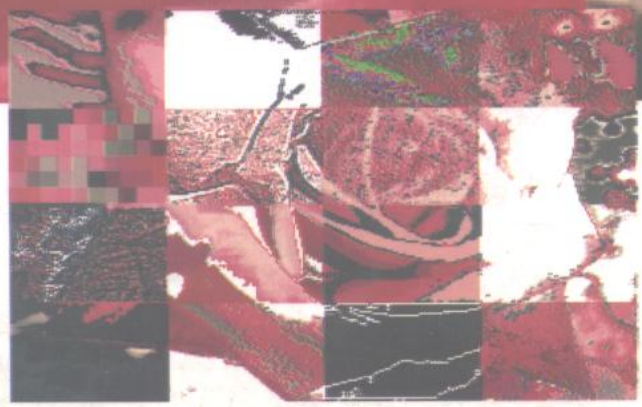
实用图象分析与处理技术

田捷 沙飞 张新生 编著



電子工業出版社

实用图象分析与处理技术

田 捷 沙 飞 张新生 编著

电子工业出版社

(京)新登字 055 号

内 容 提 要

本书全面深入地描述了图象分析与处理的基本理论与方法，重点介绍了相应的算法实现方法与实用技术。主要内容包括：图象的采集与量化，图象变换、增强和恢复，图象的边缘提取和图象分割、图象数据压缩，小波分析与图象处理，图象分析和马尔科随机场及其在图象处理中的应用等。

涉及本书算法的全部源程序，以软件形式由出版社同时出版发行。

本书适合图象处理与分析、模式识别和计算机应用的科技人员以及大专院校有关专业的师生阅读。

图书在版编目(CIP)数据

实用图象分析与处理技术/田捷等编著
——北京：电子工业出版社，1994.12
ISBN 7-5053-2566-3

Ⅰ.实…
Ⅱ.田…
Ⅲ.①图象分析－计算机应用②图象处理－计算机应用
Ⅳ.TP391.4

电子工业出版社出版
北京市海淀区万寿路 173 信箱(100036)
电子工业出版社发行　各地新华书店经销
北京天利电子出版技术公司排版
北京市顺新印刷厂印刷
*
开本：787×1092 毫米 1/16　印张：20.75　字数：500 千字
1995 年 1 月第一版　1995 年 1 月北京第一次印刷
印数：5,000 册　　定价：22.00 元
ISBN 7-5053-2566-3/TP·765

前 言

图象的处理和分析是一门迅速发展的学科。它的主要目的是在机器上实现生物特别是人类所具有的视觉信息处理和加工功能。因而无论对科学理论研究还是工程应用，图象处理和分析都具有重要的影响。一方面，在我们最终能在机器上实现各种视觉信息功能之前还需要很多理论和技术上的突破，从某种意义上来说，研究图象处理与分析也是导向智能计算机、智能机器人和其它智能系统的一条必由之路；另一方面，成熟的图象处理和分析技术已在很多方面得到了充分的应用。例如，用于处理来自地球深层的图象信息、勘探地质资源、处理和分析卫星或遥感照片、工业自动检测等等，而且其应用领域也在不断扩大。进入信息时代以来，图象通讯、办公自动化系统、地理信息系统、多媒体信息系统等得到了极大的发展。这类活动的基础是基于视觉信息的思维活动，因而同样依赖于图象处理和分析技术。

早期的图象处理和分析技术是基于专用的图象处理设备而发展起来的。在科学技术高度发展的今天，一方面是笨重的设备向小型化、微型化发展；另一方面则是随着微处理器芯片成本的下降、低价高分辨图形适配器的发展及其它数字化硬件如扫描仪的普及，往往可在微机上建立起功能强大的图象处理和分析系统，本书正是鉴于这一发展形势而编写的。

本书的主要特点是在介绍图象处理与分析理论方法的同时着重描述算法实现步骤与技巧，并给出大量关键的例程，以便读者从理论方法和实现技巧两个方面掌握本书的内容。与本书同时出版发行了含有图象输入、图象显示、图象格式转换、图象打印、各种图象处理方法和图象压缩（按 JBIG、JPEG、MPEG 标准实现）等源程序——实用图象分析与处理技术（软件），感兴趣的读者可直接与电子工业出版社联系。

本书的程序都经过调试。为了提高速度，我们没有采用任何一种 C 语言编译器所提供的图象处理和显示函数，只利用了一些标准的库函数。对微机图象处理系统来说，一个很大的困难是 DOS 操作系统对内存访问的限制。这使得对大图象的处理变得繁琐和困难，通常需要硬盘作为中介，这就大大降低了速度。本书提供的实现技巧中包括一组利用扩展内存或扩充内存进行图象数据存取的子程序。

本书的部分内容与作者承担的 863 高技术计划课题、国家自然科学基金以及国家攀登计划等课题有关，在此谨对国家科委和国家自然科学基金委给予的资助与支持致以深切的谢意。

作 者

一九九四年十一月

目　录

第一章　绪论

§1.1　引言

视觉是人类最重要的感觉器官，外部世界丰富多采的信息有百分之七十是通过视觉感知的。因此，用计算机来处理与分析图象的研究与应用一直受到人们的高度重视，并取得了丰硕的成果。目前图象处理与分析技术已广泛应用于办公自动化、工业机器人、地理数据处理、医学数据处理、地球资源遥感、交互式计算机辅助设计等领域。图象处理与分析涉及到数学、计算机科学、模式识别、人工智能、信息论、生物医学等多种学科，是一门多学科交叉应用技术。

图象处理与分析涉及到的问题很多，本书主要描述其基本概念、主要方法和实现例程，其目的在于从理论与实践两个方面使读者了解、掌握和应用本书介绍的知识。为了突出实用性，本书在简明扼要地描述图象处理与分析的基本概念与基本原理的基础上，着重以大量例程说明图象处理中内容广泛、种类繁多的各类算法。这些例程已经过调试，编译后可直接运行于 IBM PC 及其兼容机上。

§1.2　图象表示和模型化

图象的表示是与量化后的图象元素(象素)的表示相关的。一幅图象可以是自然景物中物体的光强(例如用普通照像机照的照片)，也可以是身体器官的吸收特征的量化(例如 X 光照片)；或者目标物体的雷达反射截面(雷达图象)；或者一个区域的温度场(红外线图象)；或者重力场(地球物理图象)。一般地，任意一个二维函数表示的信息均可以看成是一幅图象，图象模型是这一函数特征的逻辑与定性的描述(图 1.1)。

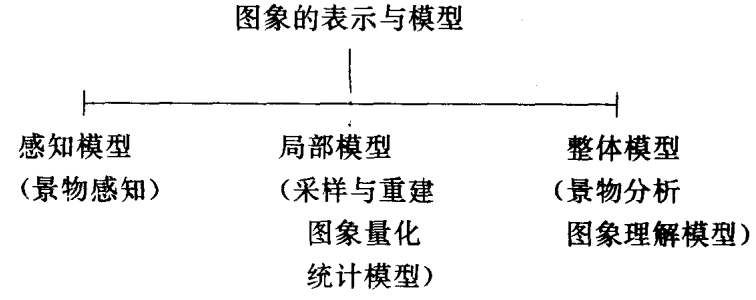


图 1.1　图象的表示与模型

1.　灰度图象的表示

灰度图象是指物体的二维光强度函数 $f(x,y)$ ($0 \leqslant f(x,y) \leqslant L-1$)，其中 x,y 是空间点的坐标，任意点(x,y) 处的函数值 f(x,y)正比于图象在该点的亮度(灰度级 L)，非负有

界二变量实函数表示一幅灰度图象，它是在空间的坐标和亮度上均已离散化的图象。我们可以把一幅灰度图象考虑为一个矩阵，其行和列表示图象中的一个点，而相应的矩阵中元素的值表示出该点的灰度级。如果灰度级仅为黑白两种，则称为二值图象。

2. **彩色图象的表示**

一般常用三原色(红色、绿色、蓝色)来产生彩色图象，所以彩色图象一般可以表示为：

$$f_c(x,y) = \{ f_r(x,y), f_g(x,y), f_b(x,y) \}$$

其中 $f_c(x,y)$ 由(r,g,b)表示，r,g,b 分别为三个灰度图象 f_r, f_g, f_b 的灰度值。

表示颜色的另一种方式是使用亮度、饱和度和色度。

亮度定义为 r,g,b 之和，即

亮度=r+g+b

饱和度表示某一颜色与全色的接近程度，

饱和度=1−3min(r,g,b)f(r+g+b)

色度近拟正比于该颜色的平均波长，即

色度 $(hue) = (\lambda_r r + \lambda_g g + \lambda_b b)/(r + g + b)$

其中 $\lambda_r, \lambda_g, \lambda_b$ 分别为红、绿、蓝三色的中心波长。

彩色图象的上述两种表示方法可以相互转换，我们在下章给出相应的转换方法与转换程序。

§1.3 本书的内容安排

如前所述，本书的目的是要使读者从实用角度掌握图象处理与分析的基本方法，所以本书一方面在描述图象处理与分析的基本理论方法的同时着重给出了实用例程，从而读者可以边学边用；另一方面也介绍了目前国际上热门的新方向——小波分析及其应用，从而使读者可以了解图象处理与分析的新方法与发展方向。

本书的内容安排是这样的，在第一章绪论之后，第二章描述图象数字化的两个概念：采样与量化，并给出了通过扫描仪采集图象的扫描仪驱动例程、图象显示与打印例程等实用程序。第三章讨论的图象变换、第四章讨论的图象增强和第五章讨论的图象恢复是图象处理的重要内容。由灰度图象到二值图象的转换可以归结为图象中物体的边缘提取。提取物体边缘的方法称为边缘检测，以区域为形式的物体抽取方法称为分割。第六章讨论的内容就是图象边缘提取与图象分割的技术。在第七章中我们给出了图象数据压缩的主要方法与例程。第八章的小波分析、第九章的图象分析和第十章的图象随机场模型介绍图象分析的基本原理与技术。

第二章　图象的采集与量化

为了便于计算机处理，图象函数 $f(x,y)$ 在空间上和幅度大小上都需要数字化。空间坐标 (x,y) 的数字化被认为是图象取样，而幅度数字化则称为灰度级量化。本章我们描述图象采集与量化的理论与方法，并给出扫描仪驱动、各种图象格式的读写与转换、图象打印等实用例程。

§2.1　采样定理

一、一维采样函数

我们的目的是用样本 $f(KT)$ 来表示一维函数 $f(t)$，其中 K 为整数值，T 为取样周期。由样本 $f(KT)$ 重构原始函数 $f(t)$ 的方法是在样本之间适当地进行插值，我们采用如下插值函数 $g(t)$：

$$f(t)=\sum_{K=-\infty}^{\infty}f(KT)g(t-KT) \tag{2.1}$$

即 $g(t)$ 是沿 t 轴移动时间 KT 的插值函数，而样本 $f(KT)$ 在 t 时刻对函数 $f(t)$ 的作用由系数 $g(t-KT)$ 加权，现假设 f 和 g 都是可作 Fourier 变换的，即：

$$f(KT)g(t-KT)=\int_{-\infty}^{\infty}f(\tau)g(t-\tau)\delta(\tau-KT)d\tau \tag{2.2}$$

把(2.2)式代入(2.1)式，我们得到：

$$f(t)=\int_{-\infty}^{\infty}f(\tau)g(t-\tau)\left(\sum_{K=-\infty}^{\infty}\delta(\tau-KT)\right)d\tau \tag{2.3}$$

其中易见 $\sum_{K=-\infty}^{\infty}\delta(\tau-KT)$ 是周期为 T 的周期函数，将其展开为 Fourier 级数：

$$\sum_{K=-\infty}^{\infty}\delta(\tau-KT)=\sum_{n=-\infty}^{\infty}a_n exp\left(\frac{j2\pi n\tau}{T}\right) \tag{2.4}$$

其中 Fourier 展式的系数 a_n 由下式给出：

$$\begin{aligned}a_n&=\frac{1}{T}\int_{-T/2}^{T/2}\left(\sum_{K=-\infty}^{\infty}\delta(\tau-KT)\right)exp\left(\frac{-j2\pi n\tau}{T}\right)d\tau\\&=\frac{1}{T}\int_{-T/2}^{T/2}\delta(\tau)exp\left(\frac{-j2\pi n\tau}{T}\right)d\tau\end{aligned}$$

在上面的积分中只在 $K=0$ 项为非零的，因此我们可以得出：

$$a_n=\frac{1}{T} \tag{2.5}$$

由(2.4)式和(2.5)式我们可以把(2.3)式改写成：

$$f(t)=\sum_{n=-\infty}^{\infty}\int_{-\infty}^{\infty}f(\tau)exp\left(\frac{j2\pi n\tau}{T}\right)\frac{g(t-\tau)}{T}d\tau \tag{2.6}$$

即 $f(t)$ 可以表示成函数

$$f(t)exp(\frac{j2\pi nt}{T}) \quad 和 \quad \frac{g(t)}{T} \tag{2.7}$$

的卷积的和。由 Fourier 变换的卷积特性可以看出(2.6)式求和号中每一项的变换均为(2.7)式中两个函数的 Fourier 变换之积。我们下面用 $F(\omega)$ 和 $G(\omega)$ 分别表示 $f(t)$ 和 $g(t)$ 的 Fourier 变换。

由 Fourier 变换的位移性质可知 $F(\omega - \frac{2\pi n}{T})$ 和$\frac{G(\omega)}{T}$ 。这样对(2.6)式两边作 Fourier 变换得到：

$$F(\omega) = \frac{G(\omega)}{T}\sum_{n=-\infty}^{\infty} F(\omega - \frac{2\pi n}{T}) \tag{2.8}$$

显然从(2.1)式到(2.8)式的上述推导步骤是可逆的，因此(2.8)式是由采用(2.1)式的样本 $f(KT)$ 正确地重构 $f(t)$ 的充分与必要条件。

下面我们用公式来表示使(2.8)式成立的 G 与 T 的充分条件。

假设 $f(t)$ 是有限带宽的，即当 $|\omega| \geqslant 2\pi f_c$ 时 $F(\omega) = 0$ ，而 $F(\omega - \frac{2\pi n}{T})$ 正好是 $F(\omega)$ 移位 $\frac{2\pi n}{T}$ ，所以对这样的 F ，只需取：

$$G(\omega) = \begin{cases} T & |\omega| < 2\pi f_c \\ 0 & 在其它\ \omega\ 处 \end{cases}$$

就可以使(2.8)式满定，易见 G 的 Fourier 反变换 g 是 sin 函数，事实上我们可得到：

$$g(t) = \frac{1}{2\pi}\int_{-2\pi f_c}^{2\pi f_c} Te^{j\omega t}d\omega = \frac{\sin 2\pi f_c t}{\pi t/T} \tag{2.9}$$

从而如 $T = f_c/2$ ，我们就有 $g(t) = \mathrm{sinc}(2\pi f_c t)$ 。

上述讨论实际上给出了 Whittaker—Kotelnikov—Shannon 定理的构造性证明。

定理 2.1 若 $|\omega| \geqslant 2\pi f_c$ 时函数 $f(t)$ 的 Fourier 变换为零，则 $f(t)$ 可以由相隔为 $f_c/2$ 或者更密的样本正确地重构。

二、二维采样定理

上述采样定理可以推广到二维情况，下面我们只给出二维采样定理的具体结果，过程略而不述。

设 $f(x,y)$ 的 Fourier 变换 $F(u,v)$ 是：

$$F(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{-j2\pi(xu+yv)}dxdy \tag{2.10}$$

并设空间频率函数 $F(u,v)$ 为有限带宽的：

$$|u| \leqslant 2\pi f_c, |v| \leqslant 2\pi f_c$$

相应于(2.3)式我们有下述二维采样定理：

$$f(x,y) = \sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty} f(mT,nT)g(x - mT, y - nT) \tag{2.11}$$

作与一维情况同样的推导，可获得使(2.11)式成立的充分必要条件：

$$f(x,y) = \frac{G(u,v)}{T^2}\sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty} F(u - \frac{2\pi m}{T}, v - \frac{2\pi n}{T}) \tag{2.12}$$

同一维情况相似，我们有如下的二维采样定理：

定理 2.2 若 $f(x,y)$ 是有限带宽的，则它可以从用间隔小于或等于 $\frac{f_c}{2}$ 的采样方格采样所得到的样本中精确地重构出来。

三、归一化正交函数采样

图象采样的目的是用有限数列或数阵表示一幅图象，只要我们有可能从这些数重现图象，那么样本就不一定要和图象平面上采样网格的灰度级相对应。下面我们将证明，如果用归一化正交函数集展开一个图象函数，那么可以取该展式的系数作为图象的样本。

设 $f(x,y)$ 是定义在 xy 平面上区域 L 的一个实函数。假设函数 $f(x,y)$ 是平方可积的：

$$\iint_L f^2(x,y)dxdy < \infty \tag{2.13}$$

假设我们给出平方可积函数集 $\{\varphi_{mn}(x,y), m = 0,1,2,\ldots; n = 0,1,2,\ldots\}$。它定义在同样的区域 L 内。

如果对 $m \neq p$ 或 $n \neq q (m,n,p,q = 0,1,2,\ldots)$，有

$$\iint \varphi_{mn}\varphi_{pq}^*(x,y)dxdy = 0 \tag{2.14}$$

则称函数集 $\{\varphi_{mn}(x,y)\}$ 为正交的。

若除满足(2.14)式之外还满足以下性质：

$$\iint_L |\varphi_{mn}(x,y)|^2 dxdy = 1,\ m,n = 0,1,2,\ldots \tag{2.15}$$

则该函数集 $\{\varphi_{mn}(x,y)\}$ 称为归一化正交的，其中函数 φ_{mn} 可以是实值的或复数的。

我们希望用形如

$$\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} a_{mn}\varphi_{mn}(x,y) \tag{2.16}$$

的求和式近似表示在 L 内所有点处的函数 $f(x,y)$，并使均方误差

$$e_{MN}^2 = \iint_L |f(x,y) - \sum_{m=0}^{M-1}\sum_{N=0}^{N-1} a_{mn}\varphi_{mn}(x,y)|^2 dxdy \tag{2.17}$$

为最小。

不难证明使 e_{MN}^2 为最小值的常数 a_{mn} 为

$$a_{mn} = \iint_L f(x,y)\varphi_{mn}^*(x,y)dxdy \tag{2.18}$$

一个归一化正交函数集 $\{\varphi_{mn}(x,y)\}$，如果对于每一个平方可积函数 f，我们有

$$\lim_{M\to\infty, N\to\infty} e_{MN}^2 = 0 \tag{2.19}$$

则称 $\{\varphi_{mn}(x,y)\}$ 为完备的。即当(2.16)式的项数趋于无限时，用(2.16)式近似表示时的均方误差趋于零。一个完备归一化正交函数集也称为归一化正交基。

综上所述可以得出如下定理：

定理 2.3 给定一个定义在 xy 平面的 L 区域内的归一化正交基 $\varphi_{mn}(x,y)$，$m,n = 0,1,2,\ldots$，则在 L 内平方可积的任何函数 $f(x,y)$ 可以展开为：

$$f(x,y) = \sum_{m=0}^{\infty}\sum_{n=0}^{\infty} a_{mn}\varphi_{mn}(x,y) \tag{2.20}$$

其中

$$a_{mn}=\iint_{\Sigma} f(x,y)\varphi_{mn}^{*}(x,y)dxdy \tag{2.21}$$

因为图象 $f(x,y)$ 可以由系数 a_{mn} 表示和重现，所以可称这些系数为图象的样本，给定一幅图象和一个定义在 xy 平面上同一区域内的归一化正交函数集，则利用(2.21)式可求得这些样本。

以上我们描述了有限带宽单个图象的采样方法，即给定一有限带宽图象，我们可以决定在图象平面上的取样方法和插值函数，使得图象可以根据样本精确地重构。上述方法可以推广到对一族或一类图象的采样方法，即推广到随机场的情况。限于本书篇幅和侧重点，我们对此情形不作详细描述。

§2.2 图象的量化

图象采样仅仅是把图象在空间上离散化，而图象的量化则是把由采样所取得的图象灰度值由模拟量转化为离散量，即完成模拟/数字变换。简单地说是把样本值的范围分为几段，落入某段中的所有灰度值用单一灰度级表示。通常，我们取量化间隔数为偶数。设样本在 $[z_k,z_{k+1}]$ 中取值，则其灰度值取为 q_k (图 2.1)。

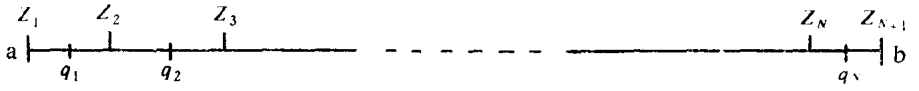


图 2.1 量化分级

图象处理中的点运算、取阈值处理都可以看成量化运算。其中落入 $[z_1,z_2]$ 中的样本量化为 q_1 (二进制零)，落入 $[z_2,z_3]$ 中的样本量化为 q_1 (二进制 1)，虽然量化级数通常为偶数，但由于样本幅度概率分布常常不是均匀的(如图 2.2)，所以量化间隔也并不是等距离的。显然对于那些较常出现的样本幅度范围，把量化间隔取细能改进图象质量，下面我们讨论获得最佳量化分级的方法。

设量化级为 $q_1,q_2,\ldots,q_L$，$h(z)$ 为样本幅度分布的密度函数，现定义均方量化误差为：

$$E=\sum_{k=1}^{L}\int_{z_k}^{z_{k+1}}(z-q_k)^2h(z)dz \tag{2.22}$$

为了提高图象质量，我们需选择一组最佳的 $(q_1,q_2,\ldots,q_L)$ 和 $(z_1,z_2,\ldots,z_L,z_{L+1})$ 使误差 E 最小。为此将误差 F 分别对 z_k 和 q_k 取偏微分，并令其结果为零：

$$\frac{\partial E}{\partial z_k}=(z_k-q_{k-1})^2h(z_k)-(z_k-q_k)^2h(z_k)=0,\quad k=2,\ldots,L \tag{2.23}$$

$$\frac{\partial E}{\partial q_k}=-2\sum_{z_k}^{z_{k+1}}(z-q_k)h(z)=0,\quad k=1,\ldots,L \tag{2.24}$$

由(2.23)式可得

$$(z_k-q_{k-1}+z_k-q_k)(z_k-q_{k-1}-z_k+q_k)h(z_k)=0$$

因此有

$$z_k = \frac{q_{k-1} + q_k}{2},\ k = 2,\ldots,L \tag{2.25}$$

(2.25)式表明，z_k 应取在 q_{k-1} 和 q_k 的中点，从(2.24)式可得

$$q_k = \frac{\int_{z_k}^{z_{k+1}} zh(z)dz}{\int_{z_k}^{z_{k+1}} h(z)dz} \tag{2.26}$$

(2.26)式说明：q_k 是 $h(z)$ 在 z_k 和 z_{k+1} 区间的重心。

最优量化问题一般叙述为：给定 $h(z)$，如何按照方程(2.25)和(2.26)来选择 z_k 和 q_k。我们可以使用下述迭代算法：首先选择一个 q_1，然后用(2.25)确定 $z_2(z_1 = 0)$，再用(2.25)式确定 $q_2,\ldots$，直到最后的 q_L，若由此得到的 z_{L+1} 不等于允许的最大值，则改变 q_1 重新开始迭代，直到满足特定条件的 z_{L+1} 出现为止。

如果 $h(z)$ 是均匀分布的，即 $h(z) = p$，则 z_k 和 q_k 可以直接计算，再由(2.26)式得到

$$q_k = \frac{0.5(z_{k+1} - z_k)}{z_{k+1} - z_k} = \frac{z_{k+1} + z_k}{2} \tag{2.27}$$

因此，若按下式选择 z_k 和 q_k，则可以满足(2.25)式和(2.26)式。

$$z_k = (k-1)/L, k = 1,2,\ldots,L+1$$

$$q_k = \frac{2k-1}{2L}, k = 1,2,\ldots,L$$

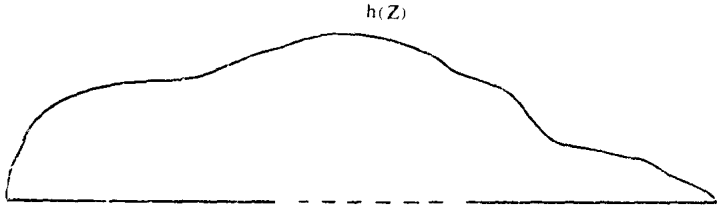


图 2.2　样本的幅度分布函数

采样与量化之间存在着一种依赖于图象细节的折衷关系。一般来说对慢变化的图象应侧重于量化的精确性，采样则可以相对粗糙一些；而对有大量细节的图象则必须精确地采样，其量化则可以相对粗糙些。

§2.3　几个实用例程

前两节我们讨论了图象的采样与量化，本节我们通过几个实用例程来描述图象的输入、显示、格式转换和打印。这些涉及到图象的输入和输出这两个主要环节，图象的处理技术将在后面的章节中介绍，有了这些例程，读者很容易将其作为输入与输出模块嵌入到自己的图

象信息系统中。

一、HP 扫描仪驱动例程

图象输入接口是图象信息系统的重要组成部分，常用的图象输入设备是扫描仪和 CCD 摄象机，下面我们给出目前国内常用的 HP 扫描仪驱动例程。

例程 2.1　HP 扫描仪驱动

```
/*
  Contents:
    SJdetect      Make sure scanner driver is present and hardware is OK
    SJerrno       Return last Scanjet error code
    SJgetcols     Determine bytes per scanline
    SJgetdwidth   Determine output data width
    SJgetodata    Determine output data type
    SJgetrows     Determine scanlines in the window
    SJmodel       Determine ScanJet model
    SJscanimage   Scan the window and transfer data to image
    SJscanwin     Scan the window
    SJsetdwidth   Set output data width
    SJsetodata    Set output data type and invert image
    SJxyres       Set X,Y resolution
    SJwinsize     Set scan window size
*/
#include <stdio.h>
#include <vicdefs.h>
#include <vicfcts.h>
#include <vicerror.h>
#include <stdlib.h>
#include <string.h>
#include <dos.h>
#include <stdarg.h>
#include <io.h>
#include <fcntl.h>
#define DEBUG 0
#define WHITE 1                    /* Image type */
#define BLACK 2
#define DITH 3
#define GRAY 4

static int cdecl SJgetval(int,char *);
static int cdecl SJsetval(int,char *,...);
```

```
static void cdecl swap(int * ,int * );
extern unsigned cdecl SJiofct(int,UCHAR);
extern UCHAR Hbuff_[];
extern int cdecl checkrange_(imgdes * );
extern int cdecl assign_gprow_(imgdes * ,int * ,int (_cdecl * * )(),int (_cdecl * * )());
extern void cdecl cmclose_(imgdes * ,int);
extern void cdecl set_raw_(int,int);
extern int cdecl check_ioctl_(int);
#if DEBUG == 1
static void sdisp_err(int,int);
#endif

/* Make sure scanner driver is present, hardware is operational, and return
a device handle. Returns NO_ERROR if everything is OK, BAD_OPN if driver
    was not detected, and SCAN_ERR if there's a hardware problem.
    For SJ, SJPlus.
*/
int cdecl SJdetect(int * sjhandle)
{
    int model, rcode=SCAN_ERR;

      /* Check that the scanner driver is installed */
      if(( * sjhandle=open("HPSCAN", O_RDWR | O_BINARY)) < 3)
        return(BAD_OPN);
      /* Make sure "HPSCAN" is a device name and not a filename */
      if(check_ioctl_( * sjhandle) == 0)
        rcode = BAD_OPN;
      else {
        /* Check for a scanner hardware error (may simply be turned off) */
          if(SJgeterrstat( * sjhandle) == NO_ERROR) {
            /* If scanner returns an error, don't continue */
            model = SJmodel( * sjhandle);
            if(inrange(0, model, 1))
              return(NO_ERROR);
            }
          }
        close( * sjhandle);          /* Done with device */
        return(rcode);
}
/* Return error number. Returns last Scanjet error code or SCAN_ERR if write
    error. For SJ, SJPlus.
*/
int cdecl SJerrno(int handle)
```

```
{
    int errno=0xff;
    if(write(handle, "\033*s259E", 6) == 6) {     /* Get errno */
        read(handle, (char *)&errno, 2);     /* Get value */
        write(handle, "\033*oE", 4);     /* Clear last error */
        return(errno);     /* Return value */
            }
        return(SCAN_ERR);
}
/* Determine bytes per scanline. Return bytes or SCAN_ERR if error.
    For SJ, SJPlus.
*/
int cdecl SJgetcols(int handle)
{
        return(SJgetval(handle, "\033*s1025E"));
}
/* Determine output data width. Return 1, 4, 8, or SCAN_ERR if error.
        For SJ, SJPlus. */
int cdecl SJgetdwidth(int handle)
{
        int model, dtype, dwidth=1;

        dtype = SJgetodata(handle);
        if(outrange(0, dtype, 4))
            return(SCAN_ERR);
        if(dtype == GRAY) {
            if((model=SJmodel(handle)) == 0)
              dwidth = 4;  /* HP ScanJet, 16 grays */
    else if(model == 1)     /* HP ScanJet Plus, 256 grays */
                dwidth = SJgetval(handle, "\033*s10312R");
            }
        return(dwidth);
}
/* Determine output data type. Return 0 — 4 or SCAN_ERR if error.
        For SJ, SJPlus. */
int cdecl SJgetodata(int handle)
{
        return(SJgetval(handle, "\033*s10325R"));
}

/* Determine scanlines in the window. Return rows or SCAN_ERR if error.
        For SJ, SJPlus. */
int cdecl SJgetrows(int handle)
```

```
{
        return(SJgetval(handle, "\033*s1026E"));
    }
/* Determine ScanJet model. Return 0 if SJ, 1 if SJPlus, SCAN_ERR if error.
        For SJ, SJPlus. */
int cdecl SJmodel(int handle)
{
        static char *sj_strs[] = {"\033*s3d5W9190A","\033*s3d5W9195A"};
        char buff[16];
        int j;

          if(write(handle, "\033*s3E", 5) == 5) {
            if(read(handle, buff, 12) == 12) {
              for(j=0; j<2; j++) {
                if(memcmp(buff, sj_strs[j], 12) == 0)
                  return(j);
                /* Match, it's a ScanJet or SJ plus */
              }
            }
          }
        return(SCAN_ERR);      /* Write/read error or unidentified string */
}
/* Scan the window into desimg. Works for 4- and 8-bit data widths.
  Returns NO_ERROR, SCAN_ERR, NO_EMM, EMM_ERR, NO_XMM, XMM_ERR, or CM_
ERR. */
int cdecl SJscanimage(int sjhandle, imgdes *desimg)
{
    int (_cdecl *getrow)(void huge *,int,int,int,int,int);
    int (_cdecl *putrow)(void huge *,int,int,int,int,int);
    int errcode=NO_ERROR, xsrows, scols, srows, cols, rows, bcols, yctr;
    int j, excols, mhandle, dwidth;
    UCHAR *sptr, *dptr=Hbuff_;

    /* Check range of start, end positions */
    if(checkrange_(desimg))
        return(BAD_RANGE);
    yctr = desimg->sty;
    /* Set raw output mode to be able to input ctrl-z's */
    set_raw_(1, sjhandle);
    /* Get the number of cols, rows to do according to the scanner */
    if((scols=SJgetcols(sjhandle)) == SCAN_ERR)
        return(SCAN_ERR);      /* Catch invalid handle error */
    srows = SJgetrows(sjhandle);
```