

# 第三讲 基于ENVI Classic 的二次开发

石雪

测绘地理信息学院

# 主要内容

---

- 1. 数据读写
  - 标准、ASCII、二进制、图像格式
- 2. ENVI功能扩展
  - 植被覆盖度函数
  - 波段运算 (Band Math) 功能扩展
  - 图像批处理

# 1 数据读写

---

- 标准输入输出
- 文件读写
  - ASCII文件
  - 二进制文件
  - 图像格式文件

# 1 数据读写

- 标准输入输出（格式化）
  - 标准输入：键盘输入
  - 标准输出：屏幕输出显示
  - 标准输入输出的函数：

函数名称	作用
Print	标准输出写出格式化数据
Read	标准输入读入格式化数据
Reads	从字符串中读取格式化数据
String()	格式化输出的字符串数据

# 1 数据读写

- 格式化输出：是按照设计的格式进行输出

- `format= ‘ ([n]FC[+][-][width]) ’`

格式化字符	含义
n	代表格式控制符的有效次数，默认为 1 次。
FC	格式控制符，详细说明见表 5-3。
+	在输出的整数前面加“+”符号，默认正数前不输出，负数前有“-”。
-	控制字符串和数字的输出为左对齐，默认输出是右对齐。
width	输出字符的宽度，对特定的宽度部分可补 0 或空格。

# 1 数据读写

## ●FC代码含义

格式代码	格式含义
A	字符
:	若没有有效的变量，则终止输入输出
\$	使下一个输出数据输出到该输出数据的所在行的后面（仅用于输出）
I	十进制整数
O	八进制整数
Z	十六进制整数
F	输出单精度浮点数
D	输出双精度浮点数
E	以指数形式（科学格式）输出浮点数
[n]G[w,d] or [n]G[w,dEe]	根据数据大小自动选择科学格式或者 F、D 格式输出
T	绝对位置输出变量的值
TL	从当前位置向左移动输出变量的值
TR	从当前位置向右移动输出变量的值
/	换行输出

# 1 数据读写---文件

## ●文件操作函数

函数名字	作用
OpenR	以只读方式打开已存在文件；
OpenW	创建一个可以读写的新文件；
OpenU	以更新模式打开已存在文件；
File_Search()	对文件名进行特定的查找；
Dialog_Pickfile()	对话框方式选择文件；
Fstat()	返回一个已打开文件的信息；
EOF()	检测是否到文件末；
CLOSE	关闭一个文件；
Free_Lun	释放一个逻辑设备号并关闭文件；

# 1 数据读写---文件

- 文件打开

- 先将一个逻辑设备号与文件进行关联，然后对逻辑设备号进行如读写或更新等操作。
- 逻辑设备号：范围是 $-2 \sim 128$ 
  - $1 \sim 99$  是可以用户任意指定
  - $100 \sim 128$  由IDL内部进行管理
  - 其他值为专用设备号



# 1 数据读写---文件

---

- 文本文件读写函数
  - Openr 文件打开
  - Readf 从文件中读入数据
  - Printf 写入数据到文件中
  - Skip\_lun (/lines) 文件中跳行
  - Free\_lun 释放逻辑设备号

# 1 数据读写---文件

- (1) ascii文件读写 (ascii.txt)

- 1-3行：元数据信息

- 列数、行数、数据类型

- 波段数、数据

- 4-9行：传感器接收的数据

1	2	6	4
2	Column 1: Band Number		
3	Column 2: X:1 Y:1~~2		
4	1.000000	0.107994	
5	2.000000	0.027035	
6	3.000000	0.085975	
7	4.000000	0.021942	
8	5.000000	0.065975	
9	6.000000	0.041942	

# 1 数据读写---文件

- (1) ascii文件读写 (ascii.txt)
  - 元数据信息：读入到字符串变量（一次读入）获取行列、数据类型
  - 数据：读入数据到浮点型数组
  - 写出数据txt格式：printf命令
  - 可视化：plot函数
  - 代码写为PRO或FUN文件

1	2	6	4
2	Column 1: Band Number		
3	Column 2: X:1 Y:1~~2		
4	1.000000	0.107994	
5	2.000000	0.027035	
6	3.000000	0.085975	
7	4.000000	0.021942	
8	5.000000	0.065975	
9	6.000000	0.041942	

# 1 数据读写---文件

- (2) 文本文件读写 ( DSM. asc )

- 1-6行: 数据描述信息

- 行列数 (1-2行)

- 7行后: 文件数据

```
1 ncols 2001
2 nrows 1601
3 xllcenter -30000.000000000000
4 yllcenter -800.000000000000
5 cellsize 0.500000000000
6 nodata value -9999.000
7 -18.233 -18.139 -17.865 -17.356 -16.999 -12.967 -3.836 5.450 15.855 19.559 19.374
8 -21.141 -21.184 -20.816 -20.215 -21.857 -18.662 -6.648 4.315 15.922 20.987 20.452
9 -22.116 -20.737 -19.266 -17.620 -19.459 -17.251 -6.541 2.321 9.282 11.719 11.786 1
10 -22.897 -21.099 -20.671 -20.362 -19.977 -15.795 -8.909 -7.633 -0.624 4.309 3.280 2
11 -23.625 -21.644 -22.389 -19.839 -18.681 -17.504 -14.225 -13.329 -10.688 -9.038 -8.
12 -23.236 -21.026 -19.537 -16.582 -16.948 -18.567 -18.258 -15.588 -17.280 -19.255 -1
13 -22.912 -20.587 -13.161 -14.952 -16.178 -16.592 -18.005 -17.852 -17.578 -16.808 -1
14 -20.690 -21.756 -20.668 -11.316 -10.729 -15.216 -15.199 -14.616 -14.339 -13.775 -1
15 -17.499 -21.007 -21.284 -17.630 -7.982 -2.251 -8.556 -10.028 -6.065 -3.782 -4.430
16 -16.869 -23.885 -23.878 -25.687 -17.559 -3.956 1.622 0.840 2.578 3.836 2.230 1.255
17 -9.997 -17.856 -17.419 -16.468 -19.430 -11.590 14.779 15.531 9.144 5.365 4.923 3.0
18 9.427 17.305 16.820 16.556 15.573 17.573 22.016 21.901 16.041 9.439 4.854 4.271 3.
19 15.914 24.154 23.845 23.600 23.759 23.560 23.042 22.544 19.565 14.923 10.164 6.837
20 16.949 22.484 22.135 21.897 21.866 21.742 22.143 23.327 22.815 20.234 15.616 10.75
21 18.145 22.876 22.617 22.491 22.859 23.541 23.055 19.380 17.936 16.388 13.440 12.69
22 19.076 22.891 22.623 22.849 22.098 22.597 22.063 6.166 2.593 2.945 4.437 13.148 5.
23 19.721 23.533 23.177 24.923 15.296 5.337 6.009 3.180 2.023 1.913 4.268 12.525 7.54
```

# 1 数据读写---文件

## ● (2) 文本文件读写 ( DSM. asc )

- 元数据信息：多次读入到字符串变量，获取行列数
- 文件数据：读入数据到浮点型数组
- 数据写出tiff格式：write\_tiff命令
- 图像可视化：image函数
- 代码写成PRO文件

```
1 ncols 2001
2 nrows 1601
3 xllcenter -30000.000000000000
4 yllcenter -800.000000000000
5 cellsize 0.500000000000
6 nodata_value -9999.000
7 -18.233 -18.139 -17.865 -17.356 -16.999 -12.967 -3.836 5.450 15.855 19.559 19.374
8 -21.141 -21.184 -20.816 -20.215 -21.857 -18.662 -6.648 4.315 15.922 20.987 20.452
9 -22.116 -20.737 -19.266 -17.620 -19.459 -17.251 -6.541 2.321 9.282 11.719 11.786 1
10 -22.897 -21.099 -20.671 -20.362 -19.977 -15.795 -8.909 -7.633 -0.624 4.309 3.280 2
11 -23.625 -21.644 -22.389 -19.839 -18.681 -17.504 -14.225 -13.329 -10.688 -9.038 -8.
12 -23.236 -21.026 -19.537 -16.582 -16.948 -18.567 -18.258 -15.588 -17.280 -19.255 -1
13 -22.912 -20.587 -13.161 -14.952 -16.178 -16.592 -18.005 -17.852 -17.578 -16.808 -1
14 -20.690 -21.756 -20.668 -11.316 -10.729 -15.216 -15.199 -14.616 -14.339 -13.775 -1
15 -17.499 -21.007 -21.284 -17.630 -7.982 -2.251 -8.556 -10.028 -6.065 -3.782 -4.430
16 -16.869 -23.885 -23.878 -25.687 -17.559 -3.956 1.622 0.840 2.578 3.836 2.230 1.255
17 -9.997 -17.856 -17.419 -16.468 -19.430 -11.590 14.779 15.531 9.144 5.365 4.923 3.0
18 9.427 17.305 16.820 16.556 15.573 17.573 22.016 21.901 16.041 9.439 4.854 4.271 3.
19 15.914 24.154 23.845 23.600 23.759 23.560 23.042 22.544 19.565 14.923 10.164 6.837
20 16.949 22.484 22.135 21.897 21.866 21.742 22.143 23.327 22.815 20.234 15.616 10.75
21 18.145 22.876 22.617 22.491 22.859 23.541 23.055 19.380 17.936 16.388 13.440 12.69
22 19.076 22.891 22.623 22.849 22.098 22.597 22.063 6.166 2.593 2.945 4.437 13.148 5.
23 19.721 23.533 23.177 24.923 15.296 5.337 6.009 3.180 2.023 1.913 4.268 12.525 7.54
```

# 1 数据读写---文件

---

- 二进制文件读写函数

- 常规文件

- ReadU读取

- WriteU写出

- 带头文件的文件

- Point\_Lun定位



# 1 数据读写---文件

## ●二进制文件读写(FY2C.AWS)

### ●序号1：文件名

### ●序号6-8：行列数，元数据的行数

1) 第一级文件头记录格式

序号	字节序号	字节数	类型	描述内容	具体数值或字符
1	1-12	12	A*12	文件名	TMG#####AWX
2	13-14	2	I*2	整型数的字节顺序	0
3	15-16	2	I*2	第一级文件头长度	40
4	17-18	2	I*2	第二级文件头长度	80
5	19-20	2	I*2	填充段数据长度	1081
6	21-22	2	I*2	记录长度	1201
7	23-24	2	I*2	文件头专用记录数	2
8	25-26	2	I*2	产品数据专用记录数	1201
9	27-28	2	I*2	产品类别	3
10	29-30	2	I*2	压缩方式	0
11	31-38	8	A*8	格式说明字符串	SAT2004
12	39-40	2	I*2	产品数据质量标记	2

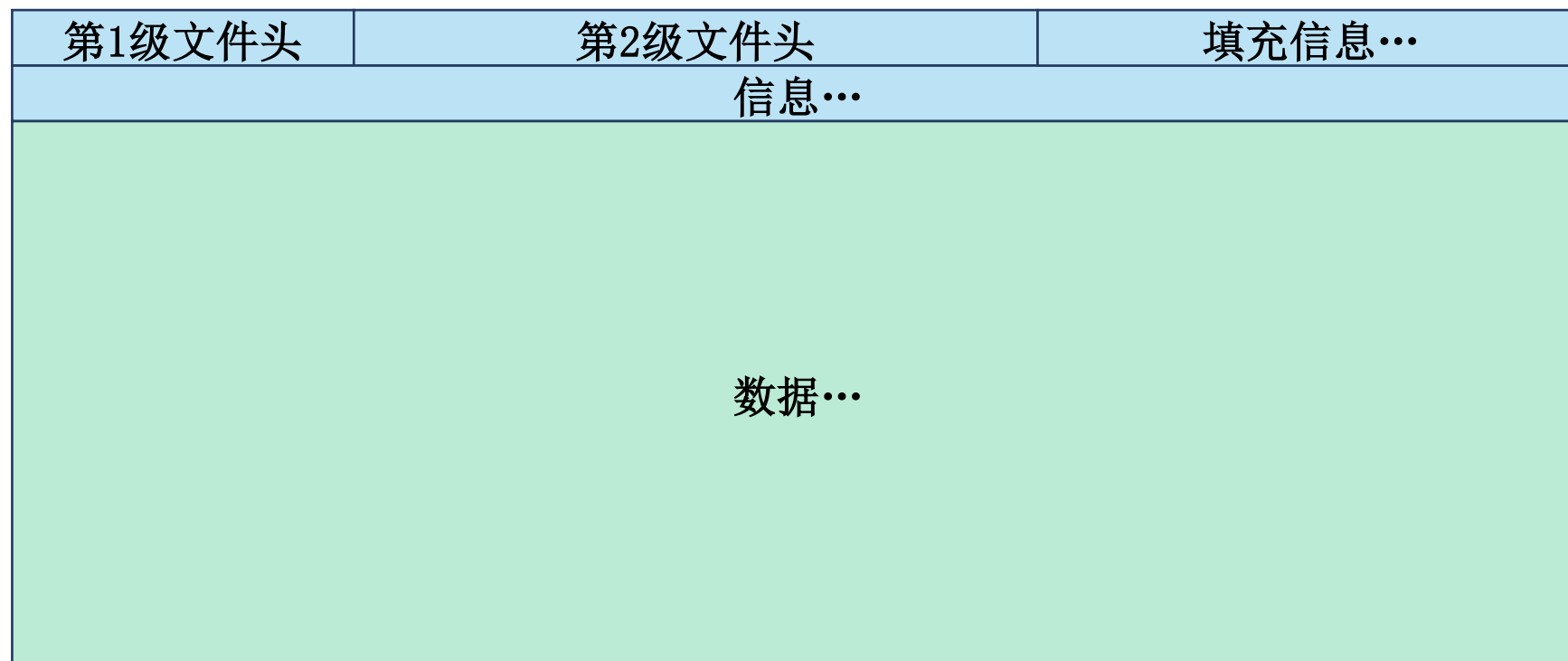
2) 第二级文件头记录格式

序号	字节序号	字节数	类型	描述内容	具体数值或字符
1	41-48	8	A*12	卫星名	FY2C
2	49-50	2	I*2	格点场要素	19
3	51-52	2	I*2	格点数据字节	1
4	53-54	2	I*2	格点数据基准值	100
5	55-56	2	I*2	格点数据比例因子	1
6	57-58	2	I*2	时间范围代码	1(或2、3、4)
7	59-60	2	I*2	开始年	####
8	61-62	2	I*2	开始月	##
9	63-64	2	I*2	开始日	##
10	65-66	2	I*2	开始时	##
11	67-68	2	I*2	开始分	##
12	69-70	2	I*2	结束年	####
13	71-72	2	I*2	结束月	##
14	73-74	2	I*2	结束日	##
15	75-76	2	I*2	结束时	##
16	77-78	2	I*2	结束分	##
17	79-80	2	I*2	网格左上角纬度	6000
18	81-82	2	I*2	网格左上角经度	4500
19	83-84	2	I*2	网格右上角纬度	-6000
20	85-86	2	I*2	网格右上角经度	16500

# 1 数据读写---文件

- 二进制文件读写(FY2C. AWS)

- 知道数据存储格式





# 1 数据读写---文件

## ●图像文件信息查询

- IDL 中提供了常用图像文件格式的查询程序。这些程序允许在没有真正读取其数据的情况下，就可以查询图像文件。
- 查询程序是返回0或1的函数，通过返回值确定是否成功地（返回值为1）读取了图像文件里的元数据。

- Query\_BMP
- Query\_DICOM
- Query\_GIF
- Query\_JPEG
- Query\_PICT
- Query\_PNG
- Query\_PPM
- Query\_SRF
- Query\_TIFF

# 1 数据读写---文件

## ●4. JPEG文件读写

### ●读JPEG文件命令格式

```
READ_JPEG [, Filename], Image [, Colortable] [, BUFFER=variable] [, COLORS=value{8 to  
256}] [, DITHER={0 | 1 | 2}] [, /GRayscale] [, /ORDER] [, TRUE={1 | 2 | 3}]  
[, UNIT=lun][, /TWO_PASS_QUANTIZE]
```

### ●写JPEG文件命令格式

```
WRITE_JPEG [, Filename], Image [, /ORDER] [, /PROGRESSIVE] [, QUALITY=value{0 to  
100}] [, TRUE={1 | 2 | 3}] [, UNIT=lun]
```

# 1 数据读写---文件

## ●BMP文件读写

### ●读BMP文件

Syntax

```
Result = READ_BMP( Filename, [, R, G, B] [, Ihdr] [, /RGB] )
```

### ●写BMP文件

Syntax

```
WRITE_BMP, Filename, Image[, R, G, B] [, /FOUR_BIT] [, IHDR=structure]  
[, HEADER_DEFINE=h{define h before call}] [, /RGB]
```

# 1 数据读写---文件

## ●通用格式文件

## ●支持的文件格式

文件格式	读函数	写函数
BMP	Read_BMP	Write_BMP
CDF	参考 CDF 库	参考 CDF 库
DICOM	IDLfndICOM 对象	IDLfndICOM 对象
DXF	IDLfndXF 对象	IDLfndXF 对象
GIF	Read_GIF	Write_GIF
HDF	参考 HDF 库	参考 HDF 库
HDF-EOS	参考 HDF 库	参考 HDF 库
Interfile	Read_Interfile	无
JPEG	Read_JPEG	Write_JPEG
netCDF	参考 netCDF 库	参考 netCDF 库
PICT	Read_PICT	Write_PICT
PBM/PPM	Read_PPM	Write_PPM
PNG	Read_PNG	Write_PNG
PostScript	无	PS 或打印设备
Sun Rasterfiles	Read_SRF	Write_SRF
SYLK	Read_SYLK	Write_SYLK
TIFF/GeoTIFF	Read_TIFF	Write_TIFF
WAVE	Read_WAVE	Write_WAVE
X11-bitmap	Read_X11_Bitmap	无
XWD	Read_XWD	无
GRIB1/GRIB2	参考 GRIB 库	参考 GRIB 库

# 1 数据读写---文件

---

## ●5. HDF文件

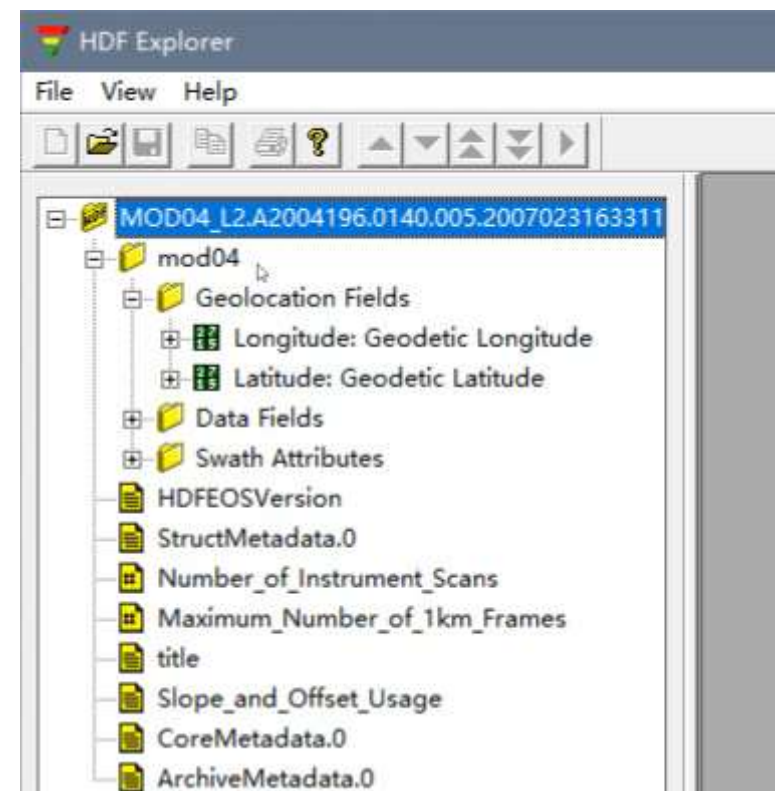
### ●层次数据格式

- HDF(Hierarchical Data Format)，可以存储不同类型的图像和数码数据的文件格式，并且可以在不同类型的机器上传输，同时还有统一处理这种文件格式的函数库

# 1 数据读写---文件

- HDF文件包括：

- 1. 栅格影像；
- 2. 调色板：8-bit调色板；
- 3. 科学数据集：多维整型或浮点数组；
- 4. 注释--描述文件和部分文件（元数据）的文字；
- 5. 定点数据（Vdata VGroup）：以表来存储多值数据；



TEST\_READHDF代码

# 1 数据读写---文件

## ●6. XML文件

- 用于标记电子文件使其具有结构性的标记语言，可以用来标记数据、定义数据类型，是一种允许用户对自己的标记语言进行定义的源语言。

- 用处：

- 数据交换
- Web服务
- 内容管理
- 系统配置

```
<?xml version="1.0" ?>
<ACFTB>
  <field name="CETAG" spec="CETAG" desc="Unique Extension Identifier" size="6"
dtype="string" encoding="ASCII">
    <values value="ACFTB" chartype="BCS-A"></values>
  </field>
  <field name="CEL" spec="CEL" desc="Length of Entire Tagged Record" size="5"
dtype="int" encoding="ASCII">
    <values value="00207" chartype="BCS-N"></values>
  </field>
  <field name="AC_MSN_ID" spec="AC_MSN_ID" desc="Aircraft Mission
Identification" size="20" dtype="string" encoding="ASCII">
    <values chartype="BCS-A"></values>
  </field>
  <field name="AC_TAIL_NO" spec="AC_TAIL_NO" desc="Aircraft Tail Number"
size="10" dtype="string" encoding="ASCII">
    <values chartype="BCS-A"></values>
```



## 考查题一：ENVI文件读写

- 读取ENVI数据 (\*\*.hdr和\*\*.img) :
  - (1) 读取头文件 (\*\*.hdr) 获取相关元数据信息;
  - (2) 利用 (1) 中信息, 读取二进制文件 (\*\*.img) 获取图像数据;
  - (3) 在IDL中显示读取图像
  - (4) 将图像数据以tiff写出
  - (5) 在ENVI中显示tiff图像



```
bhnmref - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
ENVI
description = {
  Bighorn Basin. Landsat TM calibrated to reflectance}
samples = 512
lines = 512
bands = 6
header offset = 0
file type = ENVI Standard
data type = 1
interleave = bsq
sensor type = Landsat TM
byte order = 0
map info = {UTM, 1.000, 1.000, 274785.000, 4906905.000, 3.000000000000e
+001, 3.000000000000e+001, 13, North, units=Meters}
wavelength units = Micrometers
z plot range = {0.00, 100.00}
z plot titles = {Wavelength, Reflectance}
dem file = bhdemsub.img
default stretch = 2.0% linear
band names = {
  TM Band 1, TM Band 2, TM Band 3, TM Band 4, TM Band 5, TM Band 7}
wavelength = {
  0.485000, 0.560000, 0.660000, 0.830000, 1.650000, 2.215000}
```



## 2 ENVI功能扩展

---

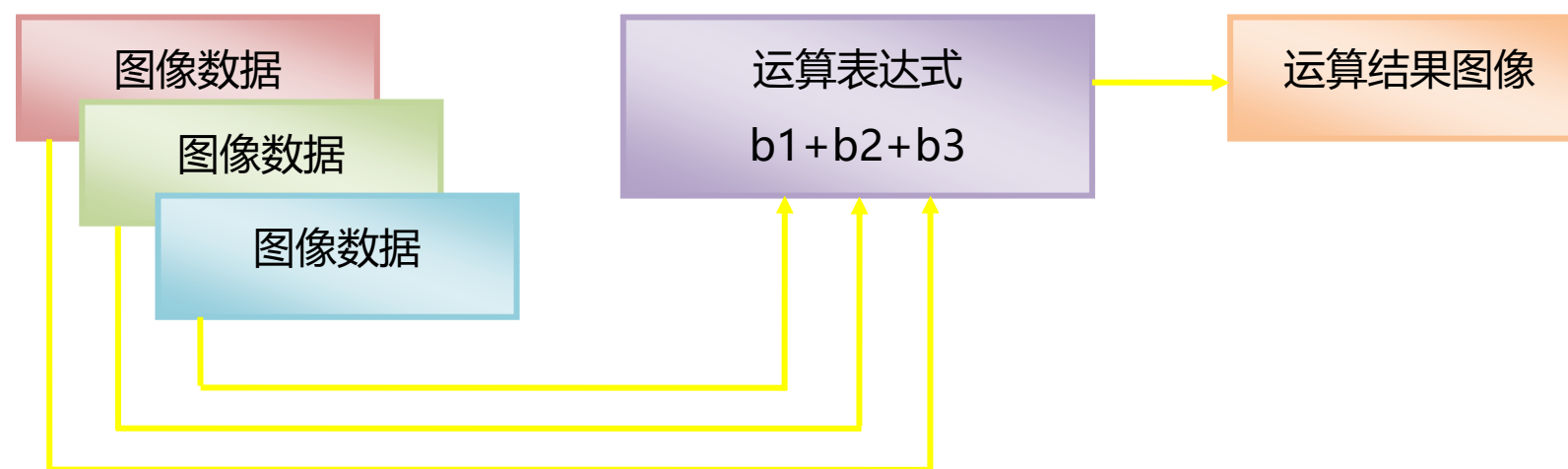
- ENVI是使用IDL语言编写的优秀的遥感影像处理平台，用户可以方便的通过IDL语言以及ENVI提供的二次开发工具对ENVI的功能进行增强，添加新的功能函数或调用ENVI已有的功能。

## 2 ENVI功能扩展

- 波段和波谱运算函数
  - ENVI提供两个功能，用户能够对影像波段或波谱进行数学运算，且无需了解ENVI内部的处理机制；
- 用户函数（扩展补丁）
  - ENVI菜单执行，通过ENVI获得输入数据，结果直接输入到ENVI中；
- 二次开发模式（批处理）
  - 将常用处理编写成为IDL程序，常用于批量处理；
  - 能够通过调用ENVI的绝大多数ENVI的功能；
  - 可以编写自己的程序界面，后台使用ENVI功能；

## 2.1 波段运算

- 波段运算对话框用来定义波段或文件作为输入，调用用户编写的表达式或函数，并将结果输出到文件或内存。



## 2.1 波段运算

- 波段运算格式

- 直接编写表达式

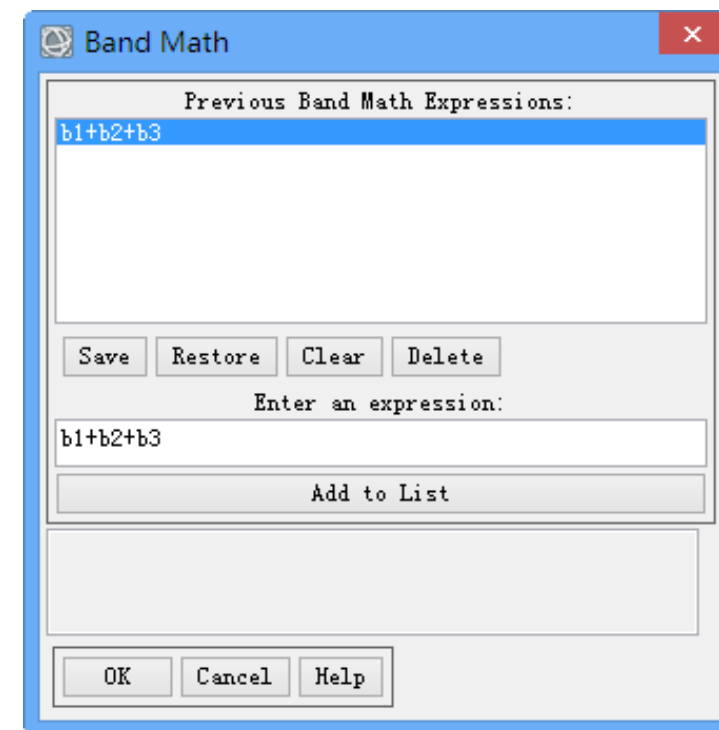
- 波段变量名必须为b1, b2...

- 表达式函数

- 波段变量名必须为b1, b2...
    - 波段运算函数Function开头，并有返回值；

- ENVI只调用SAV

- ENVI+IDL可以调用sav和pro



## 2.1 波段运算

- 植被覆盖度函数FVC

- 计算归一化植被指数 (NDVI)

$$NDVI = (NIR - R) / (NIR + R)$$

- 计算植被覆盖度 (FVC)

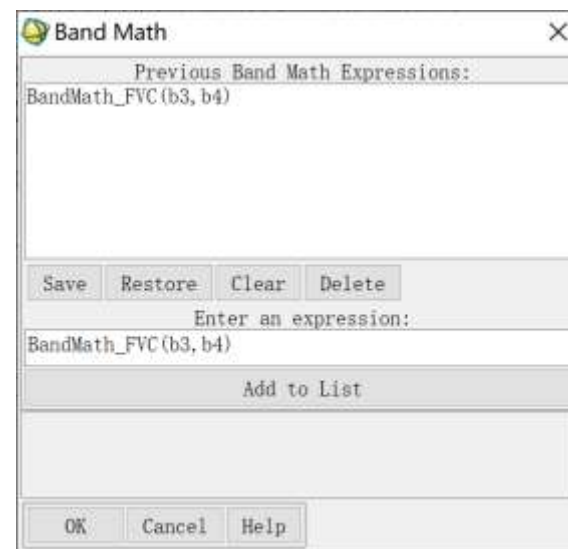
$$FC = (NDVI - NDVI_{min}) / (NDVI_{max} - NDVI_{min})$$

## 2.2 ENVI波段运算扩展

### ●在IDL中启动ENVI

```
IDL> envi  
% Restored file: ENVI.  
% Restored file: ENVI_M01.  
% Restored file: ENVI_M02.  
% Restored file: ENVI_M03.  
% Restored file: ENVI_M04.
```

### ●波段运算中调用函数FVC



## 2.2 ENVI波段运算扩展

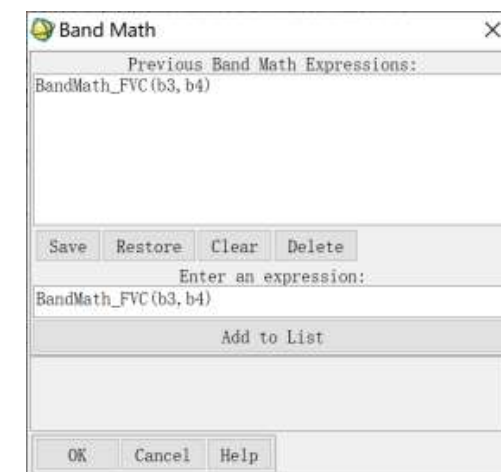
- 植被覆盖度函数编译成sav文件

```
ENVI>  
save, 'BandMath_FVC', filename='D:\BandMath_FVC.sav', /routines
```

- 直接启动ENVI进行波段运算(SAV文件)

- 复制到 Program Files > Exelis > ENVI53 > classic > save\_add

- 复制到 Program Files > Exelis > ENVI53 > custom\_code



## 2.3 ENVI Classic 批处理模式

- 批处理模式——在非GUI下使用ENVI
  - 使用IDL工作但偶尔需要用到ENVI的函数；
  - 进行大量ENVI处理而不需人工干预（批处理）；
  - 批处理模式的ENVI和正常模式下基本相同，该模式下要通过调用ENVI的函数库来执行ENVI的功能。

不启动ENVI界面，而调用ENVI功能



## 2.3 ENVI Classic 批处理模式

- 启动ENVI批处理模式

- envi, /restore\_base\_save\_files

- ENVI功能分散在大约50个左右的IDL sav文件中，存放在ENVI安装路径下的Save目录中。
    - ENVI的核心sav文件包括ENVI的基本功能函数，动态运行函数以及ENVI运行所需的内部变量。

## 2.3 ENVI Classic 批处理模式

---

- 启动ENVI批处理模式
  - 模式初始化
    - `envi_batch_init`
  - 模式关闭
    - `envi_batch_exit`

## 2.3 ENVI Classic 批处理模式

- ENVI Classic下常用的关键字
- FID
  - 一个长整型的标量。FID用于ENVI程序操作文件，如进行打开或选择，注意ENVI程序中所有对文件进行的所有操作和处理都通过该FID完成。
  - 在进行下一步处理时，要检查FID是否有效；

```
if fid eq -1 then return
```

## 2.3 ENVI Classic 批处理模式

### ●R\_FID和M\_FID

- ENVI程序产生处理的**结果**将包括一个R\_FID, 称为返回FID;
- 如结果数据是存在内存中, R\_FID是访问该数据的唯一方法;
- 进行掩模处理的程序还包括一个M\_FID, 称为**掩模FID**, 它用于确定用作掩模波段的文件。

## 2.3 ENVI Classic 批处理模式

### ●DIMS 空间裁剪

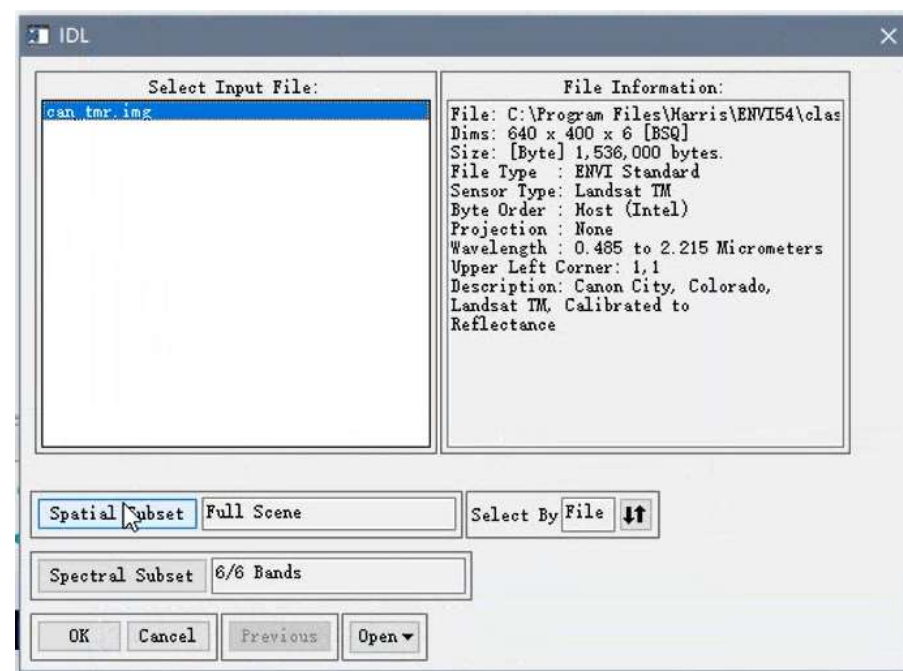
- DIMS关键字是一个5个元素长整型数组。它定义了欲处理数据的空间子集。
- DIMS[0] 存储一个打开的ROI区域的指针，仅在ROI被定义的时候使用，其它时候设为-1L
- DIMS[1] 采样的列起始位置 Sample start
- DIMS[2] 采样的列终止位置 Sample end
- DIMS[3] 行的起始位置 Line start
- DIMS[4] 行的结束位置 Line end

## 2.3 ENVI Classic 批处理模式

### ●POS

#### 光谱裁剪

- POS关键字定义处理波段位置，是一个长整型数组；
- 波段从0开始，例如，要处理第三波段，POS=[2]。



## ENVI Classic 常用函数

- 文件管理函数

- ENVI\_PICKFILE: 选择文件

```
IDL> file=envi_pickfile(filter=[ '*.*dat', '*.*img' ])
```

- ENVI\_SELECT: 选择界面

- ENVI\_OPEN\_FILE: 打开文件

```
IDL> envi_open_file, file, r_fid=fid
IDL> fid
2
```

- ENVI\_FILE\_MNG: 关闭和删除文件

- ENVI\_GET\_FILE\_IDS: 获取ENVI打开文件的ID

# ENVI Classic 常用函数

- 打开文件函数

- ENVI\_OPEN\_DATA\_FILE: 打开特定传感器文件

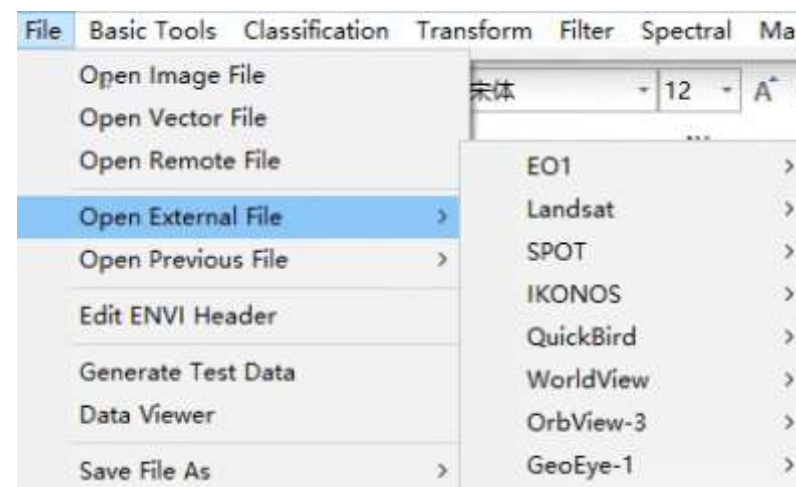
- 数据获取函数

- ENVI\_GET\_DATA

- ENVI\_GET\_SLICE

- 感兴趣区函数

- ENVI\_GET\_ROI\_DIMS\_PTR





# ENVI Classic 常用函数

---

- 数据生成函数

- ENVI\_ENTER\_DATA

- ENVI\_SETUP\_HEAD

- ENVI\_WRITE\_ENVI\_FILE

# ENVI Classic 图像处理接口

- 调用格式

- ENVI\_DOIT, 'XXXX\_DOIT', FID=FID, R\_FID=R\_FID

ENVI\_DOIT, 'RESIZE\_DOIT', DIMS=array, FID=file ID, /IN\_MEMORY, INTERP={0 | 1 | 2 | 3}, OUT\_BNAME=string array, OUT\_NAME=string, POS=array, R\_FID=variable, RFACT=array

# ENVI Classic 图像处理接口

## ●波段运算(math\_doit) （非用户界面）

### Syntax

ENVI\_DOIT, 'MATH\_DOIT', DIMS=array, EXP=string, FID=array, /IN\_MEMORY, OUT\_BNAME=string array, OUT\_NAME=string, POS=array, R\_FID=variable

- DIMS: 数组，图像起始行列
- EXP: 字符串，波段运算公式
- FID: 数值或数组，文件编码
- OUT\_NAME: 字符串，输出文件路径
- POS: 数组，波段索引
- R\_FID: 返回文件编码

# ENVI Classic 图像处理接口

---

- 波段运算批处理
  - FOR循环+math\_doit
  - 输出文件路径

## 考查题二：图像植被增强处理

- 编写植被增强函数 (TestVegEnhance)
  - 计算NDVI
  - 计算新的绿波段：
$$(ndvi > 0.2) * (0.3 * nir + 0.7 * green) + (ndvi \leq 0.2) * green$$
- 函数调用格式
  - (1) 在ENVI交互界面下，调用函数TestVegEnhance实现数据can\_tmr.img的波段运算，保存并显示波段运算结果（真彩色）；
  - (2) 在非用户交互界面下，编写过程调用函数TestVegEnhance实现数据can\_tmr.img的波段运算，保存波段运算结果。

