```
PRINT * ,        AND LESS THAN ,EPA
CALL XOMESG(IER,12,4,EZP)
IF(IER.EQ.-273) GOTO 854
IF((EZP.LT.EIP).OR.(EZP.GT.EPA)) THEN
PRINT * ,'       BEYOND NUMBER RANGE, PLEASE REENTER!
GOTO 804
```

遥感与非遥感地质信息复合应用中的计算机处理

```
IF(12.EQ.4) THEN
EL1=EZL-ESL+1
EP1=EAP
N1=EAP/NC
ENDIF

EL1=EZL-ESL+1
EP1=EZP-ESP+1
RA=EL1**2+EP1**2
IF((12.NE.0).AND.(12.NE.4)) EK=INT(SQRT(RA))
IF(12.EQ.0) EK=EL1*EP1

IF(12.EQ.4) THEN
DO 106 KK=1,NC
SI(KK)=0
DO 107 I1=1,EL1
EKA=ESL-EIL+I1
ERA=1
DO 107 J1=1,EP1
CALL XAWRLI(LUN,EADATA,EAP,EKA,ERA,EAP,1,XXXXXX,IER,IOF1,BUF1)
JK=ESP-EIP+J1
EP(I1,J1)=BUF1(JK+IOF1-1)
IF(EP(I1,J1).GE.0) EP(I1,J1)=EP(I1,J1)
IF(EP(I1,J1).LT.0) EP(I1,J1)=EP(I1,J1)+256
N1=AINT((J1-0.005)/N1)+1
SI(N1)=SI(N1)+REAL(
CONTINUE
PRINT * ,'N1=',N1
```

谭海樵　余志伟　著

地质出版社

遥感与非遥感地质信息复合应用中的计算机处理

谭海樵　余志伟　著

地　质　出　版　社
·北　京·

(京)新登字 085 号

内 容 提 要

针对遥感与地理信息系统技术在地质领域复合应用中遇到的实际问题，本书首先全面论述了各类实用地质信息的基本特征。并在此基础上，重点讨论了遥感与非遥感信息复合应用中的计算机处理方法，为常用的图形、图像软件之间的信息沟通提供了一系列实用型接口软件。用于复合处理的全部数据类型及格式的转换皆以目前通用的 386/486 微机为工作平台，并能在 DOS 操作系统下直接运行。为确保复合成果的实用性，本书以专门章节讨论了西文状态下的中文注记、输出及已有中文数据的转换问题，其中还包括了非 DOS 系统下的汉字注记问题，书中附有用 FORTRAN 语言编写的相关源程序。作为狭义的遥感、非遥感信息复合处理，本书除讨论地质模型的建立等常规方法外，还特别讨论了用三元超曲面样条函数进行信息转换的问题。

本书适用性广、实用性强，主要面向包括遥感地质在内的各类地质专业的研究生、本科生。也可供已有计算机应用基础知识的地质工作者以及环保、农林、土地管理等从事遥感应用研究的科技人员参考。

图书在版编目（CIP）数据

遥感与非遥感地质信息复合应用中的计算机处理/谭海樵，余志伟著．—北京：地质出版社，1995.1

ISBN 7-116-01819-0

Ⅰ．遥感… Ⅱ．①谭… ②余… Ⅲ．地质-遥感图象-信息处理-计算机应用 Ⅳ．P628

中国版本图书馆 CIP 数据核字（94）第 16202 号

地质出版社出版

（100013 北京和平里七区十楼）

责任编辑：白 铁

*

河北省地矿局测绘印刷厂印刷 新华书店总店科技发行所发行

开本：787×1092 1/16 印张：9 字数：205000

1995 年 1 月北京第一版·1995 年 1 月第一次印刷

印数：1—600 册 定价：10.00 元

ISBN 7-116-01819-0

P·1440

COMPUTER DATA PROCESSING FOR INTEGRATION OF REMOTE SENSING AND NON-REMOTE SENSING GEOLOGICAL INFORMATION

TAN HAIQIAO YU ZHIWEI

GEOLOGICAL PUBLISHING HOUSE

前　言

随着计算机技术的迅猛发展，以地理信息系统为代表的多元信息复合技术已发展为应用地质学领域中的一个重要分支。大量资料表明，遥感与地理信息系统相互支持、共同发展，已成为遥感与GIS技术发展必由之路。但要真正实现将遥感作为GIS的信息源和数据更新手段，还有大量的工作要做。从软件角度看，许多大型软件是从属于一定的硬件环境的，在许多微机上不能普遍使用；再者目前的软件功能虽强大，也不可能无所不包，特别对于特殊目的的处理而言，往往要广采百家软件之长，再加上自编软件，才能完成特定要求。换言之，商用软件往往是非个性化的，很难适用于特殊领域的处理，更不用说对于许多未知领域的探索，根本无现成软件可用。对于现有软件，当然是要充分利用的，问题在于各种软件的数据格式不统一（事实上也无法统一）；有些软件虽有外部数据接口（如dBASE系列数据库），但这些接口要在这些软件环境下才能使用。从严格意义讲，各种数据（信息）是互不兼容、缺乏沟通的，这是影响计算机技术发挥优势的主要障碍。

本书的主要目的就是考虑在DOS系统平台上，将各种数据（信息）沟通起来，为读者使用各种现成软件、自编软件，以及为遥感与非遥感地质信息的计算机复合处理提供一个基础。全书共分六章，第一章为绪论，简述遥感非遥感信息复合技术的基本内涵及其必要性。第二、三两章讨论遥感地质信息的基本特征和信息提取方法。第四章首先阐述了一些常用软件、处理系统的数据结构，在此基础上针对各种结构编制数据格式转换程序，以达到在各种软件系统之外，或将特定格式的数据转换成DOS平台下的数据，或将DOS平台数据完成特定操作后再写入到特定软件系统中去的目的。考虑到中文注记的特殊性和必要性，本书用独立的一章（第五章）讨论了不同西文软件环境下的各种汉字注记问题，并附有相关实用源程序。最后一章着重讨论了信息复合的各种方法及应用实例，着重介绍了用三元超曲面样条函数进行信息转换的方法。本书附有较多实用程序，但更重要的是为读者提供一个观察、分析数据结构和软件结构的方法和思路。事实上，借助PCTOOLS、DEBUG等工具，读者自己也可观察这些数据结构和软件结构，并自行编制相应程序。所以，当掌握了这些方法和思路后，读者将会感到在自己面前，所有数据结构乃至软件结构都是透明的，产生一种“豁然开朗”的感觉，由此加强自己随心所欲操纵使用各种软件的自信心。

由于作者是从事多年数学地质、遥感地质的计算机应用工作者，手边有的也大多是地质资料，故涉及到的数据、软件大都是与地质、遥感有关的，但这并不妨碍其它专业的计算机工作者将其用于自己的研究领域。事实上，与地质专业是一个多学科综合专业一样，地质信息的计算机处理也是一个涉及面十分广泛的领域；就信息来源而言，有数字的、非数字的，非数字类中有文本的、图形的、图像的；输出信息形式也是如此，光图形就有二维（如等值线）、三维（如立体图）等等。作者就遇到许多软件，用于其它领域可作十分漂亮的图件，但用于地质信息处理则无能为力。原因十分简单，地质学是以整个地球作为研究对象的，地质体数据之多、形态之复杂令人咋舌不已。反之，若将可用于地质信息处理的软件用于其它领域，则往往得心应手。所以，从这个意义上讲，本书的内容是具有普遍性的；从方法上讲，则更是如此。

本书得到煤炭院校优秀青年科学基金和煤炭系统留学回国人员科技基金的资助；在本

书撰写过程中得到中国矿业大学地质系和分析测试中心的大力支持；中国科学院地质所邵兴亚、张绍宗同志协助审校及出版事宜；戎辉和孙永战同学为书稿的打印、排版付出了辛勤的劳动，在此一并致以诚挚的谢意。

限于作者水平，书中疏漏之处在所难免，恳请读者批评指正。

作者一九九四年九月于中国矿业大学

目　录

前言

第一章　绪论…… (1)

一、 遥感非遥感地质信息复合处理的基本内涵 …… (1)
二、 遥感非遥感信息复合处理的研究现状与发展趋势 …… (2)
三、 现有信息复合处理技术的优势与不足 …… (2)

第二章　遥感地质信息的基本特征…… (3)

一、 遥感地质信息的成像方式 …… (3)
(一)摄影成像…… (3)
(二)数字扫描成像…… (5)
(三)雷达成像…… (6)
二、 遥感地质信息的几何特征 …… (6)
(一)中心投影成像…… (6)
(二)全景扫描成像…… (8)
(三)侧视成像…… (9)
三、 遥感地质信息的图像特征…… (11)
(一)比例尺 …… (12)
(二)灰度值 …… (12)
(三)反差 …… (13)
(四)分辨率 …… (13)
四、 数据格式…… (13)
(一)模拟图像类 …… (13)
(二)数字图像类 …… (14)
(三)其它类 …… (15)

第三章　遥感地质信息的提取 …… (16)

一、 遥感图像的几何特征…… (16)
(一)系统几何畸变及其校正 …… (16)
(二)随机几何畸变及其校正 …… (21)
二、 遥感图像的辐射校正与波段的选取…… (22)
(一)系统辐射畸变及其校正 …… (22)
(二)波段的选取 …… (23)
三、 基础影像图的制备…… (24)

四、 专题信息的提取…… (25)
(一)目视解译法 …… (25)
(二)自动分类法 …… (26)
五、 遥感图像中原数据的提取…… (26)

第四章 地质信息通用格式的转换 …… (35)

一、 非遥感地质信息的基本特征…… (35)
(一)数字型地质数据的基本特征 …… (35)
(二)非数字型地质数据的基本特征 …… (36)
二、 数据格式转换的基准…… (36)
(一)公用平台的统一 …… (36)
(二)数据格式的统一 …… (37)
三、 常用软件涉及的文件结构和数据格式…… (37)
(一)dBASE—FoxBase—FoxPro 系列数据库 …… (37)
(二)Grapher 系列绘图软件的文件结构 …… (38)
(三)pcARC/INFO 的基本结构与数据格式 …… (39)
(四)TIFF 文件基本结构 …… (41)
(五)地震勘探记录 SEG—Y 格式的文件结构和数据格式 …… (43)
四、 非遥感数据格式的转换…… (44)
(一)dBASE 系列数据库文件与 DOS 平台文件的相互转换 …… (44)
(二)DOS 通用格式文件向 pcARC/INFO 格式文件的转换 …… (49)
(三)地震记录 SEG—Y 格式数据向 DOS 数据文件的转换 …… (51)
(四)TIFF 文件格式的转换 …… (52)
(五)不同格式图像文件间的转换 …… (56)

第五章 数据处理中的特殊问题——汉字注记 …… (58)

一、 汉字库的结构…… (59)
(一)点阵库结构 …… (59)
(二)矢量库结构 …… (59)
二、 在 Grapher 中标注汉字 …… (61)
(一)SYM 形式调用汉字…… (61)
(二)DAT 形式调用汉字…… (70)
三、 在 C 语言中标注汉字 …… (77)
四、 pcARC/INFO 中的汉字注记 …… (85)
(一)点阵汉字的转换 …… (85)
(二)矢量汉字的转换 …… (89)
(三)手书汉字的输入及转换…… (100)

(四)汉字的生成与调用…… (101)
(五)实用效果的对比…… (103)
五、 AUTO CAD 中的汉字注记 …… (105)
(一)图层方式或命令组方式…… (105)
(二)DXF 文件方式 …… (105)
(三)LISP 语言方式 …… (105)
(四)直接数字化方式…… (106)
(五)字符集方式…… (106)
六、 VAX/VMS 系统下的汉字注记 …… (106)
(一)实用数据的传输与转换…… (106)
(二)字库传输方式…… (107)
(三)矢量汉字的注记…… (107)
(四)手书汉字的注记…… (107)

第六章 遥感非遥感地质信息的计算机复合处理…… (108)

一、 复合的基础 …… (108)
(一)地理坐标的配准…… (108)
(二)数据类型的统一…… (109)
二、 不同类型数据之间的相互转换 …… (109)
(一)矢量型数据向栅格型数据的转换…… (109)
(二)栅格数据的矢量化…… (118)
三、 复合处理方法 …… (119)
(一)多元数据统计分析…… (119)
(二)曲面样条函数插值…… (119)
四、 应用实例 …… (124)
(一)遥感与重力异常数据复合处理在徐淮地区找煤预测中的应用…… (124)
(二)用三元超曲面样条函数进行时空转换…… (124)

英文摘要…… (129)
参考文献…… (131)
图版…… (132)

CONTENTS

Preface

Chapter 1 Introduction ····· (1)

1. Fundamental Concepts of Integration of Remote Sensing and Non—Remote Sensing Geological Information ····· (1)
2. Present Situation and Tendency of Researches on Data Integration ····· (2)
3. Advantages and Disadvantages of the Present Data Integration Techniques ··· ····· (2)

Chapter 2 Basic Features of Remote Sensing Geological Information ····· (3)

1. Imaging Styles of Remote Sensing Geological Information ····· (3)
   1)Photograph ····· (3)
   2)Digital Scanning ····· (5)
   3)Radar Imaging ····· (6)
2. Geometrical Features of Remote Sensing Geological Information ····· (6)
   1)Central Projection ····· (6)
   2)Panoramic Scanning ····· (8)
   3)Side—Looking Imaging ····· (9)
3. Imagery Features of Remote Sensing Geological Information ····· (11)
   1)Scale ····· (12)
   2)Grey Values ····· (12)
   3)Contrast ····· (13)
   4)Resolution ····· (13)
4. Data Formats ····· (13)
   1)Analogical Imagery ····· (13)
   2)Digital Imagery ····· (14)
   3)Others ····· (15)

Chapter 3 Extract of Remote Sensing Geological Information ····· (16)

1. Geometrical features of Remote Sensing images ····· (16)
   1)Systematically Geometric Distortion and its Correction ····· (16)
   2)Random Geometric Distortion and its Correction ····· (21)
2. Radiometrical Correction of Remote Sensing Images and Band Selection ····· ····· (22)

1)Systematically Radiometrical Distortion and its Correction …… (22)
2)Band Selection …… (23)
3. Prepararion of Fundamental Image Map …… (24)
4. Extract of Thematic Information …… (25)
1)Visual Interpretation …… (25)
2)Automatic Classification …… (26)
5. Extract of the Primary Data from Remote Sensing Images …… (26)

Chapter 4 Conversation of Geological Information to DOS Common Format …… (35)

1. Basic Features of Non—Remote Sensing Geological Information …… (35)
1)Digital Geological Data …… (35)
2)Non—Digital Geological Data …… (36)
2. Standard for Data Format Conversation …… (36)
1)Common Interface …… (36)
2)Common Data Format …… (37)
3. File Structures and Data Formats of Common Used Softwares …… (37)
1)dBASE—FoxBase—FoxPro DataBase …… (37)
2)Graphics Software Grapher …… (38)
3)Basic Structure and Data Format used in pcARC/INFO …… (39)
4)Basic Structure of TIFF Files …… (41)
5)Structure And Data Format of SEG—Y Format Records used in Seismographic Exploration …… (43)
4. Format Conversation of Non—Remote Sensing Data …… (44)
1)Mutual Conversation of dBASE and DOS Files …… (44)
2)Conversation of pcARC/INFO Files to DOS Common Format …… (49)
3)Conversation of SEG—Y Format Seismographic Records …… (51)
4)Coversation of TIFF Files …… (52)
5)Conversation of Images with Different Formats …… (56)

Chapter 5 Special Problem in Data Processing —— Chinese Annotation …… (58)

1. Structures of Chinese Character Libraries …… (59)
1)Dot—Character Library …… (59)
2)Vector—Character Library …… (59)
2. Chinese Annotation in Grapher …… (61)
1)Call in SYM Form …… (61)
2) Call in DAT Form …… (70)
3. Chinese Annotation in Programming Language C …… (77)

4. Chinese Annotation in pcARC/INFO ····· (85)
1)Conversation of Dot—Chinese—Character ····· (85)
2)Conversation of Vector—Chinese—Character ····· (89)
3)Input and Conversation of Handwritten ChineseCharacter ····· (100)
4)Generation and Call of Chinese Characters ····· (101)
5)Comparison of Application Results ····· (103)
5. Chinese Annotation in AutoCAD ····· (105)
1)Using Layer or SCR Files ····· (105)
2)Using DXF Files ····· (105)
3)Using LISP ····· (105)
4)Direct Digitalization ····· (106)
5)Using Font Files ····· (106)
6. Chinese Annotation in VAX/VMS Operating System ····· (106)
1)Transmission and Conversation of Used Data ····· (106)
2)Transmission of Chinese Character Library ····· (107)
3)Annotation of Vector—Chinese—Characters ····· (107)
4)Annotation of Handwritten Chinese Characters ····· (107)

Chapter 6 Data Integration Processing of Remote Sensing and Non—Remote Sensing Geological Information ····· (108)

1. Basis of Integration ····· (108)
1)Match of Geographic Coordinates ····· (108)
2)Standardization of Data Types ····· (109)
2. Mutual Data Conversation of Different Types ····· (109)
1)Conversation of Vector—Type Data to Raster—Type ····· (109)
2)Conversation of Raster—Type Data to Vector—Type ····· (118)
3. Integration Methods ····· (119)
1)Multi—Verible Statistical Analysis ····· (119)
2)Interpolation Using Surface Splines ····· (119)
4. Applications ····· (124)
1)Integration of Remote Sensing and Gravity Data applied to Coalfield Prediction in Xuzhou—Huaibei Area ····· (124)
2)Coversation of TimeData to Spacial Data Using 3—Variable Hyper—Surface Splines ····· (124)

Abstract ····· (129)
References ····· (131)
Plates ····· (132)

第一章 绪 论

面对瞬息万变的信息时代，世界许多国家，特别是发达国家已把发展信息产业、建设“全球信息高速公路”视为一项重要的发展战略。与此相适应，我国也根据国情，提出了“信息国道”的计划并已经付诸实施。信息本身作为一种资源，已经被卷入市场经济的大潮。充分开发利用现有信息资源，并不断开辟新的信息源，已引起包括地质学界在内的各界人士的广泛关注。从早年的多种地质手段的综合利用，到各类遥感平台的投入运行，就是最明显的例证。而且有越来越多的资料表明，许多新矿产地的发现往往并不取决于某一种单一手段的投入，而是取决于对各种已有资料的综合评价，甚至是基于对老资料的重新评价。就地质学而言，有史以来的全部地质调查对地质体所取得的认识，相对于地质体本身时空来讲，可以说都是微不足道的。无论是最原始的路线地质调查，还是现代的遥感地质调查，无论是目视为依据的检测方法还是现代化的测试手段，都只不过是获取了地质体的某一方面的局部信息。即使是将已有的检测手段全部都投入到某一地质体的调查，也难以真正认识其全貌。正是在这个意义上，多源地质信息的复合应用就显得格外重要。

一、 遥感与非遥感地质信息复合的基本内涵

简尔言之，研究地质体的所有手段分为两大类：遥感与非遥感。前者系指经过“遥远感知手段”获取地质体的间接信息，本书中指摄影成像与扫描成像之类的探测方法，但不包括地球物理勘测。而非遥感手段，即是指各种传统地质调查手段，从地面地质调查到钻探、坑探、化探、物探等调查手段及各种模拟实验及微观测试手段的应用。

所谓遥感非遥感地质信息的复合处理，实质上就是将从研究对象获取的所有信息全部纳入统一的时空体系内所进行的综合评价。包括不同遥感信息之间，不同非遥感信息之间以及遥感与非遥感地质信息之间的综合分析与评价。换句话说，就是将从不同侧面、用不同手段得到的同一地质体的信息，皆视为统一地质体的不同表征。而复合应用的本质，则是要找出其内在规律，即最能反映地质体本来面貌的基本特征。比如，对一个矿山的认识，从普查到勘探，直至开采结束，也不能说有了全面的认识。信息复合应用只是为尽可能全面掌握现有资料，并在此基础上揭示其本来面目或预测其展布规律，提供一个有力的工具。

为实现上述目标，遥感非遥感地质信息复合处理的基本内涵应包括：

①原始数据的收集；

②数据类型与格式的统一；

③数据的复合处理。

比较而言，上述第二项任务的工作量最大。数据类型与格式的统一在这里起着承上启下的重要作用，它是原数据与复合处理之间的重要“桥梁”。所以本书将以较大的篇幅来讨论有关数据类型及格式转换的问题，对于上述第一项任务，书中将着重介绍遥感、非遥感两大类地质信息的基本特征，作为收集原始数据的参考。至于数据的复合处理，各类数学地质统计及地质模型建立方法皆已得到很大的发展，故本书只将重点放在有关矢量数据的曲面样条函数插值处理方法的讨论，对多元数据的统计分析只作一般介绍。

二、 遥感与非遥感信息复合处理的研究现状与发展趋势

随着时间的推移，人们对我们赖以生存的地球的认识正在不断深化，这一方面要感谢新理论、新方法的不断提出，另一个方面就是得益于信息复合技术的应用。如果说在40年前，广阔的中国大地上还有许多地质调查的处女地的话，那么现在几乎可以说是走向了另一个极端，过剩的地质工作者面对的是一片几乎过剩勘测过的大地。与此同时，还有许多资源卫星正在几百公里的高空时时刻刻地在探视着我们的地球。来自遥感成像系统的信息量正以百万兆计与日俱增。其中许多波段是在传统地质手段无法达到的非可见光波段区。就是没有这些卫星的存在，积几十年、上百年地质工作所形成的各种文档资料也多得令人目不暇接。如果说我们正生活在一个信息爆炸的时代也不为过，对地质学界尤其是这样。

近期发表的文献资料清楚地表明，正是在上述形势下，遥感（RS）与地理信息系统（GIS）技术都在沿着各自的固有方向飞速发展的同时，借助于日臻成熟的全球定位系统（GPS）技术，逐步朝着相互融合的方向迈进。一方面，遥感数据的信息量和精度皆在不段提高，另一方面，与GIS相关的软、硬件技术正在高速度地不断优化。各类地质信息的专业化、定量化和智能化水平的提高和上述3S技术的发展正在促使遥感和地理信息技术的不断融合。遥感、非遥感信息复合的应用研究，已成为促进遥感和地理信息系统技术共同发展的重要手段。其核心就在于为合理开发利用已有各类信息提供一个有力的工具。就研究的内容而言，遥感、非遥感信息复合的应用研究，已非常接近于美国环境系统研究所邓格蒙所提出的空间信息系统（SIS），因为所有地质信息都是在三维空间分布的。

三、 现有地质信息复合技术的优势与不足

自60年代卫星遥感技术进入实用阶段以来，遥感技术本身已有了长足的进步，无论是信息量还是工作波段，无论是几何分辨还是波谱分辨皆有了很大的提高。已发射和计划发射的遥感卫星参数充分表明了这一点。与此同时，GIS技术与遥感技术互为补充相辅相成，也得到长足的进步，相应的软件已得到迅猛发展，国内外开发的各种GIS软件已达近百种之多。同时各类输入、输出设备的性能也在不断提高。所有这些都为遥感非遥感信息复合应用提供了良好的外部条件。

与此同时，人们也注意到，在软件保护等因素的制约下，遥感与遥感信息之间、非遥感与遥感信息之间乃至不同的数据处理软件之间数据互不兼容，难以沟通，已成为影响其技术优势发挥的一大障碍。另一方面，现有各种版本的GIS软件有一个共同的不足之处，即其图形处理功能较强，但应用遥感信息的图像处理功能相对较弱。这一不足已为广大GIS和遥感地质工作者所公认，并正在为弥补这一缺陷进行各种努力，以期实现遥感与GIS的集成化，即把遥感作为GIS的信息源和数据更新手段。

此外，国内开发的中文环境下的地理信息系统，往往因使用汉字操作系统而占用了宝贵的内存资源；而国外开发的软件，如pcARC/INFO等，功能较强，颇受用户的欢迎，但又因中文注记问题而影响到其优势的发挥。至于其它多种手段获得的地质信息，更是载体不同、格式各异。要真正实现多元地质信息的沟通，当务之急，就是能够将所有各类信息的纳入统一的信息之中，即使之具有彼此可以兼容互通的数据格式，而所有这些正是本书的宗旨所在。

第二章　遥感地质信息的基本特征

如前所述，90年代的遥感技术，无论是在成像技术方面，还是在图像处理技术的实用化、智能化方面，皆会有一个迅猛的发展。在地理信息系统中各类地学信息的更新方面，遥感地质信息更是起着其它技术手段所无法替代的重要作用。单就地表信息的空间定位和实时变化的确定而言，遥感地质信息也是一种不可缺少的手段。随着“全球信息高速公路”的逐步开通，国家“经济信息国道”的投入运行，对遥感信息的需求量必将大增。而影响遥感信息在“信息高速公路”上运行的主要障碍将不是“信息高速公路”的运载能力，也不是遥感地质信息的采集，真正影响遥感信息发挥其快速、经济、信息量大等多方面优势的发挥的，恰恰是遥感地质信息本身的种种限制。由于遥感成像机理、成像几何等诸多因素的制约，直接从遥感平台获取的或是仅仅经过传统方法进行预处理的遥感地质信息，还无法满足在“信息高速公路”上运行的需要，甚至在不同类型的遥感地质及信息之间也无法直接进行数据的传输、调用。世界科学联合会国际数据委员会(CODATA)在1986年就多卫星专题制图方法问题提出的报告就是一个典型的例证。基于上述，本书在全面讨论各类遥感、非遥感地质信息的复合处理技术之前，拟在本章里，着重讨论各类遥感地质信息的基本特征。现按成像方式、成像几何、波谱特征及数据格式的顺序分述如下：

一、　遥感地质信息的成像方式

如所周知，地球表面遥感信息的成像技术的发展可以回溯到相机发明的1893年。航空摄影成像技术一直延续至今。到本世纪的60年代，在成像技术方面取得的一大突破，就是航空多光谱扫描技术，尔后还研制了侧视雷达成像技术，直到1972年陆地卫星升空之后，卫星成像技术才投入实用，并广为人知，以至卫星成像几乎成了遥感成像技术的代名词。迄今为止，随着半导体和计算机技术的迅猛发展，遥感成像技术亦随之得到飞速发展。为了便于讨论，现将遥感地质成像方式分为数字式与模拟式两大类，并将各种相关成像方式列入表2—1。

表2—1清楚地表明，无论遥感平台位于什么位置，就其成像方式而言，大致可分为两大类：即摄影成像和数字扫描成像。除此之外，还有电视摄像成像方式。

表2—1　常见遥感地质信息成像方式一览表

成像方式 / 波段	模　拟	数　字
可见光	地面摄影	机载多光谱扫描
	航空摄影	光机式星载多光谱扫描
红　外	超高空摄影	光电式星载多光谱扫描
微　波	侧视雷达	机载合成孔径雷达
	合成孔径雷达	星载合成孔径雷达

(一)　摄影成像

如果把相机的发明作为摄影成像技术的起点的话，那么摄影成像至少已有百年的历史，单就成像方式本身面言，似

乎并没有什么重大突破，整个成像系统亦然是由相机和胶片构成的。其成像平台的升高，乃至在无人操作的情况下自动拍照，丝毫也未触及其成像几何、成像波谱等基本特征。在可以预见的将来，其成像机制也不会有什么本质上的改变，而且仍不失为一种采集遥感地质信息的有效手段。

表 2－2　主要摄影成像产品基本特征对比一览表

片种		对应波段（μm）	特点	备注
黑白	常规（全色）	0.4～0.7	成本低廉、便于获得，主要用于地形图调绘、地质调查	远不及彩色照片可视性强，地物目标分辨性差等
	红外	0.7～0.9	反映的是太阳幅射热能 ·空间分辨率较高 ·水陆界线清晰 ·植被差异易于区分	
	紫外	0.3～0.4	提取可见光区段以外的紫外的信息，可用于特定目标的验测，如原油（油膜）、碳酸盐等	反差小（模糊），分辨率低
彩色	标准彩色	0.4～0.7	·真实反映地物目标的原有色彩，与人的视觉习惯一致 ·分辨能力大大提高	成本高
	彩红外	0.5～0.9	·较黑白红外更易于识别地物目标 ·对植被的差异性易于辨认 ·对水体、陆地易于区分 ·间接指示地下的岩石和土壤特性 ·病虫害的预测预报	增加地质信息量（如对岩石识别）
多波段		0.4～0.6 0.6～0.7 0.7～0.9	·多镜头同时成像 ·为用户提供更多选择自由（信息源扩大） ·可由黑白图像制备假彩色合成片	地面分辨率并未提高，但成本大幅度上升

尽管早期的摄影成像是以可见光波段（0.4－0.7μm）为主要成像区段，而且常规摄影所指的成像波段也是指可见光波段。但实际上，一部分紫外（0.3－0.4μm）和红外（0.7－0.9μm）皆可用于摄影成像。因此，用作摄影成像的遥感区段应为 0.3－0.9μm 的波长区，除非像在阿波罗计划的 SO－65 实验或是天空实验室（SKYLAB）中的S－190A实验那样，分别在指定波段成像。一般所说的摄影成像主要包括以下几类：

全色黑白照片：0.3－0.7μm

标准彩色摄影：0.4—0.7μm

红外黑白摄影：如 0.7—0.8μm，0.8—0.9μm

彩红外摄影：　0.5—0.9μm

紫外摄影：　　0.3—0.4μm

就直观特征来讲，摄影成像可分为黑白、彩色两大类。由于不同摄影成像成果所对应的波段不同，且其作用各有千秋，所以皆应作为多元遥感地质信息中的一元来看待。至于片种的选取，则应视研究目的不同进行判断。基于前人的经验，表 2—2 列举了不同类型摄影成像产品的优缺点，可作选取片种的参考（表 2—2）。

（二）　数字扫描成像

早期的光学-机械扫描成像系统，可以 1972 年发射的陆地卫星一号的多波段扫描仪系统(MSS)为代表。在 1973 年发射到轨道上的天空实验室(SKYLAB)计划中，也使用了此类多波段扫描仪，单就所获遥感地质信息的特征而言，此类成像系统与摄影成像的最大区别，就在于它获取的遥感信息是以数字方式记录的，数据的载体也不再是胶片或照片，而是磁带。另一个值得提出的重要区别，是其信息量远远大于摄影成像，如早期的 LANDSAT—MSS含 4 个波段，而 SKYLAB 的多光谱扫描仪则多达 13 个波段。1981 年欧洲宇航局的FMP计划中使用的 M^2S 多光谱扫描仪也有 11 个波段（表 2—3）。各数字扫描成像系统中皆包含了摄影遥感区之外的热红外波段，如SKYLABOR的第 13 波段 10.20—12.25μm；LANDSAT—TM中的 6 波段 10.4—12.5μm；FMP 中的 11 波段 11.0μm。

表 2—3　M^2S 多光谱扫描成像系统主要技术参数(引自 Kronberg，1985)

探视角	100°
滚动补偿	10°
扫描速度	10—100 条线/秒
几何分辨率	2.5 毫弧度

波　段	波　长(μm)	频　带　宽(μm)	灵　敏　度(%)
1	0.410	0.06	2.0
2	0.465	0.05	0.5
3	0.515	0.05	0.3
4	0.560	0.04	0.3
5	0.600	0.04	0.3
6	0.640	0.04	0.3
7	0.680	0.04	0.4
8	0.720	0.04	0.5
9	0.815	0.09	0.5
10	1.015	0.09	1.0
11	11.0	6.0	0.25k

考虑到多数多光谱扫描系统的资料皆可从有关手册或教科书中查到，无需赘述，在这里着重讨论影响到遥感信息复合处理的两大问题。

1. 数据的离散性

与摄影成像相比，多光谱扫描系统提供的遥感图像，已经不再是连续图像，而是离散图像。尽管分辨率的提高有助于提高遥感信息的可利用性，但无论如何提高其分辨率，就其记录方式而言，它所为用户提供的只能是数字化离散图像，进而要求用户使用专门的数字图像处理系统才能对其进行各种处理；后面将要讨论的遥感信息处理方法也正是以数字化离散数据为对像的。

2. 扫描方式的差异

早期的光学-机械扫描方式中，往往由扫描镜的来回摆动来获取扫描行上的全部地表信息。无论是LANDSAT－MSS中的单向扫描还是TM系统的双向扫描，都无法摆脱由于扫描镜机械运动所引起的几何畸变，并因而引起此类系统所生成的遥感信息处理中的若干特殊问题（详见本章第二节）。只是到了80年代研制的光电扫描系统，如MOMS系统和SPOT/HRV系统，才从根本上解决了这一畸变问题。

（三） 雷达成像

80年代以来，雷达成像日益受到人们的广泛重视，无论是机载系统还是星载系统，皆已成为遥感地质的重要信源。这不只是因为其全天候工作能力，和成像能源的可控制性，工作波段微波的穿透能力也是吸引人们注意的一个重要因素。

由于雷达的工作波段位于几毫米到1m的微波段，它能记录到的信息相对传统的摄影成像和数字扫描成像来讲是全新的，并且早在五、六十年代就已经有许多卓有成效的应用实例。就图像信息的特点而言，它记录的不是反射或辐射等波谱信息。雷达成像的像元灰度值是根据回波所需的时间来记录的，尽管它的载体既有胶片也有数字磁带，但与摄影成像和数字为扫描成像毫无共同之处。因而对此类图像的处理方法也不同于传统的图像处理方法，甚至图像上所看到的灰度的含义也与前两类截然不同，灰度的变化与地物目标本身的特征并非直接相关。它反映的是回波所需时间，等距离点往往有同样的灰度值。

二、 遥感地质信息的几何特征

尽管世界各国对遥感成像系统的研制各有侧重，从而已发射或准备发射的遥感卫星等遥感成像系统型号各异、功能不同；再加上飞机及至地面遥感平台也在不断采集、存储各种各样的遥感信息，使得原来数据量就很大的遥感数据又因成像方式及其数据结构类型的不同而更加复杂化。一般地讲，我们可将各类遥感平台上的成像系统按其成像几何方式分为三大类：中心投影类、全景扫描类以及侧视成像类。比如地面摄影、航空摄影及超高空摄影成像，尽管平台类型不同，成像高度迥异，但皆属中心投影方式；同样航空多光谱扫描成像及陆地卫星MSS及TM图像又都属于全景扫描图像；至于侧视成像，主要是指在包括飞机、卫星在内的各种平台上成像的雷达系统。

（一） 中心投影成像

所谓中心投影系指以摄影镜头中心光轴与成像目标平面垂直，从而将地物目标大致垂

直投影到成像底片上的成像方式（图 2—1），航空照片即为此类成像方式的典型代表。

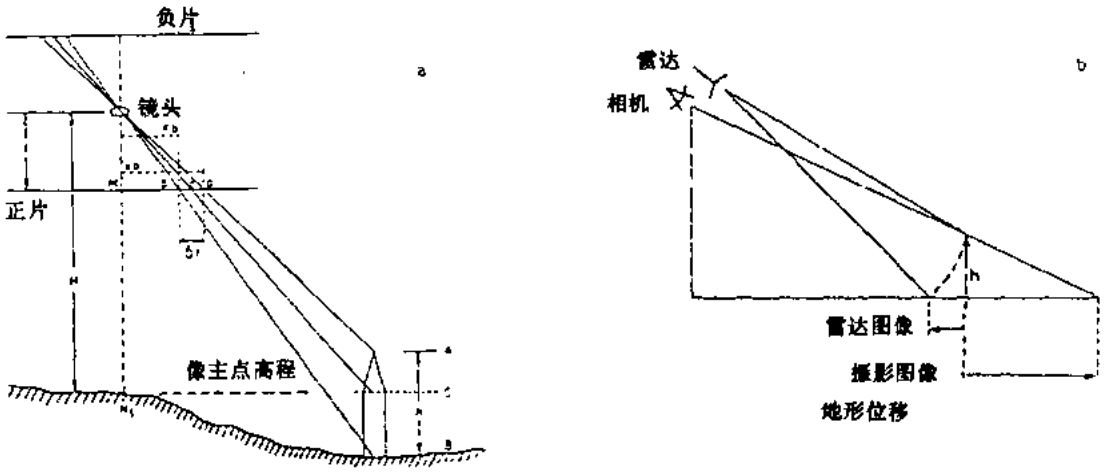


图 2—1　中心投影与侧视成像几何对比示意图（据 KRONBERG，1984，SIMONETT & DAVIS，1983）

a—中心投影方式；b—侧视成像方式

从图 2—1 不难看出，照像底片上记录到的地物特征，并不是像一般地形图那样，或是像地理信息系统中要求的那样，以正射投影方式排放的。除非正好位于像主点（像幅的正中心点，下同），否则一个直立的旗杆不可能在航空照片上记录为一个点，它总是要有阴影存在，其阴影长度是旗杆自身高度与旗杆所处位置偏离中垂线的角度的函数。

$$l_s = l \cdot \mathrm{tg}\theta$$

式中 l_s：阴影长度；

l：旗杆原高度；

θ：旗杆处偏离中垂线的角度，此角度以旗杆顶端为准，如以旗杆与地平面焦点处起算，则上式中的 θ 应计算为：

$$\mathrm{tg}\theta = (a + l_s)/H$$

式中 a 为旗杆到像主点的水平距离。

显然，在航高 H 固定的情况下，a 值不变，则 l 愈大，θ 愈大，阴影越长；而在 l 固定的情况下，θ 越大，阴影越长（图 2—1）。这一特征反映在影像上，即为高层建筑皆由像主点向四周倾斜。

以像主点为中心，地物目标的畸变是呈辐射状向四周漫延，已成为此类成像系统的一大特点，无论是地面摄影，航空摄影还是星载摄影皆无法回避。此类遥感信息的另一个特点，就是其成果为模拟图像而不是数字图像，因而也就具备了模拟图像特有的优势。如对图像处理系统的要求不是太高，可进行立体解译，可在不同平台上进行实施等；同时也带

来其不利的一面，如照片质量不稳定，不可重复等。

作为地理信息系统中的一元，为了与其它信息能够在定位几何上进行配准，有大量的前期处理工作要做，而要处理的主要问题，则是成像几何本身造成的畸变。现分述如下：

——平面上，整幅图像的比例尺变化较大，尽管在理论上，航片的比例尺可由下式决定。

$$M_0 = \frac{h_0}{c}$$

式中 h_0 为航高，c 为相机的焦距。但在实际上，图幅中各处的航高皆不仅因地物的高度变化而变化，而且，随偏离像主点的距离的大小而变化。一般来讲，愈接近像幅边缘，比例尺愈小。换句话说，像幅上两点之间的实际距离随距离像主点的远近不同而异，越接近边缘，代表的实际距离越大。就这一点而论，有些类似下面要讨论的全景扫描图像，所不同的是，在航片上，此类变化是辐射状分布的而在全景扫描图像中是逐行分布的。

——在纵向上，各点的比例尺又与高程相关。从上式中不难看出，在相机确定的情况下，航高越大，比例尺越小；如将飞行高度或更准确地讲，摄影平台的高度固定，则在同一像幅之内且距离像主点同等距离的情况下，地物目标越高，则比例尺会越大。正因为如此，在目视解释中要求找出高度相当的地物目标来用以计算像幅的比例尺，这一比例尺实际上只能作为一参考比例尺。

基于上述，在中心投影成像的遥感影像上，各点的比例尺皆不相同。只有在成像平台的高度与地物的实际尺度相比非常大的情况下，才可近似地将地物高度的影响忽略，诸如MC计划之类的超高空摄影的优势也正在于此。

（二） 全景扫描成像

与中心投影方式生成模拟影像不同，全景扫描图像系指以扫描方式生成的数字影像，其成像平台可以是飞机也可以是卫星或其它类似的载体。其扫描成像的核心部件，扫描器本身可以是光学机械方式，也可以是光学电子方式，前者可以陆地卫星MSS或TM图像为代表，后者可以SPOT图像为代表。

此类成像系统的几何机制可近似地看作为单线式中心投影（图2－2）。与一般中心投影不同的是，扫描成像并不生成连续图像，而是数字式离散图像，只在遥感平台的航迹线上，可近似地看作正射投影。基于这一特征，在每条扫描线上，几何畸变量随地物目标偏离航迹线距离的增加而增加，畸变幅度的大小可定量地表达为：

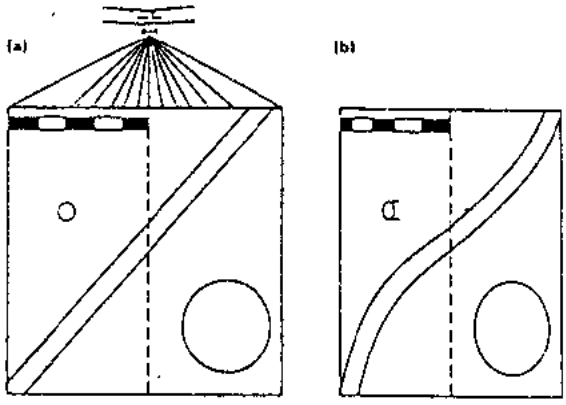


图2－2 机载全景扫描成像造成的边缘压缩现象（据DRURY，1987）

$$D = [1 - \frac{i \cdot \mathrm{tg}\theta}{\mathrm{tg}(i \cdot \theta)}] \times 100\%$$

式中 θ 为瞬时扫描角。

由于全景扫描成像过程中的采样是等时或等角进行的，所以往往造成正切畸变。从上式中不难看出，畸变量是瞬时扫描角的函数。随着瞬时扫描角的增大，畸变量急剧增长（图 2—3）。表 2—4 清楚地表明，不同全景扫描成像系统引起的系统几何畸变量，这些数据充分说明了该类图像的几何特征及系统几何校正的必要性。

为使该类图像能够作为 GIS 中的一元与其它信息进行复合处理，一个重要的先决条件就是要解决几何定位的配准问题。换句话说，就是要将由于上述几何特征所决定的系统畸变首先予以清除，以使其能够在共同的定位精度上作进一步处理。下一章将详细讨论有关校正的细节。

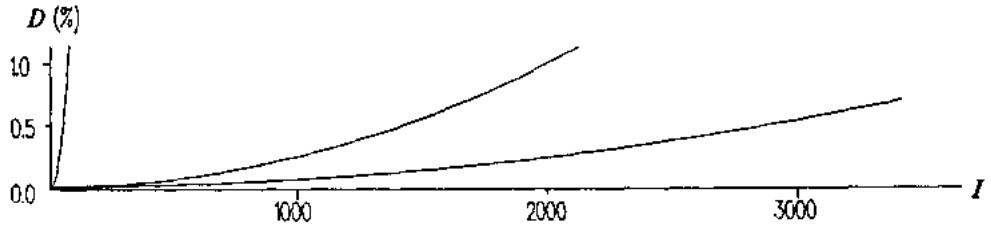


图 2—3　不同瞬时扫描角 θ 下几何畸变量 D 随像元坐标 i 变化曲线
D—几何畸变量；i—扫描线上像元坐标

表 2—4　几种典型全景扫描图像系统几何畸变特征对比表

成像系统	Landast		BENDIX	DS—1230
	MSS	TM	M^2S	
瞬时扫描角 θ（毫弧度）	0.086	0.042	2.5	2.5
每行像元总数	3240	6967	804	560
偏离航迹线 100 列处畸变量（%）	0.002	0.0008	2.09	2.09
最边部畸变量（%）	0.647	0.715	36.18	16.89

注：θ 为瞬时扫描角（据 COLWELL 1983，SCHROEDER 1978）

（三）　侧视成像

所谓侧视成像系指成像系统位于地物目标的侧上方，而不是像中心投影或全景扫描成像那样位于其正上方。目前采用这一成像方式的系统主要是指雷达成像，包括早期的机载侧视雷达RAR（SLAR）（真实孔径雷达）和应用较多的合成孔径雷达（SAR）。限于篇幅，本节只讨论二者的共性——侧视成像特征（至于二者的不同，亦可从图 2—4 中予以简要说明）。成像系统记录到的是回波所需要时间。由此引起一个重要区别就在于顶底位移的倒置（与中心投影的位移情况正好相反）以及其它特殊几何定位问题。现分述如下：

1. 地面平距与斜距的非线性关系

$$R_s = R_g \cdot \sin\Phi$$

式中 R_s 为斜距，R_g 为地面平距，Φ 为入射角，其中入射角 Φ 的确定取决于成像系统本身。

$$\Phi = \cos^{-1}\frac{H}{R}$$

式中 H 为航高，R 为成像系统到地物目标的斜距。而对于星载雷达来讲：

$$\Phi = 180^\circ - \Omega - \beta$$

式中 β 为坡角大小，以度为单位：

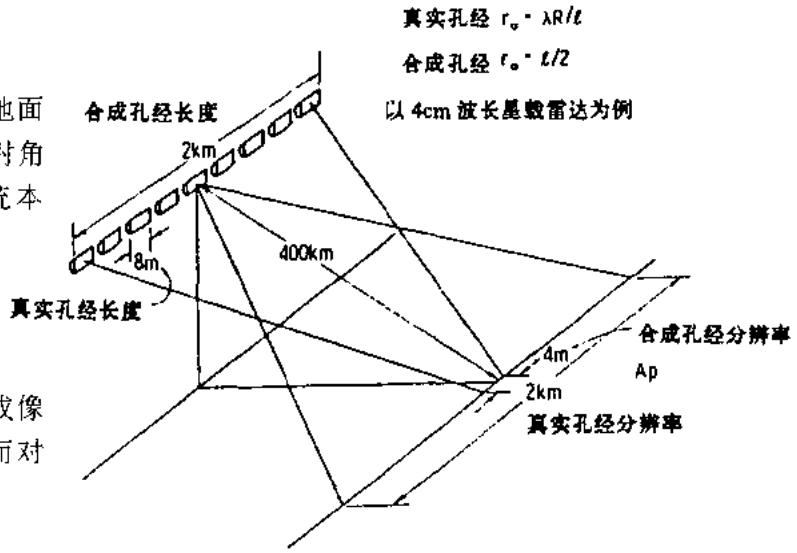


图 2－4　真实孔径雷达与合成孔径雷达成像方式示意图（据 ULABY 等，1981）

$$\Omega = \cos^{-1}\frac{R + H + h - 2R_e(H - h)}{2R(R_e + h)}$$

式中 R：雷达到地物的斜距；

R_e：地球半径；

H：航高；

h：地面高程；

2. 航向比例尺与横向比例尺相互独立

雷达发射信号到达地物目标所需的传播时间可化为斜距，此量可作为雷达图像横向比例尺的基本参数。在航向上由于航速明显低于微波脉冲的传播速度，且因系统而异，所以其航向比例尺，在理论上取决于航速。上述二者明显不统一，从而使雷达图像无固定的二维几何关系来保证其地物目标的重现。

3. 透视缩短也是雷达成像中的特殊几何问题，同样影响到地物目标在雷达图像上的几何定位，主要表现为朝向雷达天线的斜坡相对于背向天线的斜坡，所得到的记录要短。

一般来讲可记为：

$$R_s = l \cdot \sin(\theta \pm \beta)$$

式中 R_s 为斜距，l 为实际坡长，β 为坡角，θ 为俯视角余角。只有当 $\beta=0$ 时，即平坦地面，才满足

$$R_s = l \cdot \sin\theta$$

从这个意义上讲，斜距、平距的非线性关系只是坡角为 0 时的一种特例。

从上式中可以看出，θ 越大，记录到的坡长越长；另一方面，坡角越大，记录到的坡长越短。在坡角不变的情况下，缩短量可记为：

$$F_p = (1 - \sin\Phi) \cdot 100\%$$

式中 F_p 为透射缩短，Φ 为入射角，几种特殊入射角情况下的缩短量已列入表 2－5 中。表中 90°和 0°代表两种极端情况，前者为水平入射，后者为正交入射；理论上虽不多见，但在实际上，随地物坡向、坡角的变化完全可能出现上述极端情况。

表 2－5 透视缩短量和入射角对应关系一览表

入射角（Φ）(°)	缩短量（F_p）(%)
90	0.0
80	1.5
70	6.0
60	13.4
50	234
40	35.7
30	50.0
20	65.0
10	22.6
0	100.0

4. 顶底位移（layover）

有的书上将此项称之为雷达叠掩。其基本特征是在地形起伏较大时，有可能出现山顶的回波先于山底的回波到达接收机而被记录下来的情况，结果在图像上造成烟囱等高大地物目标向着飞机航迹线侧倾斜，这种情况正好与中心投影的向外倾斜相反（图 2－1）。其关系式可记为：

$$R_s = l \cdot \sin(\theta - \beta)$$

即只有在 $\beta > \theta$ 的情况下才会出现上述顶底位移的情况。而当 $\theta = \beta$ 时，顶底重合，在图像上只能记录到一个点，顶底合一。

5. 雷达阴影

阴影出现在背向天线且坡角 β 大于俯角 γ 的情况下，一般而言，阴影长度 s 是坡高 h 和俯角 γ 的函数。

$$s = h \cdot \mathrm{ctg}\gamma$$

虽然阴影部分无回波，无法看到地物的回波特征，但阴影本身却可以增加影像上地物目标的立体感从而提高其可解译度。阴影的长度是地势起伏和斜坡坡度的函数。

所有上述几何特征皆与地形起伏密切相关，所以在实际图像处理过程中往往是要以数字化高程模型（DTM）为先决条件。

三、 遥感地质信息的图像特征

遥感地质信息的诸多优势中有一个其它任何手段无法替代的特长，就是可以在可见光以外成像，为人们采集到大量肉眼看不到的信息，而这些信息又是对地质体的研究至关重要的。正因如此，在不断研制、改善成像系统的几何特性的同时，对其光谱成像的研究也在日益深化。现在不仅可以在电磁波谱中的几乎每个波段成像，从可见光到红外，到微波皆研制成功了成熟的成像系统。而且考虑到地物目标对不同波长电磁波的响应程度的可分性，对成像光谱的细分能力也在不断增强。早期投入地球资源调查的陆地卫星含 4 个波段，分别为 4、5、6、7 分别对应于 0.5μm－0.6μm、0.6μm－0.7μm、0.7μm－0.8μm、0.8μm－1.1μm 四个波段。每个波段的跨度皆在几个纳米以上，就其波谱分辨率而言并不很高，后来的TM图像已有所改进。到 1991 年 4—5 月间的第八届遥感地质专题会上，已有 24 通道、64 通道、乃至 244 通道的成像光谱仪投入使用，并取得了可喜成果。从而展现出遥感技术发展的另一特征——提高成像系统的波谱分辨率。本书不打算研究遥感图像的解译或处理，

而着重要讨论的是如何将遥感信息作为地理信息系统中的重要成员来利用。特别是如何使用海量但及时的遥感数据来更新原有地理信息系统中过时的信息，如河道的变迁，公路的改道，新建筑物的崛起等等。尽管这里主要涉及到的是几何定位问题，但不同信息来源的遥感数据的适用性问题，仍是一个不容忽视的大问题。人们无法从同一地物目标的数百个通道中直接选取其所需要的信息段，在实际上只有可数几个波段才是用户最需要的或是最适合的；另一方面，不同成像系统所得到同一波段的地物信息并不是直接可比的。迄今为止大量成像系统所记录的波谱特征皆是未经标定的，地表波谱测试结果和不同成像系统对同一地物即便在同一时刻也无法得到相同的测试结果。从这个意义上，遥感数据的可比性也是一个值得注意的问题。

SABINS（1978）曾将衡量图像特征的主要指标概括为四个：比例尺、灰度值、反差及分辨率。

（一） 比例尺

图像比例尺主要是指系统成像时所生成图像的比例尺大小。对航空摄影图像来讲，就是航高与焦距之比；对非摄影成像的图像来讲，则取决于其地面分辨率大小，该比例尺只能决定图像的最大比例尺，而不是用户使用图像的实际比例尺。无论是摄影图像或非摄影图像，比例尺都只是一个概略的指标，供用户根据不同的研究目标选择图像时参考。另外，图像不同位置的比例尺各不相同，也应引起注意。如雷达航向比例尺与横向比例尺。以 CV 580 雷达图像为例，其航向比例尺为 1：14500，而横向比例尺为 1：13300。

（二） 灰度值

图像上每个像元的灰度值直接取决于成像系统的成像方式和工作波段，如全色摄影得到的黑白照片和红外摄影得到的黑白照片，图像上所看的相同黑白程度所代表的地物目标可能有很大差异；多波段摄影所得到的地物目标的各波段影像也会迥然不同。至于多光谱扫描成像生成的不同波图像上，同一地物目标的灰度差异，则更为明显。正是这种差异，为我们识别地物目标提供了有力的工具。

值得指出的是，上述各例中灰度尽管千差万别，但有一个共同点：都是地物目标在以太阳能为光源的成像系统下生成的图像。灰度的差异可简单理解为地物目标对不同波长电磁波的反射率的差异。换句话说，这些差异本身确实与地物目标之间的差异密切相关。但对主动成像系统生成的雷达图像来说，其灰度值的含义就要另当别论。雷达成像的“光源”并非太阳光，而是由雷达发射装置发出的微波波段的电磁波。雷达图像上记录到的灰度差异，实质上是雷达回波所需时间的差异。尽管其灰度值与地物目标的组成有一定关系，但决不能在两者之间画等号。直接影响雷达图像灰度值变化的是后向散射系数，即从地物表面散射的电磁波中为雷达天线所接收到的那一部分的大小。据 ULABY（1981），后向散射系数 σ_0 可表达为一系列变量的函数：

$$\sigma_0 = f(\lambda, \varphi, \rho, \beta, \varepsilon, \Gamma_1, \Gamma_2, V)$$

式中 λ：波长；

φ：入射角；

ρ：极化方式；

β：俯视角；

ε：复介电常数；

Γ_1：表面粗糙度；

Γ_2：次表面粗糙度；

V：体散射系数。

作者在这里着重要强调的是图像上的灰度值差异，并非全都和地物目标的组成成分成必然联系，在把雷达图像用于多元信息复合时尤其应该注意。

（三） 反差

图像的反差直接影响到图像上地物目标的可识别程度。一般来讲反差大者易于识别。影响图像反差大小的主要因素为成像系统本身的波谱分辨力，即每个工作波段的频带宽度，频带窄者易于生成反差明显的图像，反之，则容易造成图像模糊不清。另一个影响反差的因素就是图像的后期制作，如图像的放大、洗印、扫描图像的回放与后期洗印等。据此不难看出，改善图像反差的途径有两条，一是成像系统本身波谱分辨能力的提高，包括通道的增加和频带的变窄；另一个就是后期制作时的反差增强，既可以采用光学手段，也可以采用图像处理中的反差增强手段来实现，其中包括以指定研究目标为对象的局部反差增强。

（四） 分辨率

图像分辨率的大小因系统而异。一般摄影图像，以图像上的单位间隔中的线对数为单位，其大小取决于系统（相机）的分辨能力及图像的比例尺；对数字扫描方式生成的多光谱图像而言，则主要取决于系统的分辨能力，即瞬时扫描视域(IFOV:Instantaneous Field Of View)的大小，如LANDSAT-TM图像的分辨率为30×30m，而航空多光谱扫描图像的分辨率可达 5×5m。至于合成孔径雷达图像的分辨率，有一个与其他遥感成像系统截然不同的地方：其分辨率与波长无关，而完全取决于雷达天线的孔径。合成孔径雷达(SAR)图像的几何分辨率为天线实际长度一半，具同样长度天线的真实孔径雷达的分辨率就低的多(图 2—4)。

四、 数据格式

无论遥感信息来自哪一种成像系统，在真正投入使用之前，必须首先了解其数据结构，并以此作为预处理的基本参数。这里所讲数据格式，除要区分模拟与数字化两大类之外，还必须深入讨论其中的有关细节。

（一） 模拟图像类

模拟图像类指用实际得到并投入使用的遥感信息是以照片为载体的，与其原始成像方式或加工处理方式虽然无直接联系，但必然要受其原始成像系统的有关参数的制约。考虑到后期处理的需要，在收集原始图像资料的同时应尽量获取原始成像参数，如航高、焦距及成像时间等等。

此类图像一般以二维平面图像方式存在，其格式可根据以下参数予以区分：

1. 规格，指照片几何尺寸。

2. 灰度分为黑白和彩色。黑白还要分为全色还是红外；彩色又分为真彩色与假彩色。

3. 成像方式，虽无直接记录但都是必须收集的资料，否则将无法对其进行必不可少的预处理。

（二） 数字图像类

数字图像的载体可以为磁盘或磁带，其中又以磁带为主。磁带上存放的数字化数据格式又因成像系统的不同而不同，比较通用的格式包括以下几点：

1. 数据记录类型：不同成像系统有其特定的记录类型，如多光谱扫描图像，可以是按波段存储，也可以按扫描行来存储，还有的是以像元方式存储。常见的 LANDSAT-TM 图像是以波段方式存储，即每个波段为一个独立完整的图像文件，而微机配用的 TH-1200 扫描仪生成的则是以像元方式存储的图像文件。每个像元按红、绿、蓝的顺序存放三个分量的灰度值。再如 SAR 580 的数据中分为四类：

- 高分辨率窄轨双通道（HR－NS）
- 高分辨率宽轨双单通道（HR－WS）
- 低分辨率窄轨双通道（LR－NS）
- 低分辨率宽轨单通道（LR－WS）

显而易见，各类数据有着各不相同的几何分辨率和航带宽度。即使是没有上述细分数据类型的成像系统，也有其固定的分辨率和覆盖区的宽度，或分帧的几何尺寸，所有这些都将直接影响到数据的处理。包括图像的初次回放。

2. 逻辑卷的划分

有些遥感数据的存储可能涉及不止一盘磁带。为数据采集的准确可靠，往往在采集之前就已确定了逻辑卷的划分方式，每一卷一般对应于一盘计算机兼容磁带。

3. 卷头文件的组织格式

一个卷头带往往有一个卷头说明文件，可包括以下几项：

- 卷内存放文件目录表，其中第一卷（第一盘带）中的该文件上还有整卷各文件的指针说明及本记录以及本卷内文件指针。
- 文件头说明文件，主要用于说明定位记录、属性记录及相关辅助记录等，仅用于每卷的卷首带中。
- 图像文件，该文件紧随文件头说明文件之后，逐行存放相关遥感数据。
- 卷尾文件，说明一个逻辑卷到此为止。

一般的卷头含 2 个头文件，卷尾带含一个卷文件，其间存放的才是有效图像文件。

卷头文件的读取，对了解整卷文件的各种参数是至关重要的。特别应指出的是，每个参数在文件中的位置，所占字节数及所用的数据格式都不是固定的，而且是互不相同的，有的占字节数一样，但格式有的是 8 进制，有的是 16 进制，只有真正了解其中的每个参数，才能真正按已有格式读取到所需要的图像——那怕还只是未经预处理的原始数据。

上述数据的记录类型、逻辑卷的划分及卷首文件的组织格式，只是就一般情况而言，对某一个特定成像系统或图像处理系统来说，其输出的图像文件皆有具体的格式说明。在遥

感图像处理中常常碰到的另一个与之相关的问题是，来自不同系统的图像数据文件的格式各异。如从I²S输出的图像文件格式就与ARIES－III输出的图像格式不同。即使是同一图像处理系统不同命令下生成的图像文件，其格式也不相同。

至关重要的是要查明与数据存放有关的几个参数的存放位置、所占字节数及数据类型，然后才能恰到好处地找到并正确读出所需的数据。

（三） 其它类

这里主要是指以各种数据文件方式存储的遥感数据，包括从其他图像处理系统转来的遥感数据文件，事先查明其结构，或记录格式直接关系到能否正确回放所需图像，有时一个数据的误差也含造成图像的错位。

如ERDAS对图像文件的存放与ARIES－III系统中对遥感文件的存放格式不一样，后者要求以512字节的整数倍求出偏移量。

即使是通过计算公式得到的数据文件也必须与生成图像所要求的源数据格式匹配，比如是一字节整型还是二字节整型；数据是256分级还是32分级等等。

第三章　遥感地质信息的提取

正如本书第一章中提到的那样，遥感地质信息已成为地理信息系统中的重要一元，而且在数据的更新方面，起着其它手段无法取代的作用。目前影响这一高新技术充分发挥其优势的主要障碍，就在于不同类型的遥感、非遥感信息之间在数据类型、格式、结构等方面互不兼容，以至难以相互沟通。鉴于地理信息系统的标准化、规范化尚需时间，至少在一个相当长的时期内，信息之间的沟通，还是要有赖于根据特定信息系统的特定要求，恰当地提取遥感地质信息。本章着重介绍遥感地质信息的提取方法，以期为信息的沟通建造一座“桥梁”，提供一些实用型接口。这里主要涉及以多元信息复合应用为目标的几何校正、灰度调整以及最基本的专题信息提取方法。关于数据格式的转换将放在下一章讨论。

一、　遥感图像的几何校正

在第二章第一节中，已详细讨论了三大类成像几何，即中心投影、全景扫描和侧视成像的基本特征。本节中要强调说明的是，无论那种成像方式皆存在两大类几何畸变，即系统几何畸变和随机几何畸变，现分述如下：

（一）　系统几何畸变及其校正

系统几何畸变系指由成像系统自身的成像几何产生的畸变，如中心投影方式下的辐射状几何畸变、全景扫描的正切畸变和侧视成像图像中的特殊畸变。由于此类畸变的空间分布并不是随机的，常规图像处理系统中提供的控制点校正法无法消除此类畸变。但由于此类畸变尚有规律可循，故可通过编制特定程序来进行有针对性地校正，以使原图像中地物目标的定位接近于GIS所需要的某种几何定位方式，如UTM坐标系。

比较而言，航空摄影照片的正射投影转换技术已相对成熟，从过去的光学转绘法已发展为微机控制的软件转绘法，有关软硬件也已得到相应的发展。从60年代的转绘仪，到80年代的正射投影转换系统，皆是比较驰名的商业化产品。究其原理，同样是在给定若干已知参数的情况下，如航高、焦距等，根据像素在图像中所处的位置、高程，计算出其畸变量，然后调整后重新定位从而得到新的校正图像。其处理过程可概括为：成像参数输入——像素在原图像上定位——计算畸变量——校正——重新组成新图像。其中的畸变量的计算既取决于成像方式，又决定于具体成像参数。

下面要讨论的全景扫描图像系统几何畸变校正程序同样是基于上述处理流程。校正中像元定位数据的转换，是基于该像元处的几何畸变量的大小来进行补偿。作者用FORTRAN语言编写、调试的程序主要以目前常见的几类全景扫描图像，包括LANDSAT－MSS、LAND－TM及航空多光谱扫描图像为校正对像。主菜单提供的就是这几种选项，由于LANDSAT－MSS及LANDSAT－TM图像有关参数可视为已知，故无需用户再输入参数；只对航空多光谱扫描图像，用户必须输入瞬时扫描角的大小，然后用户需要输入的是起始行列坐标及终止行列坐标，程序即可自动实施系统几何畸变校正操作，生成一个经过校正的新图像。

在具体算法上，主要是用下式计算各点处的系统畸变量并做相应的调整，灰度值的赋

值在程序中采用邻近点法。

经多项课题使用结果表明，此程序简便易行，方便用户，特别是对航空多光谱扫描图像的校正，可大大改善原图像的几何定位精度，为随机几何畸变的校正提供基础资料。由于严重的边缘压缩现像（图 2－2），大部分航空多光谱扫描图像的边缘部分，地物目标面目全非，以至无法找出用于随机几何校正的地面控制点，经上述校正程序处理后，地物目标的相对位置可得到明显改善（图版 1－2）。

```
C * * * PROGRAM WJ.FOR FOR GEOM. CORRECTION OF PANORAMIC SCANNED IMAGES
      IMPLICIT BYTE(B),DOUBLE PRECISION(D),INTEGER * 4(E)
      IMPLICIT INTEGER * 2(F－Q),REAL(R－S)
      CHARACTER W
      COMMON/NO9LUN/LUN
      DIMENSION BBB(26),BDVV(11),BDV(11),BFO(9),BFI(9),
     *   BFLOGI(11),BFNAME(9),BTNAME(6),IOST(4),BUF2(8000),BUF1(8000)
      DATA BFO/9 * ' '/
      DATA BFI/9 * ' '/

C   * * * * * INITIATE IMAGE PROCESSING SYSTEM   * * * * *
      CALL XOINIT(XXXXXX,IER,BBB)
      CALL XXINIT(XXXXXX,IER)
      CALL XKDFIN(IMG0,XXXXXX,BTS,BTU,BCO,BCU)
      CALL XKDISK('B',XXXXXX,IER,BDV)
      CALL XKDLNK(BDV,XXXXXX,IER,YYYYYY,LUN)
      CALL XBSPRO('N','N','Y','N','N','N','N',XXXXXX,IER,YYYYYYY,1,3)

C   * * * * * INPUT PARAMETERS OF IMAGE TO BE PROCESSED   * * * * *
      CALL XOMESG(IER,10,9,BA)
      CALL XOMESG(IER,11,9,BA)
      CALL XOMESG(IER,12,9,BA)
      CALL XOMESG(IER,13,9,BA)
      CALL XOMESG(IER,14,9,BA)
      CALL XOMESG(IER,15,9,BA)
      CALL XOMESG(IER,16,9,BA)
      CALL XOMESG(IER,17,9,BA)
      CALL XOMESG(IER,18,9,BA)
      CALL XOMESG(IER,19,9,BA)
  501 CALL XOMESG(IER,6,1,BB)
      IF(BB.EQ.4) GOTO 444
  555 CALL XOMESG(IER,1,9,BFI)
      CALL XFFSEE(BDV,BFI,'I',1,XXXXXX,IER,END,EAD,BFL,BFN,BFTY)
      IF(IER.NE.－261) GOTO 777
      PRINT * ,'       FILE NOT FOUND, REENTER FILE NAME'
      GOTO 555
  777 CALL XOMESG(IER,3,9,BFO)
      IF(IER.EQ.－273) GOTO 555
      IF(IER.EQ.－273) GOTO 777
      CALL XFFSEE(BDV,BFO,'I',1,XXXXXX,IER,END,EAD,BFL,BFN,BFTY)
      IF((IER.EQ.－261).AND.(BB.EQ.3)) GOTO 111
```

```
      IF((IER.EQ.-261).AND.(BB.NE.3)) GOTO 888
      PRINT*,'        FILE EXISTS, REENTER FILE NAME'
      GOTO 777
  111 CALL XOMESG(IER,4,4,ANG)
      IF(IER.EQ.-273) GOTO 501
      IF(IER.EQ.-273) GOTO 111
  888 CALL XOMESG(IER,9,1,BW)
      IF((BW.EQ.'n').OR.(BW.EQ.'N')) GOTO 444

C     * * * * * SEARCH THE OLD IMAGE AND READ ITS DIMENSION    * * * * *
      CALL XFFSEE(BDV,BFI,'I',1,XXXXXX,IER,EMDATA,EADATA,BFLOGI,
     *   BFNAME,BFTYPE)
      CALL XFFOPN(BDV,BFI,'I',1,XXXXXX,IER,EMDATA,EADATA,BFLOGI,
     *   BFNAME,BFTYPE)
      CALL XFRDIM(BDV,1,BFI,XXXXXX,IER,EAL,EAP,EIL,EIP,NOC)
      CALL XFWDIM(BDV,1,BFI,EAL,EAP,EIL,EIP,1,XXXXXX,IER)

C     * * * * * CALCULATE PARAMETERS USED FOR CORRECTION    * * * * *
      KH=EAP-(EAP/512)*512
      RS=FLOAT(KH)/512.
      EEL=EIL+EAL
      EEP=EIP+EAP
      SFI=0.0
      IF(BB.EQ.1) THEN
      ANG=0.042/1000
      SFI=SFI+0.05679
      EM0=3619
      END IF
      IF(BB.EQ.2) THEN
      ANG=0.083/1000
      EM0=1602
      END IF
      IF(BB.EQ.3) THEN
      EM0=EAP/2
      ANG=ANG/1000
      END IF
      EMA=EM0-AINT((EIL-1)*TAN(SFI)+0.5)
      EMB=EM0-AINT((EEL-1)*TAN(SFI)+0.5)
      EP1=EMB-EIP
      A1=ANG*ABS(EP1)
      R1=TAN(A1)/(ABS(EP1)*TAN(ANG))
      ES1=AINT(ABS(EP1)*R1+0.5)
      EX1=ES1-ABS(EP1)
      EP2=EMA-EEP
      A2=ANG*ABS(EP2)
      R2=TAN(A2)/(ABS(EP2)*TAN(ANG))
      ES2=AINT(ABS(EP2)*R1+0.5)
      EX2=ES2-ABS(EP2)
      IF(ES1.GT.ES2) EX0=EX1
      IF(ES1.LE.ES2) EX0=EX2
      EMS=EMA+EX0
```

```
      EMT=EMB+EX0
      IF(EIP.GE.EMB) EPA=EMT+ES1
      IF(EIP.LE.EMB) EPA=EMT-ES1+1
      IF(EEP.GE.EMA) EPB=EMS+ES2
      IF(EEP.LE.EMA) EPB=EMS-ES2
      EPP=EPB-EPA+1
      ELL=EAL
      EK=AINT(FLOAT(EPP)/512-0.002)
      EEK=(EK+1)*512
      EUC=ELL*(EK+1)

C     * * * * * CREATE A NEW IMAGE * * * * *
      CALL XFFCRE(BDV,BFO,'I',EUC,XXXXXX,IER,EEE)
      CALL XFRDIM(BDV,1,BFO,XXXXXX,IER,ELL,EEK,EFL,EFP,1)
      EFL=EFL+EIL
      EFP=EFP+EPA
      CALL XFWDIM(BDV,1,BFO,ELL,EEK,EFL,EFP,1,XXXXXX,IER)

C     * * * * * PERFORMING THE CORRECTION LINE BY LINE  * * * * *
      DO 100 II=EIL,EEL
      IK=II-EIL+1
      RT=RS*FLOAT(IK-1)
      RA=(RT-INT(RT))*512
      KB=2*INT(RA)+3
      EKA=II-EIL+1
      ERP=1
      EMI=EM0-AINT((II-1)*TAN(SFI)+0.5)
      IA=IABS(EMI-EEP)
      IB=IABS(EMI-EIP)
      IF(IA.GE.IB) EPC=IA
      IF(IA.LT.IB) EPC=IB
      RD=TAN(EPC*ANG)/(EPC*TAN(ANG))
      EPD=AINT(EPC*RD+0.5)
      SE=EPC*ANG/EPD
      CALL XAWRLI(LUN,EADATA,EAP,EKA,ERP,EAP,1,XXXXXX,IER,IOF1,BUF1)
      ELO=II
      EPO=1
      EMM=EMI+EX0
      DO 102 JJ=EPA,EPB
      EPJ=EMM-JJ
      EPS=IABS(EPJ)
      IF(EPS.NE.0) THEN
      AD=EPS*SE
      RC=EPS*TAN(SE)/TAN(EPS*SE)
      JE=AINT(EPJ*RC+0.5)
      JA=EMI-JE-EIP
      ELSE
      JA=EMI
      END IF
      JB=JJ-EPA+1
      BUF2(JB-IOF2)=BUF1(JA-IOF1+KB)
```

```
102   CONTINUE
      DO 103 JK=EPP+1,EEK
      BUF2(JK-IOF2)=2
103   CONTINUE
      CALL XAWWLI(LUN,EEE,EEK,ELO,EPO,EEK,IOF2,BUF2,1,XXXXXX,IER)
100   CONTINUE

C   * * * * * EXIT TO OIS   * * * * *
      CALL XFFCLO(BDV,BF0,'I',XXXXXX,IER,NOP)
      CALL XFFCLO(BDV,BF1,'I',XXXXXX,IER,NOO)
444   CALL XOEXIT
      END
```

由于上述程序是在MICRO－VAX机上进行的，程序中调用了图像处理系统ARIES－III的一系列现有模块。现首先按其在程序中出现的次序作一简单介绍，然后再具体说明程序中的主要变量及运行使用情况。需要指出的是，该程序中用到的模块在以后的程序中也有类似调用，所用同名模块以后均不再重复说明。

· XOINIT——用于建立已有交互操作服务环境（OIS）与用户命令的联系；

· XXINIT——为系统接口，用于 ARIES 任务的初始化，无需用户输入，也无输出；

· XKDFIN——为系统配置接口，返回缺省逻辑盘名；

· XKDISK——用于找到缺省盘；

· XKDLNK——用于将逻辑盘与系统盘相联接，输出指定图像盘的设备号；

· XBSPRO——用于选择处理器，所有实用程序皆必须调用此模块，代码“3”代表CPU。

以上六个模块可理解为软硬件环境的设置。下面用到的是文件处理模块：

· XFFSEE——用于查询指定文件是否存在；

· XFFOPN——用于打开一个指定文件；

· XFRDIM——用于读取指定图像文件的基本参数，含总行数、总列数、起始行、起始列等；

· XFWDIM——用于将生成图像文件的基本参数写盘，但并不存储图像文件本身。写入的参数包括图像的总行数、总列数、起始行、起始列等。上列数据可以是前一段程序生成的数据，或由其他模块输出的数据，也可以是用户直接输入的数据，只要输入数值的顺序、类型与设置变量的顺序、类型相一致即可；

· XFFCRE——用于生成一个新文件，要求输入新生成的图像文件名，系统约定为 9 个字符；同时要输入图像的大小（即所占“块数”，每 512 字节为一块，不满 512 字节的仍按 512 字节计算）；

· XFFCLS——用于文件的关闭。

至于文件中实际数据的操作，程序主要使用以下两模块：

· XAWRLI——即“读行”命令，每次从图像文件中读取一行图像数据信息，并存放于缓冲区内，供进一步调用；

· XAWWLI——为“写行”命令，能将用户生成的一行图像数据写入新生成的图像文件中的指定行，有关每行的总列数等参数，皆需在XFWDIM模块中事先确定。

以上两模块中皆用到偏移量(IOF)，其初始值为每行图像实有列数与该列数中512的整数倍之差，并随着在图像中所处行数的变化而变化。该偏移量虽不是常数，但其变化仍有规律可循。程序中的变量KH即为此而设。

除上述模块外，程序中还用到一系列与图像几何校正有关的变量，在这里也作一简要说明：

BFI——为9个字符的输入图像文件名；

BFO——为9个字符的输出图像文件名；

EAL——为图像中实有行数；

EAP——为图像中实有列数；

EIL——为图像的起始行号；

EIP——为图像的起始列号；

EEL——为图像的终止行号；

EEP——为图像的终止列号；

ANG——为扫描系统的瞬时扫描角；

EMO——为图像中航迹线所在的列数，对应于无系统几何畸变的那一类。

程序中所有的XOMESG模块皆为说明语句，既可用于提示，也可用于菜单显示。其具体内容可调用OME命令进行编辑。其优点是编辑方便、容量大，且节省程序空间；但其内容多为隐含，对程序的可读性有一定影响。如上述程序开头部分有11行的XOMESG语句，隐含的就是一个主菜单，其中包括四个选项：1—TM图像；2—MSS图像；3—航空多光谱扫描图像；4—退出。程序要求在MICRO－VAX的ARIES－III环境下运行，并可与其他已有的图像处理命令并行调用，或纳入批处理。运行中几乎不需要用户干预，只需输入待校正图像文件名、已校正图像文件名，再按菜单选择图像类型即可。考虑到用户要校正处理的图像可能只是原图像的一部分，程序中特意为用户留下了一个数据接口：选择整幅图像还是只对部分图像校正。对后者，要求用户指定其要校正的那一部分图像的起始行、列数和要校正的总行、列数。这样，用户只需选定处理区域即可，而不必实际上对图像进行“切块”操作。

（二） 随机几何畸变及其校正

随机几何畸变系指由于成像系统成像时平台姿态不稳或是载体运行速度、方向的瞬间变化产生的几何定位精度误差。由于此类畸变往往随机出现，一般皆服从正态分布，常用的校正处理原则是选取若干地面控制点，找出该点处地面坐标与图像之间的对应关系，建立线性方程组并解出多项式系数，进而生成一个包含多个系数的转换方程。该转换方程的次数一般应根据地形变化程度来决定。用此方程对原图像实施重采样，即可得到新生成的校正图像。采用此方法校正的优点是简单易行，照已有程序操作即可，但由于地面控制点的选取在校正中起到决定性作用，一般不易奏效，特别是在原图像系统几何畸变严重的情况下。另一个不容忽视的因素是，由于地形图（参照系）的更新周期较长，遥感图像上反映的地物变化较快，往往难以找到足够数量的匹配点，或是图像上清晰可见，地形图上不见踪影，或者完全相反，还有时能找到的对应点不能均匀分布于全区，这就给转换方程的建立带来诸多不变，有时直接影响校正效果，作为弥补手段目前常用方法有：

1. 基于野外调查的地面控制点选取与定位

在诸如撒哈拉大沙漠那样的无图地区，几乎找不到可用于与遥感图像进行地面控制点匹配的地形图。在这种情况下，往往要根据图像上的特征点到实地进行调查确认，进而借助测量手段来查明该点的地理坐标。

2. 基于GPS（全球定位系统，下同）技术的地面控制点选取与定位

GPS技术是近年来发展起来的卫星测量技术。到80年代末，投入使用卫星已达21个。只要使用接收系统即可在野外任意位置定出该点地理坐标，现在定位精度可达5m左右。该精度完全可以满足1：10000比例尺图件上几何定位的精度要求，前提是在实地找到图像上用作控制点的地物目标，如水系与道路交叉点等，控制点的选取原则仍为易于识别，且分布均匀。

二、 遥感图像的辐射校正与波段的选取

在图像的实际处理过程中，几何校正与辐射校正可以并行，但波段的选取则必须尽早实施，以便确定要在哪几个波段上实施几何校正和辐射校正。波段选取问题在上一章的第二节已经结合波谱特征问题进行了初步论述。这里着重讨论波段选取的基本原则与几种常用方法。

经济、快速、实效应成为整个图像处理的基本原则。面对数以十计及至百计的波谱通道，数以兆计及至十兆计的数据量．只有当我们恰当地为研究目标选取所需的波段时，才能使图像处理操作真正做到经济、快速、实效。既要能解决问题，又要尽量减少数据处理的工作量，二者不可偏废。为此，就必须在波段的选取上确定出一些行之有效的指标，一个具体可靠的标准就是待分地物目标或是研究对像的统计可分性。正如第二章所说的，可分性是在经过系统灰度校正的基础上才有效，否则，就会造成待分目标相混现象。

（一） 系统辐射畸变及其校正

同图像的几何畸变相类似，图像的灰度畸变也有系统畸变与随机畸变之分。所谓系统畸变，主要指成像系统本身造成的且与地物目标本身无关的一种灰度的系统变化。此类畸变往往又与波段密切相关，以南德某地航空多光谱扫描图像为例．在11个波段图像中皆有垂直航线方向的灰度畸变，且总体呈两侧偏高中部偏低的趋势，但变化幅度却因波段而异。图3—1清楚表明，在第8波段，扫描线的第1280列处最低，在第5波段同样，但幅度明显不同。由于此类畸变与空间位置明显相关，所以不能采用常规的全图像直方图均衡法进行处理，而要采用统计计算方法先求出总体变化幅度及其分布情况，即要求出该波段图像的辐射畸变函数，然后进行有针对性的校正。

一个实用处理程序可采用以下步骤：

步骤1：将原图像按垂直扫描线的方向均分为若干份，相当于将原图像分为几十个等分细条带；

步骤2：分别统计每一个带内的灰度平均值，忽略其中的地物目标之间的差异，即将图像内所有地物分布皆视为随机分布的；

步骤3：勾绘出扫描方向上某一特定波段图像的灰度变化曲线，即辐射畸变函数；

步骤4：确定该段图像的灰度水准总体平均值，并进而确定每一条带的校正系数；

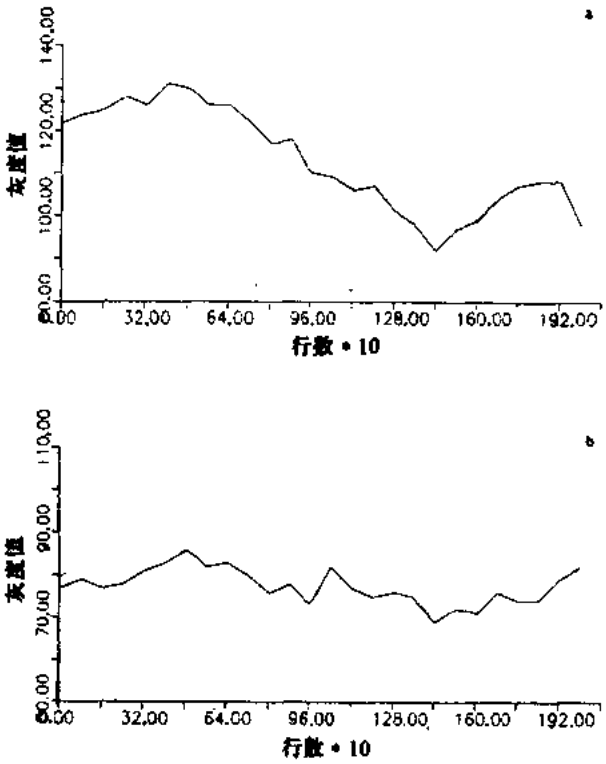


图 3－1　南德某市航空多光谱扫描图像辐射畸变曲线

a：8 波段；　　b：5 波段

步骤 5：用上述系数对每一条带进行计算并给每个像元赋以新值。

基于上述算法即可得到一幅经过灰度校正的新图像，以此作为针对研究目标进行可分性统计的采样的基础图像。

（二）　波段的选取

本节的讨论以提高图像上地物目标的可分性为主要目标，而不是泛泛地讨论图像信息量的大小等。这里首先涉及到的是波段的选择问题。目前在实用遥感图像处理中常用的波段选择方法有：

1. 依据理论值选取：LANDSAT－TM 图像各波段的功能各有侧重，如 2 波段适用于区分岩石类型和热液蚀变带，3 波段适用于区分作物类型等。但在实际上，对不同时期、不同地区的图像而言，这种理论上的选择往往难以尽如人意，有时甚至会造成混乱。从这个意义上，此法可作为研究可分性的基础资料，但绝不能当作基本依据，用以选择波段或波段组合。

2. 目视检测法：凭直觉从各个波段的回放图像中挑选易于区分待研究地物目标的波段。此法优点是易于操作，简单易行；问题是肉眼的辨别能力有限，当波段较多或回放图像效果不佳时往往会带来不理想的结果。

3. 统计可分性检测

所谓统计可分性，系指以待区分地物目标为样本，在目视效果较好的几个波段上，选取若干小试验区，主要依据是实地勘查结果或已有资料，然后分别统计出不同波段图像上，不同待分地物目标的灰度平均值及其标准离差，从而得到不同地物目标的灰度值变化情况。在波段不多的情况下，可用图示法表示。一般情况下则可用偏差值来衡量其可分性，并以此量化指标来作为选取波段的依据。

值得注意的另一个问题就是原图像上的灰度畸变——一般为系统性畸变。这一畸变的存在——无论其程度和分布方式如何，都将直接影响到波段的正确选择，即使是统计可分性也无法将本来应属两类的地物目标进行区分。在这种情况下，进行辐射校正就成为必不可少的一步。由于此类畸变的系统性，一般可使用专门程序一次性予以校正，详见第四章第二节。

研究目标的统计可分性实为波段选取的一项重要指标，一般可通过采样统计法获得。具

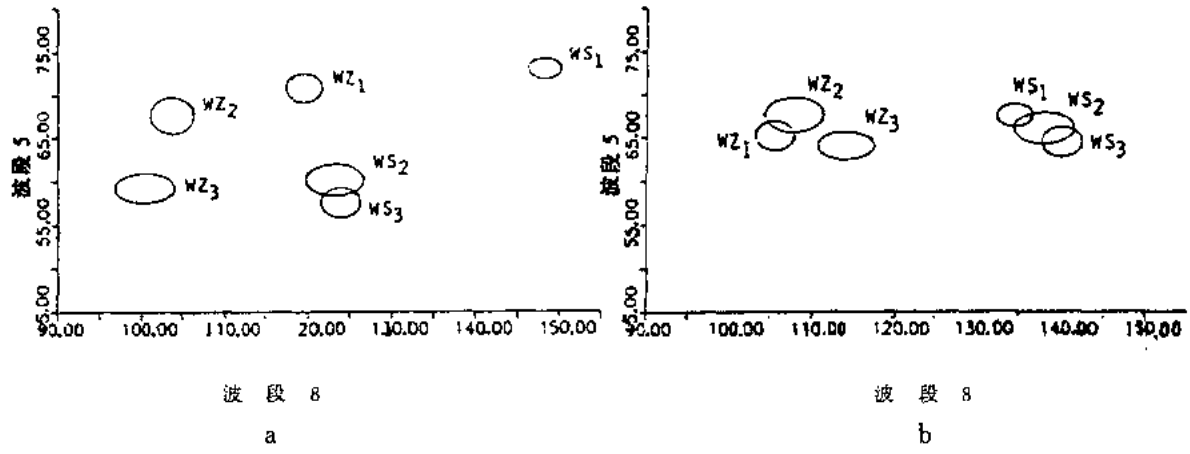


图 3－2　幅射校正前（a）后（b）多瑙尔兴恩地区 5、8 波段机载多光谱扫描图像，牧草（WS）和小麦（WZ）灰度值数值椭圆图

体实施步骤与监督分类法的前期操作类似。对每一类待分研究目标先选取若干试验区，统计其在所有各波段中的平均值和标准离差等基本参数，然后再借助绘制散点图之类的常规手段，定性或定量地确定出不同波段图像中待分研究目标的可分性。这样确定的可分性完全是针对特定研究课题中的某些特定目标而言的。只有在充分研究待分地物目标统计可分性的基础上，才有可能从众多的可利用波段中，选取尽可能少却又能切实区分识别待分研究对像的波段。作者在 1988 年的实验研究中从 11 个波段中选取 2 个波段，即取得了较为理想的分类效果——区分出 11 种不同的农作物。前人的试验结果也表明，分类结果并不是与波段个数成正比，同时使用 12 个波段的分类结果并不比使用 8 个波段的分类结果好。所以波段的选择对遥感地质信息的提取无疑是至关重要的。

三、　基础影像图的制备

传统的遥感地质解译往往是在航卫片上先进行解释，然后再附上相应图件，或转绘为正式图件。随着GIS技术的迅猛发展，特别是GIS中数据更新的需要，就图解译的方法已无法适应新形势的需要。无论是遥感图像中原始数据的采集，还是专题成果的提取皆要求与一定比例尺的地形图相匹配，并将其作为GIS中的一个数据层来看待。基于这种思想，作者等在1991年即提出基础影像图的概念与制作方法。

所谓基础影像图系指按国际分幅标准制备的、经过几何精校正的遥感影像图。这里有两条具体标准：一是按相应比例尺的国际分幅标准制备，二是几何精校正。几何精校正可通过本章第一节的各种校正手段予以实现；按国际分幅标准制备遥感图像则有必要专门讨论，因为涉及的因素比较多。

首先，就是多帧图像间的镶嵌问题。由于遥感平台生成原图像的分幅标准不可能与地形图国际分幅标准一致，在一个标准图幅范围内就难免会遇到一幅基础影像图涉及多帧原图像问题。此外，考虑到各帧图像几何精校正所用转换文件不一致，所以很难在相邻边界处确保同名点取得相同坐标。有鉴于此，作者经反复试验提出以下处理流程，实践证明可取得良好的校正效果。

1. 找出每幅基础影像图上四个顶点的对应位置（此时已可能涉及多帧图像）

2. 如已涉及一帧以上图像时，应先将未做校正的原图像先拼在一起。由于相邻卫星轨道之间往往有一定重叠部分，一般不至引起太大偏差。在极端情况下，可在重叠区内先做图像对图像的几何粗校正，以使同一图幅内各帧图像能够准确地拼在一起；

3. 原遥感图像的选取应略大于图幅边界，以便于相邻图幅之间的配准，并尽量选取共同地面控制点；

4. 对整幅图像用第一节提出的办法，实施几何精校正，建立转换文件（校正用）；

5. 重采样生成所需图像。

经过上述流程得到的基础影像图可与相应比例尺的国际分幅地形图进行配准，也就是说在此基础图上提取的任何信息，皆可与相应比例尺的其它信息进行配准或者说具有相应比例尺条件下的一一对应关系。以此图为基础解译得到的水系图、地貌图等图件或是从中提取的各波段灰度信息皆可作为GIS中的数据层直接利用，这对各类与地理信息相关的图件的更新无疑是大有裨益的。

四、 专题信息的提取

为方便下边的讨论，这里有必要先介绍几个与专题遥感信息的提取密切相关的术语，以便清楚地了解其内涵：

1. 可解译性（Interpretability）

可解译性指特定地物目标的实际检出能力。这是用户最为关心的指标，取决于原图像的几何分辨力、检出能力等综合参数，有时还取决于波段的选择与组合方式。

2. 检出能力（Detectability）

检出能力是成像系统记录一个地物目标的能力。它并不完全取决于成像系统分辨能力，即使一个地物目标小于成像系统的几何分辨能力，也可能被检测出来（指在图像上有所显示）。

3. 分辨率（Resolution）

分辨率系指一幅图像上显示出有差别并能加以区分的两个地物目标间的最小距离。它完全是系统成像能力的标志。

4. 可分性（Separatability）

可分性主要指特定地物目标从图像上实际检出能力，是一项通过统计方法得到的、以具体研究目标为对像的定量指标。一般可通过采样统计不同波段图像上某种目标的可分程度来确定。作者曾在1988、1990年著文强调可分性的重要性，它是图像的几何、波谱特征的综合指标。

在基础影像图的基础上，可以根据研究课题的需要提取所需要的专题信息，常用方法无外乎两大类。

（一） 目视解释法

目视解释的方法可分为三大类：类比法、演绎法和解译标志综合分析法。据TOWNSHEND（1981），其中的前两者直接与解译者的经验密切相关。第三种方法比较常用，有关细节几乎可在任何一本遥感教科书中查到，这里无需赘述。

在这里只强调一点，为保证解译成果在GIS中的可利用性，应尽可能在基础影像图上进行解译，至少是在经过几何精校正的遥感图像上进行。至于解译结果的格式转换问题，将在第四章专门讨论。因为几何精校正只解决了几何定位问题而并不能解决数据的格式问题。

（二） 自动分类法

遥感图像处理中的自动分类方法主要分为监督分类和非监督分类两种。常用的监督分类方法又分为：

1. 平行六面体法；
2. 最大似然率法；
3. 最小距离法。

以上三种分类方法皆是建立在统计学基础之上的。其分类结果的好坏在很大程度上取决于样品的代表性。为确保分类成果能合理的应用于多元信息复合处理，一个大前提就是该分类结果有足够的可靠性，为此，就必需在实施复合处理之前进行分类精度评价。

判断分类成果好坏的唯一标准就是地面实况，因而，要检验自动分类结果的可靠性，必需对分类成果进行实地验证。对大面积的分类成果来说，简捷迅速而又行之有效的验证方法，就是统计抽样检验法。根据HORD和BROONER(1976)提出的计算公式，在信度为95%时的分类精度 μ 可表达为：

$$N(X-2\mu+\mu^2)<(1.96)^2\mu(1-\mu)$$

式中：X——平均精度；

N——参加统计的样点个数；

μ——求得的分类精度。

其中平均精度 X 由经过验证的各类分类结果的精度平均所得。

五、 遥感图像中原数据的提取

对遥感信息的提取，除上述几种常规办法之外，还应强调原数据的提取方法，特别是在GIS之中，可将不同波段的遥感信息，按指定的坐标位置提取，将所提取的不同波段的遥感数据当作GIS中的一个数据层。按要提取数据的空间分布情况，可分为以下几种情况：

- 任意点处的（可换算为地理坐标的对应点）；
- 任意方向的线上的，主要用于剖线分析；
- 任意一个区域的，主要指由空间坐标确定的某个区域内的每一像素的、每一波段的灰度值。

上述方法提取的遥感图像的原数据主要是用于传统图像处理方法以外的各种数学变换，如数学地质方法的应用等。正如近期发表的文献资料表明的那样，数学地质方法的引入本身也是遥感地质技术发展的一大发展趋势，所以提供这样一个数据接口是十分必要的。

作者用FORTRAN语言编写，并在MICROVAX－II上调试的程序WC即具有上述功能。用户只要输入准备从中提取数据的图像文件名之后，系统自动提取原图像的基本参数，如起始行、起始列与总行数、总列数。用户可以从中选取任一行、任一列或给定指定区域的起始行列坐标及输出列，行数即可自动生成一个数据文件，供用户进一步调用。

值得一提的是，为了尽量节省存储空间和减少传输时间，生成数据一般用字节型数据（一位整型）存储。因为不同波段的遥感图像属全灰度图像，几乎每个相邻的灰度值皆不相同，可供数据压缩的余地不大，所以程序中未使用压缩编码算法。

虽然上述程序是在MICROVAX机上调试运行的，但稍加改动即可在微机上运行。其主要问题是文件调用和指定行列或区域地址的计算。唯一要注意的是原图像文件中数据的存放格式，如在MICRVAX－II上，以512字节为一块(BLOCK)这一因素务必考虑在内，否则将在数据的提取上造成混乱。程序中的以下几个参数即为此而设计：

KH——为每行图像列数与512整数倍之差，即为列偏移量；

KB——如起始行不是实际图像的第一行，从该行起就有一个行偏移量。

由此程序生成的数据文件已将此偏移量移去，全部按顺序方式存放，以便于数据的传输。

考虑到遥感图像处理系统之间的差异，在不同机型之间实施数据调用传输时，还必须有相应的异步通信软件，如用于VAX与微机之间的KERMIT之类的软件。作为一种传输工具，无需做过多的解释，只要求用户了解传输两端数据格式的差异。特别是对多波段数据的处理，要综合考虑空间、时间的消耗，选取合适的数据存储格式。

下列程序中的图形显示部分是用在VAX机上运行的RGL绘图软件编制的，并要求在VT－240终端上运行的。现仅就程序中涉及到的几个子程序作如下说明：

· INIT _ GRAPHICS () —— 用于图形方式的初始化，设置所有图形属性的缺省值；

· CLEAR _ SCREEN —— 清除屏幕上的图形实体；

· CLEAR _ TEXT —— 清除屏幕上所有正规文字；

· SET _ WINDOW —— 定义图形显示窗口；

· DRAW _ GRAPHPAPER —— 设置画数据的图形纸的类型；

· LABEL _ NUMAXIS —— 为图形的轴线单元刻度赋值，还可以给出轴线的标题字；

· SET _ COLOR —— 设置屏幕颜色；

· PLOT _ POINT —— 在已存的图形上增加单个数据点，并将这些点画在图形上；

程序运行后可在屏幕上显示出指定遥感图像中任意剖线方向的灰度值起伏变化情况，并能在CALCOMP绘图仪上输出。

```
C   * * * * * PROGRAM WC.FOR READS DN FROM AN IMAGE AND    * * * * *
C
      IMPLICIT BYTE(B),DOUBLE PRECISION(D),INTEGER * 4(E)
      IMPLICIT INTEGER * 2(F－Q),REAL(R－S)
      CHARACTER FO * 4,W,V,WX,WA,WN
      COMMON/NO1BUF/BUF1(7000)
      COMMON/NO9LUN/LUN
      DIMENSION BBB(26),BDVV(11),BDV(11),EQ(3000),EP(200,1500),
     *   BFLOGI(11),BFNAME(9),BTNAME(6),IOST(4),BFO(4),BFI(9)
      DIMENSION X(3000),Y(3000),ND(10),SI(200)
```

```
      CHARACTER * 4 DAFN,XD(10)
      DATA BFO/4 * ' '/
      DATA BFI/9 * ' '/
C
C  * * * * * INITIATES THE SYSTEM AND IMAGE DISK
C
      CALL XOINIT(XXXXXX,IER,BBB)
      CALL XXINIT(XXXXXX,IER)
      CALL XKDFIN(IMG0,XXXXXX,BTS,BTU,BCO,BCU)
      CALL XKDISK('B',XXXXXX,IER,BDV)
      CALL XKDLNK(BDV,XXXXXX,IER,YYYYYY,LUN)
      CALL XBSPRO('N','N','Y','N','N','N','N',XXXXXX,IER,YYYYYYY,1,3)
C
C  * * * * * DISPLAY THE MAIN MENU     * * * * * * * * *
C
      PRINT * ,'      * * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
      PRINT * ,'      *                                                       *'
      PRINT * ,'      *     THIS PROGRAM DUMPS DN FROM AN IMAGE AND           *'
      PRINT * ,'      *          DISPLAYS A PROFILE ON VT240                  *'
      PRINT * ,'      *                                                       *'
      PRINT * ,'      *          BY ANALYSIS AND TEST CENTER                  *'
      PRINT * ,'      *     CHINA UNIVERSITY OF MINING & TECHNOLOGY           *'
      PRINT * ,'      *                                                       *'
      PRINT * ,'      * * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
C
C  * * * * * OPEN THE OLD IMAGE FILE   * * * * *
C
  555 CALL XOMESG(IER,1,9,BFI)
      CALL XOMESG(IER,7,1,W)
      IF(IER.EQ. -273) GOTO 555
      IF(W.EQ.'N') GOTO 909
      CALL XFFSEE(BDV,BFI,'I',1,XXXXXX,IER,EMDATA,EADATA,BFLOGI,
     *  BFNAME,BFTYPE)
      IF(IER.EQ. -261) GOTO 555
  124 CALL XFFOPN(BDV,BFI,'I',1,XXXXXX,IER,EMDATA,EADATA,BFLOGI,
     *  BFNAME,BFTYPE)
      CALL XFRDIM(BDV,1,BFI,XXXXXX,IER,EAL,EAP,EIL,EIP,NOC)
      CALL XFWDIM(BDV,1,BFI,EAL,EAP,EIL,EIP,1,XXXXXX,IER)
      ELA=EIL+EAL-1
      EPA=EIP+EAP-1
C
C  * * * * * MAIN MENU   * * * * *
C
  201 PRINT *
      PRINT * ,'      * * * * * * * * * * MAIN  MENU * * * * * * * * *'
      PRINT * ,'      *                                                 *'
      PRINT * ,'      *       1. DUMP GREY VALUES                       *'
      PRINT * ,'      *       2. DISPLAY A PROFILE                      *'
      PRINT * ,'      *       3. EXIT                                   *'
      PRINT * ,'      *                                                 *'
      PRINT * ,'      * * * * * * * * * * * * * * * * * * * * * * * * * *'
```

```
      PRINT *
  301 CALL XOMESG(IER,2,1,I1)
      IF(IER.EQ.-273) GOTO 301
      IF(I1.EQ.3) GOTO 909
      IF(I1.EQ.2) THEN
      CALL XOMESG(IER,3,1,B3)
      IF((B3.EQ.'y').OR.(B3.EQ.'Y')) GOTO 103
      ENDIF
      PRINT *
C
C  * * * * * READ AND DUMP GRAY VALUES MENU  * * * * * *
C
      PRINT *
      IF(I1.NE.2) THEN
      PRINT *,'       * * * * * * * *  DUMP  MENU  * * * * * * * * *'
      PRINT *,'       *                                              *'
      PRINT *,'       *       0. M LINES * N COLUMNS                 *'
      GOTO 125
      ENDIF
      PRINT *,'       * * * * * * * *  PROFILE  MENU  * * * * * * * *'
     -PRINT *,'       *                                              *'
  125 PRINT *,'       *       1. A DIAGONAL LINE                     *'
      PRINT *,'       *       2. A RANDOM LINE                       *'
      PRINT *,'       *       3. A RANDOM COLUMN                     *'
      PRINT *,'       *       4. BAR BY BAR                          *'
      PRINT *,'       *       5. RETURN TO MAIN MENU                 *'
      PRINT *,'       *                                              *'
      PRINT *,'       * * * * * * * * * * * * * * * * * * * * * * * *'
      PRINT *
  401 CALL XOMESG(IER,4,1,I2)
      IF(IER.EQ.-273) GOTO 401
      IF(I2.EQ.5) GOTO 201
C
C  * * * * * INPUT PARAMETERS FOR DUMP THE GRAY VALUES  * * * * *
C
      IF(I2.EQ.4) THEN
  701 CALL XOMESG(IER,5,4,EIC)
      IF(IER.EQ.-273) GOTO 701
      NC=EIC
      ESP=EIP
      ESL=EIL
      EZP=EPA
      GOTO 801
      ENDIF

  815 IF(I2.EQ.2) GOTO 805
  801 PRINT *,'INPUT START LINE NUMBER'
      GOTO 806
  805 PRINT *,'       INPUT THE RANDOM LINE NUMBER'
  806 PRINT *,'       YOUR NUMBER MUST GREATER THAN',EIL
      PRINT *,'       AND LESS THAN',ELA
```

```
      IF(I2.EQ.2) GOTO 816
851   CALL XOMESG(IER,6,4,ESL)
      GOTO 827
816   CALL XOMESG(IER,10,4,ESL)
      IF(IER.EQ.-273) GOTO 851
      IF(IER.EQ.-273) GOTO 816
827   IF((ESL.LT.EIL).OR.(ESL.GT.ELA)) THEN
      PRINT*,'      BEYOND NUMBER RANGE, PLEASE REENTER! '
      GOTO 815
      ENDIF

      IF(I2.EQ.2) THEN
      EZL=ESL
      GOTO 803
      ENDIF

802   PRINT*,'      INPUT END LINE NUMBER'
      PRINT*,'      YOUR NUMBER MUST GREATER THAN',EIL
      PRINT*,'      AND LESS THAN',ELA
852   CALL XOMESG(IER,8,4,EZL)
      IF(IER.EQ.-273) GOTO 852
      IF((EZL.LT.EIL).OR.(EZL.GT.ELA)) THEN
      PRINT*,'      BEYOND NUMBER RANGE, PLEASE REENTER! '
      GOTO 802
      ENDIF

      IF(I2.EQ.1) THEN
      PRINT*,'      PLEASE STAND BY! '
      GOTO 811
      ENDIF

858   IF(I2.EQ.3) GOTO 807
803   PRINT*,'      INPUT START PIXEL NUMBER'
      GOTO 808
807   PRINT*,'      INPUT THE RANDOM COLUMN NUMBER'
808   PRINT*,'      YOUR NUMBER MUST GREATER THAN',EIP
      PRINT*,'      AND LESS THAN',EPA
      IF(I2.EQ.3) GOTO 857
853   CALL XOMESG(IER,9,4,ESP)
      GOTO 867
857   CALL XOMESG(IER,11,4,ESP)
      IF(IER.EQ.-273) GOTO 853
      IF(IER.EQ.-273) GOTO 857
867   IF((ESP.LT.EIP).OR.(ESP.GT.EPA)) THEN
      PRINT*,'      BEYOND NUMBER RANGE, PLEASE REENTER! '
      GOTO 858
      ENDIF

      IF(I2.EQ.3) THEN
      EZP=ESP
      GOTO 810
```

```
      ENDIF

804   PRINT * ,'       INPUT END PIXEL NUMBER'
      PRINT * ,'       YOUR NUMBER MUST GREATER THAN',EIP
      PRINT * ,'       AND LESS THAN',EPA
854   CALL XOMESG(IER,12,4,EZP)
      IF(IER.EQ.-273) GOTO 854
      IF((EZP.LT.EIP).OR.(EZP.GT.EPA)) THEN
      PRINT * ,'       BEYOND NUMBER RANGE, PLEASE REENTER! '
      GOTO 804
      ENDIF

810   IF(I2.NE.0) THEN
817   CALL XOMESG(IER,13,1,B4)
      IF(IER.EQ.-273) GOTO 817
      END IF

811   IF(I2.EQ.4) THEN
      EL1=EZL-ESL+1
      EP1=EAP
      N1=EAP/NC
      ENDIF

      EL1=EZL-ESL+1
      EP1=EZP-ESP+1
      RA=EL1**2+EP1**2
      IF((I2.NE.0).AND.(I2.NE.4)) EK=INT(SQRT(RA))
      IF(I2.EQ.0) EK=EL1*EP1

      IF(I2.EQ.4) THEN
      DO 106 KK=1,NC
106   SI(KK)=0
      DO 107 I1=1,EL1
      EKA=ESL-EIL+I1
      ERA=1
      DO 107 J1=1,EP1
      CALL XAWRLI(LUN,EADATA,EAP,EKA,ERA,EAP,1,XXXXXX,IER,IOF1,BUF1)
      JK=ESP-EIP+J1
      EP(I1,J1)=BUF1(JK+IOF1-1)
      IF(EP(I1,J1).GE.0) EP(I1,J1)=EP(I1,J1)
      IF(EP(I1,J1).LT.0) EP(I1,J1)=EP(I1,J1)+256
      NI=AINT((J1-0.005)/N1)+1
      SI(NI)=SI(NI)+REAL(EP(I1,J1))
107   CONTINUE
      PRINT * ,'NI=',NI

      N3=EAP-N1*(N1-1)

      DO 111 I=1,NI-1
111   Y(I)=SI(I)/(N1*EL1)
      Y(NI)=SI(NI)/(N3*EL1)
```

```
C   PRINT *,(Y(J),J=1,NI)
        NN=NI
        GOTO 331
        ENDIF

        DO 101 JJ=1,EK

        IF(I2.NE.0) THEN
        J=AINT(JJ*EP1/EK-0.005)+ESP-EIP+1
        IF((B4.EQ.'N').OR.(B4.EQ.'n')) I=ESL-EIL+AINT(JJ*EL1/EK-0.005)+1
        IF((B4.EQ.'Y').OR.(B4.EQ.'y')) I=EZL-EIL-AINT(JJ*EL1/EK-0.005)+1
        ENDIF

        IF(I2.EQ.0) THEN
        I=ESL-EIL+AINT((JJ-0.005)/EP1+1)
        I1=I+EIL-ESL
        J=ESP-EIP+JJ-(I1-1)*EP1
        J1=J+EIP-ESP
        ENDIF

        EKA=I
        ERA=1
        CALL XAWRLI(LUN,EADATA,EAP,EKA,ERA,EAP,1,XXXXXX,IER,IOF1,BUF1)

        IF(I2.EQ.0) THEN
        EP(I1,J1)=BUF1(J+IOF1-1)
        IF(EP(I1,J1).GE.0) EP(I1,J1)=EP(I1,J1)
        IF(EP(I1,J1).LT.0) EP(I1,J1)=EP(I1,J1)+256
        ENDIF

        EQ(JJ)=BUF1(J+IOF1-1)
        IF(EQ(JJ).GE.0) Y(JJ)=REAL(EQ(JJ))
        IF(EQ(JJ).LT.0) Y(JJ)=REAL(EQ(JJ)+256)
 101    CONTINUE

        IF(I2.NE.0) THEN
        PRINT *,(Y(J),J=1,EK)
        ELSE
        PRINT *,((EP(K,L),L=1,EP1),K=1,EL1)
        ENDIF

 812    IF(I1.EQ.2) GOTO 331

 833    CALL XOMESG(IER,14,1,B5)
        IF(IER.EQ.-273) GOTO 833
        IF((B5.NE.'y').AND.(B5.NE.'Y')) GOTO 201

C   * * * * * CREATES A NEW DATA FILE

 834    CALL XOMESG(IER,15,4,BF1)
        IF(IER.EQ.-273) GOTO 834
```

```
      OPEN(UNIT=2,FILE=BF1,TYPE='NEW')
      PRINT *,'NEW DATA FILE CREATED'
      IF(I1.EQ.1) THEN
      WRITE(2,*) ((EP(K,L),L=1,EP1),K=1,EL1)
      GOTO 201
      ENDIF
      IF(I1.NE.1) WRITE(2,*) (Y(J),J=1,EK)

  835 CALL XOMESG(IER,16,1,B6)
      IF(IER.EQ.-273) GOTO 835
      IF((B6.NE.'Y').AND.(B6.NE.'y')) GOTO 909
      IF((B6.EQ.'Y').OR.(B6.EQ.'y')) GOTO 201

  103 CALL XOMESG(IER,17,4,BF2)
      IF(IER.EQ.-273) GOTO 103
      OPEN(UNIT=2,FILE=BF2,TYPE='OLD')
  836 CALL XOMESG(IER,18,2,I8)
      IF(IER.EQ.-273) GOTO 836
      NN=I8
      READ(2,*) (Y(J),J=1,NN)

  331 IF(I2.EQ.1) NN=EK
      IF(I2.EQ.2) NN=EP1
      IF(I2.EQ.3) NN=EL1
      IF(I2.EQ.4) NN=NI
      DO 15 I=1,NN
   15 X(I)=I
C
C  * * * * * * *   DISPLAY PROFILE ON VT240   * * * * * * *
C
      XMI=10E15
      XMA=-10E15
      YMI=10E15
      YMA=-10E15
      DO 1 I=1,NN
      IF(XMI.GT.X(I)) XMI=X(I)
      IF(XMA.LT.X(I)) XMA=X(I)
      IF(YMI.GT.Y(I)) YMI=Y(I)
      IF(YMA.LT.Y(I)) YMA=Y(I)
    1 CONTINUE
      PRINT *
      PRINT *,XMI,XMA,YMI,YMA

      CALL INIT_GRAPHICS()
      CALL CLEAR_SCREEN
      CALL CLEAR_TEXT
      CALL SET_WINDOW(0.0,0.0,1000.0,625.0)

      CALL DRAW_GRAPHPAPER('LIN',10,5,'LIN',5,5,'WHITE')
      CALL LABEL_NUMAXIS('XB','PIXEL',0.,XMA,.TRUE.)
      CALL LABEL_NUMAXIS('YL','GREY VALUE',YMI,YMA,.TRUE.)
```

```
      CALL SET__COLOR('RED',2)

      DO 17 I=1,NN
   17 CALL PLOT__POINT(X(I),Y(I),'L',0)
C
C  * * * * * END THE PROGRAM  * * * * * *
C
      CLOSE(UNIT=2)
      CALL XFFCLO(BDV,BFI,'I',XXXXXX,IER,NOO)
  121 CALL XOMESG(IER,19,1,B7)
      IF(IER.EQ.-273) GOTO 121
C 121 PRINT *,'DO YOU WANT TO DISPLAY ANOTHER PROFILE? '
      IF(B7.EQ.'y'.OR.B7.EQ.'Y') THEN
      DO 224 K=1,EAP
  224 BUF1(K)=0
      GOTO 124
      ELSE
      GOTO 909
      ENDIF
  909 CALL XOEXIT
      STOP
      END
```

第四章　地质信息通用格式的转换

在浩如烟海的地质信息中，既有大量的遥感数据，又有种类繁多、数量可观的非遥感数据。遥感技术毕竟只是在60年代后期才真正发展起来的一门新学科，尽管从摄影角度来讲，其历史可追溯到百年以远。在地质资料馆、档案馆里存放的绝大多数资料皆来自非遥感手段，其中包括传统地质测量及采样测试结果等等。当然也包括地球物理、地球化学等方面的文字及图形资料，其中有的还是表格、图形方式记录的测试结果，除少数之外，其存储方式又都是非数字式的。作为几十年地质工作积累起来的实际资料，完全应该归并于现在的信息系统之中，并使之在地质信息的综合评价中发挥其应有的作用。那些在不同时期取自同一地区，同一孔位的的资料对我们尤为宝贵，它可以为我们提供来自地质体自身的更为翔实的信息，并成为更全面、客观认识地质体的一个重要信息源。当然这里强调的是实际资料本身，而不是各种演绎出来的推论。

由于遥感地质信息的主要特征及信息提取已经在上一章讨论过，本章着重介绍不同类型非遥感信息的基本特征，尔后讨论其通用格式的转换问题。

一、非遥感地质信息的基本特征

（一）数字型地质数据的基本特征

所谓数字型地质数据，是指其成果以数字方式存储的地质数据，如地震、测井、重力、磁法、电法、大地电测深等地球物理数据，以及与微机相连的各种现代化测试仪器所输出的数据等。现代化的大地测量手段如GPS等也存储数字型地质数据，但只是广义的地学类数据，而不是真正意义上的地质数据（应称为地理数据）。对GPS数据，本节不予讨论，感兴趣的读者可查阅相关的测量学方面的书籍。

1. 地球物理类

随着计算机技术的发展，目前绝大多数地球物理检测结果皆是以数字方式存储的，如地震、重力、磁法、电法、大地电测深、电测井等，而且皆附有配套的数据解释处理系统，用以向用户提供可资利用的数据。但地球物理手段所得到的毕竟只是有关地质体的间接信息，其解译分析结果往往因人而异，有着明显的多解性。在将此类数据纳入地质信息系统时，就必须在考虑其数据格式等因素的同时，格外注意其数据的实质内容，即收集那些更接近地质体实质内容的数据，予以处理。

2. 常用地质测试数据

一般来讲，测试数据都是根据样品采集位置来定位的，这里要讨论的是测试数据本身因测试手段不同、记录方式不同所带来的问题，而且主要是讨论自动输出结果以数字方式存盘的测试数据。不包括以表格方式抄记的那一类数据。

大约70年代末、80年代初开始，多数测试仪器皆已配上了专用计算机或是微机，测试结果皆可自动存盘，且至少可在系统内部随时调用。作为地质信息系统中的一元，有必要将其输入系统之内，为此就必须了解特定仪器生成的特定数据格式。

以显微图像处理系统为例，其图像数据格式是以2字节存放三个数据，其测试结果又

有专门的格式存放，有的还只能显示，不能存盘。要将该系统的分析成果调出，供综合分析之用，首先要做的工作就是对其数据格式的分析研究与转换。

再如热分析与 X 衍射仪皆配有微机，但自成系统。要解决其分析成果的调用问题，同样要先对其数据格式进行分析研究并实现与其它系统的转换。

所有这些测试仪器给出的都是单个样品的测试结果，如能通过适当的渠道使其按样品号及点位索引图使其并入数据库，便可在空间分析时作为一个数据层进行调用，并参与面上其它地质信息的综合评价。

（二） 非数字型地质数据的基本特征

地质调查中一切以非数字方式记录的地质数据，皆属于非数字类地质数据，其中包括原始的表格记录、文字记录、分析测试结果、各种编录图件、综合图件及野外素描、照片等。之所以在这里强调原始记录，主要是因为上述记录中已有相当一部分在不同的单位，以不同的方式转为数字式记录了，有的已建立了各种专题数据库，还有的将相关图件进行了数字化处理。只是由于不同单位使用的计算机系统不一，数据的格式有较大的差异，难以相互沟通。此外，非数字类原始地质记录中的大多数，仍处于非数字状态。根据其记录方式，可将非数字型地质数据分为：

- 文字图表类；
- 二维图形类；
- 局部数字化类。

为使此类数据能够在多元信息复合应用中发挥其应有的作用，一个首要的任务同样是格式转换。与数字类数据不同的是，对此类数据还必需进行统一格式的数字化，一方面将未数字化的原始记录输入信息处理系统，另一方面将局部数字化或已建数据库的那部分数据进行统一格式转换，以便于与同一信息系统中的其他来源的地质数据相互沟通。

非数字类地质记录的另一个特点就是缺乏统一的规范。这就意味着在真正实施数字化之前，还有大量的预处理工作要做，其中包括格式的设计、名词的统一，必要时甚至还必需建立统一的名词术语及代码词典。某些数据的归并也必须有一个统一的方案，比如，对化探数据的预处理就涉及到网格的划分、数据的算法等具体问题。有些具体参数，还必需经过大量实验才能确定。

二、 数据格式转换的基准

考虑到现有各类地质信息的千差万别以及实用软件的互不兼容，研究遥感与非遥感地质信息复合要解决的首要问题就是数据格式的转换，而且所有转换皆应以能使已有各类数据相互沟通为目的，从而保证各类数据能够在同一工作平台上实现互换，或者能直接在同一平台上实施数据的综合评价。要实现数据之间的相互沟通，必须满足以下两个基本条件：

- 统一的公用平台；
- 统一的 ASCII 数据格式。

（一） 公用平台的统一

数据格式转换引起的第一个问题是，在什么平台上实施数据格式的转换或综合评价。如

前所述，现有遥感非遥感地质信息的存放格式大致可分为数字式和非数字式两种，而信息的载体可能为表格、图纸或磁性介质。所谓数据互不兼容的主要障碍之一，就是不同介质、不同机型处理出来的数据格式差异较大。考虑到用户的原数据输入、处理以及输出大多以微机DOS系统为主，本书认为最通用的工作平台应为微机 DOS 系统。这样既可使数据的输入输出简便易行，又可确保数据格式的统一。

（二） 数据格式的统一

所谓格式的统一主要是指计算机数据处理时使用到的数据格式的统一。无论原数据以何种介质为载体，以何种格式存放，皆应使之可转为统一的数据格式。根据广大用户的实际需要，作者建议使用 ASCII 格式。这里强调的是要能在必要时转为 ASCII 格式。本书讨论的是格式的类型和转换接口，即一旦需要能做到将任何非 ASCII 格式的数据或非 DOS 平台格式的数据转为DOS平台ASCII格式数据。这里既包括了 DOS 系统下不同应用软件之间的数据格式转换，如GRAPHER、pcARC/INFO、dBASE等系列的数据格式转换，也包括了非 DOS 系统下生成的数据文件向 DOS 系统的转换，如 VAX 机与微机之间的数据转换。

三、 常用软件涉及的文件结构和数据格式

本节讨论 dBASE 系列数据库、Grapher 系列绘图软件、pcARC/INFO、地震勘探记录（SEG－Y 格式）和扫描输入数据 TIFF 的文件结构和数据格式，为地质信息的通用格式转换作准备。

（一） dBASE－FoxBASE－FoxPro 系列数据库

dBASE 系列数据库是目前微机上最通用的数据库；FoxBASE 和 FoxPro 系列是与 dBASE Ⅲ 兼容的新的数据库管理软件，是 dBASE 系列的改进型和增强型。从数据结构看，这三者是相同的，这里统称为 dBASE 系列数据库。作为数据库，dBASE 系列数据库内部功能强大，自洽性好，但从与外部数据交换性能看，则显不够方便。按常规作法，无论是将数据库中数据转换成外部数据文件，还是将外部数据文件中的数据添加到数据库中去，都必须进入数据库系统，且只能按固定的格式（非自由格式）输入或输出。这样，无论对于包含其它语言数据处理模块的大型数据库管理与处理系统，还是对于与数据库无关的用其它语言编写的数据处理软件，都会带来极大的不便。弄清了dBASE系列数据库的文件结构和数据格式．这些数据库文件将是透明的；这样，我们就能随心所欲地对数据库文件进行读写。

若将dBASE系列的数据文件（带 DBF 后缀，以下称 DBF 文件）中每 16 字节称为一行，则第一、二行包含了该数据库的总体信息，从第三行开始，每二行代表一个字段的信息。字段信息之后，为一 16 进制代码为 0D 的特殊字符，然后就是各个记录内容。每个记录的长度为所有字段长度之和加 1，这多出 1 个字节位于各记录的第 1 个字节，安放删除标志，即若该记录已在数据库中作逻辑删除，则安放“＊”标志，否则为空格。

下面我们先看 DBF 文件的总体信息，即第一、第二行。第一行的 1—4 字节由数据库内部定义，对于外部程序而言，无实质性内容，可以不管（对于类似的内容，下面不再赘

述)。5- 8字节为该文件的记录总数(包含已作逻辑删除的记录),它是一个4字节DOS系统整型数(第5字节为最低位,第8字节为最高位)。第9、10字节组成一个2字节整型数,代表记录内容起始字节数。如若第9、10字节内部码分别为c1 02,则记录起始字节为2×256+12×16+1=705。它位于第45行,前44行包含704个字节,如此计算705似乎应是45行的第1个字节,但因机器内部是从0字节开始计数的,故实际上应是45行的第2个字节(对我们习惯来说,每行从1开始计算字节)。上面已讲过,每个字段在文件头中包含2行信息,所以据此可推算出该DBF文件的字段数。设第9、10字节代表的整型数为f1,则字段数=(f1-1)/32-1,减去1是因第1、2行为文件总体信息,字段信息从第3行才开始。上例中,字段个数为(705-1)/32-1=21,即该文件有21个字段。第11、12字节为一个记录的总长度,前已讲过,它等于所有字段之和加1。至此,一个DBF文件的总体信息已经清楚。文件头的第2行无实质性内容。

对于每个字段来说,它亦有2行包含该字段的信息。每个字段信息的第1行中,第1—10字节为该字段名(dBASE系列数据库规定每个字段名不得大于10个字节)。第12字节为字段类型,为C、N、F、L、D、M、S等。其中C为字符型,N为数值型,F为浮点数型(与N型相同),L为逻辑型,D为日期型,M为备注型,S为屏幕型等。13、14字节为该字段在各记录中的起点。如上所述,各记录中第1字节为删除标志,所以第1个字段的起点肯定是2,(机器内部值为1);若第1个字段宽度为15,则第2个字段的起点为17(在机器内部为16),以此类推。第2行的第1字节为各字段宽,如果字段类型为数值型,则第2字节为小数点后的位数。例如,若某一字段信息的二行内部码为:

57 41 47 45 00 00 00 00 00 00 00 4E 12 00 00 00

07 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00

则该字段的名字为WAGE,N型(4E),由13-14字节为12 00可知该字段在每一记录中的起点字节为18(机器内部值);该字段宽为7,小数点后位数为2。

各个记录内容中,按上述字段顺序排列,其中日期型字段长度固定为8,逻辑型字段长度固定为1,备注型字段固定为10,其余字段长度由用户定义。若为备注型字段,则另有一备注文件存放备注内容,在DBF文件中这10个字节以字符形式组成一常数,指出该备注项在备注文件中的起点,以64字节为1单位。如若某一记录中的某一备注项的该10个字符中前8个都为0,第9个字符为"1"(内部码为31),第10个字节的字符为"9"(内部码为39),则在相应的备注文件中,该备注项内容的起点字节为19×64=1216,亦即从第76行开始(机器内部记数),从用户习惯计数,则为第77行。顺带指出,在备注文件中,前32行(8个64字节单位)无实际内容,第33行开始才是备注内容;在每一备注内容中,第5、6、7、8字节为备注内容实际长度(以字节为单位,与前不同的是第5字节为最高位,第8字节为最低位),第9个字节开始才是实际的备注内容。

通过上述分析,我们对DBF文件及备注文件有了全面的了解,据此就可编制相应的程序,直接对DBF文件进行读写。

(二) Grapher系列绘图软件的文件结构

这是指的Grapher系列软件为Grapher(作$x-y$曲线图)、Topo(作等值线图)和Surf(作三维立体图)软件系列的统称。该系列软件的输入文件主要有三种,即DAT、SYM和

GRD文件。DAT 文件无文件头，就是 DOS 下的文本文件格式，可直接用编辑软件或程序生成。下面阐述 SYM 和 GRD 文件。

1. SYM 文件组成一个字符集，供图件中标记或注释用，可用Grapher系列中的ALTERSYM生成。但若我们弄清楚了它的结构，可用程序将一些特殊字符（如汉字）自动转成SYM文件中的字符，供图件标注用。

SYM 文件中的数据都是占一个字节的整数，所以取值范围为－128—127；各数据中间无空格，呈紧密排列。SYM 文件的头三个字符为 SYM，以下为各字符的数据内容。

各字符数据的头 4 个字节分别是：字符代号、x 起始点、x 终点、点数。然后即是各点的数据，分别为 x 坐标、y 坐标、点性质（03 为开始点，02 为非开始点）。其中字符代号表明了该字符的调用方法，由用户指定；如若该值为 65（十进制），则在ASCII表中 65 为“A”，所以在调用该SYM文件时，在键盘上敲“A”即将该字符调入。x 终点减去 x 起点即为该字符的宽度，至于高度（y 起点，y 终点）不需指明，直接由点的 y 坐标指明。

2. GRD 文件

在 Grapher 系列软件中，GRD 文件本由 GRID 程序生成，为一网格化数据文件，供Topo或Surf使用。如我们用自己的程序生成网格化数据，亦只要写成 GRD 文件形式，即可用Topo或Surf绘制等值线图或三维立体图。

GRD 文件有两种格式，即文本格式和二进制格式，这里仅讨论文本形式文件。在 GRD 文件中，头 5 行为文件头，以后为网格中各排（固定 y 坐标）z（高度）的数据，各排中点的 x 坐标从小到大排列；第一排的 y 坐标为最小。若一排数据一行写不下，可换行续写；但每一排均需从新行开始写。数据之间用空格隔开。GRD 的第一行为顶格写的四个字符，若为 DSAA，则指该文件为文本文件，若为 DSBB，为二进制文件。GRD的第 2 到第 5 行为：

nx ny ——nx 为网格列数，ny 为网格行（排）数；

xlo xhi ——xlo 为 x 坐标最小值，xhi 为 x 坐标最大值；

ylo yhi ——yli 为 y 坐标最小值，yhi 为 y 坐标最大值；

zlo zhi ——zlo 为 z 的最小值，zhi 为 z 的最大值。

（三） pcARC/INFO 的基本结构与数据格式

美国资源环境研究所于 1988 年推出的 pcARC/INFO 软件以其强大的数据管理和图形处理功能而受到广大 GIS 用户的欢迎。其基本结构可分图形信息处理的ARC部分和属性信息管理的INFO部分。

1. 术语简介

为便于讨论 pcARC/INFO 的数据格式及其转换问题，在这里首先介绍该软件中的几个基本术语，特别强调其与别的图形处理软件的不同之处。

1）图形要素

同其它常见的图形处理软件一样，pcARC/INFO中最基本的图形要素亦为点、线、面。所不同的是pcARC/INFO中的点，既可以用来表示实际上的点要素(POINTS)，也可以当作标号(LABEL)来标识多边形(POLYGON)；线要素在pcARC/INFO中一律称之为ARC，无论是直线、折线、弧线还是矩形框或圆，每一条线要素皆包含一个起始结点(F __ NODE)和一个终止结点(T __ NODE)，介于二者之间的为一系列顶点(VERTEX)，线要素上的所有

各点皆由其坐标值来定义；至于面状要素，在pcARC/INFO中并不独立存在，即没有用于定义多边形的一组坐标值。pcARC/INFO中多边形是由与之相关的一组线要素及线要素之间的相互关系来定义的，相邻多边形的公用边(ARC)，在pcARC/NFO中只存储一组与之相关的坐标数据，在定义相邻多边形时，用到的只是相同的线要素标号。

2）拓扑关系

上面提到的“线要素之间的相互关系”实际上就是某种拓扑关系。严格地讲，所谓拓扑关系就是图形要素之间空间关系的量化或定量描述。比如线要素间的相连，线要素与线要素或多边形与多边形间的邻接，以及用一组线要素来定义一个多边形等。

3）配准点

配准点系指用以决定各图形要素的空间坐标关系的若干已知其坐标的控制点。一般要求尽可能均匀分布。除决定各图形要素的空间坐标之外，配准点还是相邻图幅间彼此匹配的控制点。

4）数字化层

在 pcARC/INFO 中，由具有一定拓扑关系的一系列图形要素组成的独立单元称之为数字化层(COVERAGE)。pcARC/INFO的大部分实际操作都是以数字化层为对像的。由于该独立单元的基本特征是所有信息的数字化，换句话说，不光其中的图形要素已数字化，而且各要素之间的拓扑关系也已数字化，所以它既不同于一般的图层，也不宜笼统地称之为信息层。为强调pcARC/INFO中该独立单元的数字化特征，本书中一律将其称之为数字化层。

5）主要处理功能

——数据的输入 ADS/ARCEDIT

——数据的编辑处理 ARCEDIT

--- 数据的空间操作 OVERLAY

——数据的输出 ARCPLOT/EDITPLOT

——属性数据的处理 TABLES

数据的转换 DATA CONVERSATION

众所周知，该软件以其结构严谨而著称，凭借图形及其属性之间的对应关系，可以任意调用处理用户存储的各类图形信息。与此同时，人们也注意到该软件提供的数据转换接口多以美国国内的相关部门，如地质调查局、人口调查局等的文件格式为准，而对其它格式的数据涉及甚少。除用于AutoCAD的DXF格式可直接与pcARC/INFO的数据进行交换之外，其它类型的数据就只有在满足用于栅格数据的GRIDPOLY命令及用于矢量数据类的GENERATE命令的格式要求的前提下，首先转为数字化层(COVERAGE)后，才能与pcARC/INFO已有数据进行交换、复合与处理。考虑到本书的重点在于研究信息复合应用中的数据转换问题，下面仅就pcARC/INFO为用户所提供的上述两个数据接口及相关命令进行讨论。

2. GRIDPOLY 命令对数据格式的要求

pcARC/INFO 的此项命令主要用于将特定格式的栅格式数据转变为数字化层。由于软件本身的限定，用户已有的栅格数据必须完全符合该命令对数据格式的各项规定，才能为pcARC/INFO所接受，否则，必须先进行格式转换。

该命令所能接受的数据格式主要有两种：即非压缩的和压缩的 ASCII 格式。二者皆用以处理整型栅格数据，并分别带有各不相同的固定格式的文件头及数据格式（图 4－1）。虽然这两个命令同样处理整型栅格数据，并同样生成数字化层，但对原数据的压缩量有明显差异，特别是对点阵汉字的处理，压缩的 ASCII 格式可将用于生成数字化层的栅格数据的存储空间压缩 50%甚至更多。

3. GENERATE 命令对数据格式的要求

对于图形处理中更常用到的实型矢量数据，pcARC/INFO 为用户提供的主要接口就是 GENERATE命令，该命令有 5 种选项，即可接受 5 种类型的数据（表 4－1）。从表中不难看出，在可接受的数据类型中，既包括点、线、圆等基本图形要素，也包括网格点数据甚至注记。与常见矢量数据不同的是，其格式中除各点坐标值之外，还要求加注结束标记“END”，对网格点型及注记型以外的数据，还必须加注各要素的序号，如表中线要素类的 101，102 等。

表 4－1　GENERATE 命令的 5 种选项及实例

线要素	点要素	圆	网格数据	注　记
101	301，2，4	202，4，3，3.5	2，3	2,1,90,15，注记　1
2,3	302，5，5.5	201，2.5,4.6,1.3	9，8	3,4,－90,15，注记　2
4，4	303，1，1	END	1.5,1	END
4,6	END		10,5	
102			END	
2，1				
3，1				
END				

（四）　TIFF 文件基本结构

所谓 TIFF 系 Tag Image File Format 的缩写。它代表一种常见的文件格式。该格式系由Aldus公司和Microsoft公司联合推出的扫描仪文件格式说明或约定。至 1988 年已推出 5.0版。1992 年又有 6.0 版。市面上有售的TH－1200扫描仪即可生成TIF文件。该文件兼顾了不同几何分辨率及不同成像方式的需要：即可用于生成二值图像、灰度图像或彩色图像。为便于讨论该类文件在遥感非遥感信息复合中的应用，下面先介绍其基本结构。

每个TIFF文件皆有一个文件头，共 8 个字节，其中的第 4—7 个字节指明了第一个图像文件的参数所在的位置。即该参数信息的偏移量，单位为字节。一般对单个图像的TIFF文件而言，该值为 8，即从第 9 个字节开始的两个字节为参数项个数信息，如此两字节为 0F00，即表示含 15 个参数项，每个参数项所占字节数固定为 12，其具体含义详见表 4－2。

现将表中各项参数的取值范围及其含义说明如下：

第一项：文件类型，为 4 字节整型数，缺省值为 0；

第二项：图像的宽度（列数），为 4 字节的整型数；

第三项：图像的高度（行数），为 4 字节的整型数；

第四项：表示每个像素所占的位数。对二值图像和灰度图像来说，其实际数值位于该参数段的最后4个字节中，一般为1个“08”；但对彩色图像而言，其最后4个字节的值，只是一个地址代码，按此代码可查明每个像素所占的位数，一般为3个“08”，表示红、绿、蓝各占8位；

第五项：说明图像是否经过压缩，缺省值为“1”，表示未经压缩；“2”表示对二值图像的哈夫曼行程压缩；“5”表示对彩色图像的LZW压缩，其中LZW表示Lempel－Ziv & Weleh算法；

表4—2　常用TIFF图像文件参数一览表

参数号	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	备　注
代　码	254	256	257	258	259	262	273	277	278	279	282	283	284	296	317	0—1字节
类　型	04	04	04	03	03	03	04	03	04	04	05	05	03	03	03	2—3字节
个　数	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	4—7字节
数　据	00	20	1F	C2	01	02	C8	03	1F	0BA0	0C68	0C70	01	02	01	8—11字节
地　址	11	23	35	47	59	71	83	95	107	119	131	143	155	167	179	相对地址
说明	文件类型	图像列数	图像行数	*每个像素位数	压缩代码	亮度解释	图像地址偏移	每像素的灰度值个数	图像行数	图像字节数	x轴分辨率	y轴分辨率	存放格式	分辨率单位	预测方式	*对彩色图像此值为地址指针，指某一个值位数所在地址。

第六项：为图像的亮度解释。对二值图像和灰度图像来说，0代表黑时，取值为“1”；反之，取值为“0”；对彩色图像取值为“2”，表示（0，0，0）代表黑色，（255，255，255）代表白色；对使用调色板的图像取值为“3”；

第七项：图像数据的实际起始地址，为4字节整型数；

第八项：说明每个像素的灰度值个数，对灰度、二值和调色板图像该值为“1”，对彩色图像该值为“3”；

第九项：表示每个图像条带中图像的行数。对单条带图像来说，该值同第三项；

第十项：表示整个图像的字节数，该值应为第二项与第三项之积；

第十一项：表示x轴方向的图像分辨率；

第十二项：表示y轴方向的图像分辨率；

第十三项：表示图像的存放格式。该值为“1”时，表示各像素相邻存放，构成一个图像平面，对彩色图像来说，数据存放格式即为RGB RGB RGB RGB……；

第十四项：分辨率的单位，即十一、十二项数据的实际单位，无单位时取值为“1”；以英寸为单位时，取值为“2”；以厘米为单位时，取值为“3”；

第十五项：表示是否使用预测方法，缺省为“1”，表不使用预测方法。此项仅当第五

项取“5”时才有意义。

在上述15个参数项之后，一般为4个字节的0，但对彩色图像而言，在这个4个字节的0之后，还有6个字节的彩色分量信息，用于告诉用户每一分量各占多少位。

紧随其后的才是真正的图像数据，表中所列的数据类型的含义详见表4－3，如表示图像行列数的数据类型为04，即表示属第4类，4字节长整型。

需要说明的是，表4－2中数据行中的数据皆是以31×32彩色图像的16进制有效数据为例，仅用于说明该值的存放地址和含义。实际图像中该处的数值各不相同。此外，表中的地址行表示的是第一个图像文件的各参数项的起始地址。

表4－3　TIFF文件数据类型一览表

类型	数据类型	例子	取值范围
01	单字节整型		－128～＋127
02	ASCII字符		0～255
03	双字节整型（短）	每像素占的位数	－32768～＋32767
04	四字节长整型	图像行、列数	－2147483648～＋2147483647
05	八字节分数（分子4B/分母4B）	分辨率(x,y轴)	两个四字节整型之商

值得注意的是，如果一个TIF文件中含多个子文件，则在文件头之后的参数段中要首先顺序列出每个子文件有关参数（每个图像15项，每项12字节），然后才是真正的图像数据。图像数据一般为单字节整型，除彩色图像占三个字节，且按红、绿、蓝的次序顺序排列外，一个像素一般占一个字节。

（五）　地震勘探记录SEG－Y格式的文件结构和数据格式

地震勘探记录经专用机与专用软件处理后，一般分为叠加和偏移两种数据，以SEG－Y格式记录在磁带上。若要在微机上使用上述数据，则要用带磁带机的计算机将磁带读入，再经异步通信转入微机。但此时传输进微机的仅是SEG－Y格式下的二制码0－1的组合，不能供DOS系统下的软件直接使用。SEG－Y格式下的浮点数，是一个IBM32位兼容浮点数，包括一位符号位，一个24位的尾数和一个7位偏置指数（见下）。

S	e_1	e_2	e_3	e_4	e_5	e_6	e_7	F_1	F_2	F_3	F_4	F_5	F_6

其中，S为符号位，e_1—e_7为指数位，F_1—F_6为16进制的尾数（每个占4位）。

$$\text{偏置指数 } E = e_1 \times 2^6 + e_2 \times 2^5 + \cdots + e_7 \times 2^0$$

$$\text{尾数 } F = F_1 \times 16^{-1} + F_2 \times 16^{-2} + \cdots F_6 \times 16^{-6}$$

整个32位数的值 $X = S \times F \times 16^{(E-64)}$。

地震记录以道为单位组成一个记录，在SEG－Y格式中，每道中另有占240字节的道头字。所以，若地震记录时间为1秒，采样间隔为1毫秒，即每道共有1000个数据（浮点数），则每道记录长度为4240个字节，从第241个字节开始，每4个字节组成一个IBM32位兼容浮点数。我们关心的是除道头字外的浮点数，故我们可从第241个字节开始转换。

四、 非遥感数据格式的转换

本节讨论非遥感数据间的格式转换问题。从前节讨论的常用软件文件结构可知，GRAPHER系列绘图软件的DAT文件就是DOS平台文件，无需转换，GRD文件有一简单的文件头，很容易转换，故这里不再讨论。所以，本节涉及的数据格式转换包括dBASE系列数据库、pcARC/INFO、地震勘探记录SEG－Y格式数据和TIFF格式数据与DOS平台数据文件之间的双向和单向转换。

（一） dBASE系列数据库文件与DOS平台文件的相互转换

1. DBF文件结构的读入

欲要对DBF文件进行操作，先要了解该文件的结构。FORTRAN子程序RDSTR即为此目的设计。该子程序中的虚参意义为：

DBNAME —— 数据库文件名，字符变量，长度为40；
FDNAME —— DBF文件中的字段名，字符变量，长度为10；
FTYPE —— 字段类型，长度为1的字符变量；
FD0 —— 字段在记录中的起点位置，2字节整型变量；
FDW1 —— 字段宽，2字节整型变量；
FDW2 —— 数值型字段小数点后位数，2字节整型变量；
FDNUM —— 字段个数，2字节整型变量；
NOFF —— DBF文件中记录起始字节数，2字节整型变量；
NRECL ——记录长度，2字节整型变量；
NUM —— 记录总数，4字节整型变量。

上述虚参中DBNAME由主程序给出，其余均返回主程序。

```
      SUBROUTINE RDSTR(DBNAME,FDNAME,FTYPE,FD0,FDW1,FDW2,FDNUM,
     &  NOFF,NRECL,NUM)
      INTEGER*4 NUM,R1,R2
      INTEGER*2 FD0(255),FDW1(255),FDW2(255),FDNUM,NRECL,NOFF,F2
      CHARACTER DBNAME*40,FDNAME(255)*10,FTYPE(255)*1,F1*1,W1*1,W2*1
      OPEN(5,FILE=DBNAME,ACCESS='DIRECT',RECL=32)
      READ(5,REC=1) R1,NUM,NOFF,NRECL,R2
      FDNUM=(NOFF-1)/32-1
      DO 40 I=1,FDNUM
      READ(5,REC=I+1)FDNAME(I),F1,FTYPE(I),FD0(I),F2,W1,W2
      FDW1(I)=ICHAR(W1)
      FDW2(I)=ICHAR(W2)
   40 CONTINUE
      CLOSE(5)
      END
```

2. 从DBF文件中读数值型字段

子程序RDDBFV用于从数据库DBF文件中读出数值型字段。其中虚参DBNA-

ME、FD0、FDW1、FDW2、NRECL、NOFF 意义同上，其余虚参意义为：

IFD —— 字段序号，即指明读哪一个字段；

X —— 实型数组，存放字段值；

N1 —— 需读的起始记录数，4 字节整型变量；

N2 —— 需读的终止记录数，4 字节整型变量；

其中 X 返回主程序，其余均由主程序调入。

子程序 RDDBFV 需调用函数 VAL2。VAL2 的功能是将一字符串转化为数值，其虚参含义为：

VSTR ——值的字符串，长度为 20；

FDW1 ——字段宽，亦即字符串长度，2 字节整型变量；

FDW2 ——值的小数点后位数，2 字节整型变量。

```
      SUBROUTINE RDDBFV(DBNAME,FD0,FDW1,FDW2,NRECL,NOFF,IFD,X,N1,N2)
      INTEGER*2 FD0(255),FDW1(255),FDW2(255),NRECL,NOFF,FDW11,FDW21
      CHARACTER DBNAME*40,VSTR(20)*1
      DIMENSION X(N2)
      OPEN(5,FILE=DBNAME,ACCESS='DIRECT',RECL=1)
      NOFF2=NOFF+FD0(IFD)
      FDW11=FDW1(IFD)
      FDW21=FDW2(IFD)
      II=0
      DO 40 I=N1,N2
      JJ=NOFF2+(I-1)*NRECL
      J1=0
      DO 20 J=JJ+1,JJ+FDW11
      J1=J1+1
   20 READ(5,REC=J) VSTR(J1)
      II=II+1
      X(II)=VAL2(VSTR,FDW11,FDW21)
   40 CONTINUE
      CLOSE(5)
      END

      FUNCTION VAL2(VSTR,FDW1,FDW2)
      CHARACTER VSTR(20)*1
      INTEGER*2 FDW1,FDW2
      VAL2=0.
      NF=1
      IF(FDW2.GT.0) THEN
        J=FDW1-FDW2-1
      ELSE
        J=FDW1
      ENDIF
      DO 10 I=1,J
      IF (VSTR(I).EQ.' ') GOTO 10
        IF(VSTR(I).EQ.'-') THEN
          NF=-1
```

```
        GOTO 10
      ENDIF
    VAL2=VAL2+(ICHAR(VSTR(I))-48)*10**(J-I)
10  CONTINUE
    DO 20 I=1,FDW2
20  VAL2=VAL2+(ICHAR(VSTR(I+J+1))-48.0)/10**I
    VAL2=VAL2*NF
    END
```

RDDBFV 仅能读数值型字段，除备注型、屏幕型字段外，其它类型字段（字符型、逻辑型、日期型）仿此均可读入，且不需经字符一数值转化。对于备注型字段，可经RDSTR读出DBF文件结构后，再打开备注文件读出。

3. 重写 DBF 文件的文件头

若要对某一 DBF 文件的文件头按指定的内容修改重写，可使用子程序WTSTR。WTSTR可对DBF文件头中的所有实质性内容进行重写，即包括字段名、字段类型、字段起点位置（字段数）、字段宽、记录长度、记录总数等。各虚参意义同 RDSTR。

```
    SUBROUTINE WTSTR(DBNAME,FDNAME,FTYPE,FD0,FDW1,FDW2,FDNUM,
   &NOFF,NRECL,NUM)
    INTEGER*4 NUM,R1,R2,NUM2
    INTEGER*2 FD0(255),FDW1(255),FDW2(255),FDNUM,F2
    INTEGER*2 NOFF,NOFF2,NRECL,NRECL2
    CHARACTER DBNAME*40,FDNAME(255)*10,FTYPE(255)*1,F1*1,W1*1,W2*1
    CHARACTER C1*16,W3(14)*1
    OPEN(5,FILE=DBNAME,ACCESS='DIRECT',RECL=32)
    READ(5,REC=1) R1,NUM2,NOFF2,NRECL2,R2,C1
    WRITE(5,REC=1) R1,NUM,NOFF,NRECL,R2,C1
    F1=CHAR(0)
    F2=0
    DO 10 I=1,14
10  W3(I)=CHAR(0)
    DO 40 I=1,FDNUM
    W1=CHAR(FDW1(I))
    W2=CHAR(FDW2(I))
    DO 20 J=2,10
    IF(FDNAME(I)(J:J).EQ.' ') THEN
      FDNAME(I)(J:J)=CHAR(0)
    END IF
20  CONTINUE
    WRITE(5,REC=I+1) FDNAME(I),F1,FTYPE(I),FD0(I),F2,W1,W2,
   &(W3(J),J=1,14)
40  CONTINUE
    CLOSE(5)
    END
```

4. 将数值型字段写入 DBF 文件

1) 按字段写

子程序 WTDBF 的功能是将数值型字段按字段写入 DBF 文件，虚参意义同上。

WTDBF需调用长度为20的字符型函数STR2，其功能是将一数值转化为字符串。STR2中的虚参FDW1、FDW2含意同上，虚参V0为需转化的值。

```
      SUBROUTINE WTDBF(DBNAME,FD0,FDW1,FDW2,NRECL,NOFF,IFD,X,N1,N2)
      INTEGER*2 FD0(255),FDW1(255),FDW2(255),NRECL,NOFF,FDW11,FDW21
      CHARACTER DBNAME*40,VSTR*20,STR2*20
      DIMENSION X(N2)
      OPEN(5,FILE=DBNAME,ACCESS='DIRECT',RECL=1)
      NOFF2=NOFF+FD0(IFD)
      FDW11=FDW1(IFD)
      FDW21=FDW2(IFD)
      DO 40 I=N1,N2
      JJ=NOFF2+(I-1)*NRECL
      J1=0
      VSTR(1:FDW11)=STR2(X(I),FDW11,FDW21)
      DO 10 J=JJ+1,JJ+FDW11
      J1=J1+1
10    WRITE(5,REC=J) VSTR(J1:J1)
40    CONTINUE
      CLOSE(5)
      END

      FUNCTION STR2(V0,FDW1,FDW2)
      CHARACTER STR2*20
      REAL*8 V,V2
      INTEGER*2 FDW1,FDW2
      DO 10 I=1,FDW1
10    STR2(I:I)=' '
      V=V0
      IF(V.GT.0) THEN
      V=V+5D0/10D0**(FDW2+1)
      ELSE IF(V.LT.0) THEN
      V=V-5D0/10D0**(FDW2+1)
      ENDIF
      IF(ABS(V).LT.1D0/10D0**(FDW2)) V=0D0
      IV=INT(V)
      IF(FDW2.GT.0) THEN
        JJ=FDW1-FDW2-1
        STR2(JJ+1:JJ+1)='.'
        V2=ABS(V-IV)
        DO 30 I=1,FDW2
        IV2=INT(V2*10**I)
        STR2(JJ+I+1:JJ+I+1)=CHAR(48+IV2)
        V2=V2-(IV2+0D0)/10**I
30      CONTINUE
      ELSE
        JJ=FDW1
      ENDIF
      IF(V.NE.0.) THEN
        IF(V.GT.0) THEN
```

```
        NF=0
        J1=1
      ELSE
        NF=1
        J1=2
        IV=-IV
      ENDIF
      IF(IV.GE.10**(JJ-J1+1)) THEN
        DO 50 I=1,FDW1
50      STR2(I:I)='*'
        GOTO 100
      ENDIF
      NJ=0
      DO 60 I=J1,JJ
      IV2=IV/10**(JJ-I)
      IF(IV2.EQ.0.AND.NJ.EQ.0.AND.I.NE.JJ) GOTO 60
      STR2(I:I)=CHAR(IV2+48)
      IV=IV-IV2*10**(JJ-I)
      NJ=1
60    CONTINUE
      IF(NF.EQ.1) THEN
        DO 80 I=1,JJ
        IF(STR2(I+1:I+1).NE.' ') THEN
          STR2(I:I)='-'
          GOTO 100
        ENDIF
80      CONTINUE
      ENDIF
    ELSE IF(JJ.GT.0) THEN
      STR2(JJ:JJ)='0'
    ENDIF
100 END
```

2) 按记录写

子程序WTDBF2将同一记录中的若干数值型字段写入 DBF 文件，其中虚参 INUM 代表记录号，IFD1、IFD2 分别代表需写字段起始号与终止号。WTDBF2亦需调用函数 STR2。

```
    SUBROUTINE WTDBF2(DBNAME,FD0,FDW1,FDW2,NRECL,NOFF,INUM,X,
   &  IFD1,IFD2)
    INTEGER*4 INUM,IFD1,IFD2
    INTEGER*2 FD0(255),FDW1(255),FDW2(255),NRECL,NOFF,FDW11,FDW21
    CHARACTER DBNAME*40,VSTR*20,STR2*20
    DIMENSION X(IFD2)
    OPEN(5,FILE=DBNAME,ACCESS='DIRECT',RECL=1)
    NOFF1=NOFF+(INUM-1)*NRECL
    DO 40 IFD=IFD1,IFD2
    FDW11=FDW1(IFD)
    FDW21=FDW2(IFD)
    JJ=NOFF1+FD0(IFD)
```

```
      J1=0
      VSTR(1:FDW11)=STR2(X(IFD),FDW11,FDW21)
      DO 10 J=JJ+1,JJ+FDW11
      J1=J1+1
10    WRITE(5,REC=J) VSTR(J1:J1)
40    CONTINUE
      CLOSE(5)
      END
```

5. 关于删除标志的处理

在 DBF 文件中，已作逻辑删除的记录在该记录的第一字节存放“*”标记，故欲判断某记录是否已被逻辑删除，需读出该标记。子程序 RDDBFD 即为此而设计。其中虚参 DSIGN 存放删除标志，其余虚参意义同上。

```
      SUBROUTINE RDDBFD(DBNAME,NRECL,NOFF,N1,N2,DSIGN)
      INTEGER*4 N1,N2
      INTEGER*2 NRECL,NOFF
      CHARACTER DBNAME*40,DSIGN(N2)*1
      OPEN(5,FILE=DBNAME,ACCESS='DIRECT',RECL=1)
      NOFF2=NOFF
      DO 40 I=N1,N2
      JJ=NOFF2+(I-1)*NRECL+1
40    READ(5,REC=JJ) DSIGN(I)
      CLOSE(5)
      END
```

同样，对于写入 DBF 文件的记录，也应写上删除标记，此时可调用子程序WTDBFD。需指出的是，对于新写入的记录，都应执行一次此类操作（删除时 DSIGN 赋“*”，否则赋空格）以确保该位置不被其它字符占领。

```
      SUBROUTINE WTDBFD(DBNAME,NRECL,NOFF,N1,N2,DSIGN)
      INTEGER*4 N1,N2
      INTEGER*2 NRECL,NOFF
      CHARACTER DBNAME*40,DSIGN(N2)*1
      OPEN(5,FILE=DBNAME,ACCESS='DIRECT',RECL=1)
      NOFF2=NOFF
      DO 40 I=N1,N2
      JJ=NOFF2+(I-1)*NRECL+1
40    WRITE(5,REC=JJ) DSIGN(I)
      CLOSE(5)
      END
```

（二） DOS 通用格式文件向 pcARC/INFO 格式文件的转换

这里要讨论的是将 DOS 平台文件按 pcARC/INFO 文件的格式要求重新读写一次。为更好的说明此项转换的一般过程，下面仅举两个转换程序为例：

1. 点类 DOS 平台文件向 pcARC/INFO 文件的转换

此项转换主要分为三步：

1）读取原数据，了解其基本结构。在本例程中，就是读取文件头，了解原点类数据中含多少行、多少列等基本参数；

2）逐行读取原数据，并生成 pcARC/INFO 所要求的数据格式，即生成序号及对应点的 x、y 坐标；

3）将已有数据按指定格式存盘。如本例程中写入点的序号 NY 并在文件结束处加入结束符“END”。至于输出文件中，存进的主要数据已在第二步转为实型坐标值。

```
C  * * * * * PROGRAM POINTSARC.FOR CONVERTS A DOS FILE TO ARC FORM
      REAL NX,NY
      DIMENSION MA(24,24)
      OPEN(1,FILE=' ',STATUS='OLD')
      READ(1,'(2I2)') NL,NP
      READ(1,'(I2)') ((MA(K,J),J=1,NP),K=1,NL)
      CLOSE(1)
      OPEN(2,FILE=' ')
      NP=100
      DO I=1,NL
      DO J=1,NP
      IF(MA(I,J).EQ.1) THEN
      NP=NP+1
      NX=REAL(J)
      NY=REAL(24-I)
      WRITE(2,'(I4,2F5.1)') NP,NX,NY
      ENDIF
      ENDDO
      ENDDO
      WRITE(2,'(A)') 'END'
      CLOSE(2)
      END
```

2. 线类数据向 pcARC/INFO 格式的转换

原数据的格式为线号及顺序排列的线上各点的坐标，要转为 pcARC/INFO 能接受的线要素数据格式，就必须在适当的位置插入线要素结束标记“END”。由于原数据中各线要素所含的点数为随机的，所以在下面的例程中插入了专门用于查明原数据中线要素及线上点的个数的语句。即生成一个一维数组RB，用于存放线要素标号出现的地址。各线要素中所含的点数，可用相临两要素标号地址之差求得。与点类数据转换程另一个不同之处就是除在每条线要素结束处加一个“END”标记外，文件尾还要再一个文件结束标记 “END”（参见表 4－1）。以下即为该例程。

```
      DIMENSION RA(5000,2),RB(1000)
      OPEN (1,FILE=' ',STATUS='OLD')
      PRINT * ,'PLEASE INPUT RESULT DATA FILE NAME! '
      OPEN (4,FILE=' ')
      I=1
      NB=0
```

```
101   READ(1,'(F9.0)',END=102) XA
      IF(XA.GT.5000.) THEN
      NB=NB+1
      RB(NB)=I
      ENDIF
      I=I+1
      GOTO 101
102   NA=I-1
      RB(NB+1)=NA+1
      REWIND 1
      DO I=1,NB
      NK=INT(RB(I))
      READ(1,'(F9.0)') RA(NK,1)
      NY=INT(RA(NK,1))
      WRITE(4,'(I4)') NY
      NL=INT(RB(I+1))-NK-1
      DO J=1,NL
      JK=NK+J
      READ(1,'(1X,2F9.2)') (RA(JK,K),K=1,2)
      WRITE(4,'(2F9.2))') RA(JK,1),RA(JK,2)
      ENDDO
      WRITE(4,'(A3)') 'END'
      ENDDO
      WRITE(4,'(A3)') 'END'
      CLOSE(2)
      CLOSE(1)
      END
```

（三） 地震记录 SEG－Y 格式数据向 DOS 数据文件的转换

SEG－Y 格式上面已经论述，利用下面的程序即可将其转换成 DOS 平台的数据文件。在该程序的 PARAMETER 语句中，KRN 为每道数据个数，可根据实际情况给定。

程序运行时，需输入原始数据文件名、结果文件名和记录个数。

```
      PARAMETER (KRN=800,KRNL=KRN*4+240)
      REAL*4 X(KRN)
      INTEGER IX(KRN*4),II(KRN*4,8)
      CHARACTER FNAME*15,IC(KRNL)*1
      WRITE(*,*) ' INPUT DATA FILE NAME'
      READ(*,'(A)') FNAME
      OPEN(4,FILE=FNAME,ACCESS='DIRECT',RECL=KRNL)
      WRITE(*,*) ' INPUT RESULT FILE NAME'
      READ(*,'(A)') FNAME
      OPEN(3,FILE=FNAME,ACCESS='DIRECT',RECL=KRN*4)
      WRITE(*,*) ' INPUT RECORD NUMBER(KR)'
      READ(*,*) KR
      FM1=16.0
      FM2=256.0
      FM3=4096.0
```

```
      FM4=65536.0
      FM5=1048576.0
      FM6=16777216.0
      DO 200 KK=1,KR
      WRITE(*,*)'K=',KK
      READ(4,REC=KK) (IC(I),I=1,KRNL)
      DO 20 I=241,KRNL
      I2=I-240
20    IX(I2)=ICHAR(IC(I))
      DO 80 K=1,KRN*4
      DO 40 J=1,8
40    II(K,J)=0
      IXX=IX(K)
      DO 60 J=8,1,-1
      IF(IXX.EQ.0) GO TO 80
      IX2=IXX/2
      II(K,J)=IXX-IX2*2
      IXX=IX2
60    CONTINUE
80    CONTINUE
      DO 100 K=1,KRN
      K4=K*4
      K3=K4-1
      K2=K3-1
      K1=K2-1
      IF(II(K1,1).EQ.0) THEN
        KE=IX(K1)-64
        KF=1
      ELSE
        KE=IX(K1)-192
        KF=-1
      END IF
      FZ1=(8*II(K2,1)+4*II(K2,2)+2*II(K2,3)+II(K2,4))/FM1
      FZ2=(8*II(K2,5)+4*II(K2,6)+2*II(K2,7)+II(K2,8))/FM2
      FZ3=(8*II(K3,1)+4*II(K3,2)+2*II(K3,3)+II(K3,4))/FM3
      FZ4=(8*II(K3,5)+4*II(K3,6)+2*II(K3,7)+II(K3,8))/FM4
      FZ5=(8*II(K4,1)+4*II(K4,2)+2*II(K4,3)+II(K4,4))/FM5
      FZ6=(8*II(K4,5)+4*II(K4,6)+2*II(K4,7)+II(K4,8))/FM6
      X(K)=(FZ1+FZ2+FZ3+FZ4+FZ5-FZ6)*16.**KE*KF
100   CONTINUE
      WRITE(3,REC=KK) (X(K),K=1,KRN)
200   CONTINUE
      CLOSE(3)
      CLOSE(4)
      END
```

（四）TIFF 文件格式的转换

上一节已以 TH－1200 扫描仪为例详细讨论了 TIFF 文件的结构，下列程序就是在此基础上用FORTRAN语言编写的读取 TIFF 文件内容的实用程序。程序中：

NP —— 图像的列数；

NL —— 图像的行数；

NB —— 每个像素所占的位数；

M —— 图像像素的地址偏移量；

FNM —— 输入 TIFF 文件名；

FNM1 —— 输出数据文件名。

程序运行时，首先提示用户输入一个 TIFF 文件名。至于该文件的行数、列数及每个像素的位数皆由程序从用户文件中自动提取。程序中部的大部分内容皆为生成pcARC/INFO的压缩ASCII文件而写，而并非通用文件格式所必需。最后用户按提示输入输出文件名，程序即可生成用户所需的数据格式文件。由于TIFF文件在本质上是点阵图像与点阵汉字无异，所以程序中将两者并为一类，如用户有点阵汉字的通用格式数据，亦可经由本程序转为pcARC/INFO格式。在整个程序运行过程中，用户所需输入的只是两个文件名而已，前提是要有一个TIFF图像文件或者点阵汉字的数据文件。至于如何由点阵汉字库生成通用格式的点阵汉字数据文件，请参阅本书下一章内容。

```
C  * * * * * PROGRAM TIF.FOR TRANSFERS TIFF DATA TO DOS FORM  * * * * *
      INTEGER * 1 H,NH
      INTEGER * 4 M,MX
      CHARACTER * 1 FA,FB,FC,FD
      INTEGER * 2 NP,NL,NB,LA(29,500),NX(2)  ,[HUGE](5000,2)
      CHARACTER CA * 4,CB * 32,CC * 16,CD * 8,FNM * 20,FNM1 * 20
      DIMENSION H[HUGE](5000),NH[HUGE](5000),MR(8),IYR(6)
      DIMENSION IHR1(6),IHR2(6),FA(6),FB(6),MY(5000)
C  * * * * READ PARAMETERS OF THE INPUT FILE
      PRINT * ,'=========================================='
      PRINT * ,'     THIS PROGRAM CONVERTS A TIFF FILE TO DOS FORM     '
      PRINT * ,'       WRITTEN BY H.Q. TAN, ATC OF CUMT, MAR. 1994      '
      PRINT * ,'COPYRIGHT (C) BY ATC OF CUMT 1994. ALL RIGHTS RESERVED'
      PRINT * ,'=========================================='
      PRINT *
  403 WRITE( * ,'(A\)') ' READ (1) TIFF FILE OR (2) C-CHARACTER FILE?
      READ( * ,'(I2)',ERR=403) NW
      WRITE( * ,'(A\)') ' PLEASE ENTER THE INPUT FILE NAME:    '
      READ( * ,'(A20)') FNM
      IF(NW.EQ.1) THEN                          ! FOR TIFF FILE
  801 OPEN(UNIT=2,FILE=FNM,STATUS='OLD',ERR=801,
     *   ACCESS='DIRECT',RECL=2,FORM='UNFORMATTED')
      READ(2,REC=16) NP          ! NUMBER OF PIXELS
      READ(2,REC=22) NL          ! NUMBER OF LINES
      READ(2,REC=28) NB          ! NUMBER OF BITS
      CLOSE(2)
      IF(NB.EQ.1) NP=NP/8
      ELSE
  802 OPEN(UNIT=2,FILE=FNM,STATUS='OLD',ERR=802)   ! FOR C-CHARACTER
      READ(2,'(I2)') (LA(1,K),K=1,4)
      NC=LA(1,1)                                ! NUMBER OF CHARACTERS
```

```
      NL=LA(1,2)                                    ! TYPE OF CHARACTÈRS
      NT=LA(1,3)                                    ! NUMBER OF BITS
      NS=LA(1,4)                                    ! SPACE LENGTH
      NL=NT*8+3
      NP=NC*NT*8+NC*NS
      READ(2,'(I2)') ((LA(I,J),J=1,NP),I=2,NL)
      CLOSE(2)
      ENDIF
C     C****  CREATES OUTPUT FILE
      CALL GETDAT(IYR(3),IYR(2),IYR(1))
      CALL GETTIM(IYR(4),IYR(5),IYR(6),I100TH)
      IYR(3)=IYR(3)-1900
      DO K=1,6
      IHR2(K)=MOD(IYR(K),10)
      FB(K)=CHAR(IHR2(K)+48)
      IHR1(K)=(IYR(K)-IHR2(K))/10
      FA(K)=CHAR(IHR1(K)-48)
      ENDDO
      FC=CHAR(47)
      FD=CHAR(58)
      CC=FA(1)//FB(1)//FC//FA(2)//FB(2)//FC//FA(3)//FB(3)
      CD=FA(4)//FB(4)//FD//FA(5)//FB(5)//FD//FA(6)//FB(6)
  803 WRITE(*,'(A\)') ' PLEASE ENTER THE OUTPUT FILE NAME:  '
      READ(*,'(A20)') FNM1
      OPEN(UNIT=1,FILE=FNM1,STATUS='NEW',ERR=803)
      NP1=NP
      IF(NB.EQ.1) NP1=NP*8
      CA='CAS '
      IF(NW.EQ.1) THEN
      CB='CREATED FROM TIF'
      ELSE
      CB='CREATED FROM CHA'
      ENDIF
      WRITE(1,102) CA,CB,CC,CD
      WRITE(1,'(4I6)') 1,NL,1,NP1
  102 FORMAT(A4,1X,A32,1X,A16,1X,A8)
      IF(NW.EQ.2) THEN
      PRINT*,NL,NP
      MX=0
      MY(MX)=0
      DO IK=1,NL
      DO JK=1,NP
        IF(IK.EQ.1) THEN
          LA(IK,JK)=1
        ELSE
          LA(IK,JK)=LA(IK,JK)+1
        ENDIF
      ENDDO
      ENDDO
      DO II=1,NL
      DO JJ=1,NP
```

```
          IF(LA(II,JJ).NE.LA(II,JJ+1)) THEN
          MX=MX+1
          MY(MX)=JJ+1
          NX(1)=MY(MX)-MY(MX-1)
          NX(2)=LA(II,JJ)-1
          WRITE(1,'(2I6)') NX(1),NX(2)
          ENDIF
        ENDDO
          MX=0
          MY(MX)=0
        ENDDO
        ENDIF
C  C* * * * READS DATA FROM INPUT TO OUTPUT FILE
        IF(NW.EQ.1) THEN
        OPEN(UNIT=2,FILE=FNM,STATUS='OLD',
     *   ACCESS='DIRECT',RECL=1,FORM='UNFORMATTED')
        MX=0
        MY(MX)=0
        DO I=1,NL
        DO J=1,NP
            M=(I-1)*NP+J
            READ(2,REC=194+M) H(J)
        IF(NB.NE.1) THEN
        IF(H(J).LT.0) THEN
        H(J)=1
        ELSE
        H(J)=2
        ENDIF
        ELSE
              MA=H(J)
              IF(H(J).LT.0) MA=H(J)+256
              DO KJ=8,1,-1
             MH=(J-1)*8+KJ
              NH(MH)=MOD(MA,2)+1
              MR(KJ)=MA/2
              MA=MR(KJ)
        ENDDO
        ENDIF
        ENDDO
        IF(NB.EQ.1) THEN
          DO KK=1,NP1
                IF(NH(KK).NE.NH(KK+1)) THEN
          MX=MX+1
          MY(MX)=KK
          NX(1)=MY(MX)-MY(MX-1)+1
          NX(2)=NH(KK)-1
          WRITE(1,'(2I6 )',ERR=101) NX(1),NX(2)   ! 1-BIT IMAGE
          ENDIF
          ENDDO
        ELSE
          DO KK=1,NP1
```

```
            IF(H(KK).NE.H(KK+1)) THEN
          MX=MX+1
          MY(MX)=KK
          NX(1)=MY(MX)-MY(MX-1)
          NX(2)=H(KK)
          WRITE(1,'(2I6)',ERR=101) NX(1),NX(2)   ! 1-BIT IMAGE
          ENDIF
          ENDDO
        ENDIF
        MX=0
        MY(MX)=0
        ENDDO
101     CLOSE(2)
        CLOSE(1)
        END
```

（五） 不同格式的图像文件间的转换

为进一步说明通用格式的必要性和实用性，下面介绍不同图像处理系统之间的格式转换问题。

上节程序说明了通用TIFF文件转为通用格式后向pcARC/INFO的转换。下面的程序则说明同样的TIFF文件还可转为另一种图像处理系统适用的格式文件。从而使扫描仪直接用作图像处理系统的数据入口。

作为例子，作者采用了 Chromatic 真彩色显微图像处理系统的数据格式。该软件原配图像数据入口只有一个显微摄像机，只能在显微镜下对光薄片进行摄像采样和图像处理，工作平台要求不高，但常规图像处理的各项功能较为齐全。要用扫描仪作为其数据入口，不仅要了解TIFF文件格式，还必需了解Chromatic图像的文件的格式。分析表明，该软件自动生成的或可以调用的图像文件，并不含任何文件头，但数据的存放有其特殊之处，那就是红、绿、蓝三个分量的灰度值是与两个字节存放的，即每个分量占 5 位，最高位用于存放覆盖文件。每个分量的取值范围为 0－31。但从对TIFF文件的分析得知，该扫描仪生成的彩色图像文件中，红、绿、蓝三个分量的灰度值，各占 1 个字节，取值范围分别为 0－255。两者之间不光灰度的分级不同，而且数据格式各异。借助对 TIFF 文件通用格式的进一步转换，不难实现向Chromatic格式的转换，以下即为用 FORTRAN 语言编写的实用转换程序：

```
C   * * * * * PIB.FOR - CONVERTS TIFF FILE TO PIB FILE    * * * * *
        INTEGER*1              R(500),G(500),H(1024),B(500)
        DIMENSION              KR(500),KG(500),KB(500)
        CHARACTER*1            Q
C   * * * * * INPUT ORIGINAL DATA
  500   OPEN (1,FILE='',STATUS='OLD',ERR=500,
     +    ACCESS='DIRECT',RECL=2,FORM='UNFORMATTED')
        READ(1,REC=16) NP
        READ(1,REC=22) NL
        CLOSE (1)
  501   OPEN(1,FILE='',STATUS='OLD',ERR=501,
     +    ACCESS='DIRECT',RECL=1,FORM='UNFORMATTED')
```

```
502   OPEN(2,FILE=' ',STATUS='NEW',ERR=502,
     +    ACCESS='DIRECT',RECL=1,FORM='UNFORMATTED')
      NN=NP*3
      NM=NP*2
      DO I=1,NL
      DO J=1,NN,3
      JJ=J/3+1
      JK=2*JJ-1
      M1=(I-1)*NM+JK
      M=200+(JJ-1)*3+(I-1)*NN
      READ(1,REC=M+1) R(JJ)
      KR(JJ)=R(JJ)
      IF(R(JJ).LT.0) KR(JJ)=R(JJ)+256
      KR(JJ)=KR(JJ)/8
      READ(1,REC=M+2) G(JJ)
      KG(JJ)=G(JJ)
      IF(G(JJ).LT.0) KG(JJ)=G(JJ)+256
      KG(JJ)=KG(JJ)/8
      READ(1,REC=M+3) B(JJ)
      KB(JJ)=B(JJ)
      IF(B(JJ).LT.0) KB(JJ)=B(JJ)+256
      KB(JJ)=KB(JJ)/8
      K1=KG(JJ)/8
      K2=KG(JJ)-8*K1
      H(JK)=KR(JJ)*4+K1
      H(JK+1)=K2*32+KB(JJ)
      WRITE(2,REC=M1) H(JK+1)
      WRITE(2,REC=M1+1) H(JK)
      ENDDO
      ENDDO
      PRINT *
109   WRITE(*,'(A\)') ' USE ANOTHER DATA FILE?
      READ(*,'(A)') Q
      IF(Q.NE.'y'.AND.Q.NE.'Y'.AND.Q.NE.'n'
     +   .AND.Q.NE.'N') GOTO 109
      IF(Q.EQ.'y'.OR.Q.EQ.'Y') THEN
      GOTO 501
      ENDIF
      CLOSE(2)
      CLOSE(1)
      END
```

第五章　数据处理中的特殊问题——汉字注记

如同其它应用领域一样，在遥感、非遥感信息复合处理中，也不可避免要涉及到中文注记问题。大部分使用效果较好的图形、图像处理软件，如 AutoCAD、pcARC/INFO、I²S、ARIES Ⅱ 等通用性较强的软件，都是建立在西文软件基础之上的。由于此类软件无法直接使用中文注记，这就给使用中文的广大用户带来诸多不便。作为补救手段，人们往往采用植字、贴字等办法来满足用户的基本要求。这种做法不仅费时费力，而且也与整个工作的自动化程度很不适应。正因为如此，关于西文软件的汉化处理已引起广泛重视。近年来，各种软件的汉化速度相当快，基本与国外软件处于准同步状态。汉化确实给国人使用带来方便，但也往往带来一些问题。一般汉化软件都需在中文 DOS 系统下操作，需占用不少内存，有的还不是能完全达到西文水平；更有甚者，有的汉化软件（或自行开发的中文软件）需特殊的中文操作系统，这就更给使用者带来不便。无论如何，汉化软件需额外开支一些内存，故与西文软件相比，总是打了一些折扣，何况对目前大多数微机来说，内存总是很宝贵的；为了节省一些内存，程序设计者们往往绞尽脑汁，所以对于大块被中文占据的内存视而不见显然是不合理的。另外，汉字占据内存的哪一部分，是否与原软件占据内存矛盾，也是一个问题。有些软件之所以要用特殊的操作系统，就是为了将汉字操作系统置于高内存中，但这样只能解决眼前问题，若该软件功能扩充，也需占据目前汉字操作系统占据的内存。这样是再搞一个处于更高内存的汉字操作系统，还是将这一块留出，软件跳过该部分内存而占据更高的内存呢？有鉴于此，有人提出部分汉化的方案，如将菜单提示部分汉化，而内核不变；也有人提出用纯硬件方式来汉化西文软件。这样做不仅造价较高，而且用户还必须重新熟悉新的工作环境。

近年来作者在处理遥感和非遥感信息的实践中，充分认识到，真正影响西文软件优势发挥的主要障碍并非操作系统本身，而是中文注记。因为大部分此类软件的用户的目标并不是中文文稿的编辑、打印，而是图形、图像处理，使用者皆有一定的英文水准，菜单的应用并不成问题，就是非专门的操作人员，只要了解图形、图像处理基本原理及软件功能也不难操作应用软件。所以对软件本身的汉化似乎意义不大，或者说投入大、收益小。问题现在归结为汉字要注记，而同时能否不占内存，保持原有软件的风范？在这里我们另辟蹊径，提出从西文操作系统下直接调入汉字的方法，前提是软件与图件中汉字不能太多。考虑到大部分软件注记汉字和图件中用到的汉字不会太多，这一方法无疑是可行的；虽然操作中涉及汉字时要复杂一些，但获得的是可节省大量内存和保持原有软件的面貌。基于上述，作者认为，更合理、方便又易于为用户接受的解决西文状态下中文注记问题的方法，应是在西文软件不动的前提下，为用户提供一个显示、加注汉字的软接口。

在西文状态下注记汉字，实质上是对汉字库直接进行读操作，再将汉字（图形）写成你所用的西文软件能接受的图形或字符格式，由此达到将汉字嵌入到西文软件中的目的。为此，该用户接口应能满足以下各项基本要求：

- 能适应任意机型、任意操作系统的需要；
- 用户不需增加任何硬件设备，原软件不需作任何改动；
- 能随意选取用户所要的不同字体的汉字并显示或加注到指定位置之上；

· 汉字本身可以无级缩放、随意旋转；

· 汉字输入方式方便可行。

为实现上述目标，需搞清的问题有两个，一为汉字库的结构，二为你所用的西文软件能接受的图形或字符格式。下面先叙述汉字库的结构。

一、汉字库的结构

要解决西文软件环境下的中文注记，就必须首先要有汉字信息源－中文字库。经过10多年的努力，各类中文字库已相对定型并趋于标准化，收入国家《信息交换汉字编码字符集——基本集》(GB2312)的汉字有6763个，已覆盖了常用汉字的99.99%，无需用户再从字库做起。

目前通常使用的汉字库可分为二类，一为点阵库，二为矢量库；每一类中皆有不同的字体库，如宋、仿、黑、楷等基本字体，现在的矢量库中还包括了魏体、行楷，甚至舒同体。所有这些字体皆可在中文环境下自由调用，并进行字形、大小等多种变换。

就西文环境下的汉字显示来讲，只了解字库的类型就远远不够了，还必须深入了解其内部结构，具体地讲就是点阵库中每个汉字的不同字节的定向摆布以及矢量汉字库中的笔画信息的存放形式。现将两大类字库的基本结构特征分述如下。

（一）点阵库结构

汉字点阵库结构比较简单，每个字的记录长度相等，每个字节8位对应8个点，若该位为1即表示该点为字的组成部分，若为0则表示该点空白。16点阵库中各字（包括符号）在字库中按区位的序号排列（符号为1—9区，汉字为16—87区，每区94个汉字）；24和40点阵的符号单独成库，汉字按字体分库。

1. 16点阵库

每个字32个字节，由上往下排成16行，每行2个字节（从左到右排列）；每个字节的高位在左，低位在右。

2. 24点阵库

每个字72字节，按列从左到右排列，共24列，即每列3字节，从上到下排列；每个字节的高位在上，低位在下。

3. 40点阵库

实际上为40（行）×36（列）个点，故每个字180个字节。它的排列分两部分：第一部分72个字节，每列2字节，构成字的上16行；第二部分108字节，每列3字节，构成字的下24行。每个字节亦是高位在上，低位在下。

（二）矢量库结构

所谓矢量汉字应是指用一系列具不同坐标的点来勾绘所需汉字。由于字体、字型的不同，每个汉字的笔画、笔顺各异，致使矢量汉字库的结构比较复杂，且因设计者而异。现仅以常用的WPS中的矢量库为例，剖析它的结构。矢量库中的符号和西文字符都单独成库，不同字体的各个纯汉字库中均含6768个汉字。矢量库与点阵库不同，不但每个汉字占

有字节长短各异，而且整个字库可分为四段：地址码段、字形码段、部件地址码段和部件字形码段。各字体库中皆包含上述前两段；后两段仅见于某些字体的汉字库，如黑体字库中。

1. 汉字地址码段

每个矢量汉字库的开头，都有 20307 字节的地址码段，用以指明每个汉字字形信息的起始地址。描述每个汉字实际笔画的字形码或其部件码的实际存放地址，即由该汉字的起始地址码来确定。将下一个汉字的起始地址减去该汉字的起始地址，即为该汉字字形码的实际长度。全字库中共含汉字 6768 个，每个地址占 3 字节，再加上最后一个汉字的终止地址占用的 3 个字节，整个地址码段的长度为 20307 字节，由此推算，易知第 1 个汉字的起始地址为 20308。

2. 汉字字形码段

已如前述，解决注记汉字字形美问题的根本出路在于对汉字本身的矢量化。矢量汉字库的字形码段存放的就是描述每个汉字实际笔画的各点坐标值或其部件码。考虑到汉字笔画的复杂多变，一般将汉字笔画分成部件字形和非部件字形两大类，并赋以不同的代码。根据地址码段提供的起始地址和长度从库中读取的就是这两类字形码。

1）非部件字形码

所谓非部件字形码，是指那些与其它汉字没有公用部分的汉字字形码。换句话说，此类字形码所勾绘的笔画只能在当前汉字中使用。常见隶书、魏碑，行楷等字体皆是由非部件字形码来勾绘的。其首码一般皆取 0 或 6。

分析表明，非部件字形码的首码可分为几类：

0——起始点坐标，后跟 2 个字节表示横、纵坐标；

1——横向平移，后跟 1 个字节，表示横线终点横坐标，其纵坐标不变；

2——纵向平移，后跟 1 个字节，表示竖线终点纵坐标，其横坐标不变；

3——斜线，后跟 2 个字节，分别表示斜线的终点的横、纵坐标；

4——三点弧线，后跟 4 字节表示后二点的横、纵坐标；

5——四点弧线，后跟 6 字节表示后三点的横、纵坐标；

6——矩形，后跟 4 字节表示矩形的二个对角点的横、纵坐标。

2）部件索引码与字形码

部件字形码主要用于勾绘某些多个汉字公用的部首、字根，或统称之为部件。考虑到汉字部首等部件的重复度较高，对有些字体往往可采用组合式矢量汉字。这样，既可提高其“保真度”，又可减少其存储量，从而将字库的冗余减少到最低限度。而部件索引码提供的只是与当前汉字有关的部件序号，该部件在当前汉字中的横、纵向的偏移量及缩放比。部件索引码的首码为 8 或 9 或 A，分别代表不同类型的部件，皆占半个字节。无论是此类部件索引首码出现在汉字字形码的开始部位还是出现在中部，其后的代码皆不能用作汉字的字形码。实际描述该部件字形的代码，还必须根据部件的序号从字库中另行查明，详见下述。

3. 部件地址码段

在诸如黑体字库之类的矢量汉字库中，紧随汉字字形码之后的就是部件地址码段。部件的序号按部件索引首码为 8、9、A 的次序顺序排列，且因字体的不同而不同。如在黑体库中，首码为 8 的共 246 个，首码为 9、A 的各 256 个，共计 758 个。同汉字地址码一样，

每个部件地址码也占 3 个字节，用以指明每个部件字形码的存放起始地址和长度。以“打”字为例，按汉字地址码指出的地址只能查到 11 个字节，其中含 2 组部件索引码：分别代表序号为 8 类 18 号的“扌”和序号为 9 类 40 号的“丁”。真正用于勾绘这两个部件笔形的代码，还必须根据部件序号查明其在下属部件笔形码中存放的地址才能读取出来。

4. 部件字形码段

部件字形码段中按部件的序号存放着当前字库中可能用到的全部部件字形信息。其中各个代码的含义与汉字字形码段中的非部件字形码完全相同，这里不再重复。程序 HZ.FOR就是基于上述分析编制的汉字调用及注记实用程序，用户只需输入所要用到的汉字的区位码，即可生成一个包含要注记汉字的通用数据文件。

二、在 Grapher 中标注汉字

在 Grapher 系列软件中标注汉字，可分二种形式，一为将一组汉字写成 SYM 文件，然后在作图时将汉字作为字符调入；二为将汉字笔画写成 DAT 文件，即将汉字作为图形输出。前者可使用点阵或矢量汉字，后者仅适合矢量汉字。

（一） SYM 形式调用汉字

Grapher 系列软件中的 SYM 文件格式已在上一章作过描述。现分别考虑点阵汉字和矢量汉字的调用。

1. 点阵汉字的 SYM 文件形式

由于 SYM 文件中字符描述是矢量形式的，故若要将点阵汉字写成 SYM 文件，必须把点阵数据转化成矢量数据。转化思路为，在将点阵数据分解成以位为单位的 0—1 数据后，以适当方式将位为 1 的点联结起来。位为 1 的点又有两种，一为非交叉点，此种点仅是线条一次通过点；二为交叉点，这类点线条要通过不止一次。将所有交叉点和非交叉点相邻 8 个方向存在的值为 1 的点的数目计数并存于一数组，在联结时各点每通过一线条，即减去连结点的数目，直至该数组的值都为 0 时，就完成了所有线条的连接。

上述联结方法有时会使点重复联结，同时 Grapher 中 SYM 文件的每个字符的点数不能超过 127 个，故适合该法的为 16×16 点阵汉字和部分笔画简单的 24×24 点阵汉字。

完成上述操作的 FORTRAN 程序如下：

```
      DIMENSION IZ(40,40),I8(40,40),NDD(100,3),IZJ(40,40,2),IH(200)
      CHARACTER FNAME*14,YS,H(200),IZC(40,40),R(20000)
      WRITE(*,*) ' Input HZK Name'
      READ(*,'(A)') FNAME
5     WRITE(*,*) ' Input Number of Dots(16,24)'
      READ(*,*) NN
      IF(NN.EQ.16) THEN
        NNI=16
        NNJ=16
        OPEN(3,FILE=FNAME,STATUS='OLD',ACCESS='DIRECT',RECL=32)
      ELSE IF(NN.EQ.24) THEN
        NNI=24
        NNJ=24
```

```
      OPEN(3,FILE=FNAME,STATUS='OLD',ACCESS='DIRECT',RECL=72)
    ELSE
      WRITE(*,*) ' Number of Dots Wrong'
      GOTO 5
    END IF
    NP1=3
    NP2=3
    R(1)='S'
    R(2)='Y'
    R(3)='M'
    NR1=3
    LK=65
10  WRITE(*,*) ' Input Region Number (NO)'
    READ(*,*) NO
    R(NR1+1)=CHAR(LK)
    R(NR1+2)=CHAR(0)
    IF(NN.EQ.16) THEN
      R(NR1+3)=CHAR(21)
    ELSE IF(NN.EQ.24) THEN
      R(NR1+3)=CHAR(29)
    END IF
    NR=NR1+4
    ND=0
    NO1=NO/100
    NO2=NO-NO/100*100
    IF(NO1.GT.87) NO1=87
    IF(NO2.GT.94) NO2=94
    IF(NO2.EQ.0) NO2=1
    IF(NN.EQ.16) THEN
      NO1=NO1-1
    ELSE
      NO1=NO1-16
    END IF
    IF(NO1.LT.0) NO1=0
    II=NO1*94+NO2
    IF(NN.EQ.16) THEN
      READ(3,REC=II) (H(J),J=1,32)
      DO 170 J=1,32
      IH(J)=ICHAR(H(J))
      IR=(J-1)/2+1
      IL=J-(J-1)/2*2
      IF(IL.EQ.1) THEN
        K1=1
        K2=8
      ELSE
        K1=9
        K2=16
      END IF
      IS=IH(J)
      DO 130 K=K1,K2
      IZ(IR,K)=0
```

```
130     IZC(IR,K)=' '
        DO 150 K=K2,K1,-1
        IF(IS.EQ.0) THEN
          GO TO 170
        END IF
        IS2=IS/2
        IY=IS-IS2*2
        IF(IY.EQ.1) THEN
        IZC(IR,K)='*'
        IZ(IR,K)=1
        END IF
        IS=IS2
150     CONTINUE
170     CONTINUE
      ELSE IF(NN.EQ.24) THEN
        READ(3,REC=II) (H(J),J=1,72)
        DO 270 J=1,72
        IH(J)=ICHAR(H(J))
        IR=(J-1)/3+1
        IL=J-(J-1)/3*3
        K1=(IL-1)*8+1
        K2=IL*8
        IS=IH(J)
        DO 230 K=K1,K2
        IZ(K,IR)=0
230     IZC(K,IR)=' '
        DO 250 K=K2,K1,-1
        IF(IS.EQ.0) THEN
          GO TO 270
        END IF
        IS2=IS/2
        IY=IS-IS2*2
        IF(IY.EQ.1) THEN
          IZC(K,IR)='*'
          IZ(K,IR)=1
        END IF
        IS=IS2
250     CONTINUE
270     CONTINUE
      END IF
      DO 380 K=1,NNI
      K1=(K-1)*2+1
      K2=K*2
380   WRITE(*,390) K,(IZC(K,J),J=1,NNJ)
390   FORMAT(1X,I2,T10,40A1)
      DO 400 K=1,2
      DO 400 I=1,NNI
      DO 400 J=1,NNJ
400   IZJ(I,J,K)=0
      IF(IZ(1,1).GT.0) THEN
        I8(1,1)=IZ(1,2)+IZ(2,1)+IZ(2,2)
```

```
      ELSE
        I8(1,1)=0
      END IF
      IF(IZ(1,NNJ).GT.0) THEN
        I8(1,NNJ)=IZ(1,NNJ-1)+IZ(2,NNJ)+IZ(2,NNJ-1)
      ELSE
        I8(1,NNJ)=0
      END IF
      IF(IZ(NNI,1).GT.0) THEN
        I8(NNI,1)=IZ(NNI-1,1)+IZ(NNI-1,2)+IZ(NNI,2)
      ELSE
        I8(NNI,1)=0
      END IF
      IF(IZ(NNI,NNJ).GT.0) THEN
        I8(NNI,NNJ)=IZ(NNI-1,NNJ-1)+IZ(NNI-1,NNJ)+IZ(NNI,NNJ-1)
      ELSE
        I8(NNI,NNJ)=0
      END IF
      DO 410 J=2,NNJ-1
      IF(IZ(1,J).GT.0) THEN
        I8(1,J)=IZ(1,J-1)+IZ(1,J+1)+IZ(2,J-1)+IZ(2,J)+IZ(2,J+1)
      ELSE
        I8(1,J)=0
      END IF
      IF(IZ(J,1).GT.0) THEN
        I8(J,1)=IZ(J-1,1)+IZ(J+1,1)+IZ(J-1,2)+IZ(J,2)+IZ(J+1,2)
      ELSE
        I8(J,1)=0
      END IF
      IF(IZ(NNI,J).GT.0) THEN
        I8(NNI,J)=IZ(NNI,J-1)+IZ(NNI,J+1)+IZ(NNI-1,J-1)+IZ(NNI-1,J)+
     &    IZ(NNI-1,J-1)
      ELSE
        I8(NNI,J)=0
      END IF
      IF(IZ(J,NNJ).GT.0) THEN
        I8(J,NNJ)=IZ(J-1,NNJ)+IZ(J+1,NNJ)+IZ(J-1,NNJ-1)+IZ(J,NNJ-1)+
     &    IZ(J+1,NNJ-1)
      ELSE
        I8(J,NNJ)=0
      END IF
  410 CONTINUE
      DO 420 I=2,NNI-1
      DO 420 J=2,NNJ-1
      IF(IZ(I,J).GT.0) THEN
        I8(I,J)=IZ(I,J-1)+IZ(I,J+1)+IZ(I+1,J-1)+IZ(I+1,J)+IZ(I-1,J+1)
     &    +IZ(I-1,J-1)+IZ(I-1,J)+IZ(I-1,J+1)
      ELSE
        I8(I,J)=0
      END IF
  420 CONTINUE
```

```
      DO 430 I=2,NNI-1
      DO 430 J=2,NNJ-1
      IF(I8(I,J).GE.4) THEN
        IF(I8(I+1,J).GE.4.AND.I8(I-1,J).GE.3.AND.I8(I,J-1).GE.4.AND.
     &    I8(I,J+1).GE.3)THEN
          I8(I+1,J)=I8(I+1,J)-2
          IF(I8(I,J+2).GE.1) THEN
            I8(I,J-1)=I8(I,J-1)-2
            I8(I,J+1)=I8(I,J+1)-2
            I8(I-1,J)=I8(I-1,J)-2
          ELSE
            I8(I,J-1)=1
            I8(I,J+1)=0
            I8(I-1,J)=0
            I8(I,J)=9
          END IF
        END IF
      END IF
430   CONTINUE
      DO 435 I=2,NNI-1
      DO 435 J=2,NNJ-1
      IF(I8(I,J).GE.4) THEN
      IF(I8(I-1,J-1).GE.2.AND.I8(I-1,J).GE.3.AND.I8(I-1,J+1).GE.2)THEN
          I8(I,J)=I8(I,J)-2
          I8(I-1,J-1)=I8(I-1,J-1)-1
          I8(I-1,J+1)=I8(I-1,J+1)-1
        END IF
      IF(I8(I+1,J-1).GE.3.AND.I8(I+1,J).GE.3.AND.I8(I-1,J+1).GE.3)THEN
          I8(I,J)=I8(I,J)-2
          I8(I+1,J-1)=I8(I+1,J-1)-1
          I8(I+1,J+1)=I8(I+1,J+1)-1
        END IF
      IF(I8(I-1,J-1).GE.3.AND.I8(I,J-1).GE.3.AND.I8(I+1,J-1).GE.2)THEN
          I8(I,J)=I8(I,J)-2
          I8(I-1,J-1)=I8(I-1,J-1)-1
          I8(I+1,J-1)=I8(I+1,J-1)-1
        END IF
      IF(I8(I-1,J+1).GE.3.AND.I8(I,J+1).GE.3.AND.I8(I+1,J+1).GE.2)THEN
          I8(I,J)=I8(I,J)-2
          I8(I-1,J+1)=I8(I-1,J+1)-1
          I8(I+1,J+1)=I8(I+1,J+1)-1
        END IF
      END IF
435   CONTINUE
      DO 440 I=2,NNI-1
      DO 440 J=2,NNJ-1
      IF(I8(I,J).GE.2.AND.I8(I,J+1).GE.3.AND.I8(I+1,J).GE.3) THEN
        IF(I8(I,J-1).EQ.0) THEN
          I8(I,J+1)=I8(I,J-1)-1
          I8(I+1,J)=I8(I+1,J)-1
        END IF
```

```
      END IF
      IF(I8(I,J).GE.2.AND.I8(I-1,J).GE.3.AND.I8(I,J+1).GE.3) THEN
        IF(I8(I-1,J-1).EQ.0) THEN
          I8(I,J+1)=I8(I,J+1)-1
          I8(I-1,J)=I8(I-1,J)-1
        END IF
      END IF
      IF(I8(I,J).GE.2.AND.I8(I-1,J).GE.3.AND.I8(I,J-1).GE.3) THEN
        IF(I8(I,J+1).EQ.0) THEN
          I8(I,J-1)=I8(I,J-1)-1
          I8(I-1,J)=I8(I-1,J)-1
        END IF
      END IF
      IF(I8(I,J).GE.2.AND.I8(I+1,J).GE.3.AND.I8(I,J-1).GE.3) THEN
      IF(I8(I,J+1).EQ.0.AND.I8(I+1,J+1).EQ.0.AND.I8(I-1,J).EQ.0)THEN
          I8(I,J-1)=I8(I,J-1)-1
          I8(I+1,J)=I8(I+1,J)-1
        END IF
      END IF
440   CONTINUE
470   NBB=0
      DO 475 I=1,NNI
      DO 475 J=1,NNJ
      IF(I8(I,J).EQ.9) THEN
        ND=ND+6
        R(NR+1)=CHAR(J+NP1)
        R(NR+2)=CHAR(NNI+1-I+1+NP2)
        R(NR+3)=CHAR(3)
        NR=NR+3
        R(NR+1)=CHAR(J+NP1)
        R(NR+2)=CHAR(NNI+1-I+NP2)
        R(NR+3)=CHAR(2)
        NR=NR+3
        R(NR+1)=CHAR(J+1+NP1)
        R(NR+2)=CHAR(NNI+1-I+NP2)
        R(NR+3)=CHAR(2)
        NR=NR+3
        R(NR+1)=CHAR(J+NP1)
        R(NR+2)=CHAR(NNI+1-I+1+NP2)
        R(NR+3)=CHAR(2)
        NR=NR+3
        R(NR+1)=CHAR(J-1+NP1)
        R(NR+2)=CHAR(NNI+1-I+NP2)
        R(NR+3)=CHAR(2)
        NR=NR+3
        R(NR+1)=CHAR(J+NP1)
        R(NR+2)=CHAR(NNI+1-I+NP2)
        R(NR+3)=CHAR(2)
        NR=NR+3
        I8(I,J)=1
        GOTO 470
```

```
      END IF
475   CONTINUE
      DO 480 I=1,NNI
      DO 480 J=1,NNJ
      IF(I8(I,J).LE.0) THEN
        II=0
        JJ=0
        GOTO 480
      ELSE
        NBB=1
        NDD(1,1)=J
        NDD(1,2)=I
        NDD(1,3)=9
        I8(I,J)=I8(I,J)-1
        II=I
        JJ=J
        GOTO 490
      END IF
480   CONTINUE
490   CONTINUE
      K=1
      IF(II.GT.0.AND.JJ.GT.0) THEN
500     CONTINUE
        IF(NBB.GE.2) THEN
          I8(II,JJ)=I8(II,JJ)-2
        END IF
        GO TO (501,501,503,503,505,505,507,507) K
501   IF(JJ.LE.NNJ-1) THEN
      IF(I8(II,JJ+1).GT.0.AND.IZJ(II,JJ+1,1).NE.-1.AND.IZJ(II,JJ+1,2)
     &  .NE.-1.AND.IZJ(II,JJ,2).NE.1) THEN
          NBB=NBB+1
          NDD(NBB,1)=JJ+1
          NDD(NBB,2)=II
          NDD(NBB,3)=-1
          IZJ(II,JJ,2)=1
          JJ=JJ+1
          IZJ(II,JJ,1)=-1
          K=1
          GOTO 500
        END IF
      END IF
503   IF(II.LE.NNI-1) THEN
      IF(I8(II+1,JJ).GT.0.AND.IZJ(II+1,JJ,1).NE.-2.AND.IZJ(II+1,JJ,2)
     &  .NE.-2.AND.IZJ(II,JJ,2).NE.2) THEN
          NBB=NBB+1
          NDD(NBB,1)=JJ
          NDD(NBB,2)=II+1
          NDD(NBB,3)=-2
          IZJ(II,JJ,2)=2
          II=II+1
          IZJ(II,JJ,1)=-2
```

```
          K=3
          GOTO 500
        ELSE IF(K.EQ.3) THEN
          K=1
          GOTO 501
        END IF
      END IF
505   IF(II.LE.NNI-1.AND.JJ.GE.2) THEN
        IF(I8(II+1,JJ-1).GT.0.AND.IZJ(II+1,JJ-1,1).NE.-3.AND.
     &    IZJ(II+1,JJ-1,2).NE.-3.AND.IZJ(II,JJ,2).NE.3) THEN
          NBB=NBB+1
          NDD(NBB,1)=JJ-1
          NDD(NBB,2)=II+1
          NDD(NBB,3)=-3
          IZJ(II,JJ,2)=3
          II=II+1
          JJ=JJ-1
          IZJ(II,JJ,1)=-3
          K=5
          GOTO 500
        ELSE IF(K.EQ.5) THEN
          K=1
          GOTO 501
        END IF
      END IF
507   IF(II.LE.NNI-1.AND.JJ.LE.NNJ-1) THEN
        IF(I8(II+1,JJ+1).GT.0.AND.IZJ(II+1,JJ+1,1).NE.-4.AND.
     &    IZJ(II+1,JJ+1,2).NE.-4.AND.IZJ(II,JJ,2).NE.4) THEN
          NBB=NBB+1
          NDD(NBB,1)=JJ+1
          NDD(NBB,2)=II+1
          NDD(NBB,3)=-4
          IZJ(II,JJ,2)=4
          II=II+1
          JJ=JJ+1
          IZJ(II,JJ,1)=-4
          K=7
          GOTO 500
        ELSE IF(K.EQ.6) THEN
          K=1
          GOTO 501
        END IF
      END IF
      IF(NBB.GE.2) THEN
        I8(II,JJ)=I8(II,JJ)+1
        ND=ND+1
        R(NR+1)=CHAR(NDD(1,1)+NP1)
        R(NR+2)=CHAR(NNI+1-NDD(1,2)+NP2)
        R(NR+3)=CHAR(3)
        NR=NR+3
        DO 520 K=2,NBB-1
```

```
          IF(NDD(K,3).NE.NDD(K+1,3)) THEN
            ND=ND+1
            R(NR+1)=CHAR(NDD(K,1)+NP1)
            R(NR+2)=CHAR(NNI+1-NDD(K,2)+NP2)
            R(NR+3)=CHAR(2)
            NR=NR+3
          END IF
520       CONTINUE
          ND=ND+1
          R(NR+1)=CHAR(NDD(NBB,1)+NP1)
          R(NR+2)=CHAR(NNI+1-NDD(NBB,2)+NP2)
          R(NR+3)=CHAR(2)
          NR=NR+3
        END IF
        GOTO 470
        END IF
        IF(ND.LE.127) THEN
          R(NR1+4)=CHAR(ND)
          NR1=NR
          LK=LK+1
        ELSE
          WRITE(*,*)'   Too Many Points.   Failed.'
        END IF
        WRITE(*,*)'Continue ? (y/n)'
        READ(*,'(A)') YS
        IF(YS.EQ.'y'.OR.YS.EQ.'Y') GOTO 10
        CLOSE(3)
        WRITE(*,*)'Input Result File Name'
        READ(*,'(A)') FNAME
        OPEN(2,FILE=FNAME,ACCESS='DIRECT',RECL=NR1)
        WRITE(2,REC=1)(R(I),I=1,NR1)
        CLOSE(2)
        END
```

运行过程如下：

程序提示：

INPUT HZK NAME：

此时输入汉字库名。

再提示：INPUT NUMBER OF DOTS (16 OR 24)：

输入你所用的汉字库的点阵数，即 16 还是 24 点阵。

以后即是你想调用汉字的区位区位码：

INPUT REGION NUMBER：

每调用一汉字后，即提示：Continue(y/n)?

若要继续输入，响应为“y”，否则为“n”。若回答为“n”，程序即要将调入的汉字写成一个 SYM 文件，故还需输入带后缀 SYM 的文件名。在程序处理汉字时，若某一汉字的笔画超过 127，则程序显示“Too many points. Failed.”。表明该汉字无法写入 SYM 文件；此种情况 16×16 点阵汉字，很少出现。SYM 文件中字符代号为 ASCII 字符集中的 32—127，为

方便调用，程序中规定 65（即 A）为第一个字符代号，所以若第一个汉字码为 5448，第二个为 2710，则在产生的字符集中，“A”代表“中”、“B”代表“华”，依次类推。

2. SYM 形式的矢量汉字

由于 SYM 文件的字符形式是矢量型的，故将矢量汉字写成 SYM 文件相对较简单些。然而由于 SYM 文件点数的限制（不多于 127），故笔画多的汉字无法显示。但对于某些字体比如黑体，大部分汉字还是可显示的。限于篇幅，具体程序从略。

（二） DAT 形式调用汉字

若将汉字生成 DAT 文件，即可用 Grapher 将汉字画出。用此种形式调用矢量汉字，可将汉字写成空心体。

```
      DIMENSION KL(759)
      INTEGER*2  KBB(600),KBB2(600)
      COMMON /HZ/XDATA(40000),YDATA(40000),X(500),Y(500)
      CHARACTER B*3,BB*1,FNAME*24,R*1,MZ*2
      WRITE(*,*) ' Input Name of Word library '
      READ(*,'(A)') FNAME
      OPEN(3,FILE=FNAME,ACCESS='DIRECT',RECL=1)
      UNITL=4
C   UNITL:  Link-line Length of Arc
      XSCALE=1
      YSCALE=1
C   XSCALE,YSCALE: Scale in X and Y Direction.
      WRITE(*,*) ' Input Angle of Words'
      READ(*,*) ANGLE
      WRITE(*,*) ' Input Angle of Lineup'
      READ(*,*) ANGLE2
      PI=ATAN(1.0)*4.0
      ANGLE=ANGLE*PI/180.
      ANGLE2=ANGLE2*PI/180.
      NDD=0
      XP0=200*COS(ANGLE2)*XSCALE
      XP=-XP0
      YP0=200*SIN(ANGLE2)*YSCALE
      YP=-YP0
      DO 10 I=1,24
      IF(FNAME(I:I).EQ.' ') I1=I
   10 CONTINUE
      BB=FNAME(I1-5:I1-5)
      IF(ICHAR(BB).GE.97.AND.ICHAR(BB).LE.122) THEN
        MZ(1:1)=CHAR(ICHAR(BB)-32)
      ELSE
        MZ(1:1)=CHAR(ICHAR(BB))
      END IF
      BB=FNAME(I1-4:I1-4)
      IF(ICHAR(BB).GE.97.AND.ICHAR(BB).LE.122) THEN
        MZ(2:2)=CHAR(ICHAR(BB)-32)
      ELSE
        MZ(2:2)=CHAR(ICHAR(BB))
```

```
      END IF
      IF(MZ.EQ.'HT'.OR.MZ.EQ.'KT'.OR.MZ.EQ.'FT'.OR.MZ.EQ.'ST') THEN
        KK=6768*3
        DO 20 J=1,3
        KKJ=KK+J
        READ(3,REC=KKJ) BB
20      B(J,J)=BB
        I1=ICHAR(B(1,1))
        I2=ICHAR(B(2,2))
        I3=ICHAR(B(3,3))
        KK=I1+I2*256+I3*256*256
        DO 50 I=1,759
        DO 40 J=1,3
        KKJ=KK+J+(I-1)*3
        READ(3,REC=KKJ) BB
40      B(J,J)=BB
        I1=ICHAR(B(1,1))
        I2=ICHAR(B(2,2))
        I3=ICHAR(B(3,3))
50      KL(I)=I1+I2*256+I3*256*256+KK
      END IF
70    WRITE(*,*) ' Input Region Number(NUM1)'
      READ(*,*) NUM1
      I1=NUM1/100
      I2=NUM1-I1*100
      IF(I1.LT.16.OR.I1.GT.87.OR.I2.LT.1.OR.I2.GT.94) THEN
      WRITE(*,*) ' Region Number Error. Input Again.'
        GOTO 70
      END IF
      XP=XP+XP0
      YP=YP+YP0
      KKK=(I1-16)*94+I2
      KK=(KKK-1)*3
      DO 90 J=1,3
      KKJ=KK+J
      READ(3,REC=KKJ) BB
90    B(J,J)=BB
      I1=ICHAR(B(1,1))
      I2=ICHAR(B(2,2))
      I3=ICHAR(B(3,3))
      LL1=I1+I2*256+I3*256*256+1
      DO 110 J=1,3
      KKJ=KK+3+J
      READ(3,REC=KKJ) BB
110   B(J,J)=BB
      I1=ICHAR(B(1,1))
      I2=ICHAR(B(2,2))
      I3=ICHAR(B(3,3))
      LL2=I1+I2*256+I3*256*256
      NK=0
      DO 130 I=LL1,LL2
```

```
      READ(3,REC=I) BB
      I1=ICHAR(BB)
      I2=I1/16
      I3=I1-I2*16
      KBB(NK+1)=I2
      KBB(NK+2)=I3
130   NK=NK+2
      NTT=1
150   CONTINUE
      IF(KBB(NTT).LE.6.AND.NTT.LT.NK) THEN
        CALL WHZ(KBB,NDD,NK,NTT,0,0,1.,1.,
     &              UNITL,XP,YP,XSCALE,YSCALE,ANGLE)
        GOTO 150
      ELSE IF(KBB(NTT).GT.6.AND.NTT.LT.NK) THEN
        I8=KBB(NTT)
        IF(I8.EQ.8) THEN
          NB1=KBB(NTT+1)*16+KBB(NTT+2)
          IX=KBB(NTT+3)*16+KBB(NTT+4)
          IY=KBB(NTT+5)*16+KBB(NTT+6)
          IB1=KBB(NTT+7)*16+KBB(NTT+8)
          IB2=KBB(NTT+9)*16+KBB(NTT+10)
          NTT=NTT+11
          II8=NB1-7
        ELSE
          NB1=KBB(NTT+9)*16+KBB(NTT+10)
          IX=KBB(NTT+1)*16+KBB(NTT+2)
          IY=KBB(NTT+3)*16+KBB(NTT+4)
          IB1=KBB(NTT+5)*16+KBB(NTT+6)
          IB2=KBB(NTT+7)*16+KBB(NTT+8)
          NTT=NTT+11
            IF(I8.EQ.9) THEN
              II8=247+NB1
            ELSE
              II8=247+256+NB1
            END IF
        END IF
        NK2=0
        LL1=KL(II8)+1
        LL2=KL(II8+1)
        DO 170 I=LL1,LL2
        READ(3,REC=I) BB
        I1=ICHAR(BB)
        I2=I1/16
        I3=I1-I2*16
        KBB2(NK2+1)=I2
        KBB2(NK2+2)=I3
170     NK2=NK2+2
        SCALE1=IB1/100.
        SCALE2=IB2/100.
        NTT2=1
        CALL WHZ(KBB2,NDD,NK2,NTT2,IX,IY,SCALE1,SCALE2,
```

```
     &     UNITL,XP,YP,XSCALE,YSCALE,ANGLE)
           GOTO 150
         END IF
         WRITE(*,*) ' Number of Point =',NDD
         WRITE(*,*) ' Continue ? (y/n)'
         READ(*,'(A)') YS
         IF(YS.EQ.'y'.OR.YS.EQ.'Y') GOTO 70
         CLOSE(3)
         WRITE(*,*) ' Input *.dat File Name'
         READ(*,'(A)') FNAME
         OPEN(2,FILE=FNAME)
         DO 200 I=1,NDD
  200    WRITE(2,210) XDATA(I),YDATA(I)
  210    FORMAT(F9.2,2X,F9.2)
         CLOSE(2)
         end

         SUBROUTINE  WHZ(KBB,NDD,NK,NTT,IX0,IY0,SCALE1,SCALE2,
     &   UNITL,XP,YP,XSCALE,YSCALE,ANGLE)
         INTEGER*2 KBB(600)
         REAL X0(4),Y0(4)
         COMMON /HZ/XDATA(40000),YDATA(40000),X(500),Y(500)
   20    CONTINUE
         IF(KBB(NTT).EQ.0.AND.NTT.LT.NK-3) THEN
           XI0=((KBB(NTT+1)*16+KBB(NTT+2))*SCALE1+IX0-90)*XSCALE
           YI0=(-1*((KBB(NTT+3)*16+KBB(NTT+4))*SCALE2+IY0-90))*YSCALE
           CALL ROTATE(XI,YI,XI0,YI0,ANGLE)
           IF(NDD.GT.0) THEN
             XDATA(NDD+1)=-90000
             YDATA(NDD+1)=-90000
             XDATA(NDD+2)=XI+XP
             YDATA(NDD+2)=YI+YP
             NDD=NDD+2
           END IF
           NDD=NDD+1
           XDATA(NDD)=XI+XP
           YDATA(NDD)=YI+YP
           NTT=NTT+5
         ELSE IF(KBB(NTT).EQ.1.AND.NTT.LE.NK-2) THEN
           NDD=NDD+1
           XI0=((KBB(NTT+1)*16+KBB(NTT+2))*SCALE1+IX0-90)*XSCALE
           CALL ROTATE(XI,YI,XI0,YI0,ANGLE)
           XDATA(NDD)=XI+XP
           YDATA(NDD)=YI+YP
           NTT=NTT+3
         ELSE IF(KBB(NTT).EQ.2.AND.NTT.LE.NK-2) THEN
           NDD=NDD+1
           YI0=(-1*((KBB(NTT+1)*16+KBB(NTT+2))*SCALE2+IY0-90))*YSCALE
           CALL ROTATE(XI,YI,XI0,YI0,ANGLE)
           XDATA(NDD)=XI+XP
           YDATA(NDD)=YI+YP
```

```
      NTT=NTT+3
    ELSE IF(KBB(NTT).EQ.3.AND.NTT.LE.NK-4) THEN
      NDD=NDD+1
      XI0=((KBB(NTT+1)*16+KBB(NTT+2))*SCALE1+IX0-90)*XSCALE
      YI0=(-1*((KBB(NTT+3)*16+KBB(NTT+4))*SCALE2+IY0-90))*YSCALE
      CALL ROTATE(XI,YI,XI0,YI0,ANGLE)
      XDATA(NDD)=XI+XP
      YDATA(NDD)=YI+YP
      NTT=NTT+2
    ELSE IF(KBB(NTT).EQ.4.AND.NTT.LE.NK-8) THEN
      X0(1)=XI
      Y0(1)=YI
      XI0=((KBB(NTT+1)*16+KBB(NTT+2))*SCALE1+IX0-90)*XSCALE
      YI0=(-1*((KBB(NTT+3)*16+KBB(NTT+4))*SCALE2+IY0-90))*YSCALE
      CALL ROTATE(XI,YI,XI0,YI0,ANGLE)
      X0(2)=XI
      Y0(2)=YI
      NTT=NTT+4
      XI0=((KBB(NTT+1)*16+KBB(NTT+2))*SCALE1+IX0-90)*XSCALE
      YI0=(-1*((KBB(NTT+3)*16+KBB(NTT+4))*SCALE2+IY0-90))*YSCALE
      CALL ROTATE(XI,YI,XI0,YI0,ANGLE)
      X0(3)=XI
      Y0(3)=YI
      CALL BEZIER(X0,Y0,3,N1,UNITL)
      DO 40 I=2,N1
      NDD=NDD+1
      XDATA(NDD)=X(I)+XP
      YDATA(NDD)=Y(I)+YP
40    CONTINUE
      NTT=NTT+5
    ELSE IF(KBB(NTT).EQ.5.AND.NTT.LE.NK-12) THEN
      X0(1)=XI
      Y0(1)=YI
      XI0=((KBB(NTT+1)*16+KBB(NTT+2))*SCALE1+IX0-90)*XSCALE
      YI0=(-1*((KBB(NTT+3)*16+KBB(NTT+4))*SCALE2+IY0-90))*YSCALE
      CALL ROTATE(XI,YI,XI0,YI0,ANGLE)
      X0(2)=XI
      Y0(2)=YI
      NTT=NTT+4
      XI0=((KBB(NTT+1)*16+KBB(NTT+2))*SCALE1+IX0-90)*XSCALE
      YI0=(-1*((KBB(NTT+3)*16+KBB(NTT+4))*SCALE2+IY0-90))*YSCALE
      CALL ROTATE(XI,YI,XI0,YI0,ANGLE)
      X0(3)=XI
      Y0(3)=YI
      NTT=NTT+4
      XI0=((KBB(NTT+1)*16+KBB(NTT+2))*SCALE1+IX0-90)*XSCALE
      YI0=(-1*((KBB(NTT+3)*16+KBB(NTT+4))*SCALE2+IY0-90))*YSCALE
      CALL ROTATE(XI,YI,XI0,YI0,ANGLE)
      X0(4)=XI
      Y0(4)=YI
      CALL BEZIER(X0,Y0,4,N1,UNITL)
```

```
      DO 60 I=2,N1
      NDD=NDD+1
      XDATA(NDD)=X(I)+XP
      YDATA(NDD)=Y(I)+YP
60    CONTINUE
      NTT=NTT+5
    ELSE IF(KBB(NTT).EQ.6.AND.NTT.LE.NK-8) THEN
      XI01=((KBB(NTT+1)*16+KBB(NTT+2))*SCALE1+IX0-90)*XSCALE
      YI01=(-1*((KBB(NTT+3)*16+KBB(NTT+4))*SCALE2+IY0-90))*YSCALE
      XI02=((KBB(NTT+5)*16+KBB(NTT+6))*SCALE1+IX0-90)*XSCALE
      YI02=(-1*((KBB(NTT+7)*16+KBB(NTT+8))*SCALE2+IY0-90))*YSCALE
      CALL ROTATE(XI1,YI1,XI01,YI01,ANGLE)
      CALL ROTATE(XI2,YI2,XI01,YI02,ANGLE)
      CALL ROTATE(XI3,YI3,XI02,YI02,ANGLE)
      CALL ROTATE(XI4,YI4,XI02,YI01,ANGLE)
      IF(NDD.GT.0) THEN
        XDATA(NDD+1)=-90000
        YDATA(NDD+1)=-90000
        XDATA(NDD+2)=XI1+XP
        YDATA(NDD+2)=YI1+YP
        NDD=NDD+2
      END IF
      NDD=NDD+5
      XDATA(NDD-4)=XI1+XP
      YDATA(NDD-4)=YI1+YP
      XDATA(NDD-3)=XI2+XP
      YDATA(NDD-3)=YI2+YP
      XDATA(NDD-2)=XI3+XP
      YDATA(NDD-2)=YI3+YP
      XDATA(NDD-1)=XI4+XP
      YDATA(NDD-1)=YI4+YP
      XDATA(NDD)=XI1+XP
      YDATA(NDD)=YI1+YP
      NTT=NTT+9
    ELSE
      GOTO 80
    END IF
    GOTO 20
80  CONTINUE
    END

    SUBROUTINE BEZIER(X0,Y0,M,N,UNITL)
    COMMON /HZ/XDATA(40000),YDATA(40000),X(500),Y(500)
    REAL C(4),X0(4),Y0(4)
    X(1)=X0(1)
    Y(1)=Y0(1)
    N=1
    IF(M.EQ.3) THEN
      D=SQRT((X0(1)-X0(3))**2+(Y0(1)-Y0(3))**2)
      N1=D/UNITL+1
      DO 40 I=1,N1-1
```

```
      T=(I+0.)/N1
      C(1)=1.-2.*T+T*T
      C(2)=2.*T-2.*T*T
      C(3)=T*T
      N=N+1
      S1=0.
      S2=0.
      DO 20 J=1,3
      S1=S1+C(J)*X0(J)
20    S2=S2+C(J)*Y0(J)
      X(N)=S1+0.5
40    Y(N)=S2+0.5
      N=N+1
      X(N)=X0(3)
      Y(N)=Y0(3)
    ELSE
      D=SQRT((X0(1)-X0(4))**2+(Y0(1)-Y0(4))**2)
      N1=D/UNITL+1
      DO 80 I=1,N1-1
      T=(I+0.)/N1
      T2=T*T
      T3=T2*T
      C(1)=1.-3.*T+3.*T2-T3
      C(2)=3.*T3-6.*T2+3.*T
      C(3)=3.*T2-3.*T3
      C(4)=T3
      N=N+1
      S1=0.
      S2=0.
      DO 60 J=1,4
      S1=S1+C(J)*X0(J)
60    S2=S2+C(J)*Y0(J)
      X(N)=S1+0.5
80    Y(N)=S2+0.5
      N=N+1
      X(N)=X0(4)
      Y(N)=Y0(4)
    END IF
    END

    SUBROUTINE ROTATE(X,Y,X0,Y0,ANGLE)
    X=X0*COS(ANGLE)-Y0*SIN(ANGLE)
    Y=X0*SIN(ANGLE)+Y0*COS(ANGLE)
    END
```

该程序包括一个主程序和三个子程序。子程序 WHZ 计算汉字各点的数值，子程序 BEZIER 利用 Bezier 曲线插值计算圆弧，子程序 ROTATE 用于旋转字的角度。

该程序生成的汉字为空心状，有以下几个特点：①可无级放大；②可自定汉字角度（使之倾斜）；③可自定一组汉字的排列角度。所以由此产生的汉字较美观，即可作为标题，

也可用于图中注记。

程序运行时，依提示分别输入汉字库名、字的角度和排列角度（逆时针为正），然后即输入你想调用的每个汉字的区位码；汉字输入完毕后再给出带后缀 DAT 的文件名。

产生结果即为一 DAT 文件，该文件有二列数据，分别为 x，y 坐标值。每个汉字的基准网格为 180×180，在用 Grapher 软件作用时可调整坐标轴长度，使之放大或缩小。同时，程序中笔画间的中断点值为（−90000，−90000），故在用 Grapher 画汉字时应使用 DATA LIMITS 项，使之限定在大于−90000 的范围内。

该程序调用的汉字库为 WPS 中的各种字体的矢量库。

三、 在 C 语言中标注汉字

在 C 语言编写的软件中标注汉字比较方便，既可采取点阵直接显示方式，也可采取与在 Gragpher 软件中 SYM 方式标注汉字相类似的方法，即汉字作为 C 语言的一个字符集供 C 语言调用输出。本节讨论后一种情况。

以 Turbo C 为例，C 语言中已有的字符集有 5 种，对应 5 种不同的字体。解剖其中的一种，将汉字亦写成类似的形式，即可产生一个新的字符集，这里我们选择 TRIP.CHR 来进行分析。C 语言中的字符集亦是采用矢量方式勾画字符笔画，矢量方式描述字符（包括汉字）需指明的每个字符的特性为：①字符高度和宽度，由此确定基本坐标系统；②字符数据的起点和终点地址，由此可正确地取出数据；③各线条的起始点，由此可确定各线条的起笔点。经分析，C 语言字符集中的 0—127 字节为文件头，其中 96—97 两个字节构成的整型数的值代表该字符集的文件长度减去 128 即文件头后的文件长度。每个字符集可容纳 ASCII 码 32—126 共 95 个字符，一般 ASCII 码 32 用于空格，故实际可包含 94 个字符。在同一个字符集中，各字符高度是相同的，由第 136 和第 138 两个字节联合决定。下面为 TRIP.CHR 中第 136 和 138 两个字节值与高度关系表（表 5—1）。

表 5—1 TRIP.CHR 中的字符高度控制表

第 136 字节	第 138 节	字符高度 H
01	F9	8
01	F9	9
08	F9	15
⋮	⋮	⋮
15	FB	26
15	FA	27
15	F9	28
⋮	⋮	⋮

由此不难对字符高度进行控制。字符宽度由第 334—428 字节决定，即第 335 字节决定第 32 号字符的宽度，第 336 字节对应第 33 号字符，第 428 字节对应第 126 号字符。

第 429 字节以后为字符数据内容，并将第 429 字节起始地址定为零。每个字符的起始地址位于第 144—333 字节区，每个字符占二个字节。

在字符数据中，依次为 x_1，y_1，x_2，y_2，…，x_n，y_n，中间无其它的控制符。这些 x 坐标都加了一个作为 128 的偏移量，故实际 x 坐标应为这些坐标减去 128。y 坐标有二种情况，若值小于 128，为起始点，若大于 128，为非起始点。非起始点的 y 坐标实际值也应减去 128，所以，由 y 值的大小可知该点是起始点还是非起始点。若连续二个 y 值为零，表明该字符到此结束。

通过以上分析，我们就可将汉字数据写成 C 语言的字符集，再在 C 语言的头文件中添加新的字符集，即达到在 C 语言中调用汉字的目的。

下面即为将矢量汉字写成 C 语言字符集的 FORTRAN 程序。

```
      DIMENSION KL(759),R(20000),X(500),Y(500) '
      INTEGER * 2   KBB(600),KBB2(600)
      CHARACTER B * 3,BB * 1,FNAME * 24,R * 1,MZ * 2
      CHARACTER F * 1,FF * 16
      INTEGER * 2 ADDR(95)
      OPEN(3,FILE='TRIP.CHR',ACCESS='DIRECT',RECL=16)
      K=0
      DO 10 J=1,9
      READ(3,REC=J) FF
      DO 10 I=1,16
      F=FF(I:I)
      K=K+1
10    R(K)=F
      CLOSE(3)
      WRITE( * , * ) ' Input name of word lib.'
      READ( * ,'(A)') FNAME
      OPEN(3,FILE=FNAME,ACCESS='DIRECT',RECL=1)
      UNITL=4
C     UNITL:  Link-line Length of Arc
      XSCALE=0.25
      YSCALE=0.25
C     XSCALE,YSCALE: Scale in X and Y Direction.
      WRITE( * , * ) ' Input angle of words'
      READ( * , * ) ANGLE
      PI=ATAN(1.0) * 4.0
      ANGLE=ANGLE * PI/180.
      IW=ABS(180. * COS(ANGLE) * XSCALE)+ABS(180. * SIN(ANGLE) * YSCALE)
      R(137)=CHAR(IW-24)
      R(139)=CHAR(232)
      DO 20 I=335,429
20    R(I)=CHAR(IW)
      R(430)=CHAR(144)
      R(431)=CHAR(0)
      R(432)=CHAR(0)
      R(433)=CHAR(0)
      R(145)=CHAR(0)
      R(146)=CHAR(0)
      ADDR(1)=4
      NR=433
```

```
      DO 30 I=1,24
      IF(FNAME(I:I).EQ.'.') I1=I
30    CONTINUE
      BB=FNAME(I1-5:I1-5)
      IF(ICHAR(BB).GE.97.AND.ICHAR(BB).LE.122) THEN
        MZ(1:1)=CHAR(ICHAR(BB)-32)
      ELSE
        MZ(1:1)=CHAR(ICHAR(BB))
      END IF
      BB=FNAME(I1-4:I1-4)
      IF(ICHAR(BB).GE.97.AND.ICHAR(BB).LE.122) THEN
        MZ(2:2)=CHAR(ICHAR(BB)-32)
      ELSE
        MZ(2:2)=CHAR(ICHAR(BB))
      END IF
      IF(MZ.EQ.'HT'.OR.MZ.EQ.'KT'.OR.MZ.EQ.'FT'.OR.MZ.EQ.'ST') THEN
        KK=6768*3
        DO 40 J=1,3
        KKJ=KK+J
        READ(3,REC=KKJ) BB
40      B(J:J)=BB
        I1=ICHAR(B(1:1))
        I2=ICHAR(B(2:2))
        I3=ICHAR(B(3:3))
        KK=I1+I2*256+I3*256*256
        DO 60 I=1,759
        DO 50 J=1,3
        KKJ=KK+J+(I-1)*3
        READ(3,REC=KKJ) BB
50      B(J:J)=BB
        I1=ICHAR(B(1:1))
        I2=ICHAR(B(2:2))
        I3=ICHAR(B(3:3))
60      KL(I)=I1+I2*256+I3*256*256+KK
      END IF
      LK=0
      ND=0
      NUMT=0
70    WRITE(*,*) ' Input Region Number(NUM1)'
      READ(*,*) NUM1
      I1=NUM1/100
      I2=NUM1-I1*100
      IF(I1.LT.16.OR.I1.GT.87.OR.I2.LT.1.OR.I2.GT.94) THEN
        WRITE(*,*) ' Region Number Error. Input Again.'
        GOTO 70
      END IF
      LK=LK+1
      NUMT=NUMT+1
      I1=I1-16
      KKK=I1*94+I2
      KK=(KKK-1)*3
```

```
      DO 90 J=1,3
      KKJ=KK+J
      READ(3,REC=KKJ) BB
 90   B(J,J)=BB
      I1=ICHAR(B(1,1))
      I2=ICHAR(B(2,2))
      I3=ICHAR(B(3,3))
      LL1=I1+I2*256+I3*256*256+1
      DO 110 J=1,3
      KKJ=KK+3+J
      READ(3,REC=KKJ) BB
110   B(J,J)=BB
      I1=ICHAR(B(1,1))
      I2=ICHAR(B(2,2))
      I3=ICHAR(B(3,3))
      LL2=I1+I2*256+I3*256*256
      NK=0
      DO 130 I=LL1,LL2
      READ(3,REC=I) BB
      I1=ICHAR(BB)
      I2=I1/16
      I3=I1-I2*16
      KBB(NK+1)=I2
      KBB(NK+2)=I3
130   NK=NK+2
      ND=0
      NTT=1
150   CONTINUE
      IF(KBB(NTT).LE.6.AND.NTT.LT.NK) THEN
        CALL WHZC(KBB,R,NR,ND,NK,NTT,0,0,1.,1.,
     &              UNITL,XSCALE,YSCALE,ANGLE,X,Y)
        GOTO 150
      ELSE IF(KBB(NTT).GT.6.AND.NTT.LT.NK) THEN
        I8=KBB(NTT)
        IF(I8.EQ.8) THEN
          NB1=KBB(NTT+1)*16+KBB(NTT+2)
          IX=KBB(NTT+3)*16+KBB(NTT+4)
          IY=KBB(NTT+5)*16+KBB(NTT+6)
          IB1=KBB(NTT+7)*16+KBB(NTT+8)
          IB2=KBB(NTT+9)*16+KBB(NTT+10)
          NTT=NTT+11
          II8=NB1-7
        ELSE
          NB1=KBB(NTT+9)*16+KBB(NTT+10)
          IX=KBB(NTT+1)*16+KBB(NTT+2)
          IY=KBB(NTT+3)*16+KBB(NTT+4)
          IB1=KBB(NTT+5)*16+KBB(NTT+6)
          IB2=KBB(NTT+7)*16+KBB(NTT+8)
          NTT=NTT+11
            IF(I8.EQ.9) THEN
              II8=247+NB1
```

```
          ELSE
            II8=247+256+NB1
          END IF
        END IF
        NK2=0
        LL1=KL(II8)+1
        LL2=KL(II8+1)
        DO 170 I=LL1,LL2
        READ(3,REC=I) BB
        I1=ICHAR(BB)
        I2=I1/16
        I3=I1-I2*16
        KBB2(NK2+1)=I2
        KBB2(NK2+2)=I3
170     NK2=NK2+2
        SCALE1=IB1/100.
        SCALE2=IB2/100.
        NTT2=1
        CALL WHZC(KBB2,R,NR,ND,NK2,NTT2,IX,IY,SCALE1,SCALE2,
     &                UNITL,XSCALE,YSCALE,ANGLE,X,Y)
        GOTO 150
      END IF
      WRITE(*,*) ' number of point=',nd
      R(NR+1)=CHAR(0)
      R(NR+2)=CHAR(0)
      R(NR+3)=CHAR(0)
      R(NR+4)=CHAR(0)
      NR=NR+4
      ADDR(LK+1)=NR+1-430
      WRITE(*,*) ' Continue ? (y/n)'
      READ(*,'(A)') YS
      IF(YS.EQ.'y'.OR.YS.EQ.'Y') GOTO 70
      CLOSE(3)
      I1=(NR-128)/256
      I2=NR-128-I1*256
      R(97)=CHAR(I2)
      R(98)=CHAR(I1)
      LK=0
      DO 220 NO=1,NUMT
      LK=LK+1
      I1=ADDR(LK)/256
      I2=ADDR(LK)-I1*256
      K=LK*2
      R(146+K-1)=CHAR(I2)
220   R(146+K)=CHAR(I1)
      WRITE(*,*) ' NR=',NR
      WRITE(*,*) ' Input Result File Name'
      READ(*,'(A)') FNAME
      OPEN(2,FILE=FNAME,ACCESS='DIRECT',RECL=NR)
      WRITE(2,REC=1)(R(I),I=1,NR)
      CLOSE(2)
```

```
      END

      SUBROUTINE  WHZC(KBB,R,NR,ND,NK,NTT,IX0,IY0,SCALE1,SCALE2,
     &  UNITL,XSCALE,YSCALE,ANGLE,X,Y)
      INTEGER * 2 KBB(600)
      REAL X0(4),Y0(4),X(500),Y(500)
      CHARACTER R(20000) * 1
20    CONTINUE
      IF(KBB(NTT). EQ. 0. AND. NTT. LT. NK-3) THEN
        ND=ND+2
        XI0=((KBB(NTT+1) * 16+KBB(NTT+2)) * SCALE1+IX0-90) * XSCALE
        YI0=(90-((KBB(NTT+3) * 16+KBB(NTT+4)) * SCALE2+IY0)) * YSCALE
        CALL ROTATE(XI,YI,XI0,YI0,ANGLE)
        IX=XI+90 * XSCALE
        IY=YI+90 * YSCALE
        R(NR+1)=CHAR(IX+128)
        R(NR+2)=CHAR(IY)
        NR=NR+2
        NTT=NTT+5
      ELSE IF(KBB(NTT). EQ. 1. AND. NTT. LE. NK-2) THEN
        ND=ND+2
        XI0=((KBB(NTT+1) * 16+KBB(NTT+2)) * SCALE1+IX0-90) * XSCALE
        CALL ROTATE(XI,YI,XI0,YI0,ANGLE)
        IX=XI+90 * XSCALE
        IY=YI+90 * YSCALE
        R(NR+1)=CHAR(IX+128)
        R(NR+2)=CHAR(IY+128)
        NR=NR+2
        NTT=NTT+3
      ELSE IF(KBB(NTT). EQ. 2. AND. NTT. LE. NK-2) THEN
        ND=ND+2
        YI0=(90-((KBB(NTT+1) * 16+KBB(NTT+2)) * SCALE2+IY0)) * YSCALE
        CALL ROTATE(XI,YI,XI0,YI0,ANGLE)
        IX=XI+90 * XSCALE
        IY=YI+90 * YSCALE
        R(NR+1)=CHAR(IX+128)
        R(NR+2)=CHAR(IY+128)
        NR=NR+2
        NTT=NTT+3
      ELSE IF(KBB(NTT). EQ. 3. AND. NTT. LE. NK-4) THEN
        ND=ND+2
        XI0=((KBB(NTT+1) * 16+KBB(NTT+2)) * SCALE1+IX0-90) * XSCALE
        YI0=(90-((KBB(NTT+3) * 16+KBB(NTT+4)) * SCALE2+IY0)) * YSCALE
        CALL ROTATE(XI,YI,XI0,YI0,ANGLE)
        IX=XI+90 * XSCALE
        IY=YI+90 * YSCALE
        R(NR+1)=CHAR(IX+128)
        R(NR+2)=CHAR(IY+128)
        NR=NR+2
        NTT=NTT+5
      ELSE IF(KBB(NTT). EQ. 4. AND. NTT. LE. NK-8) THEN
```

```
      X0(1)=XI
      Y0(1)=YI
      XI0=((KBB(NTT+1)*16+KBB(NTT+2))*SCALE1+IX0-90)*XSCALE
      YI0=(90-((KBB(NTT+3)*16+KBB(NTT+4))*SCALE2+IY0))*YSCALE
      CALL ROTATE(XI,YI,XI0,YI0,ANGLE)
      X0(2)=XI
      Y0(2)=YI
      NTT=NTT+4
      XI0=((KBB(NTT+1)*16+KBB(NTT+2))*SCALE1+IX0-90)*XSCALE
      YI0=(90-((KBB(NTT+3)*16+KBB(NTT+4))*SCALE2+IY0))*YSCALE
      CALL ROTATE(XI,YI,XI0,YI0,ANGLE)
      X0(3)=XI
      Y0(3)=YI
      CALL BEZIER(X0,Y0,3,N1,UNITL,X,Y)
      DO 40 I=2,N1
      ND=ND+2
      IX1=X(I)+90*XSCALE
      IY1=Y(I)+90*YSCALE
      R(NR+1)=CHAR(IX1+128)
      R(NR+2)=CHAR(IY1+128)
40    NR=NR+2
      NTT=NTT+5
    ELSE IF(KBB(NTT).EQ.5.AND.NTT.LE.NK-12) THEN
      X0(1)=XI
      Y0(1)=YI
      XI0=((KBB(NTT+1)*16+KBB(NTT+2))*SCALE1+IX0-90)*XSCALE
      YI0=(90-((KBB(NTT+3)*16+KBB(NTT+4))*SCALE2+IY0))*YSCALE
      CALL ROTATE(XI,YI,XI0,YI0,ANGLE)
      X0(2)=XI
      Y0(2)=YI
      NTT=NTT+4
      XI0=((KBB(NTT+1)*16+KBB(NTT+2))*SCALE1+IX0-90)*XSCALE
      YI0=(90-((KBB(NTT+3)*16+KBB(NTT+4))*SCALE2+IY0))*YSCALE
      CALL ROTATE(XI,YI,XI0,YI0,ANGLE)
      X0(3)=XI
      Y0(3)=YI
      NTT=NTT+4
      XI0=((KBB(NTT+1)*16+KBB(NTT+2))*SCALE1+IX0-90)*XSCALE
      YI0=(90-((KBB(NTT+3)*16+KBB(NTT+4))*SCALE2+IY0))*YSCALE
      CALL ROTATE(XI,YI,XI0,YI0,ANGLE)
      X0(4)=XI
      Y0(4)=YI
      CALL BEZIER(X0,Y0,4,N1,UNITL,X,Y)
      DO 60 I=2,N1
      ND=ND+2
      IX1=X(I)+90*XSCALE
      IY1=Y(I)+90*YSCALE
      R(NR+1)=CHAR(IX1+128)
      R(NR+2)=CHAR(IY1+128)
60    NR=NR+2
      NTT=NTT+5
```

```
      ELSE IF(KBB(NTT).EQ.6.AND.NTT.LE.NK-8) THEN
        ND=ND+10
        XI01=((KBB(NTT+1)*16+KBB(NTT+2))*SCALE1+IX0-90)*XSCALE
        YI01=(90-((KBB(NTT+3)*16+KBB(NTT+4))*SCALE2+IY0))*YSCALE
        XI02=((KBB(NTT+5)*16+KBB(NTT+6))*SCALE1+IX0-90)*XSCALE
        YI02=(90-((KBB(NTT+7)*16+KBB(NTT+8))*SCALE2+IY0))*YSCALE
        CALL ROTATE(XI1,YI1,XI01,YI01,ANGLE)
        IX1=XI1+90*XSCALE
        IY1=YI1+90*YSCALE
        CALL ROTATE(XI2,YI2,XI01,YI02,ANGLE)
        IX2=XI2+90*XSCALE
        IY2=YI2+90*YSCALE
        CALL ROTATE(XI3,YI3,XI02,YI02,ANGLE)
        IX3=XI3+90*XSCALE
        IY3=YI3+90*YSCALE
        CALL ROTATE(XI4,YI4,XI02,YI01,ANGLE)
        IX4=XI4+90*XSCALE
        IY4=YI4+90*YSCALE
        R(NR-1)=CHAR(IX1+128)
        R(NR+2)=CHAR(IY1)
        NR=NR+2
        R(NR+1)=CHAR(IX2+128)
        R(NR+2)=CHAR(IY2+128)
        NR=NR+2
        R(NR-1)=CHAR(IX3+128)
        R(NR+2)=CHAR(IY3-128)
        NR=NR+2
        R(NR+1)=CHAR(IX4+128)
        R(NR+2)=CHAR(IY4+128)
        NR=NR+2
        R(NR+1)=CHAR(IX1+128)
        R(NR+2)=CHAR(IY1+128)
        NR=NR+2
        NTT=NTT+9
      ELSE
        GOTO 80
      END IF
      GOTO 20
80    CONTINUE
      END
```

上述程序包括一个主程序和一个子程序 WHZC；子程序 WHZC 用于计算汉字各点的数值。除此以外，还须调用子程序 BEZIER 和 ROTATE（见 75－76 页）。

程序运行时，依提示分别输入汉字库名（指 WPS 中的矢量库），字的角度（逆时针为正），然后输入你想调用的每个汉字的区位码；汉字输入完毕后再给出带后缀 CHR 的文件名。

因程序运行时须打开 TRIP. CHR 文件以读入它的结构，故须将 TRIP. CHR 拷入当前目录中。

四、pcARC/INFO 中的汉字注记

如前所述，pcARC/INFO 为用户留下的数据接口主要有两类：一是GRIDPOLY命令用以处理整型栅格数据；另一类为GENERATE命令用以处理矢量数据。基于这一要求，汉字注记时所用的数据格式也必须以此两个命令的限定格式为基础进行转换。此外考虑到注记的实用要求，还必须区分出实体汉字与空心汉字，并充分考虑其在pcARC/INFO中汉字的存放格式，以确保调用方便。

本节讨论用点阵汉字、矢量汉字进行注记的可行性和具体实施方案，并附有用FORTRAN语言编制调试的相关程序，从而为在pcARC/INFO中各种字体汉字的任意调用提供一条切实可行的路径。

（一）点阵汉字的转换

1. 点阵式数据格式的转换

所谓点阵式数据格式的转换，是指转换后仍保留其点阵数据特点一种转换方式。转换的关键在于确定出点阵数据中每个非零点位的坐标，并使之严格按照pcARC/INFO中相关命令的格式要求存放。根据pcARC/INFO提供的数据接口，只有GENERATE命令可以完成此项转换。pcARC/INFO的GENERATE命令可接受五种不同的数据格式（表 4—1）。但汉字中的点既不属于单个注记点，也不属于独立的点要素，在进行数据格式转换时既要保证符合点阵式汉字的特点，又要满足GENERATE的特殊格式要求，就必须在具体操作中进行某种方式的变通。实验表明，比较简便易行的办法，就是直接将点阵汉字中非零点的坐标值视为一组散点数据，使其作为一个独立文件存盘，然后作为点数据调用GENERATE命令。除此而外，将各点视为长度为 0 的“线”来处理，即每条线要素中只有起点和终点，即只含F __ NODE和 T __ NODE，不含任何VERTEX，然后将其作为线要素数据调用GENERATE命令进行格式的转换。

下列程序HZ. FOR主要用于 16×16 和 24×24 点阵汉字的格式转换，其中主要变量为：

NNO——汉字区位码。

程序运行时，首先提示有无自己的汉字数据文件。这里的数据文件主要是指已存盘的具有通用格式的汉字数据文件。如用户没有此类文件，程序将自动从字库中提取汉字。然后屏幕显示三种可供选择的点阵字库，即 16 点阵简体、16 点阵繁体和 24 点阵简体，用户只需键入自己的选项，然后按提示输入区位码，输入码超限时，程序自动提示输入值出界，请重新输入。随后，屏幕将自动显示用户所选汉字，并询问要不要输入其他汉字。当用户输入“是”时，程序将重复上述操作，否则提示用户输入输出数据文件名。该文件可供用户随时调用或作其他进一步转换。该通用文件格式已由程序确定，其第一行的三个数据分别指明文件中汉字的个数、汉字的类型（空心或实体）以及字库类型，三者皆为 2 位整型数。下面的数据用两位整型格式标明点阵中各非零点的实际坐标。

```
C  * * * * * PROGRAM HZ. FOR * * * * * * *
       INCLUDE 'FGRAPH. FI'
```

```
      INCLUDE 'FGRAPH.FD'
      INTEGER*1 H(72)
      INTEGER*2 D,X,Y,MA(27,500)
      INTEGER*4 NCOLOR
      CHARACTER Q,QA,FILE__NAME*20
      DIMENSION KK(24,3),NK(24,24),NR(8),MXY(24,24)
      PRINT*,'=========================================='
      PRINT*,'  THIS PROGRAM TRANSFERS DOT-CHARACTER TO ASCII FORM  '
      PRINT*,'            WRITTEN BY H.Q. TAN, ATC, CUMT            '
      PRINT*,'  COPYRIGHT BY ATC, CUMT, 1994.   ALL RIGHTS RESERVED  '
      PRINT*,'=========================================='
  505 WRITE(*,'(A\)') ' DO YOU HAVE YOUR OWN DATA FILE? '
      READ(*,'(A)') QA
      IF(QA.EQ.'n'.OR.QA.EQ.'N') GOTO 102
      OPEN(2,FILE=' ',STATUS='OLD')
      READ(2,'(I2)') (MA(1,J),J=1,3)
      IS=MA(1,1)          ! NUMBER OF CHARACTERS
      KL=MA(1,2)          ! TYPE OF CHARACTERS
      ID=MA(1,3)          ! TYPE OF DOT-FILE
      IF(ID.EQ.3) THEN
      MK=24
      ELSE
      MK=16
      ENDIF
      KM=MK*IS
      READ(2,'(I2)') ((MA(I,J),J=1,KM),I=2,MK+1)
      CLOSE(2)
      GOTO 301
C     * * * * * INPUT PARAMETERS FOR READING DATA FROM CCLIB.DAT
  102 PRINT*,' * * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
      PRINT*,' *          1)16-DOT J  2)16-DOT F  3)24-DOT          *'
      PRINT*,' * * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
      WRITE(*,'(A\)')' WHICH KIND OF DOT-FILE DO YOU WANT TO USE? '
      READ(*,'(I2)') ID
      NL=32
      MK=16
      NP0=2
      NC=7
      ND=7
      IF(ID.EQ.1) THEN
      FILE__NAME='F:\WPSO\CCLIBJ.DOT'
      ELSE IF(ID.EQ.2) THEN
      FILE__NAME='F:\WPSO\CCLIBF.DOT'
      ELSE IF(ID.EQ.3) THEN
      FILE__NAME='F:\WPSO\CLIBS.DOT'
      NL=72
      MK=24
      NP0=3
      NC=9
      ND=1
      ELSE
```

```
      PRINT * ,'PLEASE REENTER! '
      GOTO 102
      ENDIF
      OPEN(UNIT=2,FILE=FILE__NAME,STATUS='OLD',
     *   ACCESS='DIRECT',RECL=NL,FORM='UNFORMATTED')
      IS=1
   33 WRITE( * ,'(A\)') ' PLEASE INPUT THE REGION AND POSITION NUMBER:
      READ( * ,'(I4)') NN0
      NQ=NN0/100
      NW=NN0-NQ * 100
      IF(NQ.LT.1.OR.NQ.GT.87) THEN
      PRINT * ,'INPUT BEYOND RANGE, REENTER PLEASE! '
      GOTO 33
      ENDIF
      IF(NW.LT.1.OR.NW.GT.94) THEN
      PRINT * ,'INPUT BEYOND RANGE, REENTER PLEASE! '
      GOTO 33
      ENDIF
      IF(NQ.LE.NC) THEN
      M=(NQ-1) * 94+NW                          ! CHARACTER NUMBER
      ELSE IF(ID.NE.3.AND.NQ.GT.NC.AND.NQ.LT.16) THEN
      PRINT * ,'INPUT BEYOND RANGE, REENTER PLEASE! '
      GOTO 33
      ELSE
      M=(NQ-ND) * 94+NW                         ! CHARACTER NUMBER
      ENDIF
      READ(2,REC=M) (H(J),J=1,NL)               ! BYTE-TYPE ARRAY
      DO 11 KI=1,MK
      DO 111 KJ=1,NP0
      JL=(KI-1) * NP0+KJ
      IF(H(JL).LT.0)  THEN
      KK(KI,KJ)=H(JL)-256
      ELSE
      KK(KI,KJ)=H(JL)
      ENDIF
      DO 303 K=8,1,-1
      JJ=(KJ-1) * 8+K
      NK(KI,JJ)=MOD(KK(KI,KJ),2)
      NR(K)=KK(KI,KJ)/2
      KK(KI,KJ)=NR(K)
      IF(ID.EQ.3) GOTO 303
  303 CONTINUE
  111 CONTINUE
      IF(ID.EQ.3) GOTO 11
   11 CONTINUE
      IF(ID.EQ.3) THEN
      DO 201 NX=1,24
      DO 201 NY=1,24
      KX=NK(NX,NY)
  201 MXY(NY,NX)=KX
      ENDIF
```

```
C   * * PUTTING THE DATA INTO AN ARRAY FOR A GROUP OF CHARACTERS   * * *
   32   KM=IS * MK
        MA(1,1)=IS
        MA(1,2)=1                          ! FOR SOLID CHARACTER(S)
        MA(1,3)=ID
        DO 55 MI=2,MK+1
        DO 55 MJ=1,KM
        KD=MJ-MK * AINT((MJ-0.005)/MK)
        KE=(IS-1) * MK+KD
        IF(ID.NE.3) THEN
        MA(MI,KE)=NK(MI-1,KD)
        ELSE
        MA(MI,KE)=MXY(MI-1,KD)
        ENDIF
   55   CONTINUE
C   * * * * * SHOW THE CHARACTERS ON SCREEN
  301   PRINT * , IS,KL,ID,MK
        IF(SETVIDEOMODE($MAXRESMODE).EQ.0)
     *    STOP 'ERROR: CANNOT SET GRAPHICS MODE'
        WRITE(*,*) 'PRESS ENTER TO CONTINUE'
        NCOLOR=1
        D=SETCOLOR(NCOLOR)
        X=5
        Y=50
        D=RECTANGLE($GFILLINTERIOR,X+600,Y+50,X,Y)
        NCOLOR=NCOLOR+1
        D=SETCOLOR(NCOLOR)
        DO X=2,MK+1
        DO Y=1,KM
        IF(MA(X,Y).EQ.0) GOTO 302
        D=SETPIXEL(Y-10,X+65)
  302   END DO
        END DO
        READ(*,*)
        D=SETVIDEOMODE($DEFAULTMODE)
   35   IF(QA.EQ.'y'.OR.QA.EQ.'Y') THEN
        WRITE(*,'(A\)') CONTINUE? '
        READ(*,'(A)') Q
        IF(Q.EQ.'y'.OR.Q.EQ.'Y') THEN
        GOTO 505
        ELSE
        GOTO 315
        ENDIF
        ENDIF
        WRITE(*,'(A\)') ONE MORE CHARACTER? '
        READ(*,'(A)') Q
        IF(Q.EQ.'y'.OR.Q.EQ.'Y') THEN
        IS=IS+1
        IF(IS.GT.20) THEN
        PRINT *,'BEYOND RANGE, PROGRAM STOP! '
        GOTO 335
```

```
      ELSE
      GOTO 33
      ENDIF
      ELSE
      GOTO 335
      ENDIF
335   WRITE(*,'(A\)')' WARNING! DATA NOT SAVED, SAVE OR NOT? '
      READ(*,'(A)') Q
      IF(Q.NE.'y'.AND.Q.NE.'Y') GOTO 315
      OPEN(UNIT=2,FILE=' ',STATUS='NEW')
      MA(1,1)=IS                    ! NUMBER OF CHARACTERS
      MA(1,2)=1                     ! TYPE OF CHARACTERS
      MA(1,3)=ID                    ! TYPE OF DOT-FILE
      WRITE(2,'(I2)') (MA(1,JJ),JJ=1,3)
      WRITE(2,'(I2)') ((MA(II,JJ),JJ=1,KM),II=2,MK+1)
      CLOSE(2)
315   STOP
      END
```

2. 矢量式数据格式的转换

此类处理的本质为轮廓跟踪，即提取勾画汉字字形轮廓的矢量数据信息。有关算法可在多种教科书中查到。pcARC/INFO提供的GRIDPOLY命令即具有此项功能。在调用此项命令之前，用户首先要将汉字点阵中的二值数据转为非压缩ASCII (NAS)或压缩ASCII (CAS)格式。其具体要求详见图 4－1。与前述点阵式数据转换不同的是，这里的格式转换无需先将点阵中的非零信息转为坐标信息。具体程序从略。

（二）矢量汉字的转换

这里着重介绍如何使矢量汉字库中的相关信息与 pcARC/INFO 格式相匹配的问题。现结合所附源程序 DOSHZ. FOR 说明如下：

1. 汉字整体信息的提取

由于此类汉字信息在字库中的存放方式与点阵式字库截然不同，因此，对库中汉字信息的提取就不能象对点阵汉字那样，按固定长度读取若干字节。通过对此类字库的分析表明，勾画每个汉字字形的原数据的存放地址皆由该字的地址码来决定。原则上讲，每个汉字只与其地址码存在对应关系，据此，可通过区位码查明每个汉字的地址码提示的起始地址查明汉字信息所在的位置，并根据相邻两个汉字的地址码之差计算出该字所占的字节长度，进而读取该汉字的整体信息。具体操作可分为：

1）输入区位码，找出该字的地址码及所占长度

同点阵汉字信息提取时使用区位码换算地址码一样，从矢量汉字库中提取信息同样要先换算出该汉字的地址码，计算公式也完全一样。所不同的是，此处的地址码并不提供汉字信息，而只提供该汉字信息存放的起始地址信息。其所占字节长度可由相邻两汉字的起始地址之差求得。

2）读取整体字形信息

根据上述起始地址和字节长度，即可由字库中读取整体字形信息。此处所谓整体字形

信息指的是所有字节信息，但并不一定代表描述所有字形信息，因为其中还可能隐含部件信息。

本节所附主程序 DOSHZ. FOR 即可完成上述操作。程序中相关主要变量有：

FNM —— 输入字库文件名；

NC —— 单个汉字的区位码；

NZ —— 汉字个数；

NX —— 每个汉字的起始坐标偏移量；

MY —— 用于存放汉字整体信息的一维数组。

程序运行时，用户只须输入字库文件名（FNM）和汉字的区位码（NC）并对“ONE MORE CHARACTER?”的提示作出回答，程序即可自动生成存放用户所须汉字的整体信息的一维数组。

2. 部件信息的提取

由于原字库内各码全部呈紧密方式存放，而每个汉字的组成又千差万别，用户调用的汉字中是否有部件出现、何时何处出现以及出现的次数皆是随机的且因字库而异。所以在主程序中采用从汉字的整体信息逐字节查询判断的方式，查明相关代码，并设置了若干条件语句，用于判断正在调用的汉字中是否有部件存在。当存在部件时即调用子程序 ADDRESS 读取部件信息。所谓部件信息，包括该部件的序号（NUMB）及相关字形的偏移量（XOFF，YOFF）和缩放比（XCAL，YCAL）。

3. 部件字形信息的提取

部件的字形信息是指按序号和相关参数求出的可用于描述具体笔画的字形信息。子程序 DRAW 可利用从子程序 ADDRESS 得到的汉字部件信息及时将其转为字形信息，即勾绘汉字笔画的相关坐标值，并存放于指定数组 NY 之中。从而保证了部件信息的严格定位、不漏不重。有关该子程序中调用的插值子程序 BEZIER2 的说明，详见下述。

4. 整体汉字的字形信息的获取

由于汉字笔画的多变，字库中存入的信息本身还不能直接反映汉字笔画本身的外形特点。由上述三个步骤所产生的整体汉字的字形信息有时还不能描绘出汉字的真正外形，甚至还会有很大差异。汉字中的撇捺等笔画，特别是魏碑、行楷等字体的汉字，几乎每个笔画都必须用曲线才能逼近其原形，只有经过插值处理之后，才能得到其真正的笔画有关细节。这里所说的整体汉字信息就是指包括原有字形信息、部件字形信息以及经过插值后得到的相关插值信息在内的全部字形信息。

同计算机绘图中任一曲线的生成一样，汉字中笔画本身的曲线也同样要通过插值来生成一系列直线段来逼近，这里插值的密度与参与插值的原数据点数和位置等初始条件，已在原字库信息中“内定”了。如第一节所讨论的第 4 类和节 5 类，前者是在三点之间的内插，后者是在四点之间的内插，由于原有库中提供的仅是三点或四点数据，且这些点中的过渡点大多不在曲线之上，所以插值是不可回避的一步，而且只有插值得到的数据，才是反映笔画变化的有效数据。子程序 BEZIER2 即可在必要时用于获取曲线笔画的插值信息。

当然同其它曲线的插值逼近一样，插值的密度与逼近度或保真度之间必须有所取舍，否则将会影响到程序的运行速度与生成数据的存储空间。插值越密，保真度当然越好。但数据量的增大，又将影响数据的存储和调用，程序 DOSHZ.FOR 中以经过大量试验选取的数

据为准进行内插。

5. 汉字的显示与数据的的存盘

为检查所生成数据是否为所调用的汉字，主程序中特意增加了用于屏幕显示的程序段，可是用以显示单个或一组汉字，以便用户随时了解输入是否有误，并便于随时调整。其中调用了 FORTRAN 5.0 的几个图形显示模块，如用于屏幕模式设置的 SETVIDEOMODE、用于确定显示起点的 MOVETO 和用于勾画汉字轮廓的 LINETO 等。一旦用户确认无误即可按屏幕提示进行存盘。需要说明的是，程序中人为地将原来是描绘一条曲线的若干数据，拆成了 60 个点一组这主要是为了方便字符集的生成与维护，这一点下面进行讨论。

```
C   * * PROGRAM DOSHZ.FOR READS CHARACTER DATA FROM A CHARACTER FILE   * *
        INCLUDE 'FGRAPH.FI'
        INCLUDE 'FGRAPH.FD'
        INTEGER*2 DUMMY
        RECORD /XYCOORD/ XY
        CHARACTER Q,FNM*20
        INTEGER*1 H,H1
        INTEGER*4 NH,M,KM,NA,LM
        DIMENSION H(6),MH(6),NH(2),H1(2000),MU(2000)
        DIMENSION MQ(1500,2),MW(1500),MY[HUGE](3000)
        REAL      NX,NY[HUGE](40000,2)
C   * * * * * OPEN A .PS DATA FILE
        NL=1
  102   WRITE(*,'(A\)') ' PLEASE INPUT FILE NAME:  '
        READ(*,'(A)') FNM
        OPEN(UNIT=2,FILE=FNM,ERR=102,ACCESS='DIRECT',RECL=NL)
C   * * * * * SELECT CHINESE CHARACTER
        NZ=1
        NX=0.
  103   WRITE(*,'(A\)') ' PLEASE INPUT THE REGIONAL AND POSITIONAL CODE
        READ(*,*) NC
        NW=MOD(NC,100)                  ! REGIONAL CODE
        NQ=(NC-NW)/100                  ! POSITIONAL CODE
        IF(NQ.LT.16.OR.NQ.GT.87.OR.NW.LT.1.OR.NW.GT.94) THEN
        PRINT*,' INVALID INPUT, REENTER PLEASE! '
        GOTO 103
        ENDIF
        M=(NQ-16)*94+NW
C   * * * * * GETTING ADDRESS OF THE SELECTED CHARACTER
  301   KK=3
        DO KM=M,M+1
        DO LM=1,3
        H(LM)=0
        MH(LM)=0
        LK=(KM-1)*KK+LM
        READ(2,REC=LK,END=111) H(LM)        ! BYTE-TYPE ARRAY
          IF(H(LM).LT.0) THEN
            MH(LM)=H(LM)+256
          ELSE
```

```
          MH(LM)=H(LM)
        ENDIF
111   END DO
      NA=KM-M+1
      NH(NA)=MH(3)*256*256+MH(2)*256+MH(1)
      ENDDO
C   * * * * * READING DATA FROM THE ORIGINAL DATA FILE
      NN=NH(2)-NH(1)                          ! NUMBER OF BYTES
      DO M=1,NN
      LM=NH(1)+M
      H1(M)=0
      READ(2,REC=LM) H1(M)                    ! BYTE-TYPE ARRAY
      M0=H1(M)
      MI=(M-1)*2+1
      MY(MI)=0
        IF(M0.LT.0) THEN
          MY(MI)=(M0+256)/16
          MY(MI+1)=MOD((M0+256),16)
        ELSE
          MY(MI)=M0/16
          MY(MI+1)=MOD(M0,16)
        ENDIF
      END DO
      NN=2*NN
      NCODE=MY(1)                             ! TYPE CODE OF THE CHARACTER
C   * * * * * GETTING ADDRESS OF THE UNITS
      IF(NZ.EQ.1) THEN
      DO K0=1,3
      MH(K0)=0
      H(K0)=0
      LK=20304+K0
      READ(2,REC=LK) H(K0)
      IF(H(K0).LT.0) THEN
      MH(K0)=H(K0)+256
      ELSE
      MH(K0)=H(K0)
      ENDIF
      ENDDO
      NDR8=MH(3)*256*256+MH(2)*256+MH(1)
      NN8=NN
      NK=0
      NARC=5000
      NP=0
      ENDIF
      I=1
      XCAL=1.
      YCAL=1.
      XOFF=0.
      YOFF=0.
C   * * * * * CREATE DATA FOR DRAWING ARCS
      CALL DRAW(NZ,NARC,MY,NN,NX,XOFF,YOFF,
```

```
*     XCAL,YCAL,NY,NP,I,NN0)
      IF(I-1.EQ.NN0.OR.I.EQ.NN0) GOTO 136
      KU=0
      KD=0
145   IF(MY(I+KD).GE.8) THEN
      KU=KU+1
      KD=KD+11
      GOTO 145
      ENDIF
      NN8=NN0-I+1
      IF(KU.EQ.0) GOTO 136
      DO KL=1,NN8
      MW(KL)=MY(I+KL-1)
      ENDDO
      NNN=KU*11
      NN9=NN8-NNN
      DO KI=1,KU
      DO NI=1,11
      NJ=(KI-1)*11+NI
      MU(NI)=MW(NJ)
      ENDDO
      CALL ADDRESS(NDR8,MU,MY,NN,NUMB,XOFF,YOFF,XCAL,YCAL)
      I=1
      CALL DRAW(NZ,NARC,MY,NN,NX,XOFF,YOFF,
*     XCAL,YCAL,NY,NP,I,NN8)
      ENDDO
      IF(NN9.GT.2) THEN
      DO KK=1,NN9
      MY(KK)=0
      KJ=NNN+KK
      MY(KK)=MW(KJ)
      ENDDO
      I=1
      XCAL=1.
      YCAL=1.
      XOFF=0.
      YOFF=0.
      NN=NN9
      CALL DRAW(NZ,NARC,MY,NN,NX,XOFF,YOFF,
*     XCAL,YCAL,NY,NP,I,NN8)
      ENDIF
136   WRITE(*,'(A\)') ' ONE MORE CHARACTER?     '
      READ(*,'(A)') Q
      IF(Q.EQ.'y'.OR.Q.EQ.'Y') THEN
      IF(NP.GT.39500) THEN
      PRINT*,'TOO MANY CHARACTERS! DATA OVERFLOW! '
      GOTO 136
      ENDIF
      NZ=NZ+1
      NX=REAL((NZ-1)*200.)
      GOTO 103
```

```
      ENDIF
      NARC=0                                          ! NUMBER OF ARCS
      DO I=1,NP
      IF(NY(I,1).GT.5000) THEN
      NARC=NARC+1
      MQ(NARC,1)=I
      ENDIF
      ENDDO
      DO J=1,NARC+1
      IF(J.LT.NARC) THEN
      MQ(J,2)=MQ(J+1,1)-MQ(J,1)-1
      ELSE
      MQ(J,2)=NP-MQ(J,1)
      ENDIF
      ENDDO
      DUMMY = SETVIDEOMODE($MAXRESMODE)
      NA=10
      F=REAL(NZ/3.)
      K=1
      K0=MQ(K,1)+1
      KD=MQ(K,2)
  507 CALL MOVETO(NY(K0,1)/F+NA,NY(K0,2)/F+NA,XY)
      DO JJ = 1,KD-1
      DUMMY = LINETO(NY(K0+JJ,1)/F+NA,NY(K0+JJ,2)/F+NA)
      ENDDO
      K=K+1
      IF(K.LE.NARC) THEN
      K0=MQ(K,1)+1
      KD=MQ(K,2)
      GOTO 507
      ENDIF
      READ(*,*)
      DUMMY = SETVIDEOMODE($DEFAULTMODE)
  122 WRITE(*,'(A\)') ' SAVE THE DATA?    '
      READ(*,'(A)') Q
      IF(Q.EQ.'y'.OR.Q.EQ.'Y') THEN
      WRITE(*,'(A\)') ' PLEASE INPUT RESULT FILE NAME:  '
      READ(*,'(A)') FNM
  123 OPEN(1, FILE=FNM,ERR=123)
C   WRITE(1,'(1X,2F9.2)') ((NY(K,L),L=1,2),K=1,NP)
      MQ(NARC+1,1)=NP+1
      DO K=1,NARC
      KJ=MQ(K,1)
      KI=MQ((K+1),1)-KJ-1
      KM=KI/60
      KN=KI-60*KM
      IF(KM.GE.1) THEN
      KY=KY+1
      IF(K.EQ.1) KY=INT(NY(KJ,1))
      DO IA=1,KM
      WRITE(1,'(1X,I4)') KY
```

```
IC=MA
IF(IA.EQ.1) IC=KJ+1
DO IB=1,60
MA=IB+IC-1
IF(IA.EQ.1.OR.IB.GT.1) NY(MA,2)=200.0-NY(MA,2)
WRITE(1,'(1X,2F9.2)') (NY(MA,K1),K1=1,2)
ENDDO
KY=KY+1
WRITE(1,'(1X,A3)') 'END'
ENDDO
DO J=1,KN-KM+1
   IF(J.EQ.1) THEN
   WRITE(1,'(1X,I4)') KY
   ELSE
   K2=MA+J-2
   IF(J.GT.2) NY(K2,2)=200.0-NY(K2,2)
   WRITE(1,'(1X,2F9.2)') (NY(K2,K1),K1=1,2)
   ENDIF
ENDDO
WRITE(1,'(1X,A3)') 'END'
ELSE
   KY=KY+1
   IF(K.EQ.1) KY=INT(NY(KJ,1))
DO J=1,KI+1
   IF(J.EQ.1) THEN
   WRITE(1,'(1X,I4)') KY
   ELSE
   K2=KJ+J-1
   NY(K2,2)=200.0-NY(K2,2)
   WRITE(1,'(1X,2F9.2)') (NY(K2,K1),K1=1,2)
   ENDIF
ENDDO
WRITE(1,'(1X,A3)') 'END'
ENDIF
ENDDO
WRITE(1,'(1X,A3)') 'END'
ENDIF
WRITE(*,'(A\)') ' CONTINUE WITH THE SAME DATA FILE?   '
READ(*,'(A)') Q
IF(Q.EQ.'y'.OR.Q.EQ.'Y') THEN
NP=0
NARC=0
NZ=1
NX=0
GOTO 103
ELSE
CLOSE(2)
WRITE(*,'(A\)') ' CONTINUE WITH ANOTHER DATA FILE?   '
READ(*,'(A)') Q
IF(Q.EQ.'y'.OR.Q.EQ.'Y') THEN
CLOSE(2)
```

```
      GOTO 102
      ENDIF
      ENDIF
  108 CLOSE(2)
      CLOSE(1)
      END

C   * * * * * CREATING DATA ARRAY FOR CHARACTER(S)
      SUBROUTINE DRAW(NZ,NARC,MY,NN,NX,XOFF,YOFF,XCAL,YCAL,
     *   NY,NP,I,NN0)
      REAL NX,NY[HUGE](40000,2)
      DIMENSION X0(4),Y0(4),X(2500),Y(2500),MY[HUGE](3000)
      IF(NZ.EQ.1) NX=0.
  121 IF(MY(I).EQ.0) THEN
      NARC=NARC+1
      NP=NP+1
      NY(NP,1)=REAL(NARC)
      NY(NP,2)=REAL(MY(I))
      NY(NP+1,1)=XCAL*REAL(16.*MY(I+1)+MY(I+2))+NX+XOFF
      NY(NP+1,2)=YCAL*REAL(16.*MY(I+3)+MY(I+4))+YOFF
      I=I+5
      NP=NP+1
      IF(I.GE.NN) GOTO 135
      ENDIF
      IF(MY(I).EQ.1) THEN
      NP=NP+1
      NY(NP,2)=NY(NP-1,2)
      NY(NP,1)=XCAL*REAL(16.*MY(I+1)+MY(I+2))+NX+XOFF
      I=I+3
      IF(I.GE.NN) GOTO 135
      ENDIF
      IF(MY(I).EQ.2) THEN
      NP=NP+1
      NY(NP,1)=NY(NP-1,1)
      NY(NP,2)=YCAL*REAL(16.*MY(I+1)+MY(I+2))+YOFF
      I=I+3
      IF(I.GE.NN) GOTO 135
      ENDIF
      IF(MY(I).EQ.3) THEN
      NP=NP+1
      NY(NP,1)=XCAL*REAL(16.*MY(I+1)+MY(I+2))+NX+XOFF
      NY(NP,2)=YCAL*REAL(16.*MY(I+3)+MY(I+4))+YOFF
      I=I+5
      IF(I.GE.NN) GOTO 135
      ENDIF
      IF(MY(I).EQ.4) THEN
      NP=NP+1
      NY(NP,1)=XCAL*REAL(16.*MY(I+1)+MY(I+2))+NX+XOFF
      NY(NP,2)=YCAL*REAL(16.*MY(I+3)+MY(I+4))+YOFF
      NY(NP+1,1)=XCAL*REAL(16.*MY(I+5)+MY(I+6))+NX+XOFF
      NY(NP+1,2)=YCAL*REAL(16.*MY(I+7)+MY(I+8))+YOFF
```

```
I=I+9
M=3
X0(1)=NY(NP-1,1)
Y0(1)=NY(NP-1,2)
X0(2)=NY(NP,1)
Y0(2)=NY(NP,2)
X0(3)=NY(NP+1,1)
Y0(3)=NY(NP+1,2)
XA=X0(1)
YA=Y0(1)
XB=X0(3)
YB=Y0(3)
CALL BEZIER2(X,Y,X0,Y0,M,N)
NY(NP-1,1)=XA
NY(NP-1,2)=YA
DO J=2,N
KJ=NP+J-2
NY(KJ,1)=X(J)
NY(KJ,2)=Y(J)
ENDDO
NP=NP+N-2
IF(I.GE.NN) GOTO 135
ENDIF
IF(MY(I).EQ.5) THEN
NP=NP+1
NY(NP,1)=XCAL*REAL(16.*MY(I+1)+MY(I+2))+NX+XOFF
NY(NP,2)=YCAL*REAL(16.*MY(I+3)+MY(I+4))+YOFF
NY(NP+1,1)=XCAL*REAL(16.*MY(I+5)+MY(I+6))+NX+XOFF
NY(NP+1,2)=YCAL*REAL(16.*MY(I+7)+MY(I+8))+YOFF
NY(NP+2,1)=XCAL*REAL(16.*MY(I+9)+MY(I+10))+NX+XOFF
NY(NP+2,2)=YCAL*REAL(16.*MY(I+11)+MY(I+12))+YOFF
I=I+13
M=4
X0(1)=NY(NP-1,1)
Y0(1)=NY(NP-1,2)
X0(2)=NY(NP,1)
Y0(2)=NY(NP,2)
X0(3)=NY(NP+1,1)
Y0(3)=NY(NP+1,2)
X0(4)=NY(NP+2,1)
Y0(4)=NY(NP+2,2)
XA=X0(1)
YA=Y0(1)
XB=X0(4)
YB=Y0(4)
CALL BEZIER2(X,Y,X0,Y0,M,N)
NY(NP-1,1)=XA
NY(NP-1,2)=YA
DO J=2,N
KJ=NP+J-2
NY(KJ,1)=X(J)
```

```
      NY(KJ,2)=Y(J)
      ENDDO
      NP=NP+N-2
      IF(I.GE.NN) GOTO 135
      ENDIF
      IF(MY(I).EQ.6) THEN
      NARC=NARC+1
      NP=NP+1
      NY(NP,1)=REAL(NARC)
      NY(NP,2)=REAL(MY(I))
      NLFX=REAL(16.*MY(I+1)+MY(I+2))
      NLFY=REAL(16.*MY(I+3)+MY(I+4))
      NURX=REAL(16.*MY(I+5)+MY(I+6))
      NURY=REAL(16.*MY(I+7)+MY(I+8))
      NY(NP+1,1)=XCAL*REAL(NLFX)+XOFF+NX
      NY(NP+1,2)=YCAL*REAL(NURY)+YOFF
      NY(NP+2,1)=XCAL*REAL(NURX)+XOFF+NX
      NY(NP+2,2)=YCAL*REAL(NURY)+YOFF
      NY(NP+3,1)=XCAL*REAL(NURX)+XOFF+NX
      NY(NP+3,2)=YCAL*REAL(NLFY)+YOFF
      NY(NP+4,1)=XCAL*REAL(NLFX)+XOFF+NX
      NY(NP+4,2)=YCAL*REAL(NLFY)+YOFF
      NY(NP+5,1)=NY(NP+1,1)
      NY(NP+5,2)=NY(NP+1,2)
      I=I+9
      NP=NP+5
      IF(I.GE.NN) GOTO 135
      ENDIF
C     *****FOR COMBINED CHARACTERS
      IF(MY(I).GE.8) THEN
      NN0=NN
      NN=I-1
      ENDIF
  135 IF(I.LT.NN) GOTO 121
      END

      SUBROUTINE ADDRESS(NDR8,MU,MY,NN,NUMB,XOFF,YOFF,XCAL,YCAL)
      INTEGER*1 H,H1
      INTEGER*4 NDR1,NDR2,LK0,LK,LM
      DIMENSION H(6),MH(6),H1(2000),MY[HUGE](3000),MU(2000)
      JJ=1
      IF(MU(JJ).EQ.8) THEN
      NUMB=16*MU(JJ+1)+MU(JJ+2)-7
      XOFF=REAL(16.*MU(JJ+3)+MU(JJ+4))
      YOFF=REAL(16.*MU(JJ+5)+MU(JJ+6))
      XCAL=REAL((16.*MU(JJ+7)+MU(JJ+8))/100.)
      YCAL=REAL((16.*MU(JJ+9)+MU(JJ+10))/100.)
      ENDIF
      IF(MU(JJ).EQ.9) THEN
      NUMB=16*MU(JJ+9)+MU(JJ+10)+247
      ELSE IF(MU(JJ).EQ.10) THEN
```

```
      NUMB=16*MU(JJ+9)+MU(JJ+10)+503
      ENDIF
      IF(MU(JJ).EQ.9.OR.MU(JJ).EQ.10) THEN
      XOFF=REAL(16.*MU(JJ+1)+MU(JJ+2))
      YOFF=REAL(16.*MU(JJ+3)+MU(JJ+4))
      XCAL=REAL((16.*MU(JJ+5)+MU(JJ+6))/100.)
      YCAL=REAL((16.*MU(JJ+7)+MU(JJ+8))/100.)
      ENDIF
      LK0=NDR8+(NUMB-1)*3
      DO K3=1,6
      H(K3)=0
      MH(K3)=0
      LK=LK0+K3
      READ(2,REC=LK) H(K3)
      IF(H(K3).LT.0) THEN
      MH(K3)=H(K3)+256
      ELSE
      MH(K3)=H(K3)
      ENDIF
      ENDDO
      NDR1=MH(3)*256*256+MH(2)*256+MH(1)
      NDR2=MH(6)*256*256+MH(5)*256+MH(4)
      NN1=NDR2-NDR1
C     ***** READING DATA OF UNITS
      DO M=1,NN1
      H1(M)=0
      LM=NDR1+NDR8+M
      READ(2,REC=LM) H1(M)
      M0=H1(M)
      MI=(M-1)*2+1
      MY(MI)=0
        IF(M0.LT.0) THEN
          MY(MI)=(M0+256)/16
          MY(MI+1)=MOD((M0+256),16)
        ELSE
          MY(MI)=M0/16
          MY(MI+1)=MOD(M0,16)
        ENDIF
      END DO
      NN=2*NN1
      END

      SUBROUTINE BEZIER2(X,Y,X0,Y0,M,N)
      REAL C(4),X0(4),Y0(4),X(2500),Y(2500)
      N=1
      IF(M.EQ.3) THEN
        D=SQRT((X0(1)-X0(3))**2+(Y0(1)-Y0(3))**2)
        N1=D/4+1
        DO 40 I=1,N1-1
        T=(I+0.)/N1
        C(1)=1.-2.*T+T*T
```

```
      C(2)=2. * T-2. * T * T
      C(3)=T * T
      N=N+1
      S1=0.
      S2=0.
      DO 20 J=1,3
      S1=S1+C(J) * X0(J)
20    S2=S2+C(J) * Y0(J)
      X(N)=S1 + 0.5
40    Y(N)=S2+0.5
      N=N+1
      X(N)=X0(3)
      Y(N)=Y0(3)
    ELSE
      D=SQRT((X0(1)-X0(4)) * * 2+(Y0(1)-Y0(4)) * * 2)
      N1=D/8+1
      DO 80 I=1,N1-1
      T=(I+0.)/N1
      T2=T * T
      T3=T2 * T
      C(1)=1.-3. * T+3. * T2-T3
      C(2)=3. * T3-6. * T2+3. * T
      C(3)=3. * T2-3. * T3
      C(4)=T3
      N=N+1
      S1=0.
      S2=0.
      DO 60 J=1,4
      S1=S1+C(J) * X0(J)
60    S2=S2+C(J) * Y0(J)
      X(N)=S1+0.5
80    Y(N)=S2+0.5
      N=N+1
      X(N)=X0(4)
      Y(N)=Y0(4)
    END IF
    END
```

（三） 手书汉字的输入及转换

在多元信息复合应用中，有时手书汉字的输入也是需要的，特别是一些名人题字、专有名词或有特定涵义的标记如旅游景点的特殊标记等。在这种情况下，无论是点阵汉字还是矢量汉字皆无法满足这一方面的需求，唯一的“入口”就是对手书汉字进行数字化，使其转变为矢量数据或是点阵数据，常用的方法有两类：

1. 手扶跟踪扫描法

手扶跟踪扫描主要是将待输入汉字的原件置于数字化桌之上，用与之配套的数字化软件逐点跟踪要输入汉字的轮廓。即可生成所需矢量化数据，并要在配套软件上先进行预处理，使之能符合pcARC/INFO的GENERATE命令对数据的格式要求。

由于手书汉字的笔画变化较大，手扶跟踪扫描生成的汉字字形信息不但数据量大，而且受手扶跟踪方法的限制，以其较低的分辨率很难以体现汉字中常见的笔锋之类的细微变化。此外，手扶跟踪的分辨率还因人而异，可重现度很低。因此，这不是一个值得推荐的输人方式。

2. 扫描输人

扫描输入与手扶跟踪相比，其最大优点就是分辨率高，可重现度好。一般扫描仪的分辨率皆在300dpi以上，而一些较高档的扫描仪甚至可达600dpi、1200dpi。这就使扫描得到的手书汉字信息，既有点阵汉字的特点，又有接近矢量汉字的精度。

扫描所得的文件一般多以点阵式 TIF 文件存盘输出，为适应pcARC/INFO的要求．这种 TIF格式文件还必须经过格式转换，使之能够满足pcARC/INFO中GRIDPOLY命令所能接受的格式要求，才可利用 GRIDPOLY 命令生成独立的汉字数字化层。有关转换程序详见程序 4—1。

由于 TIF 文件数据量较大，比较适宜于用下述独立数字化层方式进行汉字注记；隐含的问题是不易为字符集所接受，解决的办法之一就是在扫描输人时，尽量在分辨率与存储空间两者之间进行权衡，即在保证精度的前提下，尽量减少数据量。实际上并非分辨率越高越好，因为 2 倍的数据量并不等于能将扫描汉字的质量提高两倍，有时甚至可能适得其反，所以扫描输入的关键所在是其分辨率的选取。

（四）汉字的生成与调用

根据 pcARC/INFO 为用户提供的数据接口的特点，汉字生成与调用的方式可分为以下两大类：独立数字化层方式和字符集方式。现分述如下：

1. 独立数字化层方式

无论是点阵式还是矢量式数据，皆可在 pcARC/INFO 中通过适当的命令调用生成独立的数字化层。该层中可以只含一个汉字，也可以包括一组汉字。其原数据皆可通过调用汉字转换程序生成，而其注记位置的坐标则可通过改变配准点坐标的方式进行调整。依据经过预处理得到的数据的格式，可分别调用相应的命令来生成数字化层。每个或每组汉字皆作为一个独立的数字化层存在，并可以像对其它数字化层一样实施各种操作。汉字的调用也必须通过数字化层的各种操作来实现。

1) GENERATE 命令既可用于利用源于点阵汉字库的数据，也可用于利用源于矢量汉字库的数据。实际操作中需要注意的是，对于预处理中获取的点式数据，其选项应为 POINTS；而对于预处理中获取的线类数据，包括线长为 0 的线类数据，其选项则应为 LINES。

2) GRIDPOLY 命令仅用于将点阵式数据转变为数字化层。需要用户提供的参数除数据文件名、生成数字化层名外，主要的就是起点坐标，即该数字化层左下角的坐标值。只要起点坐标设置得当，该层即可直接调用，将汉字加注在恰当的位置上；否则的话，就要修改配准点并对整个数字化层的进行坐标转换。

2. 字符集方式

与独立数字化层方式不同的是，字符集方式将单个汉字直接存人 pcARC/INFO 的指定字符集文件，可供用户随时调用。考虑到这方面的实际需要，作者经过大量试验，确定

了一整套在pcARC/INFO中生成中文字符集的参数和方法，用户可据此随时生成自己的汉字字符集，并可在ARCPLOT等环境下，随意调用。即可在用户指定的任意位置、以任意大小方向进行调用。为了说明其工作步骤，现将有关内容作一扼要介绍。

1）字符集的生成

①汉字数据的预处理，生成具pcARC/INFO格式汉字字形信息文件；

②调用GENERATE命令生成一个独立的单个汉字的数字化层；

③调用TABLES命令，修改新生成数字化层的TIC文件，使其中配准点的排列顺序符合字符文件的要求，使四个配准点按左上、右上、左下、右下的顺序排列；

⑤调用FONTEDIT命令，直接读取改过TIC文件的汉字数字化层，即可在指定字符集中存放单个汉字。

2）字符集的更新、维护

由于汉字本身的笔画较西文字符要复杂得多，而pcARC/INFO中字符集文件的大小、格式皆已固定，其中的预留空间不可能容纳任何一个汉字库中的所有汉字，而用于注记的汉字的数量一般也不是很大，因此，汉字字符中以存放注记时常用汉字为宜。大量试验证明，在每个字符集空间内存放26个汉字为宜，并使之分别对应于26个大写英文字母，即ASCII码的65--80号。如一次不能存满，可在下一次重新调用、添加。如感到某些字需要修改，可先将该字删除，再重新生成新字的数字化层再读入字符集中。pcARC/INFO的字符文件中共有8个预留的空集可供用户使用，全部填满即可生成包含200个左右汉字的8个字符集。一般来讲，完全可以满足汉字注记的需要。

鉴于每个集中只能存放26个以下的汉字，当需要注记的汉字较多，而且字体变化较大无法在已有的8个空集之中全部存放时，亦可分别生成几个同样的字符集文件，更名后存放备用，在调用时再将原有文件更名，即可同其它字符同样调用。

字符集的更新可直接按照FONTEDIT的有关命令进行删改或补充，此处不再重复。

3）字符集中汉字的调用

为了能从新生成的汉字字符集中调出所需汉字，用户还必须用TABLES命令生成一个扩展名为TXT的字符调用文件，如MY.TXT。其中最为重要的两项就是字符集调用号和该集在字符集文件中的实际序号，如MY.TXT文件规定调用号1对应的是第20集。这就表明，要使用第20集中的汉字，其调用号TEXTSYMBOL必须为1。以下就是在ARCPLOT环境下的部分调用实例：

```
      ……
:     TEXTSET MY.TXT
:     TEXTSYMBOL 1
:     MOVE *
:     TEXT 'AK'
      ……
```

上述第一句用于字符集文件的调用。第二句则是指明调用的是第几集，这里调用号为1，隐含的实际字符集号已由MY.TXT所确定。第三句可用于为注记汉字定位。第四句即为实际调用到的注记汉字，既可以为单个汉字，也可以为一组汉字，只是对调用码及其与该字符集中汉字的对应关系必须格外小心。至于注记汉字的大小、方向乃至修改、删除皆

与西文字符的调用相同，无需赘述。

（五）实用效果的对比

大量试验表明，经过预处理的汉字字形信息可以不同的方式生成数字化层，并能以不同的方式进行调用。为便于对比各注记方式的优劣，现将 pcARC/INFO 中的汉字注记方式汇总如表 5－2，并就其实用效果对比如下。

1. GRIDPOLY 与 GENERATE 命令的对比

GRIDPOLY 与 GENERATE 命令皆可用于生成数字化层，但前者可直接生成多边形的数字化层；而后者则只能生成点或线的数字化层。对汉字的注记来讲，经由GRIDPOLY生成的数字化层中，往往会有一些非汉字笔画所需要的多边形注记。如“口”字的内多边形对该字的描述无疑是多余的，在调用该字，特别是进行充填时，必须首先将这个多余的多边形注记删除。由于此类多边形往往随机出现且因字而异，删除只有交互式进行。经由GENERATE命令生成的数字化层则与之相反，其中只含点要素或线要素，不含任何多边形注记。在调用该字，特别是进行充填时，必须首先为其中的多边形添加注记，由于同样的原因，多边形注记的添加也只能交互式进行。

表 5－2　pcARC/INFO 中的汉字注记方式一览表

调用命令		存放、调用形式	
		独立 COVERAGE	字符集
原数据类型	点阵（手书）	GRIDPOLY GENERATE	FONTEDIT CREATEFONT READCOVERAGE
	矢量	GENERATE	

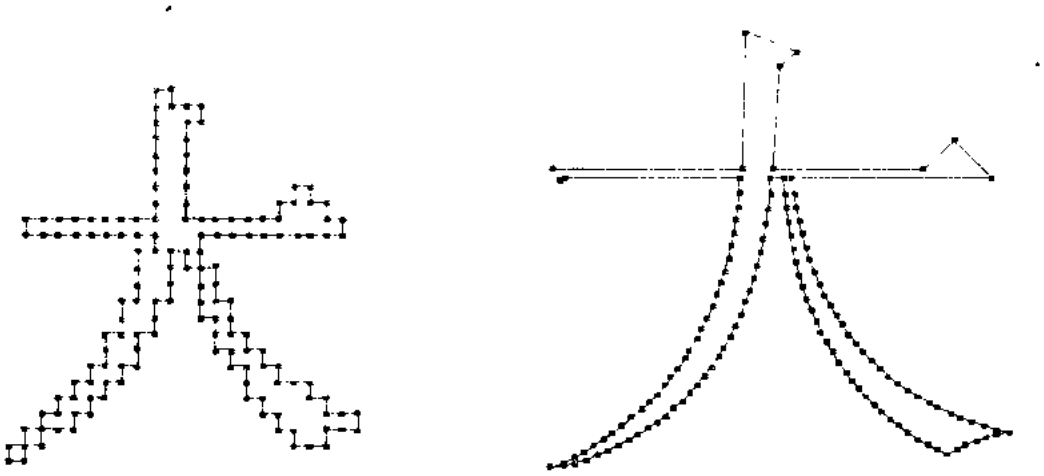


图 5－1　GRIDPOLY（左）与 GENERATE（右）命令生成的汉字数字化层对比图

另一个不同之处就在于，在GRIDPOLY生成的数字化层中，在每一个网格线与汉字笔画交叉处皆自动生成一个顶点(VERTEX)。这样一来，当汉字的某个单笔画较长，往往造成大量的顶点冗余，以至会影响到字符集汉字的生成。以宋体的“大”字为例，在此命令下生成的“大”字将根据原网格的多少不同而生成一系列的顶点（图5—1)。虽然通常不能直接看到这些隐含的顶点，但在读入字符集，往往出错而不能生成一个完整的汉字。消除这些隐含顶点的唯一办法就是在ARCEDIT环境下，调用UNSPLIT命令。

基于上述，GRIDPOLY命令比较适合于生成包含一组汉字的独立数字化层，而不宜用于生成汉字字符集使用的单个汉字数字化层。

2. 空心汉字与实体汉字的对比

从汉字注记的实用角度来看，空心汉字与实体汉字各有其长。前者可无级放大并能充填不同的纹饰，充分体现不同字体汉字的字形美，特别适宜于用作图件的标题字。但就其生成方式而言，只宜于使用矢量数据生成独立的数字化层。用矢量数据生成的字符集中往往只含空心汉字，虽然调用方便，但不便于后期的纹饰充填；至于实体汉字，无论是作为独立的数字化层还是存放于字符集，皆不宜放大，否则离散点将过于明显，终至字不成字。不过，此类格式的汉字对图面上地名之类的小字注记不无好处。当字号较小时，离散点不易识别，且字本身呈实体状，无需充填，其注记效果明显优于空心汉字。

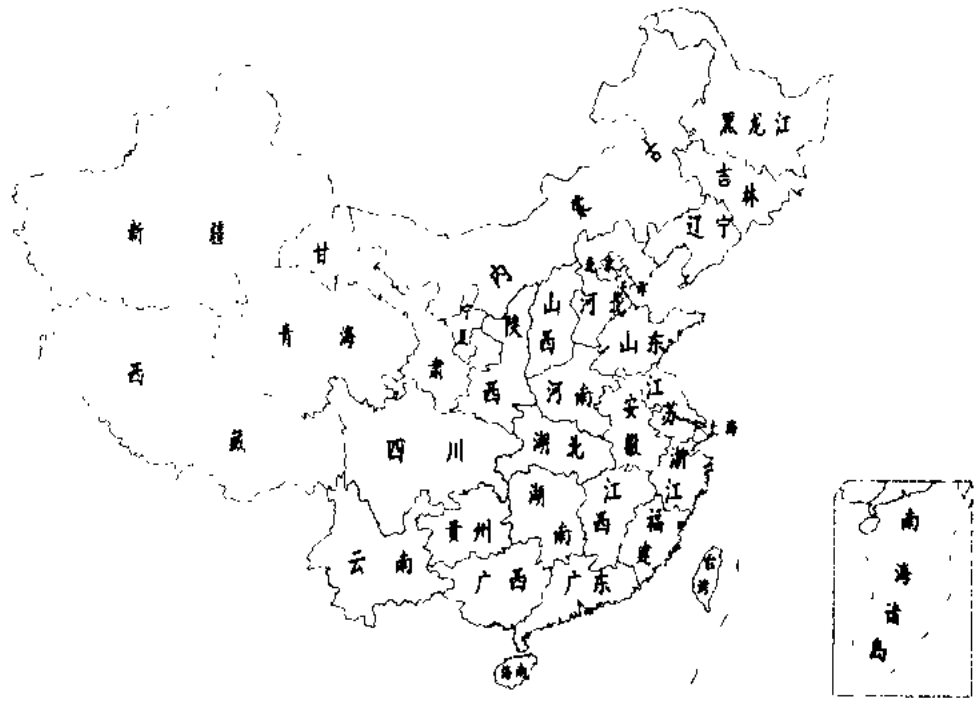


图5—2　独立数字化层与字符集汉字实用效果对比图

另一方面，由点阵汉字库的信息经GRIDPOLY命令生成的空心汉字数字化层也不宜

放大，因为此法生成的汉字虽不是点式而是矢量式，但本质属轮廓跟踪生成的汉字。其锯齿状边界皆保留无遗，一旦超越一定显示尺度，将导致边缘锯齿过于明显以至不便使用(图 5—1)，所以此法虽然方便，但绝非汉字注记的最佳方式。作为小号注记汉字它不如点阵式实体汉字，而作为大号标题汉字它又不如基于矢量汉字信息生成的空心汉字。

3. 独立数字化层方式与字符集方式的对比

图 5—2 中的图名系用独立数字化层方式生成的一组汉字注记的，而图中的省名则是用字符集方式生成的单个汉字注记的。实用效果表明，前者便于放大、充填，生成不同字体的图名和标题，但不易于改动，且只能一次性使用；而后者可根据用户的需要随时改变其字体、大小、方向及注记位置，每个汉字皆可多次调用到不同的位置上。

五、 AutoCAD 中的汉字注记

AutoCAD 自 1982 年由美国 Autodesk 公司推出以来，一直是一个深受用户欢迎的微机交互式绘图软件包。AutoCAD 中的汉字注记除采用与上几节相同的方法外，还可将AutoCAD本身的诸多功能，直接应用于汉字的显示注记。与在pcARC/INFO里的应用一样，AutoCAD 中的汉字注记也分为几种方式。下面分别阐述。

(一) 图层方式或命令组方式

这里的图层方式与 pcARC/INFO 中的数字化层相似，即将经预处理后得到的汉字ASCII数据文件读入AutoCAD，而将汉字笔画视为一组线型图元，其实现途径可以是用LISP语言编程；或者借助一个程序，直接将汉字数据文件转换为命令组文件；甚至可在任何文本编辑状态下，将已有汉字数据文件经编辑处理，生成一个扩展名为 SCR 的命令组文件，即可在ACAD下进行调用。下面即为“北”字的命令组文件的一部分：

```
Line  54.00,189.00  71.00,187.00  69.00,181.00  69.00,40.00  54.00,40.00
54.00,64.00  50.69,62.48  46.94,61.42  43.25,60.31  39.61,59.17  36.03,57.98
32.50,56.75  29.03,55.48  25.61,54.17  22.25,52.81  18.94,51.42  15.69,49.98
```

因为只涉及笔画轮廓，所以就图元而言，只涉及线型图元LINE，所有的内插数据，皆可在运行本书提供的原程序时自动生成，无需用户再次插值。与pcARC/INFO的GENERATE命令对数据格式的要求的主要区别在于，SCR 文件中对数据间空格个数的要求十分严格，这就为数据格式的转换带来不便。

(二) DXF 文件方式

将原汉字数据文件直接或间接转为 DXF 文件格式亦能为 AutoCAD 所接受。但该文件格式较为复杂，代码较多，稍有偏差就可能字不成字。所以这里提出一种间接方式，即利用其它软件进行转换，或者直接用某种高级语言编程来实现此格式转换。比如前述pcARC/INFO中就可以实现与 DXF 格式的互换。编程转换这里不作进一步讨论。

(三) LISP 语言方式

AutoCAD 提供的 LISP 语言亦可用于汉字的显示，但考虑到本书讨论的矢量汉字或点

阵汉字的数据已经经过了预处理，即可直接调用LISP语言编程进行绘图。由于LISP语言对数据格式的限制不如 SCR 文件那样严，所以其实用性要优于命令组方式。该语言的一大优势是可在程序运行过程中生成数据，以供绘图时使用。

（四） 直接数字化方式

将图名等注记汉字直接由数字化仪输入，当然也不失为一种实用的输入方式。但此法生成的汉字一般质量不高，而且操作太繁，除非是在其它手段皆不具备的情况下，才于以使用。使用本书提供的汉字转换程序又可生成多达 8 种字体的可无级放大的矢量汉字及四种常见的字体的点阵汉字，所以直接数字化方式只对一些特殊的手书体的输入有用。

（五） 字符集方式

同 pcARC/INFO，在 AutoCAD 中也可以生成用户字符集方式。与前述几种方式生成的图层方式相比，字符集方式调用方便，且可供多次使用，可大大减少重复劳动。所不同的是AutoCAD中的字符集对线段的限制较多，以至一些复杂汉字无法按通常方式存入字符集。按AutoCAD要求生成用户自己的 FONT.SHX 文件也不失为一种实用有效的汉字注记方式。虽然也是形文件方式，与一般形文件不同的是，这里的源数据已经经过预处理，完全可以避免单线化线条汉字的诸多弊端而满足用户对不同汉字字体变化的要求。有关形文件的格式等已见本书第一章。在这里值得一提的是，形文件中使用的整型数，而且是按方向进行编码，所以对矢量数据而言，这就意味着要重新进行删格化，并按方向进行编码。对每个字形来讲，其字节数不得超过 1000，即只能有 1000 个以下的小线段，考虑到汉字笔画的内插，此项限定，无疑会影响到某些复杂笔画汉字的生成。

六、 VAX/VMS 系统下的汉字注记

以上所述的汉字注记，尽管软件不同，但其操作系统皆为 DOS 系统，而国内 80 年代乃至近年引进的图像处理系统中的大多数使用的是非 DOS 系统。如何解决非 DOS 系统下汉字的注记问题，也是广大中文用户所关心的问题。在这里仅以VAX/VMS系统为例，介绍有关非 DOS 系统下汉字的注记问题。

与 DOS 系统一样，非 DOS 系统西文状态直接注记汉字，亦要首先将用到的汉字信息转为通用格式，然后再根据各个不同的机型对数据格式的要求进行转换。基于大量试验，其实际可行的途径可分为以下几类：

（一） 实用数据的传输与转换

众所周知，对非 DOS 系统与 DOS 系统之间的通讯必须依赖于某种异步通讯软件，如 VAX/VMS与 DOS 之间的 Kermit 软件等。此类软件一般只能模拟对方的终端，调用对方的文件或数据，并不能直接运行在对方系统下运行的可执行程序。所以说到底，主要用于进行数据的传输。所谓实用数据的传输主要是指将已组织好并经预处理的通用格式汉字数据，直接传到VAX机中，并使之能在VMS操作系统下调用。经预处理的汉字信息已经为通用的ASCII数据文件，且其格式为已知。所以无论是在 DOS 一端处理成对方接受格式，还

是传至对方后再作变动，皆可使之满足非DOS系统，如VAX/VMS操作系统的要求。就汉字显示注记而言，可调用本书第六章提到的 WT 图像生成程序直接运行，即可生成一个与要注记遥感图像同样大小的图像文件或是二值的专题覆盖文件。至于遥感图像中的地名等零散分布的汉字的注记，既可先生成一个统一的覆盖文件，也可分别生成单个汉字图像文件，再经常规的图像处理合成为一个统一的包含所有注记汉字在内的图像文件。问题的关键并不在图像的生成，而在于每组注记汉字的定位，即确定每组注记汉字的行列坐标。

实用型数据传输速度快，且不在 VMS 系统中占太多磁盘空间。缺点是传输后所用到的汉字已经固定，如有改动，还必须回到 DOS 系统下重新生成汉字数据文件，重新传输。实践证明，尽管这样，这种传输方式仍不失为一种行之有效的汉字注记方式。

（二） 字库传输方式

字库传输方式，就是将原本在 DOS 系统下使用的各类字体的字库全部传至VAX/VMS下，虽然数据量大，但可以“一劳永逸”，一次传输，反复使用。至于汉字调用，则可将DOS下的源程序传至 VAX/VMS 机中，略加改动，重新编译后运行。这样，即可在VAX/VMS系统下调用不同字体的汉字，并用图像文件或是注记用的覆盖文件的方式直接生成所需的注记。此种方式灵活方便，但要占据较多的存储空间。就注记汉字的效果而言与实用数据传输方式无异。

（三） 矢量汉字的注记

作为狭义的图像文件，只能接受点阵汉字，所以对矢量汉字而言，或者将其栅格化，或者只用其生成非图像文件，或图形文件，然后作为覆盖文件叠加于图像文件之上，以达到注记之目的。实验对比表明，此法以注记标题字或是连同图框一同生成为宜。至于图像内部的汉字注记仍建议使用点阵汉字。

（四）手书汉字的注记

在 VAX/VMS 的实际应用中，早有人通过数字化方式将手书汉字（也可以是印刷体汉字）将汉字输入到系统之中，只是操作繁锁且精度难以得到保障，所以很难付诸实用。现在扫描仪的投入使用可从根本上解决汉字的数字化精度问题，因为对手扶跟踪法来讲，100dpi的精度都是望尘莫及的，更不用说300dpi，600dpi乃至1200dpi的精度了。至于扫描仪生成的文件格式，因设备而异，现仅以 TH-1200 为例说明之。

TH－1200 扫描仪生成的 TIF 文件的格式已在有关章节详细讨论过。这里只强调该文件向图像文件的转换问题，实际上，TIF4程序生成的已经是通用ASCII文件，只需在格式上略加改动，使之满足 WT 程序对源数据格式的要求即可，所以在实现上并无任何障碍。唯一值得注意的是在分辨率提高的同时，注记用手书汉字数据所占空间无疑也要加大，二者之间的权衡必须由用户在数据输入之时即行考虑。一旦生成TIF文件，只需调用TIF4 程序即可转为VAX/VMS所需格式，然后经KERMIT将数据传入VAX机即可。当然如有与VAX配套的扫描仪，可节省传输之劳，但对其格式仍需分析研究，并研制相应的格式转换程序。

第六章　遥感非遥感地质信息的计算机复合处理

一、 复合的基础

本书的第四章第二节曾讨论了数据格式转换的基准，特别强调了以微机DOS系统为统一的数据平台。从实际应用角度来看，格式转换只不过是个中间过程，其目的还在于不同信息来源的地质数据的综合评价或复合应用。正如第一章有关发展现状与趋势部分所指出的那样，遥感非遥感地质信息复合应用势在必行，它可使不同地质手段的优势互补，并在地质调查中发挥出综合实力。格式的转换只是将计算机处理的数据结构、格式进行了统一，并未对反映地质体信息的空间关系进行统一定位，而且数据的总体类型也未进行统一。本章要讨论的数据复合问题，就是以在统一定位坐标系内，使用统一的数据类型为基础的。前述格式转换已为此奠定了基础。为确保不同地质探测手段所获取的信息得以相互沟通，并进而借助不同的数学变换，实现复合处理，首先要解决的就是复合的基础问题，即

- 地理坐标的配准；
- 数据类型的统一。

（一） 地理坐标的配准

前已述及，任何地质探测手段获得的信息，都是统一地质体中的某一侧面的局部信息，而这些局部信息皆应隶属于统一的主体。那么何以确定这些局部信息之间的相互关系呢？一个重要的基准就是数据采集点的空间分布。即使是在几百公里高空得到的遥感信息，其中的每个象元也必然能与地面上的某点相对应。缺少定位记录的任何地质信息，都会像野外地质调查中没有点位的地质记录一样，失去其实际意义。这里的定位记录就是其地理坐标。在通常所用的坐标系中，空间任一点的位置皆可由其 x、y、z 三个坐标值来确定，数据采集点的定位也不例外。但是由于许多无法回避的原因，不同探测系统使用的坐标系不统一造成了彼此之间的不可比性，致使实际上同一点位的坐标值在不同的数据采集系统中并不一致；甚至在同一矿区内，不同时间所测得的同一孔位的坐标值也不一致。这里有可能为数据的合理评价带来不便，甚至混乱。所以作为信息复合应用的第一步，就是要统一坐标系。这里的统一既可以是狭义的，也可以是广义的。前者可以70年代不同遥感图像之间的配准为例，如将MSS图像与TM图像配准，或将雷达图像与TM图像配准。这里的所谓配准只是两种数据之间的点位配准，并不一定与地理坐标系相统一；后者则可以本书第三章提到的基础影像图为例，因为采用了相应比例尺的国际分幅标准，各幅图像皆有统一的地理坐标，只要提供图幅编号或某点的地理坐标，即可在相应的位置查询到相关地质信息。下面要讨论的数据的复合，就是以统一地理坐标系为大前提的。这样，每一点位皆可能有一组来自不同信息源的数据。比如，在平面上纵横坐标完全相同的一个点，就可能与从地表到地下一系列不同标高处的数据相对应；另一方面，在地下任一标高处又可能查到多源数据的平面分布特征。如果没有地理坐标的统一，所有的数据转换就可能失去其意义，多元信息的复合也就无从谈起。

（二） 数据类型的统一

所谓数据类型的统一是指参与数据复合应用的多元信息的总体类型的统一。比如在统一的地理坐标系给定某点坐标时，可以遥感图像上找出不同波段的象元灰度值，但在矢量格式生成的等值线图上却不一定找到与之对应的确定的值。这样一来，二者之间尽管已有统一的地理坐标系，也不便进行复合处理。简而言之，数据类型的转换就是要用栅格类数据与同比例尺的栅格类数据进行复合处理；用矢量类数据与同比例尺的矢量类数据进行复合处理。如pcARC/INFO中，数学模型的建立，实际上就是在经过匹配之后的多个数字化层的矢量类数据之间建立某种数学模型；而不同遥感图像之间经过配准后，进行的合成处理，则可视之为栅格类数据之间的复合处理。那么如何实现矢量类与栅格类数据之间的复合处理呢？问题的关健就在于数据类型的统一，要么要统一为矢量类，要么统一为栅格类，要么在二者之间架起某种联系的桥梁，其基本原则就是在不同来源的地质信息之间，经过这种转换，建立起数据之间的一一对应关系。

二、 不同类型数据之间的相互转换

如前所述，实现复合处理的基础是地理坐标的配准和数据类型的统一。其实际可行的处理方式可大致分为：矢量类向栅格类的转换和栅格类向矢量类的转换。

（一） 矢量类向栅格类数据的转换

绝大多数非遥感地质数据皆属矢量类数据。无论其成果是以平面图，还是剖面图方式，要想与相同地区的栅格类遥感数据进行复合处理，首先就要将矢量类数据转换为栅格类数据。以常见的各种等值线图为例，其原数据多为散点式矢量数据，无论手绘还是机助制图，得到的等值图上只有在线上才有相应的特征值，而在线与线之间，只能靠估值来定。要实现与栅格类数据的复合，就必须在两者之间建立一一对应关系，即图幅的范围大小一致，定位尺度一致，象元大小（即栅格大小）一致。程序 SPL1. FOR 即为用 FORTRAN 编制的样条插值程序，并在MICROVAX机上调试运行成功。该程序可直接将含 x、y、z 值的散点数据转为栅格数据。程序运行中用户可根据原数据的分布自定图幅边界，即 x_{min}、x_{max}、y_{min}、y_{max}，和象元大小，即 x、y 方向的间隔大小。从而生成一个拟与之复合的遥感图像的象元多少及排列次序完全一致的栅格数据。至于图像的生成可直接调用程序 WT. FOR 即可。该程序只需输入一个数据文件名和一个 9 个字符的图像文件名，即可生成栅格数据的图像，并可以在VAX机的ARIES－Ⅱ中显示。

```
C  * * * * * SURFACE SPLINE PROGRAM  SPL1. FOR  * * * * * *
C  * * * * * DATA TYPE AND DIMENSION
       BYTE KY(20,32256)
       CHARACTER FI * 8,FO * 12,W
       DIMENSION Z(645120),KZ(32256)
       COMMON /OS/X(200,3),F(203)
       COMMON /OA/A(20706)
       COMMON /SJ/B(203),C(203)
```

```
C  * * * * * INPUT PARAMETERS OF OLD AND NEW DATA FILE
      PRINT * ,'   * * * * *      WARNING!       * * * * *'
      PRINT * ,'   THE NUMBER OF DATA GROUP N MUST BE LESS OR EQUAL 200'
      PRINT * ,'   THE TOTAL DATA QUANTITY TO BE GENERATED MUST'
      PRINT * ,'   NOT BE GREATER THAN 645120(1260 BLOCKS)'
      PRINT * ,'   WILL YOU CONTINUE(Y/N)? '
      READ( * ,'(A)') W
      IF((W.EQ.'Y').OR.(W.EQ.'y')) GOTO 909
      IF((W.NE.'Y').OR.(W.NE.'y')) GOTO 999
  909 TYPE 22
   22 FORMAT(1X,'INPUT SOURCE DATA FILE NAME, E.G. DI0:XXXX.DAT')
      READ( * ,77) FI
   77 FORMAT(A8)
      TYPE 23
   23 FORMAT(1X,'INPUT TARGET DATA FILE NAME, E.G. DI0:YYYY.DAT')
      READ( * ,'(A)') FO

C  * * * * * READS OLD DATA FILE
      OPEN(UNIT=1,FILE=FI,STATUS='OLD')
      TYPE 24
   24 FORMAT(1X,'INPUT NUMBER OF DATA GROUPS N=')

C  * * * * * INPUT PARAMETERS FOR GENERATING A NEW DATA FILE
      READ( * , * ) N
      READ(1, * )((X(I,J),J=1,3),I=1,N)
      TYPE 33
   33 FORMAT(1X,'INPUT EPS=, E.G. EPS=0.001')
      READ( * , * ) EPS
      CALL SSPL1(N,EPS)
      XMI=10E30
      XMA=-10E30
      YMI=10E30
      YMA=-10E30
      DO 30 I=1,N
      IF(XMI.GT.X(I,1)) XMI=X(I,1)
      IF(XMA.LT.X(I,1)) XMA=X(I,1)
      IF(YMI.GT.X(I,2)) YMI=X(I,2)
      IF(YMA.LT.X(I,2)) YMA=X(I,2)
   30 CONTINUE
      PRINT * ,'XMI=',XMI,'XMA=',XMA,'YMI=',YMI,'YMA=',YMA
      PRINT * ,'  DO YOU WANT TO CHANGE THESE VALUES (Y/N)? '
      READ( * ,'(A)') W
      IF((W.EQ.'n').OR.(W.EQ.'N')) GOTO 367
      PRINT * ,'  PLEASE ENTER THE NEW VALUES. (XMI,XMA,YMI,YMA)'
      READ * ,XMI,XMA,YMI,YMA
  367 WRITE( * ,40) XMI,XMA,YMI,YMA
   40 FORMAT(1X,4HXMI=,F11.1,5X,4HXMA=,F11.1,5X,4HYMI=,F11.1,5X,
     *   4HYMA=,F11.1)
      WRITE( * ,25) XMA-XMI,YMA-YMI
   25 FORMAT(1X,6HXLONG=,F15.2,5X,6HYLONG=,F15.2)
      TYPE 26
```

```
 26   FORMAT(1X,'INPUT PIXEL SIZE PX=')
      READ * ,PX
      M=(INT((XMA-XMI)/PX+1)/4) * 4
      L=(YMA-YMI)/PX+1
      PRINT * ,'NUMBER OF LINES L=',L
      PRINT * ,'NUMBER OF PIXELS M=',M

C   * * * * * GENERATING THE NEW DATA FILE
      DX=PX
      DY=PX
      CALL SSPL2(N,EPS,M,L,DX,DY,K,XMI,YMA,Z)
      ZMI=10E30
      ZMA=-10E30
      DO 36 IZ=1,K
      IF(ZMI.GT.Z(IZ)) ZMI=Z(IZ)
      IF(ZMA.LT.Z(IZ)) ZMA=Z(IZ)
 36   CONTINUE
      WRITE( * , * ) ZMI,ZMA
      XK=ZMA-ZMI
      PX=255./XK
      LL=K/512+1
      NN=LL/62+1
      IF(MOD(LL,NN).EQ.0) NB=LL/NN
      IF(MOD(LL,NN).NE.0) NB=LL/NN+1
      IF(NN.LE.1) MUC=LL
      IF(NN.GT.1) MUC=NB * NN
      MK=NB * 512
      LK=LL * 512
      PRINT * ,LL,LK
      IF(LK.LE.645120) THEN
      PRINT * ,'NEW DATA FILE CREATED'
      ELSE
      GOTO 989
      END IF

C   * * * * * WRITING THE NEW DATA FILE
      OPEN(UNIT=2,FILE=FO,TYPE='NEW',FORM='UNFORMATTED')
      DO 201 MA=1,NN
      DO 202 MB=1,MK
      JM=(MA-1) * MK+MB
      IF(JM.LE.K) KZ(MB)=AINT((Z(JM)-ZMI) * PX+0.5)
      IF(JM.GT.K) KZ(MB)=1
      IF(KZ(MB).LE.127) KY(MA,MB)=KZ(MB)
      IF(KZ(MB).GT.127) KY(MA,MB)=KZ(MB)-256
 202  CONTINUE
      WRITE(2) (KY(MA,M),M=1,MK)
 201  CONTINUE
      PRINT * , (KY(1,J),J=1,100)
      CLOSE(UNIT=1)
      CLOSE(UNIT=2)
 989  PRINT * ,'    NO ENOUGH SPACE FOR NEW DATA FILE'
```

```
  999  STOP
       END
C   * * * * * * * * * * * * SSPL1(N,EPS) * * * * * * * * * * *
C   ---------------------------------------------------
       SUBROUTINE SSPL1(N,EPS)
       COMMON /SJ/B(203),C(203)
       COMMON /OS/X(200,3),F(203)
       COMMON /OA/A(20706)
       DO 20 I=1,N
   20  C(I)=0.
       N1=N+1
       N2=N+2
       N3=N+3
       DO 60 I=1,N
       A(N*N1/2+I)=1.
       A(N1*N2/2+I)=X(I,1)
       A(N2*N3/2+I)=X(I,2)
       A(I*(I+1)/2)=C(I)
       B(I)=X(I,3)
       DO 40 J=1,I-1
       R2=(X(I,1)-X(J,1))**2+(X(I,2)-X(J,2))**2
   40  A((I-1)*I/2+J)=R2*LOG(R2+EPS)
   60  CONTINUE
       DO 80 I=1,3
       DO 80 J=1,I
   80  A((N+I)*(N1+I)/2-I+J)=0.
       DO 100 I=N1,N3
  100  B(I)=0.
       CALL HOUSE(N3)
       RETURN
       END
C   ==================================================
       SUBROUTINE SSPL2(N,EPS,M,L,DX,DY,K,XMIN,YMAX,Z)
       COMMON /OS/X(200,3),F(203)
       DIMENSION Z(650000)
       K=0
       Y=YMAX+DY
       DO 20 I=1,L
       X1=XMIN-DX
       Y=Y-DY
       DO 20 J=1,M
       K=K+1
       X1=X1+DX
       Z(K)=F(N+1)+F(N+2)*X1+F(N+3)*Y
       DO 10 I1=1,N
       R=(X1-X(I1,1))**2+(Y-X(I1,2))**2
   10  Z(K)=Z(K)+F(I1)*R*LOG(R+EPS)
   20  CONTINUE
       RETURN
       END
```

```
C  * * * * * * * * * * * * HOUSE  * * * * * * * * * * *
C  ---------------------------------------------
      SUBROUTINE HOUSE(N)
      DIMENSION U(2),V(2)
      COMMON /SJ/B(203),D(203)
      COMMON /OS/X(200,3),F(208)
      COMMON /OA/A(20706)
      WRITE(*,'(A)') '     HOUSEHOLDER PROCEDURE'
      NN=N*(N+1)/2
      N2=N-2
      DO 140 K=1,N2
      S=0.
      K1=K+1
      DO 20 I=K1,N
   20 S=S+A((I-1)*I/2+K)**2
      E=SQRT(S)
      L=K*(K+3)/2
      H1=S+ABS(A(L))*E
      D(K+1)=SIGN(1.,-A(L))*E
      A(L)=A(L)-D(K+1)
      DO 60 I=K1,N
      F(I)=0.
      DO 40 J=K1,N
      IF (I.GE.J) THEN
      II=I
      ELSE
      II=J
      END IF
      JJ=I+J-II
   40 F(I)=F(I)+A((II-1)*II/2+JJ)*A((J-1)*J/2+K)
   60 F(I)=F(I)/H1
      E=0.
      S=0.
      DO 80 I=K1,N
      L=(I-1)*I/2+K
      E=E+A(L)*F(I)
   80 S=S+A(L)*B(I)
      E=E/(2.*H1)
      S=S/H1
      DO 100 I=K1,N
      L=(I-1)*I/2+K
      F(I)=F(I)-E*A(L)
  100 B(I)=B(I)-S*A(L)
      DO 120 I=K1,N
      DO 120 J=K1,I
      L=(I-1)*I/2
  120 A(L+J)=A(L+J)-A(L+K)*F(J)-A((J-1)*J/2+K)*F(I)
  140 CONTINUE
      D(N)=A(NN-1)
      IF (ABS(A(NN)).LE.1.E-10) THEN
      DO 180 K=2,1,-1
```

```
      M=N-K
      H1=B(M)
      S=D(M)
      T=A(M*(M+1)/2)
      DO1 60 I=M,3,-1
      H1=H1-T*B(I-1)/D(I)
      E=S
      S=-T*D(I-1)/D(I)
160   T=E-T*A((I-1)*I/2)/D(I)
      U(K)=(A(1)*T/D(2)-S)/D(M+1)
180   V(K)=(H1-B(1)*T/D(2))/D(M+1)
      F(1)=(B(N)-D(N)*V(2)-A(NN)*V(1))/(D(N)*U(2)+A(NN)*U(1))
      F(2)=(B(1)-A(1)*F(1))/D(2)
      DO 200 I=2,N-1
200   F(I+1)=(B(I)-D(I)*F(I-1)-A(I*(I+1)/2)*F(I))/D(I+1)
      ELSE
      B(N)=B(N)/A(NN)
      A(NN)=-D(N)/A(NN)
      DO 220 I=2,N
      M=N+1-I
      M1=M+1
      A(M*M1/2)=D(M1)*A(M1*(M1+1)/2)+A(M*M1/2)
      B(M)=(B(M)-D(M1)*B(M1))/A(M*M1/2)
220   A(M*M1/2)=-D(M)/A(M*M1/2)
      F(1)=B(1)
      DO 240 I=N-1,1,-1
      M1=N+1-I
240   F(M1)=A(M1*(M1+1)/2)*F(N-I)+B(M1)
      END IF
      DO 280 K=N-2,1,-1
      K1=K+1
      E=0.
      H1=ABS((D(K+1)*A(K*(K+3)/2)))
      DO 260 I=K1,N
260   E=E+A((I-1)*I/2+K)*F(I)
      E=E/H1
      DO 280 I=K1,N
280   F(I)=F(I)-E*A((I-1)*I/2+K)
      RETURN
      END
```

上述程序生成的数据经运行下列程序即可生成相应的图像。程序 WT. FOR 中调用的系统模块同程序 WC. FOR（见第三章），程序中主要变量有：

EK —— 生成图像的像元总数；

ETP —— 生成图像的总列数；

ETL —— 生成图像的总行数；

KH —— 为偏移量，其值等于 ETP 与其中 512 的整数倍之差；

RS —— 为偏移系数，随图像的行数变化而变化。

```
C   PROGRAM WT.FOR CTREATS AN IMAGE FROM A DATA FILE IN [AOS.OIS]
C   DATA TYPE AND DIMENSIN:
      IMPLICIT BYTE(B),DOUBLE PRECISION(D)
      IMPLICIT INTEGER*4(E),INTEGER*2(F-Q),REAL(R-S)
      BYTE EQ(7000),EY(600,1024)
      INTEGER IE
      CHARACTER WW,FNM*4
      COMMON/NO1BUF/BUF1(8000)
      COMMON/NO9LUN/LUN
      DATA BFO/9*' '/
      DATA BFI/9*' '/
      DIMENSION BFI(9),BFO(9),BBB(26),BDVV(11),BDV(11),
     *   BFLOGI(11),BFNAME(9),BTNAME(6),IOST(4),P(100),
     *   BDA(4),BDB(13),BDC(3),EP(600,1024),EBU(512),EBW(512)

C   INIT IATES THE SYSTEM AND IMAGE DISK
      F11LUN=2
      CALL XOINIT(XXXXXX,IER,BBB)
      CALL XXINIT(XXXXXX,IER)
      CALL XKDFIN(IMG0,XXXXXX,BTS,BTU,BCO,BCU)
      CALL XKDISK('B',XXXXXX,IER,BDV)
      CALL XKDLNK(BDV,XXXXXX,IER,YYYYYY,LUN)
      CALL XBSPRO('N','N','Y','N','N','N','N',XXXXXX,IER,YYYYYYY,1,3)

C   INPU T PARAMETERS FOR GENERATING AN IMAGE
      PRINT*
      PRINT*,'      *********************************'
      PRINT*,'      *                               *'
      PRINT*,'      *    THIS PROGRAM CREATES DIFFERENT IMAGES    *'
      PRINT*,'      *      BY ANALYSIS AND TEST CENTRE, CUMT      *'
      PRINT*,'      *                               *'
      PRINT*,'      *********************************'
      PRINT*
  111 CALL XOMESG(IER,1,9,BFO)
      DO 5 K=1,9
    5 IF(BFO(K).GE.97.AND.BFO(K).LE.122) BFO(K)=BFO(K)-32
      CALL XFFSEE(BDV,BFO,'I',1,XXX,IER,END,EAD,BFL,BFN,BFTY)
      IF(IER.EQ.-261) GOTO 222
      PRINT*,'      FILE EXISTS, REENTER FILE NAME'
      GOTO 111
  222 CALL XOMESG(IER,2,4,ETL) ! INPUT NUMBER OF LINES TO BE CREATED
      IF(IER.EQ.-273) GOTO 111
  333 CALL XOMESG(IER,3,4,ETP) ! INPUT NUMBER OF PIXELS TO BE CREATED
      IF(IER.EQ.-273) GOTO 222
  401 CALL XOMESG(IER,4,1,BX)  ! DO YOU HAVE A DATA FILE?
      IF(IER.EQ.-273) GOTO 333

      IF(BX.EQ.'y'.OR.BX.EQ.'Y') THEN  ! YOU HAVE A DATA FILE!
  444 CALL XOMESG(IER,5,4,BDB) ! INPUT DATA FILE NAME!
      FNM=CHAR(BDB(1))//CHAR(BDB(2))//CHAR(BDB(3))//CHAR(BDB(4))
      IF(IER.EQ.-273) GOTO 401
```

```
      CALL XOMESG(IER,7,1,BY)   ! IS YOUR DATA BYTE--TYPE OR NOT?
      IF(IER. EQ. -273) GOTO 444
      ELSE                      ! YOU HAVE NOT A DATA FILE
      GOTO 402
      ENDIF
      IF((BY. EQ. 'y'. OR. BY. EQ. 'Y'). AND. (ETP. GT. 1024. OR.
     *   ETL. GT. 600)) THEN
      PRINT *,'       BEYOND RANGE, PLEASE REENTER! '
      GOTO 222
      ENDIF

  402 CALL XOMESG(IER,8,1,BWW) ! CONTINUE?
      IF(BWW. EQ. 'n'. OR. BWW. EQ. 'N') GOTO 555
      IF((BWW. EQ. 'y'. OR. BWW. EQ. 'Y'). AND. (BX. EQ. 'n'. OR. BX. EQ. 'N')) GOTO

C     OPEN AN OLD DATA FILE
      IF(BY. EQ. 'y'. OR. BY. EQ. 'Y') THEN
      OPEN (UNIT=2,NAME=FNM,STATUS='OLD',FORM='UNFORMATTED')
      READ(2) ((EY(I,J),J=1,ETP),I=1,ETL)
      ELSE
      OPEN (UNIT=2,NAME=FNM,STATUS='OLD')
      READ(2,*) ((EP(L,K),K=1,ETP),L=1,ETL)
      ENDIF

  565 ETP=INT(ETP/4)*4
      EK=ETP*ETL
      EUC=INT(ETP*ETL-0.005)/512+1
      KH=ETP-(ETP/512)*512
      RS=FLOAT(KH)/512.
      IF(BX. NE. 'n'. AND. BX. NE. 'N') GOTO 707

      PRINT *
      PRINT *,'       * * * * * * * * * * MAIN MENU  * * * * * * * * * * *'
      PRINT *,'       * *                                          * *'
      PRINT *,'       * *     1. RANDOM IMAGE     2. BARS IMAGE    * *'
      PRINT *,'       * *                                          * *'
      PRINT *,'       * *     3. CHESS IMAGE      4. EXIT TO OIS   * *'
      PRINT *,'       * *                                          * *'
      PRINT *,'       * * * * * * * * * * * * * * * * * * * * * * * * *'
      PRINT *

  403 CALL XOMESG(IER,6,1,E6) ! GIVES YOUR SELECTION
      IF(E6. EQ. 4) GOTO 555
      IF(E6. EQ. 3) GOTO 555

      IF(E6. EQ. 1) THEN
  404 CALL XOMESG(IER,9,4,IE)
      END IF

      IF(E6. EQ. 2) THEN
```

```
405   CALL XOMESG(IER,10,4,E10) ! INPUT NUMBER OF COLUMNS

      PRINT * ,'       THERE ARE',E10,'      COLUMNS IN THE IMAGE'
      PRINT * ,'       INPUT GREY VALUE FOR EACH COLUMN PI[0/255]='
      READ( * , * ) (P(J),J=1,E10)

      DO 77 I=1,E10
      IF(P(I). LE. 127) P(I)=P(I)
      IF(P(I). GT. 127) P(I)=P(I)-256
 77   CONTINUE

      ELM=AINT((ETP-0. 005)/E10)+1
      EPP=ELM * (E10-1)
      ENDIF


C   CREA TES A NEW IMAGE
 707  CALL XFFCRE(BDV,BFO,'I',EUC,XXXXXX,IER,EEE)
      IF(IER. NE. 1) GOTO 555
      EFL=1
      EFP=1
      CALL XFWDIM(BDV,1,BFO,ETL,ETP,EFL,EFP,1,XXXXXX,IER)

      DO 1000 II=1,ETL
      EKK=II
      EKP=1
      RT=RS * FLOAT(II-1)
      RA=(RT-INT(RT)) * 512
      IOF1=INT(RA)

      RT0=RS * FLOAT(II)
      RA0=(RT0-INT(RT0)) * 512
      IOF0=INT(RA0)

      DO 1001 JJ=1,ETP+IOF1

      IF(BX. EQ. 'y'. OR. BX. EQ. 'Y') THEN
      IF(BY. EQ. 'y'. OR. BY. EQ. 'Y') THEN
      EQ(JJ)=EY(II,JJ)
      ELSE
      CALL LIB$CVT--DX--DX(%DESCR(EP(II,JJ)),%DESCR(EQ(JJ)))
      ENDIF
      IF(EQ(JJ). GT. 127) EQ(JJ)=EQ(JJ)-256
      ENDIF

      IF(E6. EQ. 1) THEN
      EQ(JJ)=AINT(128. 0 * RAN(IE))
      IF(EQ(JJ). GT. 127) EQ(JJ)=EQ(JJ)-256
      ENDIF

      IF(E6. EQ. 2) THEN
```

```
      K=(JJ-0.005)/ELM+1
      IF(JJ.GT.EPP) THEN
      EQ(JJ)=P(E10)
      ELSE
      EQ(JJ)=P(K)
      ENDIF
      ENDIF

      K0=JJ/512

      IF(JJ.GT.IOF1) THEN
      BUF1(JJ)=EQ(JJ-IOF1)
      ELSE
      BUF1(JJ)=EBU(JJ)
      ENDIF

1001  CONTINUE

      DO 22 JK=1,IOF0
      MK=JK+K0*512
      EBU(JK)=BUF1(MK)
  22  CONTINUE

      CALL XAWWLI(LUN,EEE,ETP,EKK,EKP,ETP,IOF1,BUF1,1,XXXXXX,IER)
1000  CONTINUE

 555  CALL XFFCLO(BDV,BFO,'I',XXXXXX,IER,NOP)
      IF(BX.EQ.'y'.OR.BX.EQ.'Y') CLOSE(UNIT=2)
      CALL XOEXIT
      STOP
      END
```

（二） 栅格数据的矢量化

与一般教科书中栅格数据的矢量化不同，遥感、非遥感地质信息复合处理中的矢量化，还必需考虑数据的地质意义，即不是纯数学上的转换，而是与实用相结合的转换。本书第五章提到的TIFF文件的汉字转换可属于标准的栅格数据矢量化，即勾绘出轮廓线。对大量的遥感信息而言，除非经过诸如纹理增强等特殊处理，一般图像中各象元之间，皆有较大的灰度差，泛泛的矢量化必然会使最终成果杂乱无章。所以在这里强调的矢量化主要是指自动分类和目视解译成果的矢量化。前者可用于土地利用或环境调查等易于实现自动分类的研究目标。分类之后的遥感图像可视为有若干“图斑”或不规则多边形的栅格图像，可直接用轮廓跟踪法实现其数据的矢量化；对后者，即目视解译成果，如在第三章所提出的基础影像图得到的目视解译成果，实际上已经转为平面图形信息，需借助数字化的办法，才能将其转为矢量类数据，并与其他地质信息进行复合处理。需要强调指出的是无论是自动分类结果，还是目视解译成果，皆应加注其地质特征（属性数据），还不只是简单的数据类型转换。

三、 复合处理方法

（一） 多元数据统计分析

对于非遥感地质信息，由于已转换成通用格式，故已可以进行综合处理了。对于遥感信息，若点位置用 x，y 坐标描述，灰度值用另一变量描述，则亦构成一个三元数据集。所以从数据处理角度考虑，所有地质信息是等价的，常用的多元数据统计分析方法——趋势面分析、回归分析、聚类分析、因子分析、判别分析等对此都适用。但由于它们的信息来源不同，包含的地质含意也不同，所以在复合、叠加前，应作必要的预处理或确定各自的权因子。

在多元统计分析中，常用的预处理方法有标准化（新数据的均值为 0，标准差为 1）和正规化（新数据的区间为－1—1）。采用这二种预处理方法，可消除数据间的量级差别，但不能确定各类数据的权因于。

权因子的确定因视实际情况而定，有的可据经验而定，有的可借助多元统计方法如回归分析、判别分析等来确定。

（二） 曲面样条函数插值

许多地质数据是从空间曲面采样而得到的，如由钻孔资料得到的煤层底板或顶板标高，反映了煤层底板或顶板曲面的变化情况；还有一些数据并不代表真实存在的空间曲面，如由遥感数据采集得到的灰度值，但加上 x，y 坐标，它们代表了一个反映某些地质特性的数值曲面。对上述这些数据进行复合处理的基础之一是将它们所代表的曲面形态“模拟”出来，以统一定位标准。曲面“模拟”有拟合和插值计算两种，前者如趋势面分析等，后者有线性、样条插值等。在二元数据插值方法中，曲面样条函数方法是较好的一种，目前已广泛应用于地质数据处理中。它的特点是：已知点平面坐标不必按规则点阵排列，采取自然边界条件而不需边界导数的信息，得到的曲面任意阶可微。所以曲面样条函数比较适合各种数值曲面的模拟。

1. 数学方法

已知条件：在待估区域内，已知一组曲面坐标 $(x_i,y_i,z_i)(i=1,2,3,\ldots,n)$。曲面样条函数的表达式为：

$$Z(x,y)=a_0+a_1x+a_2y+\sum_{i=1}^{n}F_ir_i^2\ln(r_i^2+\varepsilon) \tag{1}$$

其中 x,y 为待估点坐标，$r_i^2=(x-x_i)^2+(y-y_i)^2$，$\varepsilon$ 为一小量（取 10^{-2}—10^{-8}），$F_i(i=1,2,\cdots,n)$、a_0、a_1、a_2 为待求系数（共 $n+3$ 个）。

$F_i(i=1,2,\cdots,n)$、a_0、a_1、a_2 可用已知点数据通过下列方程组求出：

$$
\begin{cases}
Z_j = a_0 + a_1x_j + a_2y_j + \sum_{i=1}^{n} F_i r_{ij}^2 \ln(r_{ij}^2 + \varepsilon) + C_jF_j \\
\qquad (j = 1,2,\cdots,n, r_{ij}^2 = (x_i - x_j)^2 + (y_i - y_j)^2) \\
\sum_{i=1}^{n} F_i = 0 \\
\sum_{i=1}^{n} x_iF_i = 0 \\
\sum_{i=1}^{n} y_iF_i = 0
\end{cases}
\tag{2}
$$

其中 $C_j \geqslant 0(j = 1,2,\cdots,n)$ 决定曲面是否通过 (x_j, y_j, z_j) 点，若 C_j 等于零，则曲面通过该点；否则通过 $(x_j, y_j, z_j - C_jF_j)$。在实际计算中，一般都取 $C_j = 0$。若令

$$
\boldsymbol{A} = \begin{bmatrix}
C_1 & r_{12}^2\ln(r_{12}^2+\varepsilon) & \cdots & r_{1n}^2\ln(r_{1n}^2+\varepsilon) & 1 & x_1 & y_1 \\
r_{12}^2\ln(r_{12}^2+\varepsilon) & C_2 & \cdots & r_{2n}^2\ln(r_{2n}^2+\varepsilon) & 1 & x_2 & y_2 \\
\vdots & & \ddots & \vdots & \vdots & \vdots & \vdots \\
\vdots & & C_{n-1} & r_{n-1,n}^2\ln(r_{n-1,n}^2+\varepsilon) & 1 & x_{n-1} & y_{n-1} \\
r_{1n}^2\ln(r_{1n}^2+\varepsilon) & \cdots & r_{n-1,n}^2\ln(r_{n-1,n}^2+\varepsilon) & C_n & 1 & x_n & y_n \\
1 & \cdots & 1 & 1 & 0 & 0 & 0 \\
x_1 & \cdots & x_{n-1} & x_n & 0 & 0 & 0 \\
y_1 & \cdots & y_{n-1} & y_n & 0 & 0 & 0
\end{bmatrix}
$$

$X = (F_1, F_2, \cdots, F_n, a_0, a_1, a_2)^T$

$B = (Z_1, Z_2, \cdots, Z_n, 0, 0, 0)^T$

则上述方程组可以写成矩阵形式：$\boldsymbol{AX} = \boldsymbol{B}$

2. 计算程序

上述方程式中 $\boldsymbol{A}$ 为一对称矩阵，且一般而言主对角线元素都为零，需用特殊方法求其逆矩阵。为此可用 Householder 镜像变换法求解。这里我们附了三个子程序。子程序 SSPL0 用于生成矩阵 $\boldsymbol{A}$、$\boldsymbol{B}$，子程序 HOUSE 的功能是用 HOUSEHOLDER 法来求解 $\boldsymbol{A}$ 的逆矩阵，子程序 SSPL1 的功能是利用已求得的系数，对给定的 x，y 点求出 z 值。这三个子程序都用到了一个公用块，其中 X（400，3）用来存放原始点坐标 x_i、y_i、z_i，F（403)用来存放系数 F_i（i=1，2，…，n）、a_0、a_1、a_2，一维数组 A 用来存放矩阵 **A** 的数据。

子程序 SSPL0 的虚参意义为：

N——原始数据数；

EPS——即 ε 。

子程序 HOUSE 中虚参 N 代表方程组阶数。

子程序 SSPL1 中虚参 N、EPS 同 SSPL0，其余虚参意义为：

XX—待求点 x 的值；

YY—待求点 y 的值；

WXY —待求点 z 的值。

```
      SUBROUTINE SSPL0(N,EPS)
      COMMON /OS/X(400,3),F(403),B(403)
      COMMON /OA/A(81406)
      DIMENSION C(400)
      DO 20 I=1,N
   20 C(I)=0.
      N1=N+1
      N2=N+2
      N3=N+3
      DO 60 I=1,N
      A(N*N1/2+I)=1.
      A(N1*N2/2+I)=X(I,1)
      A(N2*N3/2+I)=X(I,2)
      A(I*(I+1)/2)=C(I)
      B(I)=X(I,3)
      DO 40 J=1,I-1
      R2=(X(I,1)-X(J,1))**2+(X(I,2)-X(J,2))**2
   40 A((I-1)*I/2+J)=R2*LOG(R2+EPS)
   60 CONTINUE
      DO 80 I=1,3
      DO 80 J=1,1
   80 A((N+I)*(N1+I)/2-I+J)=0.
      DO 100 I=N1,N3
  100 B(I)=0.
      END

      SUBROUTINE HOUSE(N)
      COMMON /OS/X(400,3),F(403),B(403)
      COMMON /OA/A(81406)
      DIMENSION U(2),V(2),D(403)
      WRITE (*,'(A)') ' HOUSEHOLDER PROCEDURE '
      NN=N*(N+1)/2
      N2=N  2
      DO 140 K=1,N2
      S=0.
      K1=K+1
      DO 20 I=K1,N
   20 S=S+A((I-1)*I/2+K)**2
      E=SQRT(S)
      L=K*(K+3)/2
      H1=S+ABS(A(L))*E
      D(K+1)=SIGN(1.,-A(L))*E
      A(L)=A(L)-D(K+1)
```

```
      DO 60 I=K1,N
      F(I)=0.
      DO 40 J=K1,N
      IF (I.GE.J) THEN
        II=I
      ELSE
        II=J
      END IF
      JJ=I+J-II
   40 F(I)=F(I)+A((II-1)*II/2+JJ)*A((J-1)*J/2+K)
   60 F(I)=F(I)/H1
      E=0.
      S=0.
      DO 80 I=K1,N
      L=(I-1)*I/2+K
      E=E+A(L)*F(I)
   80 S=S+A(L)*B(I)
      E=E/(2.0*H1)
      S=S/H1
      DO 100 I=K1,N
      L=(I-1)*I/2+K
      F(I)=F(I)-E*A(L)
  100 B(I)=B(I)-S*A(L)
      DO 120 I=K1,N
      DO 120 J=K1,I
      L=(I-1)*I/2
  120 A(L+J)=A(L+J)-A(L+K)*F(J)-A((J-1)*J/2+K)*F(I)
  140 CONTINUE
      D(N)=A(NN-1)
      IF (ABS(A(NN)).LE.1.E-10) THEN
        DO 180 K=2,1,-1
        M=N-K
        H1=B(M)
        S=D(M)
        T=A(M*(M+1)/2)
        DO 160 I=M,3,-1
        H1=H1-T*B(I-1)/D(I)
        E=S
        S=-T*D(I-1)/D(I)
  160   T=E-T*A((I-1)*I/2)/D(I)
        U(K)=(A(1)*T/D(2)-S)/D(M+1)
  180   V(K)=(H1-B(1)*T/D(2))/D(M+1)
        F(1)=(B(N)-D(N)*V(2)-A(NN)*V(1))/(D(N)*U(2)+A(NN)*U(1))
        F(2)=(B(1)-A(1)*F(1))/D(2)
        DO 200 I=2,N-1
  200   F(I+1)=(B(I)-D(I)*F(I-1)-A(I*(I+1)/2)*F(I))/D(I+1)
      ELSE
        B(N)=B(N)/A(NN)
        A(NN)=-D(N)/A(NN)
        DO 220 I=2,N
        M=N+1-I
```

```
      M1=M+1
      A(M*M1/2)=D(M1)*A(M1*(M1+1)/2)+A(M*M1/2)
      B(M)=(B(M)-D(M1)*B(M1))/A(M*M1/2)
220   A(M*M1/2)=-D(M)/A(M*M1/2)
      F(1)=B(1)
      DO 240 I=N-1,1,-1
      M1=N+1-I
240   F(M1)=A(M1*(M1+1)/2)*F(N-I)+B(M1)
      END IF
      DO 280 K=N-2,1,-1
      K1=K+1
      E=0.
      H1=ABS((D(K+1)*A(K*(K+3)/2)))
      DO 260 I=K1,N
260   E=E+A((I-1)*I/2+K)*F(I)
      E=E/H1
      DO 280 I=K1,N
280   F(I)=F(I)-E*A((I-1)*I/2+K)
      WRITE(*,'(A)') ' The Solution of Linear Equation Group'
      WRITE(*,300) (F(I),I=1,N)
300   FORMAT (5(2X,G11.5))
      END

      SUBROUTINE SSPL1(N,XX,YY,WXY,EPS)
      COMMON /OS/X(400,3),F(403),B(403)
      WXY=F(N+1)+F(N+2)*XX+F(N+3)*YY
      DO 40 I=1,N
      X1=XX-X(I,1)
      Y1=YY-X(I,2)
      R2=X1*X1+Y1*Y1
40    WXY=WXY+F(I)*R2*LOG(R2+EPS)
      END
```

我们从上面可以看到，曲面样条函数可对二个自变量的空间（数值）曲面进行插值计算，是解决此类问题的有力工具。目前地质探测手段十分丰富，同一地区可用不同手段获得许多种信息。若要对这些信息进行复合、转换、叠加，往往涉及三个或更多的自变量。由三个或更多的自变量决定的曲面，在数学上称之为超曲面。由于曲面样条函数具有良好的插值计算特性，我们可以利用它的形式来进行超曲面的插值计算，即将曲面样条函数扩展成超曲面函数。下面给出三元问题的数学表达式。

设该三个自变量为 x、y、t，因变量为 z，则三元超曲面样条函数为：

$$Z(x,y,t) = a_0 + a_1x + a_2y + a_3t + \sum_{i=1}^{n} F_i r_i^2 \ln(r_i^2 + \varepsilon) \tag{3}$$

其中 $r_i^2 = (x - x_i)^2 + (y - y_i)^2 + (t - t_i)^2$，此时共有 $n+4$ 个待定系数，即 $F_i(i = 1,2,\cdots,n)$ 和 a_0、a_1、a_2、a_3。

四元以上问题可作类似推广。

利用超曲面样条函数可以进行信息反演、时空转换、建立地质模型等多项工作。

四、 应用实例

（一） 遥感与重力异常数据复合处理在徐淮地区找煤预测中的应用

地处苏、鲁、豫、皖四省交界处的徐淮地区，作为我国重要的煤炭、电力基地之一，急需大量的煤炭资源。但这个素以富煤著称的徐淮盆地，却面临着煤炭后备基地不足的严重威胁。由于该地区工作程度较高，且多为厚达百米以上的第四系所覆盖，这就为煤炭资源的预测增加的很大的难度，传统的地质调查手段已很难取得突破性进展。作为遥感技术应用的新尝试，1989 至 1991 年在本地区立项开展了 1：20 万遥感地质调查。作者在参加该项课题工作中，探讨了将遥感地质信息与重力异常信息复合处理的可行性，并在该地区找煤预测中取得了较为理想的效果。

该地区除淮阳山脉外，皆为第四系所覆盖，区内的煤系地层皆分布于第四系覆盖之下。裸露区出露的主要为寒武系、奥陶系。尽管如此，遥感影像仍能较好的展示出区域性构造格架。在目视解译、野外调查和构造分析的基础上，课题组对该地区的找煤前景进行了预测。在没有任何其它勘探手段验证的情况下，作者尝试使用信息复合技术检验找煤预测的可靠性。基于经过几何精校正的遥感影像图，和经过栅格化处理，并与遥感影像图配准的重力异常数据，准确的将两者按统一坐标系叠合在一起。复合结果表明，它不只是指示了重力异常梯度带与遥感解译的线性构造，在空间位置上的相关性，而且还可结合重力异常等值线的分布，将该线性构造准确定位，如图版 2 中的废黄河断裂。与此同时，通过与已知矿区重力异常分布特征的对比，为区内的找煤预测提供了有力的佐证。

图版 2 所展示的虽然只是遥感与非遥感复合处理在找煤预测中应用的一种初步尝试的结果，但也充分说明了该方法在找煤预测中的实用性。

1. 复合处理结果除较清楚地显示出诸如废黄河断裂之类的线性构造的影像特征和实际位置外，还提供了重力异常梯度带的空间分布特征，从面在确保遥感信息用于构造解译的优势的同时，进一步明确了线性构造的几何定位问题。

2. 根据重力异常的空间分布规律，特别是与已知矿区的对比，为基于构造分析推测的隐伏含煤区提供了有力的佐证，从而实现了两类数据间的优势互补，提高了构造分析与找煤预测的可靠性。运用此项技术在徐淮地区提出的几个含煤预测区，均得到专家们的认可，有的已开始投入勘探工程。

（二） 用三元超曲面样条函数进行时空转换

作者选择某煤田勘探区作为实例，收集了钻孔资料和地震勘探资料作为二种不同类型的数据，用三元超曲面样条来实现它们之间的复合、转换，以达到将地震勘探资料的时间数据转换为空间（深度）数据的目的，进而达到准确地推断煤层形态和地质构造的目的。

1. 地震勘探资料的获得和预处理

1） 时间剖面数据的获得

地震勘探原始资料经专用计算机和专用软件处理后，可综合成叠加剖面和偏移剖面两

种成果数据。我们的工作即从这里开始。首先将载有这两种数据的磁带用附有磁带机的计算机读出，再经异步通信将数据传输到微机上。然后用第四章的程序SEGYT16将原来为SEG－Y格式数据转换为DOS格式数据。至此，便得到地震时间剖面数据。在叠加剖面上，可读出煤层反射波旅行时间（t_0）。在我们的实例中，即以这套数据作为基本数据，其测线上采样点距为 5m。

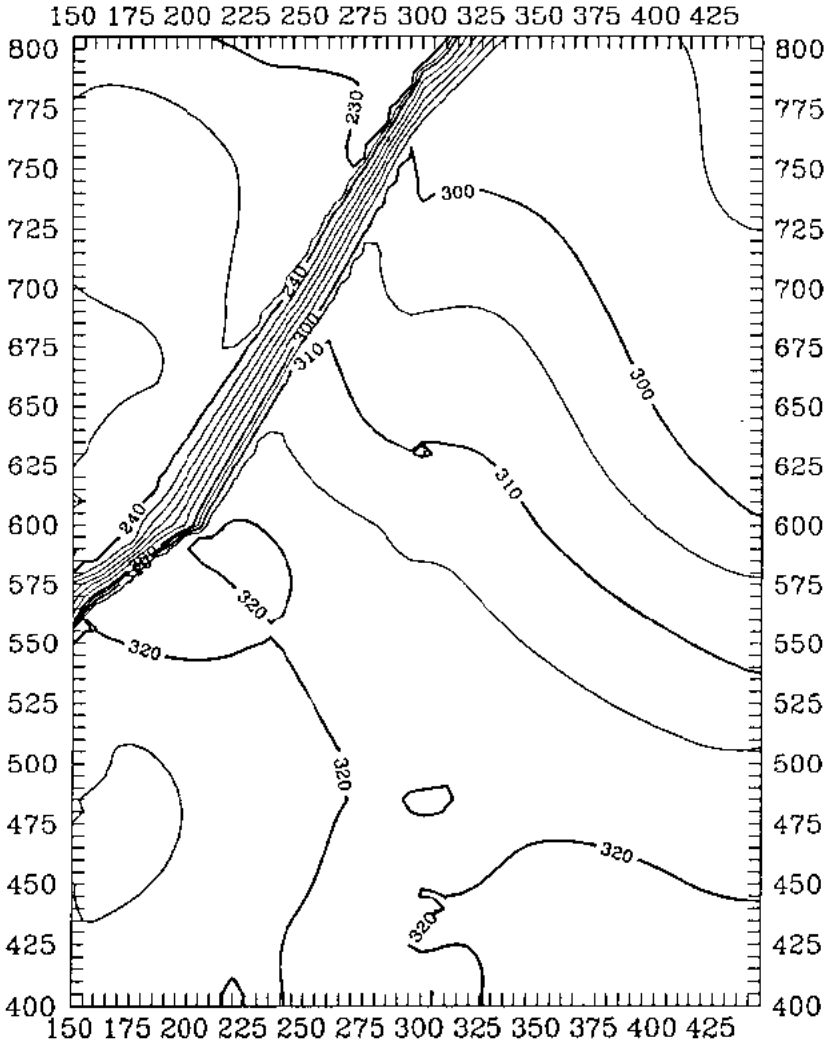


图 6－1　煤层反射波旅行时间曲面等值线图

2）　网格数据的形成

上述数据代表了在这些点上反射波从地面到煤层底板垂直往返的时间。为得到整个区域的网格点数据（x，y，t），我们用二元曲面样条函数进行插值计算。其作法是：首先在有明显断点（在测线上表现为跳跃点）的地方，将这些断点连成线作为补充数据，以弥补由测线组成的网格中可数值的不足。再挑一网格用曲面样条函数插值形成该网格的旅行时间曲面数据。然后以此网格为中心，对每个网格进行计算，扩展至整个区域。为保证整个区

域的连续协调，先形成的网格边界作为资料参加到后形成的网格计算中；这样，由于曲面样条函数的良好特性，可保证后形成的网格与先形成的网格保持连续特性，从而得到整个区域连续协调的时间曲面。图 6—1 即为由 4 个网格合成的旅行时间曲面等值线图。

2. 钻孔数据与时间数据的关系

钻孔资料经孔斜校正后，能较准确地确定钻孔所在位置的煤层底板深度（标高），即可得出代表煤层底板坐标的 x、y、z 值。n 个钻孔即可得到一套煤层底板坐标数据（x_i，y_i，z_i，$i=1$，2，…，n）。钻孔数据较精确，但数据量小，不足以构成整个区域的空间曲面。另一方面：由地震勘探数据产生的时间曲面数据分布均匀，但却不是深度数据，不能直接反映煤层底板变化状况。所以，需将二者结合起来完成时深转换，即以钻孔数据为基准，将时间曲面转换成空间曲面。为此我们先考察在钻孔位置 z 与 t 的关系，表 6—1 列出了这些点的 x，y，t，z 值，图 6—2 为这些点上旅行时间 t 与煤层底板标高 z 的关系图。从表 6—1 和图 6—2 中都可看出，从趋势角度看，随着旅行时间 t 的增加，深度 z 也增加，但它们之间不是一个单调增加或下降的函数关系，存在着局部反转现象，即时间较长的点反而深度较小；同时，图中各线段斜率差别很大。造成这种现象的原因很多。但我们认为，主要是与钻孔所在位置有关，即由于各点的地层结构不同，反射波的速度因地因深度而异。由此可以认为，若要从地震勘探数据反演出煤层深度数据，三个自变量（x，y，t）都须考虑，即须形成一个三元反演函数 z（x，y，t）。

表 6—1　钻孔位置煤层底板的 x、y、t、z 值

x	y	t	z
71. 35	578. 01	242. 07	—180. 52
365. 71	819. 83	292. 39	—273. 67
414. 52	604. 89	303. 39	—299. 70
509. 16	956. 79	272. 73	—242. 28
622. 67	1199. 44	247. 90	—214. 94
775. 59	410. 05	312. 79	—316. 48
899. 12	788. 60	276. 86	—253. 13
1104. 92	1258. 10	256. 51	—231. 37
1113. 50	1704. 98	216. 88	—165. 90
1203. 64	1698. 54	261. 46	—238. 92
1256. 28	1592. 45	276. 36	—254. 09
1350. 79	1510. 81	339. 87	—335. 89
1505. 02	1292. 91	447. 86	—553. 61
1708. 55	1616. 78	437. 75	—530. 95
—441. 04	191. 25	261. 00	—218. 14
—31. 07	—311. 26	486. 00	—579. 61
—28. 14	—130. 49	375. 00	—437. 76
—14. 60	21. 69	352. 70	—394. 19
—7. 68	281. 95	328. 00	—335. 61
175. 88	928. 98	221. 00	—166. 56
1125. 67	448. 61	516. 00	686. 95
1275. 98	767. 41	502. 30	—638. 86
1110. 40	393. 75	528. 00	—700. 00

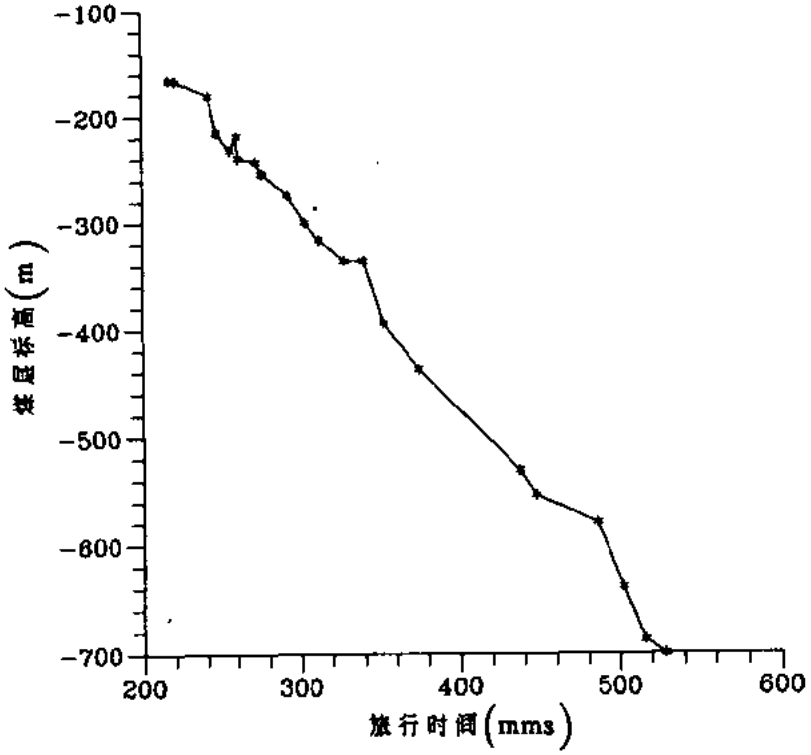


图 6－2　钻孔位置的旅行时间与煤层底板标高关系图

3. 用三元超曲面样条函数进行时深转换

由于我们以钻孔资料作为基准，故上述三元函数 z（x，y，t）在钻孔点上的数值应与钻孔数据完全一致；同时，该函数应从全局上控制整个区域，应有明确的函数表达式。在本章第三节中，我们曾将二元曲面样条函数推广成三元超曲面样条函数。可以看到，将三元超曲面样条函数用于此，可满足上述要求。

表 6－2 为采用表 6－1 中的数据计算产生的三元超曲面样条函数中 $n+4$ 个系数 $F_i(i=1,2,\cdots,n)$，a_0，a_1，a_2，a_3 的值。

表 6－2　三元超曲面样条函数系数表

. 8571933E－04	. 2118989E－04	－. 7997574E－04	. 3750042E－04	－. 3425417E－04
. 6029225E－04	. 4462766E－04	－. 3641258E－04	. 1034116E－03	－. 1370011E－03
－. 1142482E－03	. 2708332E－03	－. 1107931E－03	. 5171648E－05	. 8285398E－05
. 9625519E－04	－. 1642229E－03	. 5350053E－04	－. 2942994E－04	－. 5098218E－04
－. 3059281E－03	. 6657299E－04	. 2087723E－03	－. 8740843E－01	－. 2107454E－01
－. 2143461E－01	－1. 578090			

4. 结果

应用表 6－2 中的系数，即可将全部时间曲面值（x，y，t）转换成（x，y，z），完成时间－空间数据的转换，达到反演煤层底板深度的目的。图 6－3 为与图 6－1 相同区域的由

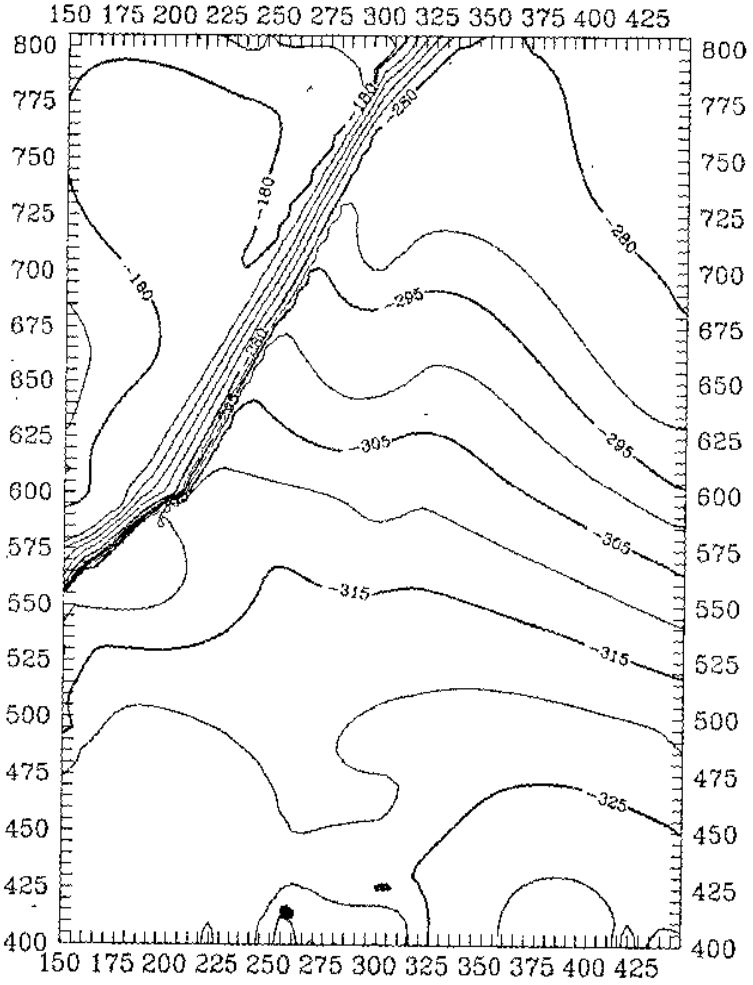


图 6—3　煤层底板标高等值线图

上述反演得到的煤层底板等值线图。图 6—3 与图 6—1 的基本形态相似，但二者的局部等值线形态不同，反映了反演函数中 x，y 自变量的作用。与实际资料对比，图 6—3 较好地反映了煤层底板形态。

5. 讨论

将地震勘探数据由专用机经磁带转到微型机上，再经格式转换程序将这些数据转换成 DOS 格式数据，就使得这些数据可以在微机上进行处理，增加了应用这些数据的灵活性。对于其它类型数据，也可循这个思路进行格式转换。

不同类的数据之间的复合、转换，须视具体情况而定。本例将煤层反射波旅行时间与钻孔数据结合起来，使用三元超曲面样条函数反演出煤层底板深度（标高），较充分地应用了这二类不同信息。为此类问题的解决开辟了一个新思路。

ABSTRACT

With the rapid development of computer techniques both remote sensing and GIS(Geographical Information System) techniques have made considerable progress and trend to come together. Remote sensing serves as information resources for GIS and GIS serves as an effective tool for application of remote sensing data. Nevertheless it is well—known that there is a long way to go before the practical combination of remote sensing and GIS techniques come true. Just under the circumstances the emphasis of this book is paid on the computer data processing for integration of remote sensing and non—remote sensing geological information, especially on the mutual conversation of different kinds of remote sensing and non—remote sensing geological information. The purpose of this book is to build a "software bridge"between the data with different formats, in order that all kinds ofgeological information could be linked up throuth suitable conversations and then have a compatible format to be used in data integration.

Generally geological information could be devided into two types: remote sensing and non—remote sensing. Both analogical and digital imagery, including aerophotos, LANDSAT—MSS, LANDSAT—TM SPOT and SAR images etc., belong to the former, which normally have a image form; and others, obtained by means of relative conventional geological methods, such as geophysical, geochemical methods and so on, have the features of non—remote sensing information, which normally have a text, table or graphic form. Because of the wide diferences in data form and data format, even the analysis results using the same primary data but different software could not be linked up directly. In order to solve such problems basic features of remote sensing geological information are briefly analysed at first, which cover imagine styles, geometrical features, imagery features and data formats.

Some special problems about extract of remote sensing geological information are discussed detailed in chapter 3, especially the correction of systematic geometrical distortion in panoramic scanning images, the correction of systematical radiometrical distortion of remote sensing images, preparation of fundamental image maps and the extract of primary data from remote sensing images. Two main programs that are related to geometrical correction and extract of primary data respectively are attached.

With respect of the main abstacles for data integration — data incompability, interior structures and data formats of common used sofware, such as dBASE, Grapher, pc ARC/INFO, TIFF and seismographic records SEG—Y are detailed analysed. A series of conversation programs, mainly for the non—remote sensing data have been written in FORTRAN based on the thoroughly understanding of the interior structures of the softwares mentioned above. All converted data have a common data format, which can be called easily and used conveniently in different software environments.

For Chinese users of data integration they have to face an unavoidable problem, annotaion in Chinese, which has obviously affect the application of many famous graphics and image processing softwares in China to a considerable extent. A special chapter has been included in this book, which deals with the Chinese annotation under non—Chinese operating system, including normal DOS system and VAX/VMS system. All problems such as features of Chinese character library, generation of Chinese character under different software environment and the call of Chinese characters in annotation are covered in this chapter.

Chapter6 deals mainly with some concret integration techniques, especially the interpolation method using 3—vriablke hypersurface splines and some examples that applied data integration techniques in coalfield prediction and coversation of time data to spacial data.

参 考 文 献

陈述彭、赵英时，1990，遥感地学分析，296 页，测绘出版社

(美) J. 邓格蒙，1992，地理信息技术向何处去？《遥感信息》，1992 年第 2 期，36—38 页

何钟奇、王学佑，1992，国外遥感地质技术若干新进展，《遥感信息》，1992 年第 2 期，32—35 页

兰运超、利光秘、袁　征，1991，地理信息系统原理，220 页，广东地图出版社

李昌国，1991，90 年代世界遥感技术的发展，《国土资源遥感》，1991 年第 2 期，53—59 页

李德仁、龚建雅　等，1993，地理信息系统导论，131 页，测绘出版社

吕凤翥、蔡建新，1985，微机绘图软件 AUTO CAD，432 页，清华出版社

彭望禄，1991，遥感数据的计算机处理与地理信息系统，390 页，北京师范大学出版社

(美) W. K. 普拉特，1984，数字图像处理学，478 页，科学出版社

谭海樵，1991，全景扫描图像几何畸变特征及其校正，《国土资源遥感》，第 3 期，37—42 页

谭海樵、王　颖、张大顺，1991，1：20 万遥感地质调查中基础影像图的制备，《江苏地质》(增刊)，117—121 页

谭海樵，1992，TM 图像的系统几何畸变与几何精校正，《地球科学进展》学术论文集（中国地质大学编），354—355 页

谭海樵，1991，机载多光谱图像在西德多瑙尔兴恩地区土地利用调查中的应用，《石油实验地质》，第 1 期，99—106 页

谭海樵、余志伟，1993，遥感非遥感信息复合技术及其在徐州地区找煤预测中的应用，《中国矿业大学学报》(增刊)，8—12 页

(日) 田村秀行等，1988，计算机图像处理技术，303 页，北京师范大学出版社

徐　蔓、邵建林、赵　坚，1992，C 语言实用图像处理，493 页，海洋出版社

余志伟，1987，一种新的地质曲面插值计算法——曲面样条函数方法，《中国矿业学院学报》，第 4 期，69—76 页

张寿萱、徐建毅、张建生，1984，中文信息的计算机处理，294 页，宇航出版社

Drury, S. A., 1987, Image Interpretation in Geology, 243 pp., London.

Harder, R. L. & Desmarais, R. N., 1972, Interpolation Using Surface Splines, Journal of Aircraft, Vol. 9, p. 189—191.

Sabins, F. F., 1978, Remote Sensing — Principal and Interpretation, 381 pp., San Francisco.

Simonett, D. & Davis, R. E., 1983, Image Analysis — Active microwave. Manual of Remote Sensing (Colwell, R. N. ed.), Vol. 1, pp. 1125—1181, Falls Church.

ESRI, 1990, Understanding GIS, the pc ARC/INFO Method

Ulaby, F. T. et al, 1981, Microwave remote sensing fundamentals and radiometry, in Microwave remote sensing, Vol. 1, pp. 1—456, London.

Kronberg, P., 1984, Fernerkundung der Erde, 394 S., Stuttgard

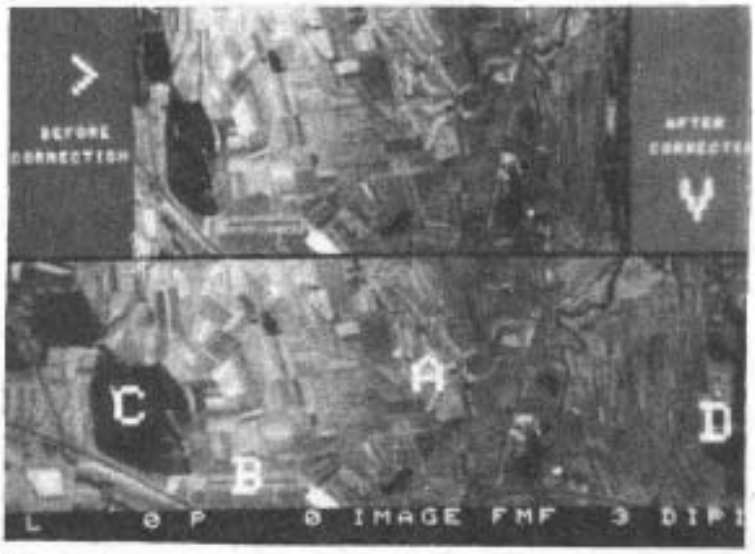
Haiqiao Tan, 1988, Vergleich der Aussagekraft von SAR— und anderen Fernerkundungsdaten fuer Land nutzung und Geologie im humiden und ariden Klimabereich, 106 S., Detrich Reimer Berlin.

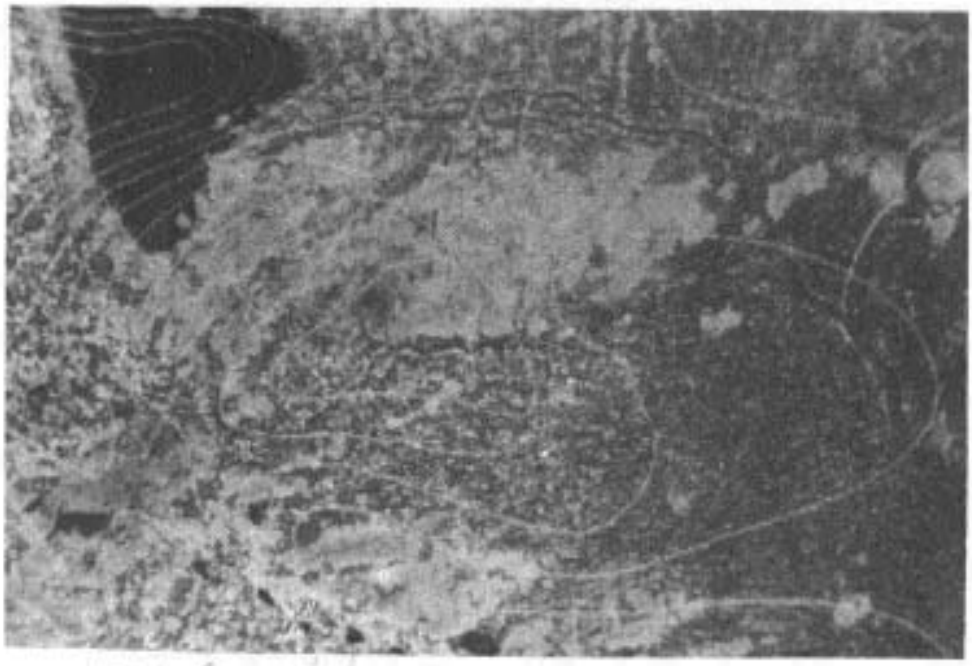
Tan Haiqiao & Xu Peifen, 1993, Features and Correction of the systematic Distortion in Panoramic Scanning Imagery, 《Scientia Geologica Sinica》, Vol. 2, No. 4, pp. 391—196

Hord, R. M. & Brooner, w., 1976, Land Use Map Acouracy Criteria, Photogramm. Eng. Vol. 28, No. 5, pp. 671—677

List, F. K., Meissner, B. & Poehlmann, G., 1986, Landsat—MSS Remote Sening and Satellite Cartography — Integrated Approch to the Preparation of a New Geological Map of Eygpyt at a Scale of 1:500 000, Proceedings of IGARSS 86 Symposium, pp. 1503—1510, Zurich.

Townshend, J. R. G., 1983, An Introduction to the Study of Terrain, In: Townshend, J. R. G. (ed.): Terrain Analysis and Remote Sensing, pp. 1—15, London





35-9-081
P203
440154
2.0406