[width 00 - height 0 height]。而那些从左到右,自下而上的图像的 matrix 为[width 00 - height 00]。

datasource 是二进制位图数据源。这个过程是用数据填写任意长度的字符串。填写数据的过程则是将数据压入堆栈顶。图像运算符调用该过程,直到出现文件结束标志。

image 运算符为图像组成运算符,是位图图像产生符。它所使用的数据在 PostScript 文件中,用名字 currentfile 来标记。而 image 还可以从其它地方接受图像数据。

图像数据用 ASCII 字符编码,一般用 16 进制。位图读取与存储的过程用下列语句表示之:

file stringreadhexstring

数据存储在 file 中,读过程将数据写入缓冲区 string 之中,将图像数据当作 16 进制数读取,再转换为二进制数。string 的长度要大于每个象素的位数(bits/sample)。读取数据直到缓冲区填满,或遇到文件终止符。这样在 image 运算符执行后,可以立即找到位图图像数据。

9.3 彩色位图图像

PostScript 支持彩色位图图像。对于 Leve 1 PostScript 来说,加上一条扩展性能功能就可实现彩色图像,而 Live 2 是可以输出彩色图像的,将 image 运算符换成 colorimage 运算符即可实现。输出彩色图像的标准语句格式为如下:

width height bits/component matrix datasource(s) multisources ncomponents colorimage

在标准格式中 width, height 和 matrix 与灰度位图图像格式中的含义一样。

bits/component 则给出每个分量中象点的位数,例如彩色的 3 个分量 R、G、B,每个分量为 8 位,3 个分量为 24 位。

datasource(s)是一个源或多个源,对于彩色位图平面结构,就有 3 个或 4 个。

multisources 是一个布尔数，如果值为真，则为多个数据源。

ncomponents 是色彩分量的数目，如为灰度图是整数值 1，红绿蓝(RGB)或色调、饱和度和光强(HSI)图则为整数值 3，青、朱红、黄三种减底色再加黑色(CMYK)为整数值 4。

colorimage 彩色运算符和 image 运算符本质相同，不同的是它用于彩色位图图像。

9.4 PostScript 程序的执行

PostScript 语言程序由 ASCII 字符和数组成。ASCII 字符首先被分析为一个“对象”，该对象分可执行的与不可执行两大类。可执行对象是 PostScript 的命令，它包括各种运算符和用户定义的过程，如用户子程序。运算符是英语单词，很容易理解与记忆，它是存放在字典栈的系统部分，用户定义的过程则放在字典栈的用户部分。运算和它的操作的数据被放在花括号{…}之中。该括号定义了一个可执行的数组。在花括号之前加上/name，name 是用户自定义的过程名，并在花括号后用 def 运算符结束，就建立了用户定义的过程。PostScript 的数包括带符号整数或实数、字符串和布尔数。程序语句的注释由一个百分号(%)开始。

PostScript 采用堆栈方式来执行操作。PostScript 拥有四个堆栈：操作数、字典、执行和图形状态。PostScript 运行到一个运算符和用户定义的过程时，先将操作数弹出操作数栈，然后执行运算操作。当遇到一个不可执行的数据对象，如一个数或一个数据数组，就把该对象压入操作数栈。

PostScript 程序对一个图形的执行，取决于某些系统参数，这些参数指出了绘图色彩，所画线的宽度，线、弧等所画的位置，坐标变换矩阵等，这些参数构成了图形状态。有些图形状态可使用 PostScriptPost 运算符存储到图形状态栈中，在需要时，从栈中弹出。

PostScript 既支持灰度图像，又支持彩色图像。为了设置色彩空间，PostScript 提供一些专用的运算符。PostScript 是一种编码语言，除了语法和语义的正确性需要检查外，它不要求其它特殊结构。现举一例来看 PostScript 的语句执行。若要填充在坐标为 x、y 处，高为 height，宽为 width 的矩形，PostScript 提供了如下的运算符：

```
x y width height rectfill
```

若要将填充一个长方形的任务定义用户过程，则写为

```
Square {x y width height rectfill} def
```

Square，PostScript 解释器遇到 Square，就知道要执行在 Square 和 def 之间的花括号中定义的过程。具体过程则是从操作数栈中弹出 x，y，width，height 等操作数，在左下角坐标为 x，y 处画上并填充宽为 width 和高为 height 的长方形。

9.5 PostScript 文件硬拷贝输出

众所周知，HP 激光打印机是具有基本页面描述语言的激光打印机，应用非常普及。所以一提起 PostScript，人们往往想起激光打印机。然而事实是，PostScript 能够用来驱动几乎任何一种硬拷贝设备。例如 PostScript 激光照排系统、PostScript 液晶打印机、PostScript 点阵打印机。PostScript 的优点之一是当输出一幅图像时，不需要知道所用的是哪一种类型的 PostScript 设备，从而实现了输出设备的相对独立性。

10 生物医学图像处理

●显微图像　●噪声处理　●过滤效应
●美化图像　●三维重建　●识别分类

生物医学图像是图像处理的一大门类，应用非常广泛。其应用焦点有二：一是在生物体内构成部件观察，在许多情况下是困难的，比如医师为患者诊断，若要了解肺、心、肝、肠、肾等脏器的情况，除了在体外听诊触摸之外，就靠放射科的X光透视，B超声波的透视，或其它射线状装置的透射。这就产生了大量的医学图像，医学专家除运用其专业知识和进行目测等常规的作法外，现时在计算机辅助诊断中，采用计算机图像处理及分析辅助医学专家进行图像分析识别。第二点是生物医学的学科要求，需要进行微观的观测，为观察细胞组织的图像，光镜与电子显微镜（简称电镜）则是最常用的手段。光镜可以由观测人员目测，而电镜出来的图像则必需要应用计算机进行处理、分析。本章主要讨论需要进行计算机处理的微观图像，特别是电子显微镜产生的图像特征及处理。

10.1 电子显微图像特点

观察微观世界，比如分子结构、细胞结构，电子显微镜显示出它的高超的能力。现代电子显微镜的分辨率已经达到了纳米级或更高，能用来观测分子甚至是原子的成像。在实际用电子显微镜观测时，限于样本制作技术，往往达不到这个水平。为了充分发挥电子显微镜在观测中的潜能，必须改善被观测样本的制造技术。先用样本的负染技术代替超薄切片技术，进而又用葡萄糖法替代了负染技术，从而把分辨率提高到5～7 nm的水平。再者，则是采用先进的图像处理技术来改进图像质量。

电子显微镜的放大倍数高，成像系统中的噪声也会被放大，因而会导致图像质量下降。在制备观测样本的过程中，为了提高样本的图像对比度，往往需要染色处理，染料的质量不稳定，或者染料分布的不均匀性，会导致背底噪声，图像变得模糊。另外，电子源发射出来的电子，加速后到达终点的不一致性会引起电子噪声。样本在受电子照射之后，其细胞结构会变粗，使样品丢失某些细节，即所谓辐射噪声。拍照时，照相器材本身质量不好，乳胶涂得不均匀，或乳胶颗粒大小不合适，也会引起底片噪声。所有这些噪声都会影响到图像清晰的程度。

电子显微镜物镜的焦距若调节不当，所得到的图像就不能完全反映其真实图像，原像中有些细节可能是假像，有些细节可能完全丢失，这称为过滤效应。再者，任何一张电子显微镜的照片都是二维的平面像，任何一个三维物体的复杂结构需要通过若干张二维图像才能得到，这就需要用图像处理的技术来解决。

10.2 电子显微图像的计算机处理

10.2.1 二维显微图像的处理

电子显微图像是二维图像，图像处理的任务包括两项：一项是美化图像，经过处理，从而提高图像质量；另一项是消除显微图像中的过滤效应和电子光学中的"假像"问题。

在进行电子显微取像时，有时可能因为照明不均匀，比如说，一边黑，一边变白，这时的图像是很不理想的。从图像处理的角度，可以把图像理解为被摄样品与不均匀的背底的叠加。可以应用计算机定出照明不均匀的方向和程度，对图像进行减法运算，把不均匀的图像减去，就可得到均匀的图像。电子显微镜的被摄样品的切片制作时，往往会有刀口、抓痕等缺陷，影响显微图像的质量。在电子显微成像后，印成照片时，会有爆光不均匀而形成亮、暗悬殊。这些图像缺陷都可以用常规的图像增强方法解决。至于显微图像上的噪声完全可以用低通滤波的方法消除。

消除电子显微成像过程中的"假像"，运用图像处理的办法要远比光学的方法强得多。在解决这类问题时，往往需要两张电子显微图像。在电子束成像的过程中，被倒置的反差和随之产生的畸变，都可以借助两张电子显微图像，很容易应用图像处理的办法来解决。在电子成像的过程中，由于物镜球差、物镜欠焦量和使用的波长的原因，会发生丢失信息的情况。如果有被观测的同一图像区域，一张图像上没有这信息，而另一张图像上又有这信息，那么用两张图像相结合很容易修复丢失的细节信息。

10.2.2 显微图像的三维重建

电子显微图像都是二维图像，可以采用傅立叶变换的方法进行三维重建。任何一个三维物体都可以看成是无数个小点的组成，每个小点的傅立叶变换就是一个简单的平面波，因而整个物体的傅立叶变换可看成无数个平面波叠加的总和。一幅电子显微图像是三维物体在三维平面上的投影，这个投影的二维傅立叶变换正好是这个物体的三维傅立叶变换在这个方向上过中心的截面。因而，如果有足够多的，并且是在不同方向上的投影，都进行二维傅立叶变换，就可得到三维物体在不同截面上的空间分布频率，再对二维变换进行综合运算，并进行反变换，就可以重构出原来物体的三维结构。

10.2.3 显微图像的识别分类

在生物医学图像中，需要测定神经纤维的长度，各种细胞内颗粒大小及分布频度；在进行细胞诊断，特别是癌细胞识别时，需要得出定性和定量的结果。在完成这些工作时，人眼难以胜任，利用计算机图像处理、模式识别的技术，可以完成这些工作。为要识别图像必须先抽取出图像的特征。例如癌细胞的识别可以选择的代表特征有：癌细胞的胞核很大（核质比例与正常细胞恰好相反），癌细胞的外形不规则，正常细胞外形规整，多为圆形或椭圆形；核内染色质出现团块，正常细胞内染色质分布均匀。这些特征在图像分析中是可以抽取出来的。抽取出生物的特征，就可以找出正确的识别分类的规则，进行图像识别的工作。

11 金属结构图像的处理

●金属构件　●无损探伤　●射线底片
●自动处理　●模式识别　●缺陷分类

焊接是制造和连接各种工程构件最重要的方法，无损探伤则是检验焊接质量好坏的重要手段。焊接技术的发展是和无损探伤技术的提高分不开的。随着焊接方法和工艺的改进，目前焊缝的质量已能完全达到母材的水平。但是，由于影响焊接质量的因素太多，诸如电流、焊件、焊材、焊剂、环境以及人为因素等等，即使是十分成熟的焊接工艺，也难免在焊缝中残留这样那样的缺陷。所以，要检验构件的可靠性和安全性，焊缝自然是检查的重点。

射线探伤是应用较早的检验方法。利用射线照相检验焊缝内部缺陷具有准确、可靠、直观等优点，射线照相底片不仅可以用于缺陷的分析，而且还能作为质量凭证存档，这是其它无损检验方法所无法比拟的。

早期，对焊缝 X 射线底片的评定一般是人工进行的。因底片质量受光源、被测材质和冲洗水平等因素的影响，以及检测人员的水平、经验不一，所以误判、漏判的现象难免发生。从 90 年代起，人们开始运用计算机模式识别技术，对焊缝缺陷进行分类识别。

11.1 系统设备及工作原理

焊缝缺陷计算机自动识别系统组成如图 11.1 所示。图像转换装置负责把 X 射线底片变成模拟图像(图像采集)，具体设备可以是高精度转鼓式发片机，也可是负片扫描仪等。A/D 转换器将模拟图像量化成 8 比特灰度图像送至计算机处理。

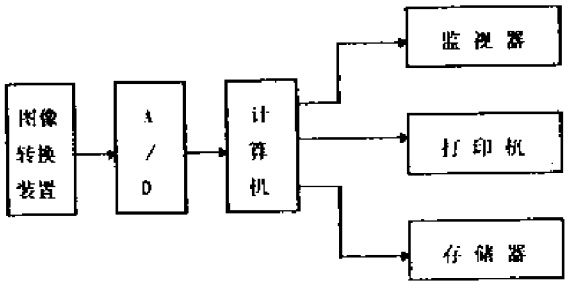


图 11.1 系统组成框图

11.2 图像预处理

在量化后的底片图像数据中，不可避免地存在多种噪声。为正确抽取缺陷特征，需在底

片自动评定前进行适当处理，其中包括：

11.2.1 灰度变换

焊缝X射线照相底片图像的灰度范围一般较窄，对比度差，尤其是在缺陷部位。经过灰度变换，拉宽图像灰度域。

11.2.2 滤波

噪声使图像模糊，分析困难。滤波操作可有效地去除多种干扰，突出图像中的目标对象，是图像预处理最重要的工作。图像滤波的算法较多（详见第3章），使用中需视具体图像灵活运用。

11.2.3 图像二值化

只有两个灰度级的图像称为二值图像。

金属结构图像二值化时，最常用的方法是设定某一阈值 T，大于 T 的像素群以1表示，小于等于 T 的像素群以0表示。

二值图像有利于对图像边界的跟踪。

11.2.4 焊缝缺陷的轮廓线跟踪

计算机对经过预处理、二值化后的缺陷图像进行光栅扫描，寻找还未打上已跟踪标记的边界点B(0)，如发现这样的点，就开始一条边界线的跟踪。在B(0)的8邻域中，按逆时针的顺序，判断像素的值，将最先遇到的1像素B(1)作为下一个边界像素进行跟踪。在B(1)的8邻域中，继续按逆时针方向寻找1像素，并把它定义为B(2)。用同样的方法可求出B(3)、B(4)、…。如果 $B(m+1)=B(1)$，$B(m)=B(0)$，则一条边界跟踪结束，B(0)、B(1)、…、$B(m-1)$形成一个缺陷边界线。

在跟踪过程中，对每个 $B(i)$ 分别赋给一个已跟踪标记。

11.3 物理量计算

根据国家GB3323—87标准，焊缝缺陷分为五类：

(1) 裂纹：具有尖锐的端部和大的长宽比，表现为一条直线或曲线状的缺陷。

(2) 未焊透：在射线底片上呈连续的规则黑线。

(3) 未熔合：形状近似于未焊透和线状夹渣。

(4) 圆形夹渣：指残留在焊缝中的熔渣，缺陷长、宽之比一般小于等于3。

(5) 长形缺陷：长宽之比大于3。

为了能较好的识别上述缺陷，主要需要计算以下物理量：

(1) 缺陷周长（L）。

(2) 缺陷长径（L_1）。

(3) 缺陷短径（L_2）。

(4) 缺陷面积（S）。

(5) 周长平方面积比(P)。

周长平方面积比 $P = L^2/S$,这是一个能很好反映边界特征的参数。当缺陷为圆形时,L^2/S 最小。缺陷越长,边界越不规则,L^2/S 越大。周长平方面积比对缺陷边界形状非常敏感。

(6) 长宽比。

L_1/L_2 是国标规定的使用标准。实验结果表明,L_1/L_2 小于等于 3,可以比较准确地判别出圆形缺陷。

(7) 面积像素数与周长像素数之比 (F)。

$F(=S/L)$反映单位边界长度所围缺陷面积的大小,对 L_1/L_2 大于 3 的长形缺陷,若 $F<1.2$,则一般是裂纹。

11.4 计算机自动识别

利用前述分析,焊缝缺陷计算机自动识别过程可简述如下:

(1) 用缺陷的长宽比是否小于等于 3,来判断是不是圆形缺陷,若不是,则进行(2)。

(2) 用面积像素数除以边界像素数,若比值小于 1.2 为裂纹,否则进行(3)。

(3) 用边界水平方向像素数除以边界总像素数,如比值大于等于 0.8,该缺陷为未焊透,否则进行(4)。

(4) 若缺陷长宽比大于等于 5.0,该缺陷为未焊透,否则为长形夹渣。

12 遥感图像处理

●遥感技术　●遥感信息处理　●辐射校正
●几何校正　●图 像 解 译　●图像分类

遥感技术是高新技术领域中的一个重要分枝。在几百米的空中或几百公里的高空，在飞行器上(如气球、飞机或人造卫星)安装了传感器设备，它接收从地球表面辐射或反射出来的电磁波信息，经过计算机处理从而实现对地表目标或物体的观察与检测。

12.1 遥感技术与遥感信息的获取

遥感技术包括三个方面的内容，即遥感平台、传感器和遥感信息处理。遥感平台指的是运载工具，分为航空平台和卫星平台。航空平台是指气球和飞机，它所获得的地表图像，称之为航片。卫星平台则是以人造卫星为运载工具，在卫星上记录的地表图像，称之为卫片。航空平台飞行较低，在大气层以下。卫星平台往往在数百公里的高空。传感器是指安装在运载工具上的遥感仪器，在气球和飞机上安装的是大像幅航空相机，在卫星上安装的多光谱扫描仪和红外扫描仪。传感器直接接收不同种类的地物反射和幅射的能量，并将接收到的能量用数据或影像方式记录下来。

遥感技术中采用的电磁波谱为可见光、红外线和微波。可见光的波长范围在 0.38 ~ 0.76μm 之间，由红、橙、黄、绿、青、蓝、紫七色光组成，人眼对可见光的组成很敏感，是鉴别地物的主要波段。红外线的波长范围在 0.76 ~ 1000μm 间。自然界中的任何物体不仅能反射太阳幅射来的红外线，而且能在常温下向外发射红外线。微波的波长范围是 1mm ~ 1m，它能直线传播，并且不受雨、云、雾等气象条件的限制，是全天候的遥感波谱。

众所周知，太阳光照射地物后，地物表面接收一部分，反射一部分。由于地物表面组成的复杂性，不同的地物具有不同的光谱特性。所谓地物的光谱特性是指不同的地物表面的反射率与波长的关系，反射率高的地物反射入射光的能力强，在遥感仪器上记录的图像的亮度值就高，图像的色调就浅；反之，反射率小的地物表面反射入射光的能力弱，图像的亮度值就低，色调就深。地物的光谱特征是利用专用的光谱仪测定的，不同地物具有不同的光谱特性。只有很好地掌握了地物的光谱特征，才能很好地为传感器设计最佳波段，就是在某个波长范围内很好地监测所需要观察的地物。掌握了光谱特性，才能很好地进行遥感图像的判读和计算机的识别分类。

12.2 卫星遥感图像的特点和分类

人造地球卫星的对地观察是它的重要任务之一。卫星的制造与发射的费用是非常之高

的。一旦卫星升空进入轨道之后，它就可以为对地观察长时间地提供卫片。卫星对地观察图像资料具有全局宏观性、周期性的特点，而且费用价格要低于飞机提供的航片资料。所以，卫星遥感图像资料随着各行各业的广泛应用而很快成为商品性资料进入高技术领域市场。下面介绍国际、国内广泛应用的几种卫星图像资料的特点。

12.2.1 美国陆地卫星的图像资料

美国陆地卫星从1972年升空，已经历了一个发展过程。早期使用的是多光谱扫描仪，获得被称之为MSS的图像资料。陆地卫星在高空飞行扫描时一个扫描行为185公里，扫描行的宽度为79m。在扫描过程中每次采样是地表$(57\times79)m^2$大小的灰度值，这个面积称为一个像元(pixel)。因为一个扫描行有185km，它包含$185km\div0.057km=3240$个像元。一幅卫星图像有$(185\times185)km^2$，它包含$185\div0.079=2340$扫描行，卫星摆头每摆动扫描一次为6行，所以，一幅图像有$2340\div6=390$个扫描带。因此每幅图像有$3240\times3240=10.5\times10^6$个像元。从扫描镜引入的地物光谱信息经聚光滤色光学系统后，分成MSS4，MSS5，MSS6和MSS7四个波段，分别对应于波长(0.5～0.6μm)，可见光的绿色波段(0.6～0.7μm)，可见光的红色波段(0.7～0.8μm)和(0.8～1.1μm)的近红外波段。

在80年代改进的陆地卫星上采用了一种新的遥感仪器，称为专用绘图仪，它所获得的图像称为TM图像，相对于MSS图像，它具有空间分辨率高的特点，每个像元对应的地表面积从$(57\times79)m^2$减小到$35m^2$。除此而外，它的光谱分辨率由于采用窄波段传感而明显地提高。它的波段数从4个波段增加到7个波段，因而合成的颜色更具有目视的直观性，并且可以很好地用于植被、水面的检测。

陆地卫星可以扫描世界任何一个地区的图像，人们由它的图像检索中心进行目录检索，了解它的卫星图像。图像检索中心编制了世界性的卫星图像索引图，在这种图上标出图像的中心位置。该中心位置就是由北向南的轨道线与东西向扫描行线的交点。用户可以提出该中心位置的编号，就能得到所需图像。我国从美国购买来的陆地卫星图像，保存在国家测绘局遥感服务中心，供我国多种行业的用户选用。

12.2.2 法国SPOT卫星图像资料

法国空间研究中心(CNES)研制的地球观测系统SPOT卫星，从传感器的设计、光谱段的选择和分辨率的提高都有很大的进展。SPOT卫星有两台相同的可见光高分辨率传感器，收集系统为一透镜，探测器为CCD。SPOT卫星图像的分辨率由相邻光敏元件中心点的距离确定，故称为像素分辨率。在可见光全色波段部分为13μm，红外部分为16μm，所以地表的分辨率，也就是每一像素所代表的面积为$(10\times10)m^2$。SPOT卫星图像以它的分辨率高而著称。SPOT卫星的飞行高度为832km，它的传感器可在垂直于轨迹方向上扫描±27°，这样就有可能在地表±400km的范围里，确定60km的扫描带，因而有可能对某一规定地区连续观测数天。它具有双传感器的双轨迹运行，所观测的资料，可获得立体覆盖。由于SPOT卫星图像的地表分辨率很高，运用它的全色片图像，可以制出质量很高的1∶10万的地图。

12.2.3 我国国土卫星的遥感图像

我国也成功地发射了试验性国土卫星，它的主要任务是完成国土资源的普查。国土卫

星的传感器为全景扫描摄影机，它由地物摄影机和星空摄影机两部分组成。当卫星运行到需要摄影成像的地区上空时，特控系统控制摄影机工作，镜头对准地面扫描摄影成像。在地物摄影机工作的同时，星空摄影机对星空拍摄具有星位的星空照片，以确定卫星在摄影瞬间的空间位置。国土卫星每天绕地球运行 16 圈。飞越我国上空时的高度为 180～200km，它的遥感成像地表分辨率，在图像中心较高为 10～15m，边缘部分减小到 30～40m。国土卫星在完成观测任务后，连同影像按计划返回地面，回收成功率 100%，为世界航天界所称颂。

12.3 遥感信息处理

遥感图像具有两大特点：一是信息量特大，二是图像信息中存在着噪声和失真。为了节省遥感信息的存储容量，提高数据传输效率，需要对遥感信息进行编码压缩。为了去除噪声必须运用图像处理的方法进行预处理。多光谱扫描仪在录取图像的过程中产生误差和几何畸变失真。为了补偿这些失真，必须利用计算机进行矫正，包括辐射校正、几何校正等。因而遥感信息处理的框图如图 12.1 所示。

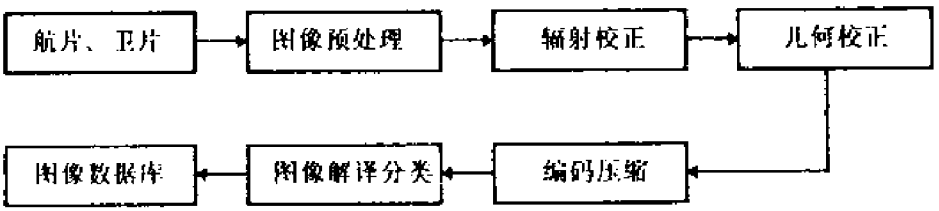


图 12.1 遥感信息处理流程图

12.3.1 辐射校正

多光谱扫描仪获取遥感信息时，每扫描一次如陆地卫星(landsat)将会获得 6 行信息，若有一个探测器故障，会导致失漏一行数据，在图像上产生一条黑线。若探测器的响应频率产生漂移，即某探测器的增益或振幅比其它探测器高，结果在扫描的过程中，在图像信息上产生明显的条带，与全幅图像完全不协调匹配。航空摄影也有类似的问题，往往会产生同样的地物出现不同的灰度，或者同样的灰度点，而实际上并非同样的地物。为了恢复它的原始图像就要用计算机对它进行辐射校正处理，往往采用相邻的扫描线的平均灰度值和去除黑线或条带状失真，还可以用直方图的操作处理，解除灰度的的失真。

12.3.2 几何校正

遥感图像的几何失真，分内部畸变和外部畸变。内部畸变是由传感器的内在问题所引起的，如镜头的像差，比例尺的变化，歪斜，中心移动，扫描非线性等原因造成的几何扭曲。外部畸变是由于运载工具的姿态和目标引起的畸变，如卫星飞行中的滚摆，俯仰，偏航所引起的畸变。对于上述原因所引起的图像失真的校正，往往是很复杂的，要具体分析畸变的原因，才能找出算法去解决失真。

12.4 遥感图像的解译与分类

遥感图像应用于各行各业，如地质矿产的探测，农作物分类与估产，森林分类长势和林火防治，湖泊河流污染、洪涝灾害估计，沙漠化及其扩张，城市发展及国土利用等，最终目的是要运用遥感图像解决地表的问题，通过对遥感图像的解译，进行多种定性定量的分析。地学专业人员用仪器（如立体镜、放大镜、量积仪等），结合地面测定资料，对遥感图像进行判读，将判读结果标注在遥感图像上。这种判读的准确程度取决于专业知识和判读经验。现时较好的遥感判读是人机交互进行的，已有许多计算机软件，让专业人员建立标记，训练待识的目标，按照软件中所给出的分类算法进行分类。在人机对话的过程中，出现错误可进行人工的干预。专业人员根据遥感图像的灰度特征、形状特征、纹理特征及其相互关系，加上专业经验，可以对地物很好地判读。它充分地利用专业人员的知识经验，又可避免人为的主观因素。利用计算机快速分类的算法，模式识别理论的决策方法，进行分类识别。计算机模式识别理论与技术的发展，采用人工智能的方法，建立专家系统、知识库、数据库、推理规则，结合具体的专业范围，当输入遥感图像之后，即可进行计算机自动判读与分类识别。

12.5 遥感图像的应用

12.5.1 地质遥感图像处理与应用

在人烟稀少的崇山峻岭之中进行地质勘探和寻找矿藏是极其艰苦的工作，但这项工作对于国家的发展又是极其重要的，因而地质工作者就深入研究用遥感技术来进行地质勘探的工作。因为，不同的地质构造，不同的矿床的表面光谱特性是完全不相同的，这在遥感图像上的反映也是很明显的。地质遥感工作有进行地貌、地物特征信息的研究，进行地球物理与化学特征光谱的测定与分析，并进行矿区岩类和矿区构造的识别的研究。有了这些基础研究工作，再运用遥感图像进行目视解释和人机交互解释，就可制成地质遥感的分类图样本。特别是在大范围的地质勘探工作方面，国内外均取得了重要成果。地质工作者运用遥感技术，研制了丰富的软件系统，进行计算机辅助找矿取得了成功。

12.5.2 森林遥感图像处理与应用

我国国土上，13%的土地面积上复盖着大片的森林。森林不仅是重要的自然资源，它提供了国家建设所需要的木材，另一方面它调节了大自然，是生态环境保持平衡的重要保障。因而充分利用森林资源和认真保护森林资源，是人类面临的重要任务。为了完成这一任务，需要人们采用多种新技术来实现。其中遥感技术以它的宏观性、周期性、长期性等特点而倍受青睐。在国内外许多森林管理部门均已建立了遥感系统，定期地利用遥感卫星资料分析森林样本、森林长势。在管理部门有60多项指标，其中的一些指标可以从遥感图像资料上获得，因而可以构成森林的遥感信息库。这个庞大遥感信息库对于管理部门制定砍伐，种植，保护，抗虫灾害等方案均具有重大意义。众所周知，林火是一项很严重的灾害，对于林火的观测，运用遥感图像制定灭火方案是一项很得力的办法。森林工作者根据对周期的森林

遥感图像的分析，掌握森林长势、气候干旱的程度。对于林火的预测及防护已成功地进行了大量的工作。

12.5.3 国土资源遥感图像处理与应用

任何一个国家的发展都离不开对本国资源的利用。在众多资源中，土地资源是第一位的，它涉及到工业、农业、经济的多个方面。世界上不少国家已经利用遥感技术，摸清了本国的土地资源，如土壤类型，多种地貌的分析细节，多种自然条件，如土、水、林、草的分布和它们的关系，对于农业区域的规划、开发、利用发挥了很重要的作用。在农业资源的开发利用方面，应用遥感技术进行农作物估产，诸如小麦、水稻、玉米、大豆的长势及产量预测，显然对国民经济是极为重要的。对于大片草场的调查与监测，对于生态环境，对于畜牧业均是非常重要的。遥感图像对于大面积的草原的观测尤其显出它的优越性，估计草的长势及产量和草场潜力。国土自然灾害的监测，遥感技术又是大有作为，大面积的洪涝灾害和干旱，遥感图像提供了准确的信息，对于抗灾、救灾发挥了重要的作用。

12.5.4 海洋遥感图像处理与应用

海洋占据了地球大部分的表面，海洋资源是人类共同的财富。拥有海岸线的国家，对于海洋的开发与利用已取得了巨大的经济效益。其中利用海洋遥感进行大面积的实时调查与监测是这些国家开发和利用海洋的极为重要的手段。80 年代以后，不少国家发射了海洋观测卫星，专门从事海洋利用及开发，可见海洋遥感事业的重要性及其发展。在海洋遥感的应用上有下面重要的方面：海浪观测，它涉及到航海及海洋渔业、海岸变迁、海岸河口研究、海滩利用、海洋石油及其它海洋资源，如渔业资源的调查监测；海洋水文，水温，海洋动力学研究；海洋环境污染监测等。

12.5.5 气象卫星云图的处理与应用

1)利用 GMS 红外云图资料估测降水

气象和水文部门一般都是利用常规雨量站测定降水量，但由于混合性、对流性降水分布不均匀，下大雨的地方不一定设置雨量站，因而有时真正的强降水未能测出。再者，受通信条件的限制，雨量计资料一般要滞后若干小时才能传送到有关部门。因此，仅仅用常规雨量站资料作降水量预报和指挥抗洪防汛，不但在实时性上不能满足需要，而且也难以反映降水实况。

地球同步卫星能准确地提供大范围内气象信息资料，其中，红外云图反映被测对象的表面温度，若测量目标是云区，则又反映了云顶的高度。通常，积雨云和发展中的锋面云系都有较高的云顶，而这些云系又是产生降水的主要天气系统。因此，用静止卫星云图定量估测降水，是近 10 年国内外气象工作者探讨的重要课题。这里提出的云顶温度降雨方法是笔者 90 年代的研究成果之一。

(1) 方法

数字化 GMS 红外云图资料与地面降水之间还没有直接的对应关系式，不少学者曾就此作了许多有益的研究，如 Liljas 分别以红外云图上的云顶温度和标准化后的可见光云图的反照率为坐标，对相应的 1 小时降水量作点聚图；Follasubee 根据卫星云图的识别原理和云

移动模式，结合天气学知识，得到了估计 12 小时降水量 $P_{12}=0.007D\cdot P_N$，D(是降水云的持续时间，P_N 是月平均降水量)。这些方法有一定效果，但也存在不少问题，如估测精度不高，强降水估计不出等。

① 建立云顶温度(T)与地面实测降水(R_g)曲线。

② 选取订正因子。T—R_g 曲线是建立在多样本基础上的拟合关系曲线。由于降水受多种因素制约，同一类型天气因云况不同，对应的地面降水量也不完全相同。为了提高估测精度，在 $T-R_g$ 曲线基础上，再选取下列因子作为计算降水订正量的参数。

r_i 为云顶温度于 $-5\sim-16$℃、$-16\sim-26$℃和低于 -26℃的云区面积占窗区负温度云总面积的比率($i=1,2,3$)。在汛期降水过程中，基本上可以排除不产生降水的高云的影响，上述 3 个温度区间大体上对应该季节江淮流域上空层状云、层积云和对流云的云顶(由 0℃等温度线平均高度和温度随高度的递减率，计算出云顶高度分别是 4.5～5.6、5.6～6.6 和大于 6.6km)。在某一类降水系统中，3 种云系的比例差异在一定程度上影响着同一云顶温度对应的地面降水强度。

r_d 为可能产生强降水云(一般把云顶温度低于 -48℃(去掉非降水高云)的云，视为可能产生强降水的云)的云区面积占窗区总负温度区云面积的比率。

F 为系统形状因子，反映锋面降水、切变线降水、局地降水等不同降水系统。

③ 地面降水估计方程

$$R(\Delta t)=R_1(\Delta t)+R_2(\Delta t,r_i,r_d,F) \tag{12.1}$$

式中 R_1 是根据 $T-R_g$ 曲线，由红外云图资料反演的降水，R_2 是降水修正量。

(2) 实施步骤

① 预处理

地球同步卫星云图资料在离开星下点处，图像在不同程度上产生了畸变。为获得客观的降水分布，首先将卫星资料作校正，使图像恢复到正常状态。然后滤去非降水资料，以减小运算量，提高分析速度和精度。

② 降水修正量计算

先计算 r_i、r_d 及 F，然后求出降水修正量

$$R_2(\Delta t,r_i,r_d,F)=Q_iR_1+Q_dR_1+Q_FR_1 \tag{12.2}$$

Q_i、Q_d、Q_F 分别是与 r_i、r_d、F 相对应的订正系数，或大于零，或小于零，或等于零，视订正因子值而定，$R_1=R_1(\Delta t)$。

③ 降水估测

计算 r_i、r_d、F 值，再确定 Q_i、Q_d、Q_F，然后根据(12.2)式求取降水修正量 $R_2(\Delta t,r_i,r_d,F)$。Q_i 为考虑不同云顶温度的云产生的降水与 $T-R_g$ 曲线值之间的差异而给予的修正；Q_d 为由于在降水系统不同阶段整体降水水平不一样而给予的调整；Q_F 则是针对天气类型相同、降水系统不同(如带状的锋面、切变线降水及局地对流降水等)而提出的订正。

根据(12.2)式即得当前时次各网格点上的降水量 $R(\Delta t)$。

将本次计算的格点降水量与过去时段降水量相加，得到格点累计降水量。

求取分析时段内窗区降水总量 P_T：

$$P_T=\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{q}R_{i,j,k}\cdot\Delta T_k\cdot\Delta s_{i,j} \tag{12.3}$$

式中　$m \times n$ = 窗区面积像元个数；

q 为时段数；

ΔT_k 为图像时间间隔；

$R_{i,j,k}$是像元上的平均降水；

$\Delta S_{i,j}$是面积元面积。

卫星资料星下点分辨率最高，离开星下点分辨率逐渐降低。因此，在云图上不同经纬度处，一个格点所代表的实际云区面积是不同的。在计算降水时，必须根据格点所处的地理位置进行面积修正。

2) 卫星遥感资料与云迹风

通过追踪遥感资料中气象目标物图像而获得的风场称之为云迹风。

风场是最重要的气象要素之一。用云图资料估计风场，一方面可大致了解降水区内部流场结构，加强对中、小尺度天气系统的监测和预报；另一方面可了解云的发展，为人工影响天气和云雾物理研究提供一个探测手段。同时，对民航飞行安全亦有十分重要的意义。

(1) 基本原理

令 $f(x,y)$为某二维连续图像函数，则其$(p+q)$阶原点矩和中心矩分别定义为

$$m_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^p y^q f(x,y)\mathrm{d}x\mathrm{d}y \tag{12.4}$$

$$\mu_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (x-\bar{x})^p (y-\bar{y})^q f(x,y)\mathrm{d}x\mathrm{d}y \tag{12.5}$$

式中

$$\bar{x} = m_{10}/m_{00} \qquad \bar{y} = m_{01}/m_{00}$$

$$p,q = 0,1,2,\cdots$$

考虑到使用的是离散的数字化图像资料，所以(12.4)、(12.5)式可用下列形式代替：

$$m_{pq} = \sum_{i=0}^{M}\sum_{j=0}^{N} i^p j^q F(i,j) \tag{12.6}$$

$$\mu_{pq} = \sum_{i=0}^{M}\sum_{j=0}^{N} (i-\bar{i})^p (j-\bar{j})^q F(i,j) \tag{12.7}$$

式中　M,N 为图像尺度；$F(i,j)$ 为图像灰度；i,j 为网格点坐标。

定义

$$\eta_{pq} = \mu_{pq}/\mu_{00}^{\gamma} \tag{12.8}$$

式中

$$\gamma = \frac{p+q}{2} + 1, \qquad p+q = 2,3,\cdots$$

η_{pq}称为标准化中心矩。

根据上述定义，可导出7个矩不变量如下：

$$\varphi_1 = \eta_{20} + \eta_{02}$$

$$\varphi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2$$

$$\varphi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2$$

$$\varphi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2$$

$$\varphi_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2]$$
$$+ (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$
$$\varphi_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})$$
$$\varphi_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2]$$
$$- (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2$$

以上 7 个矩不变量不仅对连续函数图像发生平移、旋转和放大缩小时都是不变的，而且对离散的图像也具有良好的保守性。为减少 φ_i 的动态范围，取 $\psi_i = \ln\varphi_i (i = 1,2,\cdots,7)$。实际上无需用 7 个矩不变量跟踪模块。如可对 ψ_i 取以下代数组合：

$$\xi = \sum_k |\psi_k| \qquad (k = 1,3,4,6) \tag{12.9}$$

作为模块特征量。大量的数值试验结果表明，ξ 对相邻时刻的同一回波模块具有很好的保守性，同时对不同的回波模块又有明显的差异性，故可用 ξ 作为跟踪回波模块的特征量。

按一定方式将相邻时刻的两图像区划分成若干个大小相同的模块。设 t_1 时刻在 M_1 处以 (i_1, j_1) 为中心的模块在 t_2 时刻移动到 M_2 处(以 i_2, j_2 为中心)，则判别因子为

$$d(i,j) = |\xi(i_1,j_1) - \xi(i_2,j_2)| \tag{12.10}$$

严格地说，不同时刻的同一模块，其判别因子 d 应为零，但由于模块在随流场、移动过程中发生形变转动等原因，$\xi(i_1,j_1)$ 和 $\xi(i_2,j_2)$ 不可能完全相等。不过这种差别很小，远小于不同模块之间的差别。故在所示的搜索区内，判别因子最小的 t_2 时刻的模块，即可认为是 t_1 时刻被跟踪的模块。根据已知时间间隔、两时刻的模块中心坐标和网格距离，即可求得所跟踪的模块速度矢量。按上述方法处理其它模块，则流场就可以估计出来。其位移分量 u, v 由下式求之：

$$\left.\begin{aligned} u &= (i_2 - i_1)\cdot\Delta l/\Delta t \\ v &= (j_2 - j_1)\cdot\Delta l/\Delta t \end{aligned}\right\} \tag{12.11}$$

式中 Δl 是相邻格点间的距离；Δt 是相邻资料的时间间隔。

(2) 处理流程

使用卫星云图资料测量云迹风，其处理过程一般包括预处理、特征量计算、模式识别、流场计算、合理性分析、产品输出几部分。预处理主要是对原始资料进行校正，划分模块。计算特征量时，先计算低阶原点矩 m_{pq}，接着计算 μ_{pq}、η_{pq}、φ_i，最后求取特征量 ξ。

在知道了 u, v 分量后，合成风向、风速分别为

$$\beta = \text{arctg}\frac{u}{v} \tag{12.12}$$

$$v = (u^2 + v^2)^{\frac{1}{2}} \tag{12.13}$$

13 3S 系统与图像处理

●遥感系统 ●全球定位系统 ●地理信息系统
●信息融合 ●图像数据库 ●3S 应用

遥感系统(RS)、全球定位系统(GPS)和地理信息系统(GIS)统一组合成为一个综合的信息系统,称为 3S 系统。

13.1 地理信息系统(GIS)和遥感图像处理

地理信息系统(GIS)以区域地理环境和地图为背景,加入社会经济发展的统计数据,形成方便各行各业使用的咨讯系统,已成为工业、农业、交通运输、军事等多种领域发展的重要手段。国内外研制了各种不同的地理信息系统已达到了成熟的阶段。常规的地理信息系统中包含了大的图形数据库、社会经济统计数据库,还有些系统中包含了图形标记工具库、知识库等。在做总体规划的部门,如区域经济规划,工农业发展规划,土地利用规划,环境综合治理规划等部门,对于地理信息系统的要求是非常迫切的。方便使用,人机交互对话的友好界面,使地理信息系统的结构越来越复杂,可以说是信息科学与地理科学的结合。在国家机关的领导部门,在许多涉及地理环境的部门,许多地理信息系统已在很好地运行,发挥了重要作用。

从 1972 年开始的对地观测卫星的发射,卫星从几百千米的高度对地球作全面宏观的观测,卫星遥感获取的图像已成为另一个重要的信息源。尤其是各种对地的专用卫星的发射,如海洋卫星、国地资源卫星、气象卫星等,定时地、周期性地、持续地向地面发回获取的图像。而人们熟知的陆地卫星,SPOT 卫星的图像更是从未间断过。从各种卫星图像上得到的丰富的信息,如国土利用,土壤类型,农作物长势,城市发展,海岸变迁,矿藏资源,森林种类及长势,湖泊污染等等,都需要经过遥感图像的处理后,从图像中抽取出来,再将这些重要的信息纳入到地理信息系统之中。所以,在现有的计算机地理信息系统中,已经加入了遥感图像库,将遥感信息直接纳入地理信息系统,由计算机软件进行处理,抽取信息和信息融合,使得地理信息系统成为一个包含遥感信息源的运行系统。

13.2 全球定位系统(GPS)和遥感图像处理

1993 年,美国发射了 24 颗地球同步卫星,环绕地球形成了一个全球定位系统。由于这 24 颗卫星与地球同步运转,所以在高空的位置相对于地球是不变的,而且这 24 颗卫星的发射是高精度的,可以认为这 24 颗卫星所形成的坐标系相对于地球是静止的,并且是准确的。

它的主要功能则是构建快速、准实时的定位系统，及根据系统提供的信息，计算地面高程。在地面可移动的运载机械上安装 GPS 接收机，就可快速地得到该运载机械的地理坐标，它的定位不依赖于地面控制点，而是在高空的 GPS 卫星系统。目前，GPS 定位系统已经进入商业运用阶段，例如，在汽车上安装了 GPS 接收系统，在屏幕上电子地图里就可以看到该汽车在电子地图中的位置。在轮船、飞机、火车等一切移动目标上均可安装 GPS 接收系统，成为理想的导航设备。GPS 系统与遥感图像系统相结合，其应用范围是很广泛的。例如结合遥感图像进行地质勘探，可以利用 GPS 进行同步工作，在遥感图像上标记要检测的目标，标记图像中的等高曲线，使得遥感图像与行政地图的配准更加方便准确。将全球定位系统、遥感图像系统组合进地理信息系统，现在已经有了专门的计算机软件来实现，形成可运行的 3S 系统。

例如，一个专门用于抗洪涝灾害的 3S 综合应用系统，在地理信息系统中有各种地理、经济、环境数据，遥感图像则反映了洪涝大水的灾情，GPS 系统提供的快速实时的水面救灾船队的信息，为抗洪救灾的指挥部门进行洪涝灾情分析和救灾提供了各种信息，形成了一个抗洪救灾的辅助决策环境。

14 马尔可夫随机场及其在图像分析与处理中的应用

●邻域系·连通系 ●吉布斯分布 ●模拟退火
●噪声处理 ●纹理分析 ●视觉效果

近年来，用于模拟空间中随机变量之间相关性的马尔可夫随机场(MRF)理论越来越引起人们的关注。马尔可夫随机场模型是由 D.Geman 和 S.Geman 在 1984 年提出的，作为一种描述图像结构的概率模型，它已广泛应用于图像分割、图像识别、图像恢复、纹理分析、图像解译等领域，其突出特点是通过适当定义的邻域系和相应的连通系上的能量函数引入了结构信息。由于它有模拟退火算法的支持，在图像处理中有特殊的用途。随着马尔科夫随机场理论和方法的不断发展，它在图像处理领域中所起的作用显得越来越突出。

14.1 马尔可夫随机场的定义和基本性质

马尔可夫随机场(以下简称 MRF)是马尔可夫随机过程在二维参数集中的推广。

定义 1：随机场

设(Ω,F,P)为一概率空间，$\lambda=\{1,2,\cdots,l\}$为一指标集或参数集，$S=\{0,1,2,\cdots,M\}$为状态空间，$X_{ij}:(\Omega,F,P)\to S(i,j\in\lambda)$是随机变量，则称其全体 $X=\{X_{ij},i,j\in\lambda\}$是 S 上的随机场。

定义 2：邻域系

对给定的指标集 λ，若 $N=\{N_{ij}\subset\lambda\times\lambda,i,j\in\lambda\}$满足：

(1) $(i,j)\notin N_{ij}$

(2) 若$(i,j)\in N_{i_2j_2}$ 则$(i_2j_2)\in N_{i,j}$

则称 N 为$\lambda\times\lambda$ 上的一个邻域系，N_{ij}称为(i,j)的邻域。在图像中，最常用的对称邻域系形式是：

已知图像 Ω，Z 为Ω 平面上的格点，即 $Z=\{(i,j)|\quad i,j$ 为整数 且 $1\leqslant i,j\leqslant N\}$，则 $N_{ij}=\{(k,l)|(k,l)\in Z\quad$ 且 $0<(k-i)^2+(l-j)^2\leqslant c\}$。

当 $c=1$ 时 $N_{ij}^{(1)}=\{(i-1,j)(i+1,j),(i,j-1),(i,j+1)\}$

当 $c=2$ 时 $N_{ij}^{(2)}=\{(i-1,j-1),(i,j-1),(i+1,j-1),(i-1,j),(i+1,j),(i-1,j+1),(i,j+1),(i+1,j+1)\}$

定义 3：连通系

设 $N=\{N_{ij},i,j\in\lambda\}$为一邻域系，若 $C=\{C_a,a\in\lambda\}$满足：

(1) $Ca\subset\lambda\times\lambda$

(2) 对一切$(i_1,j_1),(i_2,j_2)\in Ca$,则有$(i,j)\in N_{i_2j_2}$

则称 C 是关于 N 的连通系。

定义 4:邻域的 MRF

设 $X=\{X_{ij},i,j\in\lambda\}$是随机场,$N=\{N_{ij},i,j\in\lambda\}$是邻域系,则如果对一切$(i,j)\in\lambda\times\lambda$,有

$$p\{X_{ij}=x_{ij}\mid X_{kl}=x_{kl},(k,l)\neq(i,j)\}=p\{X_{ij}=x_{ij}\mid X_{kl}=x_{kl},(k,l)N_{ij}\}$$

就称 X 是关于邻域系 N 的 MRF。

MRF 定义的直观意义是,如果把(i,j)看作为"将来",而把 N_{ij}看作"现在",所有其它的(i',j')看作"过去",则在已知"现在"的状态条件下,过程的"将来"状态的概率与其"过去"的状态无关。换个角度来讲,(i,j)只受到其周围点即 N_{ij}的影响,而与其它的点无关。

定义 5:吉布斯(Gibbs)分布

对于邻域系 N,$C=\{Ca,a\in\lambda\}$是它的连通系,若随机场 $z=\{x_{ij},i,j\in\lambda\}$的联合分布为

$$p\{X=x\}=\frac{1}{k}\exp(-v(x)/t)$$

其中 k、t 为常数,$V=\sum_{a\in\lambda}V_{Ca}(x)$,函数 V_{Ca} 只依赖于 x 中 Ca 有关的那些(k,l)的 x_{kl} 值。则称这种形式的分布为 C 上的 Gibbs 分布,V 称为能量函数。

定理 1:$X=\{x_{ij},i,j\in\lambda\}$是关于 N 的 MRF 的充要条件,是它的联合分布为 C 上的 Gibbs 分布。这是 MRF 的一条基本定理,它将确定 MRF 的局部特征转化为确定 Gibbs 分布的能量函数 $V(x)$,使得规定一个具体的 MRF 有了实现的可能。

14.2 模拟退火算法

模拟退火(以下简称 SA)算法是基于 Monte Carlo 迭代求解的一种启发式随机搜索算法,它通过模拟退火过程,可找到全局(或边似)最优解,SA 算法的思想最早是由 N.Metropolis 等人于 1953 年提出的,但把它用于组合优化是在 1983 年由 S.kirkpatrick 等人和 V.Cerny 分别提出的。

设 $S=\{S_1,S_2,\cdots,S_n\}$为所有可能的状态构成的集合。$C:S\to R$ 为非负标函数,即 $C(S_i)\geqslant 0$。它表示了取状态 S_i 为解的代价,则组合优化问题可形式地表述为:寻找 $S^*\in S$,使得:

$$C(S^*)=\min\{C(S_i)\},\forall S_i\in S$$

令 $S_{\min}=\{x\in S:C(x)=\min\{C(S_i)\},\forall S_i\in S\}$

$D=\sup\{\mathrm{d}x:S\in S_{L\min}-S_{\min}\}$,其中 $S_{L\min}$表示其解使 $C(x)$达到局部极小值的 x 的全体。

定理:$\lim_{k\to\infty}P\{X(k)\in S_{\min}\mid X(0)=y\}=1,\forall y\Leftrightarrow\sum_{k=1}^{\infty}\exp(-D/T_k)=+\infty$ 也就是说,模拟退火的速度 T_k 应具有形式:

$$T_k=\frac{C}{\lg k}\qquad k\geqslant 2$$

更进一步有:

$$P\{X(k)\in S_{\min}\}\to 1\Leftrightarrow C\geqslant D$$

上述定理说明了 SA 算法可以用于优化。

SA 算法的基本思想是：把 S_i 看成某一物质体系的微观状态，$C(S_i)$看成该物质体系在状态 S_i 下的内能，并用控制参数 T 类比温度，让 T 从一个足够高的值慢慢下降，对每个 T，用 Metropolis 抽样法在计算机上模拟该体系在温度 T 下的热平衡状态。即对当前状态 S 随机扰动产生一个新状态 S'，计算增量 $\Delta C' = C(S') - C(S)$并以概率 $\exp(-\Delta C/kT)$接受 S' 作为新的当前状态。当重复地如此扰动足够次数后，状态 S_i 出现为当前状态的概率将服从玻尔兹曼分布，如下式：

$$f(S_i) = Z(T)\exp(-C(S_i)/kT) \tag{14.1}$$

式中 $Z(T) = \dfrac{1}{\sum_i \exp(-C(S_i)/kT)}$

k 为玻尔兹曼常数。

若 T 下降得足够慢，且 $T \to 0$，由定理可见当前状态将有最小 $C(S_i)$的状态。

可将上述思想写成算法形式，包括 Metropolis 抽样和退火过程实现算法两部分：

(1) 实现退火过程算法

第一步：任选初始状态 S_0，$S(0) = S_0$，并设初始温度 T_0，令 $i = 0$。

第二步：令 $T = T_i$，以 T 和 S_i 调用 Metropolis 抽样算法，返回其最后所得到的当前解作为当前解 $S_i = S$。

第三步：按一定方式将 T 降温。即 $T = T_{i+1}$，$T_{i+1} < T_i$，$i = i + 1$。

第四步：检查退火过程是否结束，是则转第五步，否则转第二步。

第五步：以当前解作为最优结果输出，结束。

(2) Metropolis 抽样算法

第一步：令 $k = 0$ 时的当前解为 $S(0) = S$，在 T 下进行以下各步。

第二步：按某一规定方式，根据当前解 $S(k)$所处状态 S，产生一近邻子集 $N(S(k) \subset S$，$N(S(k)) \neq \Phi$，同 $N(S(k))$随机得到一个新状态 S'作为下一当前解的候选解，且计算：

$$\Delta C' = C(S') - C(S(k))$$

第三步：若 $\Delta C' < 0$，则接受 S'为下一个当前解；若 $\Delta C' > 0$，则按概率 $\exp(-\Delta C'/kT)$接受 S'为下一个当前解，若 S'被接受，则令 $S(k+1) = S'$，否则令 $S(k+1) = S(k)$。

第四步：$k = k + 1$，由某个收敛准则检查算法是否结束，是则转向第五步，否则转向第二步。

第五步：将当前解 $S(k)$返回调用它的退火算法。

在实际应用中，初始温度、内循环次数和算法终止条件是影响 SA 算法性能的重要参数，对它们常用的处理方法如下：

(1) 初始温度选择：

(a) 均匀地随机抽样$\{S_i\}$，取此时 $C(S_i)$的方差为 T_0；

(b) 在所有可能组合状态中，选 两个状态使 $\Delta C'$最大，取 T_0 为最大值的若干倍；

(c) 按经验给出。

(2) 检验 Metropolis 抽样是否稳定：

(a) 检验代价函数的 $C(\cdot)$的均值是否稳定；

(b) 继续若干步 $C(\cdot)$变化很小；

(c) 按一个固定步数抽样。

(3) T 减小的方式：$T \leftarrow \lambda T, 0 < \lambda < 1$，常取 $\lambda \in [0.2, 0.9]$。

(4) 算法终止：

(a) 取 T 小于某阈值；

(b) 检验系统的熵是否已达到最小。

模拟退火法是否能达到 $C(\cdot)$ 的最小值，取决于 T_0 是否足够高和温度 T 下降的充分慢，以及对应于每个 T。Metropolis 抽样能达到稳定，但是这些正好和计算时间相反，因此合适地确定上述参数，对算法的性能有很大的影响。

14.3 MRF 在图像处理中的应用

14.3.1 用于图像恢复的 MRF 模型

上面介绍 D.Geman 和 S.Geman 给出的用于图像恢复的 MRF 模型。设 $S^P = \{(i,j): 1 \leqslant i,j \leqslant N\}$，$S^E$ 是“对偶格点”，S^E 中元素为 S^P 中元素的线段状态值。此 MRF 主要由一对过程组成，其一是密度过程，记作 $X^P = \{X_s^P, S \in S^P\}$；另一是边缘过程，记作 $X^E = \{X_s^E, S \in S^E\}$。通常 $X_s^P \in E = \{0,1,\cdots,L-1\}$，$X_r^E \in \{0,1\}$。$X_r^E = 0$ 表明在 r 不存在边缘，$X_r^E = 1$ 表明在 r 存在边缘。把上述两个过程联合起来，考虑如下过程：$X = (X^P, X^E) = \{(X_s^P, X_s^E), (X_s^P, X_s^E \in \Omega, s \in S\}$ 而且 $\Omega = E^{S^P} \times \{0,1\}^{S^E}$，$S = S^P \cup S^E$，其上的邻域系 $N = \{N_s, s \in S\}$，当 $s \in S^P$ 时，邻域 N_s 为四个紧邻的像素及与之相对偶的四个边。当 $s \in S^E$ 时，N_s 为线段的两个端点及与此两端点相连的 6 个线段。

描述图像的一般数学模型为：$Y_s = \psi\{\varphi(kx)_s, \eta_s\}$。其中 $x = \{x_s, s \in S\}$ 和 $Y = \{y_s, s \in S\}$ 分别表示“真实”的图像及观察到的数据，η_s 为噪声干扰。在给定 Y 的的条件下，我们考虑 X 的概率分布，称之为 X 的后验分布，它表明在观测到图像数据 Y 之后对“真实”图像 x 的重新认识，而得到的修正模型，后验分布 $P(x|y)$ 可以证明是相对于 (S,N) 的吉布斯分布，需要找出 X 的值使 $P(x|y)$ 的值最大，可采用前述的模拟退火算法加以实现。

14.3.2 用于图像噪声处理的 MRF 模型

设图像的大小为 $N \times N$，记 $f(i,j)$ 为像素 (i,j) 的灰度值。$h(i,j) = 1$ 表示像素 (i,j) 下面有一边素，$h(i,j) = 0$ 表示没有。$v(i,j) = 1$ 表示像素 (i,j) 右边有一边素，$v(i,j) = 0$ 表示没有。

$$X_{ij} = (f(i,j), h(i,j), v(i,j)) \qquad i,j \in \lambda$$

由于图像上各像素的灰度值不是相互独立的，各边素之间有一定的概率依赖关系，因此 $X = \{X_{ij}, i,j \in \lambda\}$ 是一个马尔可夫随机场。

当图像观察有噪声时，可以观察到：

$$G = H[\Phi(F)] + n$$

其中 $F = \{f(i,j), i,j \in \lambda\}$、$G = \{g_{ij}, i,j \in \lambda\}$ 分别表示真实图像及实际观测值。从而噪声处理及边缘提取的问题，可以描述为已知 $\{g_{ij}, i,j \in \lambda\}$，求 $X_{ij} = \{(f(i,j), h(i,j), v(i,j),$

$i,j\in\lambda\}$。

这是一个典型的统计决策问题，可采用 Bayes 方法来解决。Bayes 方法的核心是求出后验概率，即

$$P\{X = x \mid G = g\} = P\{f = \hat{f}, h = \hat{h}, v = \hat{v} \mid G = g\} \tag{14.2}$$

记为：$P_{X|G}(x \mid g) = P_{X|G}(\hat{f}, \hat{h}, \hat{v} \mid g)$

$$P_{X|G}(x \mid g) = \frac{P(X = x, G = g)}{P(G = g)} = \frac{P(X = x)P_{G|X}(g \mid x)}{P(G = g)} \tag{14.3}$$

式中

$$\begin{aligned} P_{(G|X}(g \mid x) &= P(G = g, X = x) = P(n = g - \hat{f}) \\ &= \prod_{i,j\in\lambda} P(n_{ij} = g_{ij} - \hat{f}_{ij}) = \exp\{-\sum_{i,j\in\lambda} -\log P(g_{ij} - \hat{f}_{ij})\} \\ &= \exp\{-\sum_{i,j\in\lambda} \hat{V}_0(i,j)\} \\ \hat{V}_0(i,j) &= -\log(Y_{ij} - \hat{P}_{ij}) \end{aligned}$$

由于 $P(x)$ 是吉布斯分布，因此有 $P(x) = \frac{1}{K}\exp(-a\sum_a V_a(x)/T)$

所以 $$P_{X|G}(x \mid g) = \frac{1}{K \cdot P(G = g)} \cdot \exp(-\sum_a \hat{V}_a(x)/T) \tag{14.4}$$

其中

$$\hat{V}_a(x) = \begin{cases} V_0(x) + \hat{V}_0(x)XT & a = 0 \\ V_a(x) & a \neq 0 \end{cases}$$

$P_{(X|G)}(x|g)$作为 X 的分布仍是吉布斯分布，因而仍是 MRF。

D.Geman 和 S.Geman 使用最大后验概率准则来恢复图像，但它对于抑制噪声不利。在噪声处理领域，采用后验均值方法来求 $f(i,j)$、$h(i,j)$、$v(i,j)$。而实际上后验均值与最大后验概率一样无法直接计算，应采用随机模拟方法计算。

14.3.3 用于纹理分析的 MRF 模型

纹理分析指的是通过一定图像处理技术取出纹理特征，从而获得纹理的定量或定性描述的处理过程。为了进行纹理分析，必须首先建立起适当的模型。

设 $S = \{G(i,j), 1 \leqslant i,j \leqslant N\}$ 是平面上的格点，G 表示图像灰度级数，图像纹理 $\{Xs, s\in S\}$ 为 MRF，其局部特征定义如下：

$$P(X_s = k \mid N_i) = \binom{G-1}{k}\theta(T_i)^k(1 - \theta(T_i))^{G-K-1} \tag{14.5}$$

式中 $k = 0,1,\cdots,G-1$；N_i 为 S 上的四种邻域（$i = 1,2,3,4$）

$$\theta(T_i) = \frac{e^{T_i}}{1 + e^{T_i}}$$

而 T_i 的值由不同的模型而定：

对于一阶模型，$T_1 = a + b(1,1)(t + t') + b(1,2)(u + u')$

对于二阶模型，$T_2 = T_1 + b(2,1)(v + v') + b(2,2)(X + X')$

对于三阶模型，$T_3 = T_2 + b(3,1)(m + m') + b(3,2)(l + l')$

对于四阶模型，$T_4 = T_3 + b(4,1)(O_1 + O_1' + O_2 + O_2') + b(4,2)(q_1 + q_1' + q_2 + q_2')$

式中各项为 N_i 邻域中之元素，而系数 $b(i,j)$则反映在不同方向上是否有聚类产生。

利用 MRF 模型进行图像的纹理分析可分为如下几个步骤：

(1) 选定模型的阶数，利用观察到的自然图像的纹理数据，估计模型中的参数 $a, b(i, j)$。

(2) 对参数 $a, b(i,j)$进行假设检验。

(3) 从已综合的纹理模型中抽取样本，从而得到综合纹理模型中的参数。

(4) 这样得到两个纹理图像都可用同样的 MRF 表示，并且有相同的参数，然后根据两个纹理图像进行视觉比较，看其中是否有相同的视觉效果。

15 小波分析及其在图像处理中的应用

●小波变换　●二进小波　●多尺度分析
●纹理分析　●编码压缩　●自适应处理

众所周知，经典的傅立叶分析给信号分析和图像处理开辟了道路。傅立叶分析的核心是将任一函数表示为不同频率的波函数的叠加。有了这个工具，信号和图像处理领域就非常活跃，对各种不同的组成频率进行各种操作。傅立叶变换又将空域的函数与时域的频率挂钩，使得信号与图像处理更向前推进一步，这些方面的应用前面已讨论很多了。然而，傅立叶分析对于非周期函数就显得软弱无力了。特别是对于表示陡然变化的信号，诸如语音和电视信号，傅立叶分析就遇到了困难。人们注意到傅立叶分析的弱点，Gabor 于 1946 年提出了信号的时频局部化的分析方法，就是人们所说的 Gabor 变换，也称为加窗的傅立叶变换。信号 $f(t)$的 Gabor 变换的定义如下：

$$Wf(\omega,\sigma) = \int_R g(t-\sigma)f(t)e^{-i\omega t}dt \tag{15.1}$$

其中，函数 $g(\tau) = \frac{1}{\sqrt{2\pi}}e^{-\frac{\tau^2}{2}}$ 被称为窗函数。变换被限定在窗口中进行，比之在无限大空间的傅立叶变换有所进步，但仍有局限性。因为无论 $g(\tau)$是什么样的窗函数，时窗 $g(t)$的宽度与频窗 $\hat{g}(\omega)$ 的宽度之积不小于 $\frac{1}{\pi}$。所以，当确定某一个窗函数之后，若其频宽对应于某一频段，其时宽则不能太窄。为了提高局部的可观察性，则需要加大窗口，这将导致计算量大增，以致无法具体实现。

80 年代中期逐步发展起来的小波分析，一种函数基很小的波分析方法的产生与发展解决了这个问题。顾名思义，“小波”就是小的波形，在 $\pm\infty$ 处衰减为 0，它之所以也称为波，是由于它的波动性，它的波形具有震荡的形式。小波分析的发展，使信号系统科学更前进了一大步。

小波分析同时具有理论精深与应用广泛的双重意义。它的应用范围涉及到了许多学科，如信号分析、图像处理、量子力学、计算机识别、地震数据处理、曲线曲面构造、微分方程求解、控制论等。在信号分析方面，信号的分解与重构、去噪声、滤波等；图像处理方面如图像增强、图像压缩、边缘检测等。

15.1 连续小波变换及其性质

小波是一个满足条件 $\int_R \psi(t)dt = 0$ 的函数 $\psi(t)$ 通过平移和伸缩而产生的一个函数簇

$\psi_{a,b}(t)$，如下式：

$$\psi_{a,b}(t) = |a|^{\frac{-1}{2}}\psi\left(\frac{t-b}{a}\right), \quad a \in R, \quad b \in R, \quad a \neq 0 \tag{15.2}$$

按上式定义的函数 $\psi(t)$也称基本小波或母小波，函数簇 $\psi_{a,b}(t)$也称为分析小波，(15.2)式中 b 为平稳的距离，a 为伸缩的尺度。

若任一函数 f 作周期延伸，则可成为实直线上有定义的函数，也就是说：

$$\int_0^{2\pi} f^2(x)\mathrm{d}x < \infty$$

那末，每个 f 函数均属于一个平方可积函数 $L^2(0,2\pi)$，也可写为 $L^2(R)$，R 为实数空间。对于任一函数 $f \in L^2(R)$，且基本小波 $\psi \in ^2(R)$，那末 f 的连续小波变换定义为

$$W_f(a,b) = <f,\psi_{a,b}> = |a|^{-\frac{1}{2}}\int_R f(t)\,\overline{\psi\left(\frac{t-b}{a}\right)}\mathrm{d}t \tag{15.3}$$

式中 $\overline{\psi\left(\frac{t-b}{a}\right)}$ 为基本小波的共轭函数，其中 $\psi \in L^2(R)$，且满足如下条件：

$$C_\psi = \int_R \frac{|\hat{\psi}(\omega)|^2}{|\omega|}\mathrm{d}\omega < \infty \tag{15.4}$$

式中 $\hat{\psi}(\omega)$ 为 $\psi(t)$的傅立叶变换，该式条件称为允许性条件，而 $\psi(t)$则称为允许小波。对于由允许小波产生的信号进行小波变换，可以有如下关系式：

设 $\psi(t)$是允许小波，则对一切 $f \in L^2(R)$，$h \in L^2(R)$，$W_f(a,b)$和 $W_h(a,b)$分别表示 f，h 关于同一基本小波 ψ 的小波变换。则有下式：

$$\int_R\int_R W_f(a,b)\,\overline{W_h(a,b)}\,\frac{\mathrm{d}a\mathrm{d}b}{a^2} = C_\psi <f,h> \tag{15.5}$$

式中 $C_\psi = \int_{-\infty}^{\infty} |\omega|^{-1}|\hat{\psi}(\omega)|^2\mathrm{d}\omega$

另外，对任意 $f \in L^2(R)$ 及 $t \in R$，若 $f(t)$ 在 t 处连续，则可由其小波变换得到反演公式，或称反变换，得到原函数为

$$f(t) = C_\psi^{-1}\iint_{R^2} W_f(a,b)\psi_{a,b}(t)\,\frac{\mathrm{d}a\mathrm{d}b}{a^2} \tag{15.6}$$

设 $f \in L^2(R)$，$W_f(a,b)$是 f 的小波变换，则

$$\int_R |f(t)|^2\mathrm{d}t = C_\psi^{-1}\int_R\int_R |W_f(a,b)|^2\,\frac{\mathrm{d}a\mathrm{d}b}{a^2} \tag{15.7}$$

由此可以看出，小波变换是一种信息保持型的可逆变换，原来的信息保存于小波变换系数中，反变换后又可恢复。

小波变换的变换基的选择只要满足(15.4)式，所以在各个学科应用领域中若要采用小波变换，就有一个变换基的选择问题。这点与经典的傅立叶变换不一样，傅立叶变换基是固定的。在具体应用中需要根据原函数 $f(t)$的特点来选择小波变换基 $\Psi(t)$，使得小波变换能更好地反映 $f(t)$的特征。对小波理论发展有重大贡献的科学家们构造了正交小波基库，供选择之用。除此之外，他们又构造了非正交小波基。下面给出一些小波基的例子。

例 1 Haar 正交小波基

$$\Psi(x) = \begin{cases} 1 & 0 \leqslant x < \frac{1}{2} \\ -1 & \frac{1}{2} \leqslant x < 1 \end{cases} \tag{15.8}$$

例 2 Meyer 正交小波基，选一基本小波 Ψ 使其傅立叶变换为

$$\hat{\Psi}(\omega)=\begin{cases}\frac{1}{\sqrt{2}}e^{i\omega/2}\sin[v(3/4\mid\omega\mid-\pi/2] & \frac{2\pi}{3}\leqslant\mid\omega\mid<\frac{4\pi}{3}\\ \frac{1}{2}e^{i\omega/2}\cos[v(3/8\mid\omega\mid-\pi/2] & \frac{4\pi}{3}\leqslant\mid\omega\mid<\frac{8\pi}{3}\\ 0 & \text{其 它}\end{cases}\tag{15.9}$$

例 3 二阶墨西哥草帽小波基

$$\Psi_2(x)=(1-x^2)\frac{1}{\sqrt{2\pi}}e^{-\frac{x^2}{2}}\qquad -\infty<x<+\infty\tag{15.10}$$

三阶墨西哥草帽小波基

$$\Psi_3(x)=(3x-x^3)\frac{1}{\sqrt{2\pi}}e^{\frac{-x^2}{2}}\qquad -\infty<x<+\infty\tag{15.11}$$

15.2 二进小波变换及其性质

在实际应用中，需把上面提到的连续小波及其小波变换离散化才有意义。作为一种方便的形式，计算机实现中往往对小波进行二进制离散，并把经这种离散化后的小波及相应的小波变换，称之为二进小波和二进小波变换。

函数 $\psi\in L^2$ 被称为一个二进小波，若存在两常数 $0<A\leqslant B<+\infty$，使得

$$A\leqslant\sum_{k=-\infty}^{+\infty}\mid\hat{\psi}(2^{-k}\omega)\mid^2\leqslant B\tag{15.12}$$

上式被称为稳定性条件。若 $A=B$，则称为最稳定性条件。函数序列 $\{W_2{}^kf(x)\}_{k\in Z}$ 则叫作函数 f 的二进小波变换，其中

$$W_2{}^kf(x)=f^*\Psi_{2^k}(x)=\frac{1}{2^k}\int_R f(t)\Psi(\frac{x-t}{2^k})dt\tag{15.13}$$

15.3 多尺度分析的概念及其性质

所谓构造小波正交基，就是要寻找一个小波函数 Ψ，使得 $\{2^{\frac{j}{2}}\Psi(2^jx-n)\}_{j,n\in Z}$ 成为 L^2 中的标准正交基。为了有效地寻找空间 L^2 的基底，先从 L^2 的某一个子空间出发，在这个子空间建立基底，然后利用极其简单的变换，再把基底扩展到 L^2 中去，这就是多尺度分析的思想方法。下面给出多尺度分析的一些基本概念。

空间 L^2 中的多尺度分析是指 L^2 中的满足如下条件的一个空间序列 $\{V_j\}_{j\in Z}$，这些条件是：

(1) 一致单调性

$$\cdots V_{-1}\subset V_0\subset V_1\subset\cdots$$

(2) 渐进完全性

$$\bigcap_{j\in Z}V_j=\{0\};\qquad\overline{\bigcup_{j\in Z}V_j}=L^2(R)$$

(3) 伸缩规则性

$$f(x)\in V_j \Leftrightarrow f(2x)\in V_{j+1}, \qquad \forall j\in Z$$

(4) Riesz 基存在性

存在 $\psi\in V_0$，使得 $\{\psi(x-n)\}_{n\in Z}$ 是 V_0 的无条件基。若 $\{\psi(x-n)\}_{n\in Z}$ 是 V_0 的一组无条件基，则 $\{2^j\Phi(2^jx-n)\}_{n\in Z}$ 是 V^j 的一组无条件基。

为得到标准化正交基，可以对 $\{\Phi_{0n}\}$ 施行正交化手续。下面的定理提供了检验一组基是否为正交化的方法。

定理 1 设 $\Phi\in L^2$，则下列命题等价：

① $\{\Phi(x-n)\}_{n\in Z}$ 是标准正交系

② $\sum_{k\in Z}|\hat{\Phi}(\omega+2k\pi)|^2=1$，其中 $\hat{\Phi}$ 为 Φ 的傅立叶变换

③ $\frac{1}{2\pi}\int_R|\hat{\Phi}(\omega)|^2e^{ik\omega}d\omega=\delta_{0,k}$

定理 2 存在函数 $\tilde{\Phi}(x)\in V_0$，使得 $\{\tilde{\Phi}(x-k)\}_{k\in Z}$ 构成 V_0 的标准正交基。

定理 3 记 $H(\omega)=\frac{1}{2}\sum_{k\in Z}h_ne^{ik\omega}$，则有

$$|H(\omega)|^2+|H(\omega+\pi)|^2=1$$

总起来说，若已知一个多尺度分析，就有一个多尺度分析的生成元 $\Phi\in L^2$，通过正交化手续，可以使 Φ 的正交化 $\tilde{\Phi}$ 产生 V_0 乃至 V_j 的正交基 $\{\tilde{\Phi}_{0n}\}$ 及 $\{\tilde{\Phi}_{jn}\}$。这样，就得到了一种特殊的渐进空间序列 $\{V_n\}_{n\in Z}$ 及每个子空间 V_j 上的正交基，利用空间正交分解，就可得到 $L^2(R)$ 中的小波正交基了。

15.4 二维连续小波变换

为了使小波变换能应用于图像处理，必须将一维情形推广到二维，甚至三维的情形。鉴于二维与三维无实质性区别，现只考虑二维情形，如下式：

$$\left.\begin{aligned}\Psi_{2^j}^1(x,y)&=\frac{1}{2_{2^j}}\Psi^1\left(\frac{x}{2^j},\frac{y}{2^j}\right)\\ \Psi_{2^j}^2(x,y)&=\frac{1}{2^{2^j}}\Psi^2\left(\frac{x}{2^j},\frac{y}{2^j}\right)\end{aligned}\right\} \tag{15.14}$$

则在尺度 2^j 下，运用多尺度扩展的原理，函数 $f(x,y)\in L^2(R^2)$ 的小波变换如下式：

$$\left.\begin{aligned}W_{2^j}^1f(x,y)&=f(x,y)*\Psi_{2^j}^1(x,y)\\ W_{2^j}^2f(x,y)&=f(x,y)*\Psi_{2^j}^2(x,y)\end{aligned}\right\} \tag{15.15}$$

从(15.15)式得出 $Wf=\{W_{2^j}^1f(x,y),W_{2^j}^2f(x,y)\}$，它称为二维小波变换。这种二维小波变换由于具有空间多向滤波性，在图像处理方面有着广泛的应用。利用小波变换可进行图像的边缘检测、纹理分析以及图像增强等。现举例来说明二维小波变换。

设 $L^2(R^2)$ 空间中，取小波基如下：

$$\Psi^1(x,y)=\frac{-x}{\sqrt{2\pi}}e^{\frac{-(x^2+y^2)}{2}}$$

$$\Psi^2(x,y)=\frac{-y}{\sqrt{2\pi}}e^{\frac{-(x^2+y^2)}{2}}$$

经推算，其傅立叶变换为

$$\hat{\Psi}^1(\omega_1,\omega_2)=i\sqrt{2}\omega_1 e^{\frac{-(\omega_1^2+\omega_2^2)}{2}}$$

$$\hat{\Psi}^2(\omega_1,\omega_2)=i\sqrt{2}\omega_2 e^{\frac{-(\omega_1^2+\omega_2^2)}{2}}$$

并且，可见

$$\int_0^{\infty}\left[|\hat{\Psi}^1(s\omega^1+s\omega^2)|^2+|\hat{\Psi}^2(s\omega_1,s\omega_2)|^2\right]\frac{\mathrm{d}s}{s}=1,$$

所以，$\{\Psi^1(x,y),\Psi^2(x,y)\}$是 $L^2(R^2)$空间的规范化基本小波，并且 Ψ^1、Ψ^2 分别可视为沿垂直方向与水平方向的滤波器。这样定义的二维小波变换可作为二维图像边缘检测算子，检测到的边缘分别是图像边缘的垂直分量和水平分量。

15.5 图像的小波分解编码

生成二维小波基可利用两个一维小波基做张量积。设$\{V_n\}_{n\in Z}$是一个已知的 $L^2(R)$的多尺度分析，Φ 是它的标准正交生成元。对于 $L^2(R)$，其张量积 $V_j\otimes V_j$ 是 $L^2(R^2)$的子空间，且有

$$\cdots\subset V_{-1}\otimes V_{-1}\subset V_0\otimes V_0\subset V_1\otimes V_1\subset\cdots$$

记 $\tilde{V}_j=V_j\otimes V_j$，则有

$$\overline{\bigcup_{j\in Z}\tilde{V}_j}=L^2(R^2),\quad \text{且}\bigcap_{j\in Z}\tilde{V}_j=\{0\}$$

由于$\{\Phi_{j,k}(x)=2^{j/2}\Phi(2^jx-k)\}_{k\in Z}$是 V_j 的标准正交基，因而$\{\Phi_{j,k}(x)\Phi_{j,l}(x)\}_{l,k\in Z}$构成 $\tilde{V}_j$ 的标准正交基。因此可得到一个二元的多尺度分析，其生成元为 $\Phi(x,y)=\Phi(x)\Phi(y)$。下面来构造 $L^2(R^2)$ 的小波子空间的正交分解。

设 $L^2(R)$ 的小波子空间的正交分解如下：

$$L^2(R)=\bigoplus_{j\in Z}W_j,\quad \text{其中 } W_j=\overline{span_{L^2}\{\Psi_{j,n}:n\in Z\}}$$

由于 $\forall j,l\in Z$，

$$W_j\perp V_j,\ W_j\perp W_l,\quad \text{且 } V_j\oplus W_j=V_{j+1}$$

因此，
$$\begin{aligned}\tilde{V}_{j+1}&=(V_{j+1}\otimes W_{j+1})=(V_j\oplus W_j)\otimes(V_j\oplus W_j)\\&=(V_j\otimes V_j)\oplus(V_j\otimes W_j)\oplus(W_j\otimes V_j)\oplus(W_j\otimes W_j)\end{aligned}$$

然而，对 $\forall f\in L^2(R^2)$ 有

$$\begin{aligned}f(x,y)=&\sum_{j,k,l\in Z}d_{j,k,l}^1\Phi_{j,k}(x)\Psi_{j,l}(y)+\sum_{j,k,l\in Z}d_{j,k,l}^2\Psi_{j,k}(x)\Phi_{j,l}(y)\\&+\sum_{j,k,l\in Z}d_{j,k,l}^3\Psi_{j,k}(x)\Psi_{j,l}(y)\end{aligned}\tag{15.16}$$

式中

$$\left.\begin{aligned} d_{j,k,l}^{1} &= \iint_{R^2} f(x,y)\Phi_{j,k}(x)\Psi_{j,l}(y)\mathrm{d}x\mathrm{d}y \\ d_{j,k,l}^{2} &= \iint_{R^2} f(x,y)\Psi_{j,k}(x)\Phi_{j,l}(y)\mathrm{d}x\mathrm{d}y \\ d_{j,k,l}^{3} &= \iint_{R^2} f(x,y)\Psi_{j,k}(x)\Psi_{j,l}(y)\mathrm{d}x\mathrm{d}y \end{aligned}\right\} \tag{15.17}$$

下面给出上式的离散算法，记 $L^2(R^2)$到 $\bar{V}_j$ 的正交投影算子为 P_m，由于当 m 充分大时，$P_m f \approx f$，设 $m=0$，则有

$$\bar{V}_0 = \bigoplus_{j=-1}^{-N} \bar{W}_j^1 \bigoplus_{j=-1}^{-N} W_j^2 \bigoplus_{j=-1}^{-N} W_j^3 \oplus \bar{V}_{-N}$$

由于 $f \in \bar{V}_0$，$g_{-j}^i \in \bar{W}_{-j}^i$，$i=1,2,3$，$f_{-j} \in \bar{V}_{-j}$，记

$$\left.\begin{aligned} f_{-j}(x,y) &= \sum_{k,l\in Z} C_j(k,l)\Phi_{j,k}(x)\Phi_{j,l}(y) \\ g_{-j}^1(x,y) &= \sum_{k,l\in Z} d_j^1(k,l)\Psi_{j,k}(x)\Phi_{j,l}(y) \\ g_{-j}^2(x,y) &= \sum_{k,l\in Z} d_j^2(k,l)\Phi_{j,k}(x)\Psi_{j,l}(y) \\ g_{-j}^3(x,y) &= \sum_{k,l\in Z} d_j^3(k,l)\Psi_{j,k}(x)\Psi_{j,l}(y) \end{aligned}\right\} \tag{15.18}$$

于是类似于一维的情形，则有如下的分解公式：

$$\left.\begin{aligned} C_j(n,m) &= \frac{1}{2}\sum_{k,l\in Z} C_{j-1}(k,l)h_{k-2n}h_{l-2m} \\ d_j^1(n,m) &= \frac{1}{2}\sum_{k,l\in Z} C_{j-1}(k,l)g_{k-2n}h_{l-2m} \\ d_j^2(n,m) &= \frac{1}{2}\sum_{k,l\in Z} C_{j-1}(k,l)h_{k-2n}g_{l-2m} \\ d_j^3(n,m) &= \frac{1}{2}\sum_{k,l\in Z} C_{j-1}(k,l)g_{k-2n}g_{l-2m} \end{aligned}\right\} \tag{15.19}$$

类似的重构算法为

$$\begin{aligned} C_{k-1}(n,m) = \frac{1}{2}\{&\sum_{j,l\in Z} C_k(j,l)h_{n-2j}h_{m-2l} + \sum_{j,l\in Z} d_k^1(j,l)h_{n-2j}g_{m-2l}\} \\ &+ \sum_{j,l\in Z} d_k^2(j,l)g_{n-2j}h_{m-2l} + \sum_{j,l\in Z} d_k^3(j,l)g_{n-2j}g_{m-2l}\} \end{aligned} \tag{15.20}$$

对 $\forall A = |a(m,n)|_{m,n\in Z} \in l^2$，则定义如下算子：

$$\left.\begin{aligned} (H_C A)(m,n) &= \frac{1}{\sqrt{2}}\sum_{k\in Z} a(k,n)h_{k-2m} \\ (G_C A)(m,n) &= \frac{1}{\sqrt{2}}\sum_{k\in Z} a(k,n)g_{k-2m} \\ (H_r A)(m,n) &= \frac{1}{\sqrt{2}}\sum_{k\in Z} a(k,n)h_{k-2n} \\ (G_r A)(m,n) &= \frac{1}{\sqrt{2}}\sum_{k\in Z} a(k,n)g_{l-2n} \end{aligned}\right\} \tag{15.21}$$

则

$$\left.\begin{aligned} C_j &= H_c H_r C_{j-1} \\ d_j^1 &= G_c H_r C_{j-1} \\ d_j^2 &= H_c G_r C_{j-1} \\ d_j^3 &= G_c G_r C_{j-1} \end{aligned}\right\} \tag{15.22}$$

(15.22) 式中，G、H 为两个一维滤波算子，下标对应于图像的行和列。如果把二维信号 $f(x,y)$ 视为一幅图像，则算子 H_cH_r 相当于二维低通滤波器。因此，C_j 是 C_{j-1} 的低频成份，称为 C_0 的 j 水平模糊图像。算子 G_cH_r 相当于先对列作平滑，然后检测行的差异；在有水平边缘的地方，d_j^1 的幅度应当比较大，因此 d_j^1 显示 d_{j-1}^1 的竖直方向上高频成份，相当于图像的水平边缘；同样，d_j^2 是水平方向的高频成份，相当于图像的竖直边缘；d_j^3 是两个方向的高频成份，即角点和 45° 边缘。经过小波变换分解，C_j 为图像的亮度分量，d_j^1 和 d_j^2 分别为图像边缘的水平分量和垂直分量，而 d_j^3 为图像边缘在对角方向的分量。

15.6 图像纹理的小波分析

纹理是图像的一种非常重要的特征，在自然景物图像、遥感图像和医学图像等许多种类的图像处理中得到了应用。在图像边缘检测和区域分割的分析中，是把灰度和颜色的一致性作为区域的一致性来考虑的。初看上去物体表面的纹理可认为是灰度和颜色的二维变化的图案。

小波分解是提取纹理特征的有效方法，由于它反映了在水平、垂直和角上的高频变化，纹理的差别可在一定程度上得到体现。Mallat 提供了一种利用两个一维卷积实现两维图像小波分解的简便快速算法，并且在分解后的图像上仅利用简单的统计方法便可较好的提取纹理特征。将一幅图像 $f(x,y)$ 看作为二维信号，根据前述的二维小波分解算法，得到

$$\left.\begin{aligned} C_j &= H_c H_r C_{j-1} \\ d_j^1 &= G_c H_r C_{j-1} \\ d_j^2 &= H_c G_r C_{j-1} \\ d_j^3 &= G_c G_r C_{j-1} \end{aligned}\right\} \tag{15.23}$$

式中 j 为图像的分辨率为 2^j，H 和 G 为两个一维滤波算子，下标 c 和 r 对应于图像的列和行。且有

$$\left.\begin{aligned} (H_a)_k &= \sum_n h(n-2k)a_n \\ (G_a)_k &= \sum_n g(n-2k)a_n \end{aligned}\right\} \tag{15.24}$$

(15.24) 式中的 h 和 g 分别为

$$\left.\begin{aligned} h(n) &= \frac{1}{\sqrt{2}}\int \Phi\left(\frac{1}{2}x\right)\Phi(x-n)\mathrm{d}x \\ g(n) &= \frac{1}{\sqrt{2}}\int \Psi\left(\frac{1}{2}x\right)\Phi(x-n)\mathrm{d}x \end{aligned}\right\} \tag{15.25}$$

(15.25)式中，Φ 称为比例函数，$\Psi(x)$就是小波。二维小波变换可以通过两个独立的一维变换来实现。$\Phi(x)$和 $\Psi(x)$分别看作为低通和带通滤波器。小波变换可解释为在一

个独立空间方向上的频率通道。当原始图像分解为 C_j, d_j^1, d_j^2 和 d_j^3 时，C_j 是原始图像在 2^j 分辨率上的近似，d_j^1 对应于垂直方向上的高频变化，即水平的边缘信息；d_j^2 对应于水平方向的高频变化，即垂直的边缘信息；而 d_j^3 则给出垂直和水平两个方向上，即对角方向上的高频变化信息。由于在不同方向上的频率变化被反映，也就反映了图像纹理特征。在取得了纹理特征之后，利用统计方法或神经网格的方法，可以实现对纹理图像的分类与分割。

采用小波分解的方法对于许多纹理图像都能得到较好的分类结果。但是特征值的构造是很重要的。目前大多数方法主要考虑了纹理在水平、垂直和角上的高频变化，在某些情况下，当两种纹理上述所有特征都非常相似，且图像中纹理种类较多，或各纹理区域面积相差较大的情况下，用这类特征，分割结果就会不太理想。因而除了选取更好的变换基外，还可在变换后的图像中加入其它特征，有可能在一定程度上得到改善，

15.7 图像小波分析中的自适应处理

图像的边缘是图像的一个重要特征。它往往意味着图像灰度的迅速变化，因而这些边缘点带有很多图像信息。边缘检测的实现往往是用到一些多尺度算子，这里给出一种多尺度小波变换算法，它的尺度具有随着图像平面坐标的不同而自调整，具有自适应的特点。这种算法中运用一种能量函数并使之最小化来得到最优化尺度，多尺度小波变换滤波器由能量函数导出，而小波变换模的最大值点则构成了图像的边缘。

对于一幅图像，任何类型的信息结构均以像素灰度的不连续性表现现来。而计算机视觉的目的正是从一幅或多幅图像中获取信息。图像灰度的不连续性描述了目标的外沿轮廓，这对识别目标是很重要的。人们做了很多努力来寻找检测此种不连续性的方法，即便是最简单的图像也会包含一些细节信息。灰度的不连续性可由许多原因构成：如不同的表面纹理、照度，表面的方向、深度等，均有许多不同的局部强度。在实践中，数据有可能被噪声所污染，从而导致这种不连续性会不明显。现在的问题是如何检测灰度不连续的存在和决定它们的位置。

在许多应用中，仅有单尺度平滑是不够的。多尺度边缘检测算子因具有多样的空间尺度而得到应用。多尺度边缘检测算子用不同的尺度来平滑图像，利用它们的一阶和二阶导数来检测图像的尖锐变化点。一阶导数的极值点即意味着二阶导数的零交叉点和平滑图像的拐角点，其中的尺度定义了计算信号变化的邻域的大小。

多尺度边缘检测技术中要涉及的两个重要问题是：

(a) 对一幅图像，怎样选择算子的尺度？

(b) 怎样从不同的尺度来有效地恢复边缘信息？

这里提出的算法中得到的小波变换的尺度随着图像的局部区域特征的不同能够自适应地调整，并在这些尺度上检测图像的边缘。

首先看一看 Mallat 提供的二维图像的小波变换的算法和特性。这里先定义两个小波分别为一个二维连续函数 $\theta(x,y)$ 沿 x 和 y 方向的偏导数

$$\Psi^1(x,y) = \frac{\partial\theta(x,y)}{\partial x}, \quad \Psi^2(x,y) = \frac{\partial\theta(x,y)}{\partial y} \tag{15.26}$$

令 $\Psi^1(x,y) = (\frac{1}{s})^2\Psi^1(\frac{x}{s},\frac{y}{s})$ 和 $\Psi^2(x,y) = (\frac{1}{s})^2\Psi^1(\frac{x}{s},\frac{y}{s})$，则对任意两数 $f(x,y) \in$

$L^2(R)$，由两个小波 $\Psi^1(x,y)$ 和 $\Psi^2(x,y)$ 定义的小波变换具有两个分量：

$$\begin{aligned} W^1 f(x,y) &= f * \Psi^1(x,y) \\ W^2 f(x,y) &= f * \Psi^2(x,y) \end{aligned} \tag{15.27}$$

可以证明

$$\begin{bmatrix} W^1 f(s,x,y) \\ W^2 f(s,x,y) \end{bmatrix} = s \begin{bmatrix} f * \Psi_2^1(x,y) \\ f * \Psi_2^2(x,y) \end{bmatrix} = s \vec{\nabla} (f * \theta)(x,y) \tag{15.28}$$

由此可以看出，小波变换的两个分量与 $f(x,y)$（被 $\theta(x,y)$ 所平滑）的梯度向量的坐标成正比。在 $f(x,y)$ 尺度 s 的边缘点就是上面这一梯度向量模的最大值点，而这一梯度向量的方向正是 $f(x,y)$ 的偏导数变化最迅速的方向。因此边缘点实际上是曲面 $(f * \theta)(x,y)$ 的拐点。对于二维情形，尺度空间实际上是一个三维空间 (s,x,y)。由于受到计算复杂性和计算内存的限制，要尽可能地保留较少的尺度。基于上面的考虑，可以定义二维二进小波变换，其中的尺度沿着二进序列 $\{2^j\}_{j\in Z}$ 遍历。如下的函数集被称为 $f(x,y)$ 的二进小波变换：

$$\{W^1 f(2^j,x,y), W^2 f(2^j,x,y)\}_{j\in Z} \tag{15.29}$$

令 $\hat{\Psi}^1(\omega_x,\omega_y)$ 和 $\hat{\Psi}^2(\omega_x,\omega_y)$ 分别为 $\Psi^1(x,y)$ 和 $\Psi^2(x,y)$ 的傅立叶变换，于是就有

$$\begin{aligned} \hat{W}^1 f(2^j,\omega_x,\omega_y) &= \hat{f}(\omega_x,\omega_y)\hat{\Psi}^1(2^j\omega_x,2^j\omega_y) \\ \hat{W}^2 f(2^j,\omega_x,\omega_y) &= \hat{f}(\omega_x,\omega_y)\hat{\Psi}^2(2^j\omega_x,2^j\omega_y) \end{aligned} \tag{15.30}$$

一个二进小波变换是完全的和稳定的，当且仅当二维傅立叶平面被 $\hat{\Psi}^1(\omega_x,\omega_y)$ 和 $\hat{\Psi}^2(\omega_x,\omega_y)$ 的二进膨胀所覆盖。这即是说，存在两个严格为正的常数 A 和 B，以使得

$$A \leqslant \sum_{j=-\infty}^{\infty} (|\hat{\Psi}^1(2^j\omega_x,2^j\omega_y)|^2 + |\hat{\Psi}^2(2^j\omega_x,2^j\omega_y)|^2 \leqslant B \tag{15.31}$$

小波变换在尺度 2^j 的模和幅角分别为

$$\left.\begin{aligned} Mf(2^j,x,y) &= \sqrt{|W^1 f(2^j,x,y)|^2 + |W^2 f(2^j,x,y)|^2} \\ Af(2^j,x,y) &= \operatorname{arctg}[W^2 f(2^j,x,y)/W^1 f(2^j,x,y)] \end{aligned}\right\} \tag{15.32}$$

总的来说，小波变换的模 $Mf(2^j,x,y)$ 正比于梯度向量 $\vec{\nabla} f(*\theta)(x,y)$ 的模，而小波变换的幅角 $Af(2^j,x,y)$ 是梯度向量 $\vec{\nabla}(f * \theta)(x,y)$ 与水平方向的夹角，它正是图像边缘的方向。所以，如要检测边缘只需寻找梯度向量 $\vec{\nabla}(f * \theta)(x,y)$ 的模的局部最大值点。在每一个尺度 2^j，小波变换的模的最大值都定义为模 $Mf(2^j,x,y)$ 在沿着梯度方向 $(f * \theta)(x,y)$ 的局部最大值点。需要注意的是，在小尺度上，许多模最大值往往是由图像噪声和细纹理所造成的；而在大尺度上，由图像噪声和纹理产生的模最大值已经消失了，但也没避免牺牲边缘的精确位置。为了实现二进小波变换计算的数字化，介绍如下一种方法。

两个小波分别是

$$\left.\begin{aligned} \Psi^1(x,y) &= \Psi(x)\xi(y) \\ \Psi^2(x,y) &= \Psi(y)\xi(x) \end{aligned}\right\} \tag{15.33}$$

令 $\Psi(x)$ 是一个一维小波，它的傅立叶变换等于

$$\left.\begin{aligned}\hat{\Psi}(\omega) &= i\omega\left[\frac{\sin(\frac{\omega}{4})}{\frac{\omega}{4}}\right]^{2n+2}\\ \hat{\Psi}^1(2\omega) &= e^{-i\omega}G(\omega)\hat{\Phi}_0(\omega)\\ \hat{\Phi}_0(\omega) &= e^{-i\omega}\prod_{p=1}^{\infty}H(2^{-p}\omega)\end{aligned}\right\} \tag{15.34}$$

式中

$$H(\omega) = e^{i\omega/2}[\cos\frac{\omega}{2}]^{2n+1}$$

$$G(\omega) = 4ie^{i\omega/2}\sin\frac{\omega}{2}$$

于是平滑函数 $\theta(x,y)$ 可以写为

$$\theta(x,y) = \hat{\Phi}_0(x)\hat{\Phi}_0(y) \tag{15.35}$$

二进小波边缘检测是一种多尺度边缘检测方法，下面来看看小波边缘检测中的最优尺度决定方法。这里需要解决一个问题，那就是怎样将不同尺度的输出集成为一个简单边缘图像。这里给出一种替代方法。它的滤波器的变量范围可以利用图像的局部光度特征进行自调整。具体为

$$E(s) = \iint_{R^2}|(f-\theta*f)^2 - \alpha|\nabla\frac{1}{s(x,y)}|^2|dxdy \tag{15.36}$$

式中 s 为图像区域。光度能量函数 $E(s)$包含两项，显然第一项当 $s\to 0$ 时变得极小，当 $s\to\infty$时变得极大；第二项的情形刚好相反。因此可以肯定 $E(s)$在开放空间 $0<s<\infty$ 具有极小值。若 s 使 $E(s)$为极小化能量函数 $E(s)$，它必须满足如下条件：

1) 在图像的缓变区域 s 中 $E(s)$为极大，这样可以平滑随机噪声；

2) 在图像的显著变化区域 s 中 $E(s)$为极小，这样可以精确定位边缘的位置；

3) 在像素与像素之间，s 不能发生跳变以阻止间断边缘的产生。

前两个特性对应于能量函数的第一项，而第 3)项特性对应于能量函数的第二项。对于二进小波变换，由于尺度 s 只能得到离散值，所以也仅能得到能量函数在某些离散点上的较小值，而不是全局极小值。因而可以导出利用离散尺度的小波滤波器如下：

$$G(\omega) = 4ie^{\frac{i\omega}{2}}\sin\frac{\omega}{2} = 2(e^{i\omega}-1)$$

$$h(n) = 2[\delta(n+1)-\delta(n)]$$

$$H(\omega) = e^{i\omega}[e^{\frac{i\omega}{2}}+e^{-\frac{i\omega}{2}}]^3 = \frac{1}{8}[e^{i2\omega}+3e^{i\omega}+3e^{-i\omega}]$$

$$h = \frac{1}{8}[\delta(t-1)+3\delta(t)+3\delta(t+1)+\delta(t+2)]$$

……

综合以上的讨论，可以总结如下算法：

1) 对每一个像素

①计算 $E(2^j)$，找到使能量函数 $E(2^j)$最小的尺度 j；

②用图像 $S_2{}^jf(n,m)$和滤波器 $h_j(n)$作卷积。

2) 用第 1)步的结果和 $g_j(n)$作卷积。

3) 检测二进小波变换模的局部极大值，在该尺度 j 下构成图像边缘。

将上述算法应用于法国 SPOT 卫星遥感图像的边缘抽取，获得了较好的效果。

15.8 小波变换在图像编码压缩中的应用

80年代中期出现的小波变换是一种新的时频信号分析工具，受到了广泛的重视并被应用于图像编码研究。借助于小波变换，可以把图像信号分解成许多具有不同空间分辨率、频率特性和方向特性的子带信号，实现低频长时特征和高频短时特征的同时处理，有效地克服了傅立叶分析在处理复杂图像时所存在的局限性，使得图像信号的分解更适合于人的视觉特性和特征数据压缩的要求，因此近年来利用小波变换进行图像数据压缩的研究成为一个热点。通过对于小波基的合理选择，可使小波变换具有正交或双正交特性。前述已知，利用离散小波变换可将二维图像 $f(x,y)$ 分解为亮度分量、水平边缘、垂直边缘和对角边缘分量，它对应于 C_j, d_j^1, d_j^2 和 d_j^3。对于所得到的4个子图分量，结合人的视觉生理和心理特点分别作不同策略的量化和编码处理。比如，对于亮度分量子图，采取余弦变换，并结合 Huffman 编码方法进行压缩。对于三个方向的边缘分量，采取去掉高频成份，阈值量化和均匀量化，结合 Huffman 编码进行压缩。压缩了的图像编码，经过信道传输到接收端之后，通过两个步骤还原图像。第一步解码，对于不同的编码采取不同的解码方法。第二步是小波逆变换，还原成原来的图像。这方法的特点是图像的分解和方向滤波，正交小波变换则正好能提供图像的多尺度分解表示，从而为常规的压缩方法提供了理想的变换域。

任何正交小波基对应的滤波 $\{h,g\}$ 都能实现图像的分解，然而并非任何分解均能满足要求。小波基的选择要满足分解后，亮度与图像比原来的图像具有更强的相关性，而三个边缘子图像要有更高的局部相关性，整体相关性被大部分甚至被完全解除。Daubechies 正交小波基正满足这些条件，可作为理想变换基的选择。

因此可以看出小波编码是一种很有潜力的图像编码方法，这种方法被广泛的使用在静止图像编码中。近年来小波分解编码研究主要集中在以下几个方面：

(1) 研究小波基的构造方法，寻求适合于信源特性的小波基；

(2) 建立变换数据的最佳表示结构；

(3) 小波分解与已有的编码技术的有机结合；

(4) 用小波分析来开发和利用图像的几何特征等。

虽然小波分析在图像编码中的应用是初步的，但研究结果已经显示出了它的优越性。

742293

参考文献

[1] Rafad C Gonzalez, Paul Wintz. Digital Image Processing(Second Edition). Addison – Weslez Publishing Company, 1987.

[2] Kenneth R Castliman. Digital Image Processing. Prentice Hall, 1995.

[3] Ajriel Rosenfeld, Avinash C Kak. Digital Picture Processing. Academic Press, 1982.

[4] Jan Teuber. Digital Image Processing. Prentice Hall, 1992.

[5] R.C. 冈萨雷斯, P. 温茨著; 李叔梁等译. 数字图像处理. 北京: 科学出版社, 1983.

[6] 田捷, 沙飞, 张新生编著. 实用图像分析与处理技术. 电子工业出版社, 1995.

[7] 王新成编著. 多媒体实用技术(图像分册). 成都: 电子科技大学出版社, 1996.

[8] 荆仁杰等编. 计算机图像处理. 杭州: 浙江大学出版社, 1990.

[9] 吴健康编著. 图像处理与分析. 北京: 人民邮电出版社, 1992.

[10] 李介谷, 施鹏飞, 刘重庆, 谢式绚编著. 图像处理技术. 上海: 上海交通大学出版社, 1988.

[11] 徐建华编著. 图像处理与分析. 北京: 科学出版社, 1992.

[12] 傅京孙主编. 程民德, 石青云, 戴汝为, 边肇祺译. 模式识别应用. 北京: 北京大学出版社, 1990.

[13] 罗耀光, 盛立东编著. 模式识别. 北京: 人民邮电出版社, 1989.

[14] 边肇祺等编著. 模式识别. 北京: 清华大学出版社, 1988.

[15] 郭德方编著. 遥感图像的计算机处理和模式识别. 北京: 电子工业出版社, 1987.

[16] 许殿元, 丁树柏编著. 遥感图像信息处理. 北京: 宇航出版社, 1990.

[17] 濮静娟主编. 遥感图像目视解译原理与方法. 北京: 中国科学技术出版社, 1992.

[18] 华瑞林编著. 遥感制图. 南京: 南京大学出版社, 1990.

[19] Daivd C Kay, John R Levine 著; 柏东, 闫世尊译. 20 种位图矢量图文件格式与实践. 北京: 学苑出版社, 1994.

[20] 董士海, 郑全战, 徐曦, 余晓萍编著. 图像格式编程指南. 北京: 清华大学出版社, 1995.

[21] 夏德深, 郑阿奇, 谈惠峰编著. PC 机系统大全. 南京: 东南大学出版社, 1995.

[22] 秦前清, 杨宇凯编著. 实用小波分析. 西安: 西安电子科技大学出版社, 1994.

[23] 崔锦泰著, 程正兴译. 小波分析导论. 西安: 西安交通大学出版社, 1995.

0000001090516