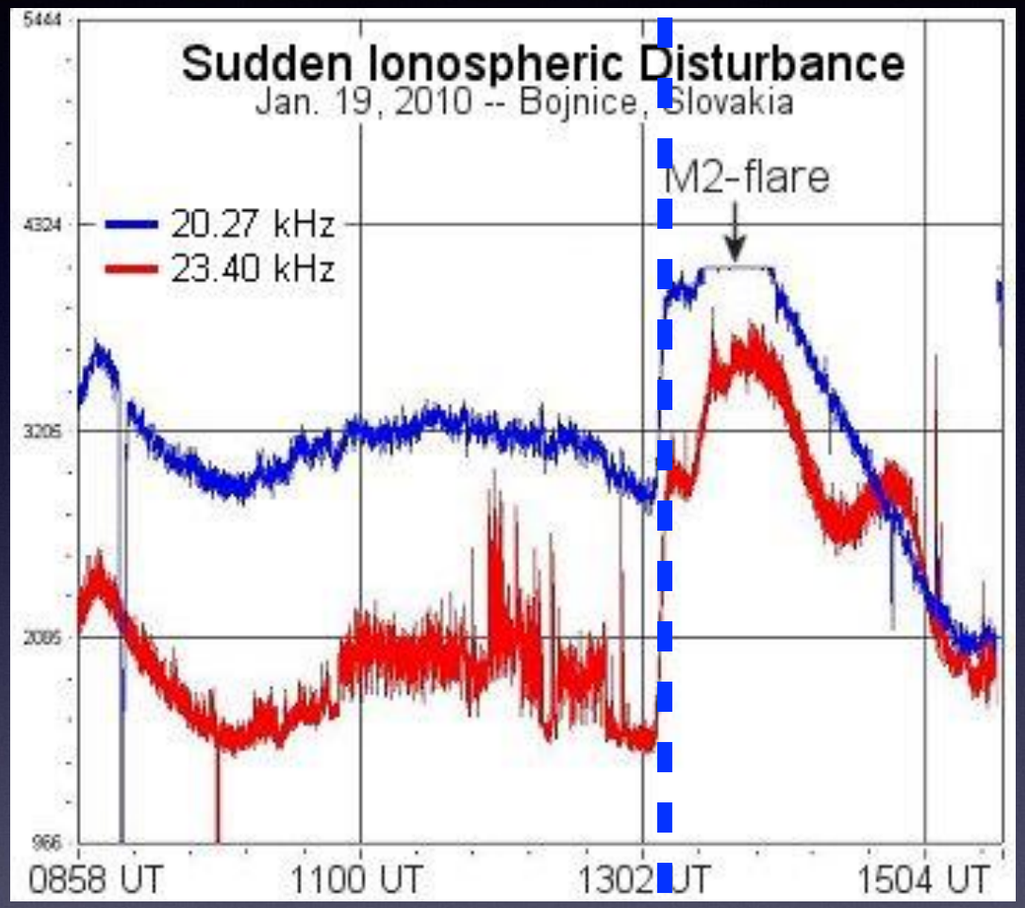
第四章

空间天气因果链分析及数据处理方法

1、空间天气事件因果链分析

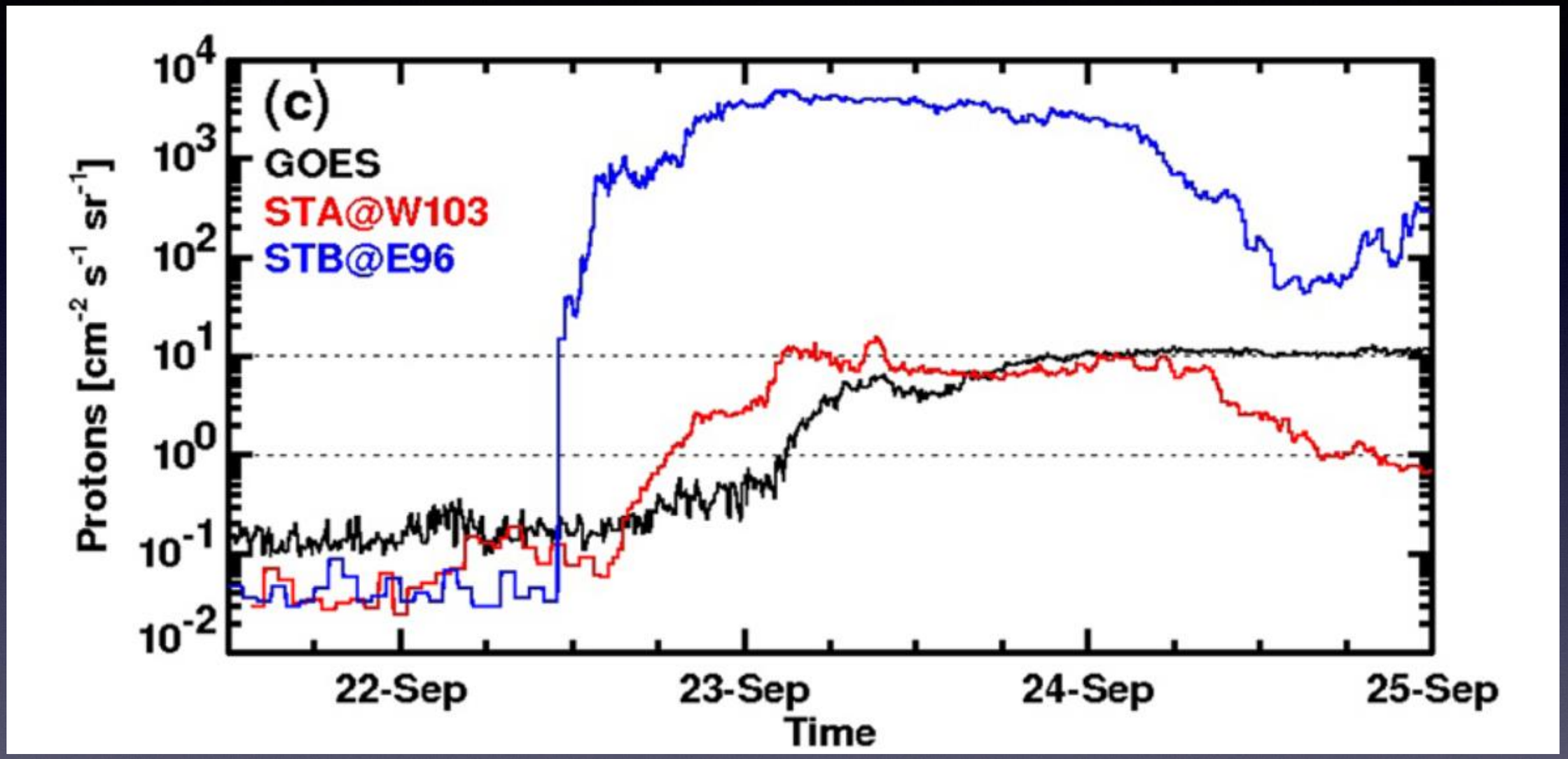
电离层扰动的源分析




```
:Product: 20100119events.txt
:Created: 2010 Jan 22 0332 UT
:Date: 2010 01 19
# Prepared by the U.S. Dept. of Commerce, NOAA, Space Weather Prediction Center
# Please send comments and suggestions to SWPC.Webmaster@noaa.gov
#
# Missing data: ////
# Updated every 30 minutes.
#                            Edited Events for 2010 Jan 19
#
#Event    Begin    Max       End  Obs  Q  Type  Loc/Frq   Particulars       Reg#
#-----------------------------------------------------------------------------

5610       0128   0130      0132  LEA  3   FLA  N24W88    SF                1040

5620 +     0614   0618      0636  G14  5   XRA  1-8A      B5.8    6.1E-04   1041

5630 +     0703   0711      0717  G14  5   XRA  1-8A      B8.3    5.0E-04   1040

5640 +     0834   0845      0914  G14  5   XRA  1-8A      C5.2    9.2E-03   1040

5650 +     1025   1029      1036  G14  5   XRA  1-8A      C1.1    5.5E-04   1041

5660 +     1303   1341      1350  G14  5   XRA  1-8A      M2.3    3.9E-02   1041

5670 +     1532   1536      1543  G14  5   XRA  1-8A      C2.3    1.0E-03   1041
```

电离层扰动 → 太阳耀斑

太阳高能粒子事件的源分析

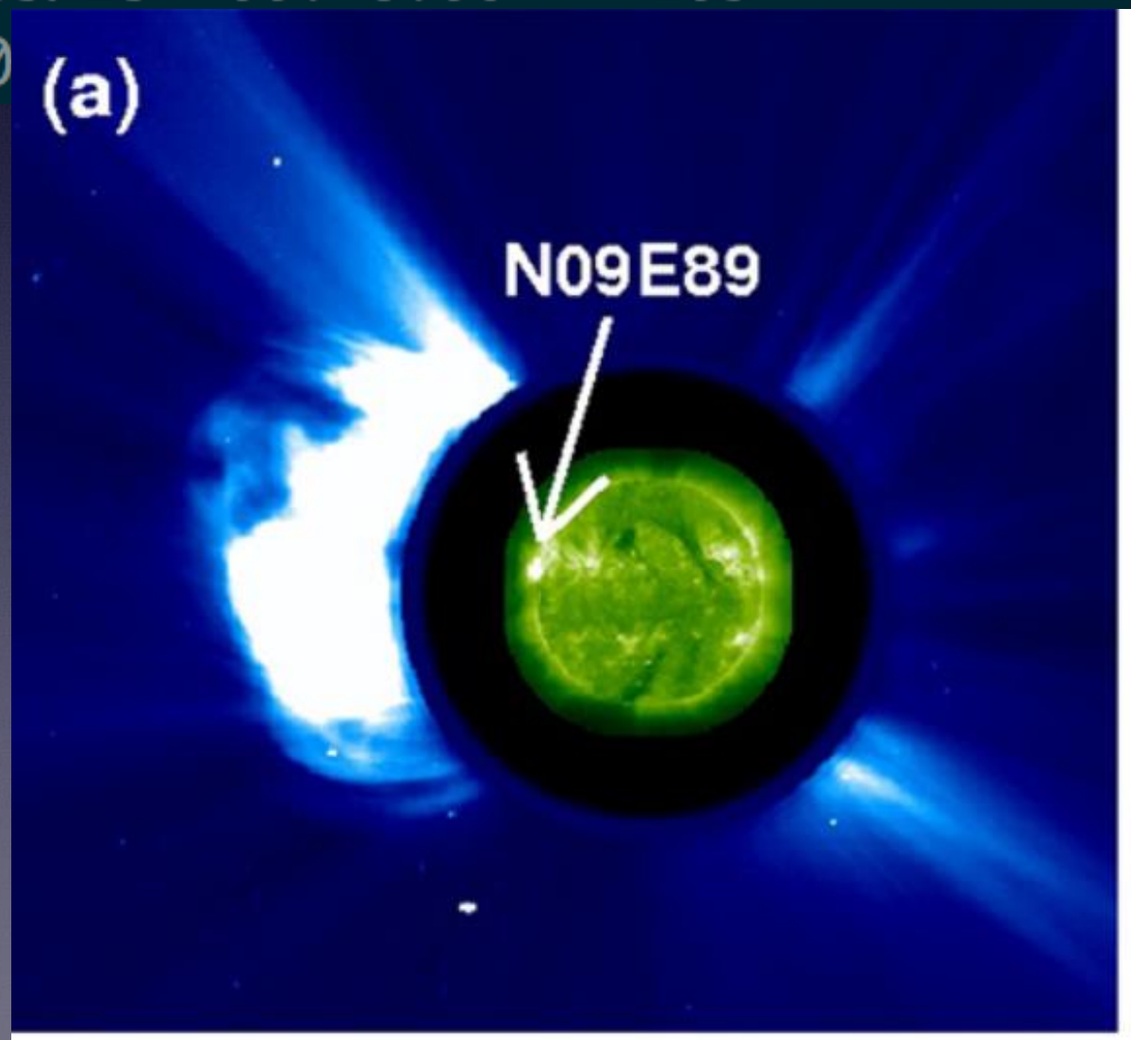


太阳高能粒子事件

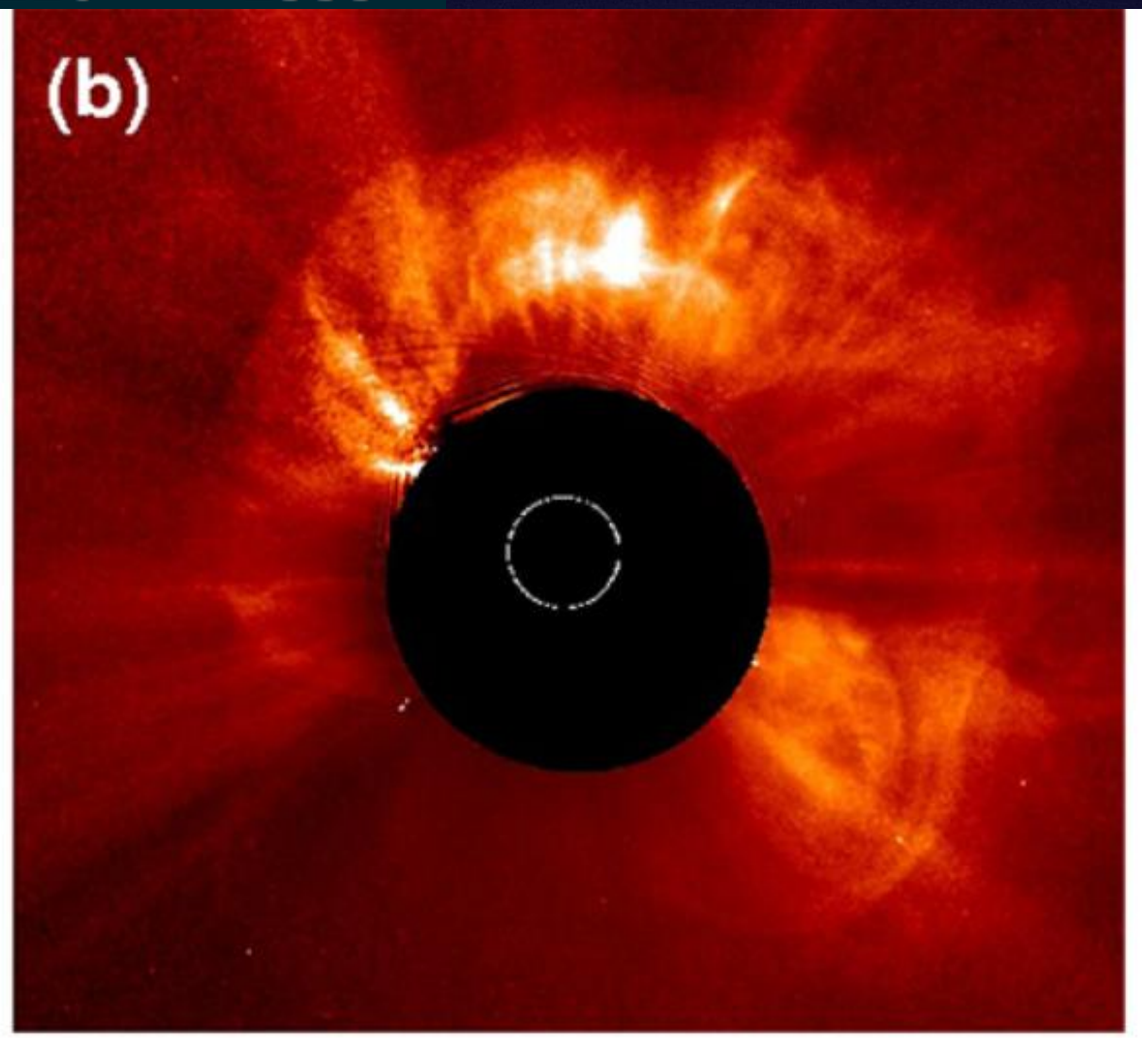
- 渐进事件 → CME
- 脉冲事件 → 耀斑

太阳高能粒子事件的源分析

2011/09/22	06:48:05	297	46	452	353
2011/09/22	08:48:06	298	53	508	425
2011/09/22	10:48:06	Halo	360	1905	2211
2011/09/22	18:36:07	42	46	374	382
2011/09/23	00:12:05	47	40	327	231
2011/09/23	00:48:06	105	44	1116	1333
2011/0					

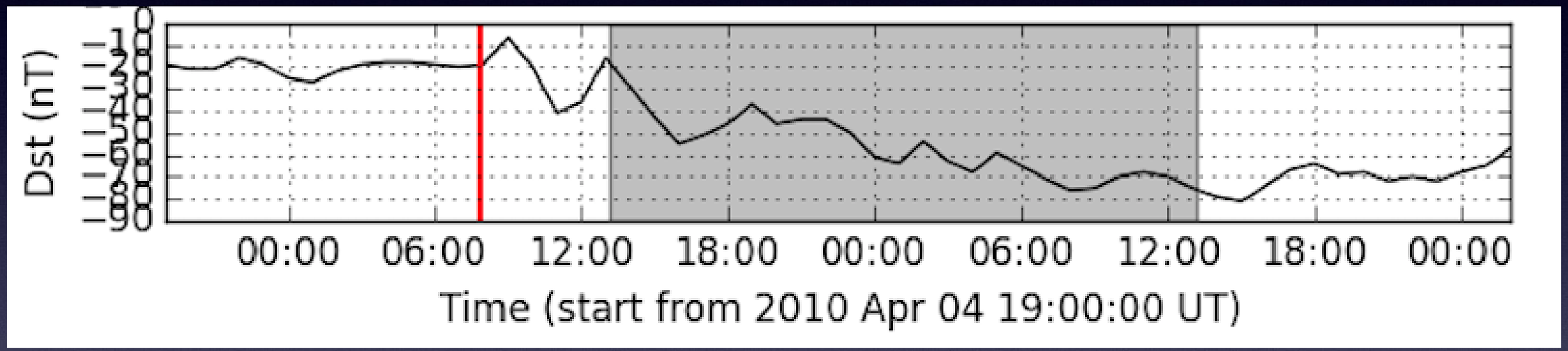


SOHO: 2011/09/22 11:00



STEREO-B: 2011/09/22 11:54

地磁暴的源分析



地磁爆 → 行星际源

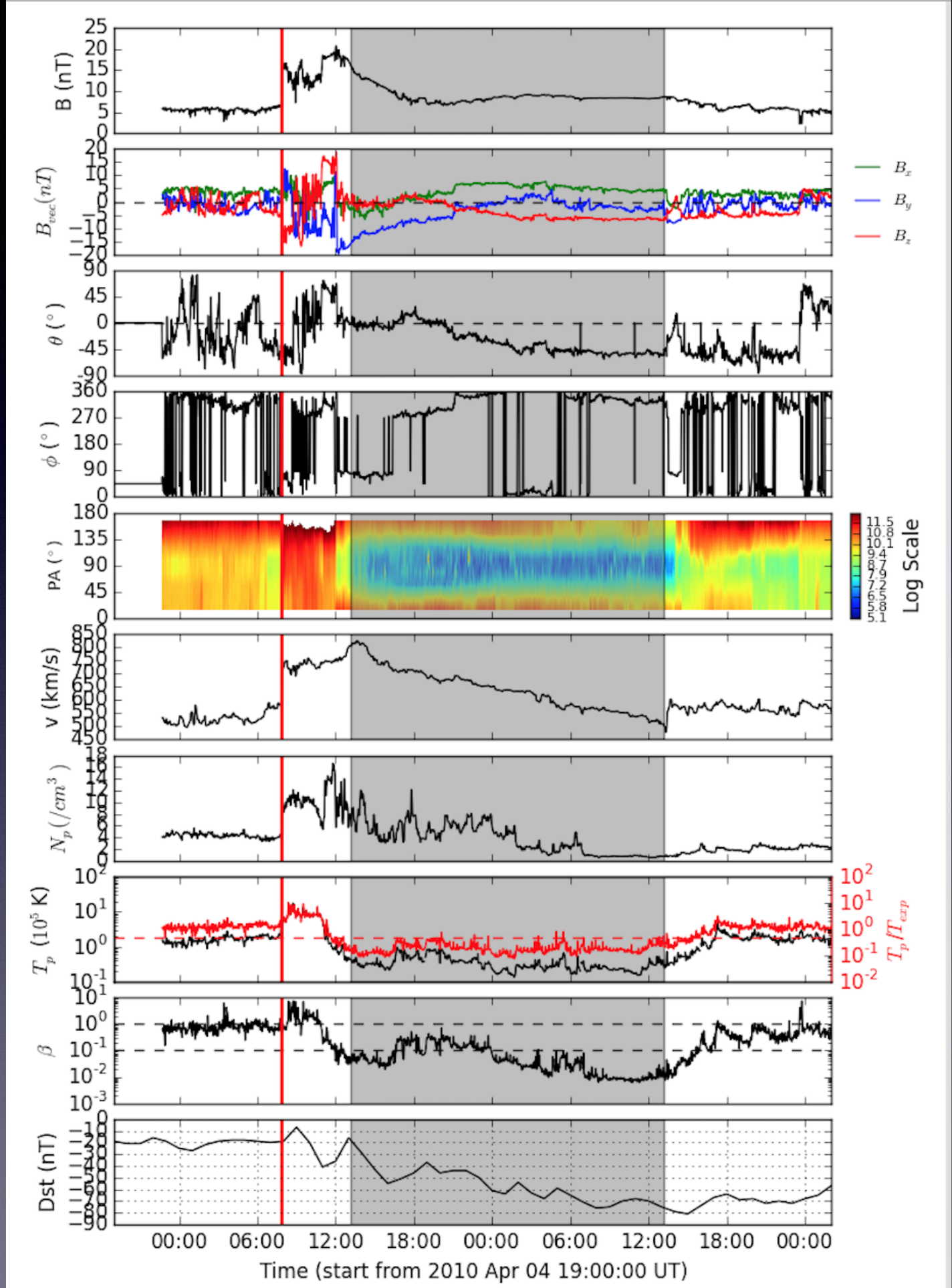
地磁爆的源分析

地磁爆

行星际源

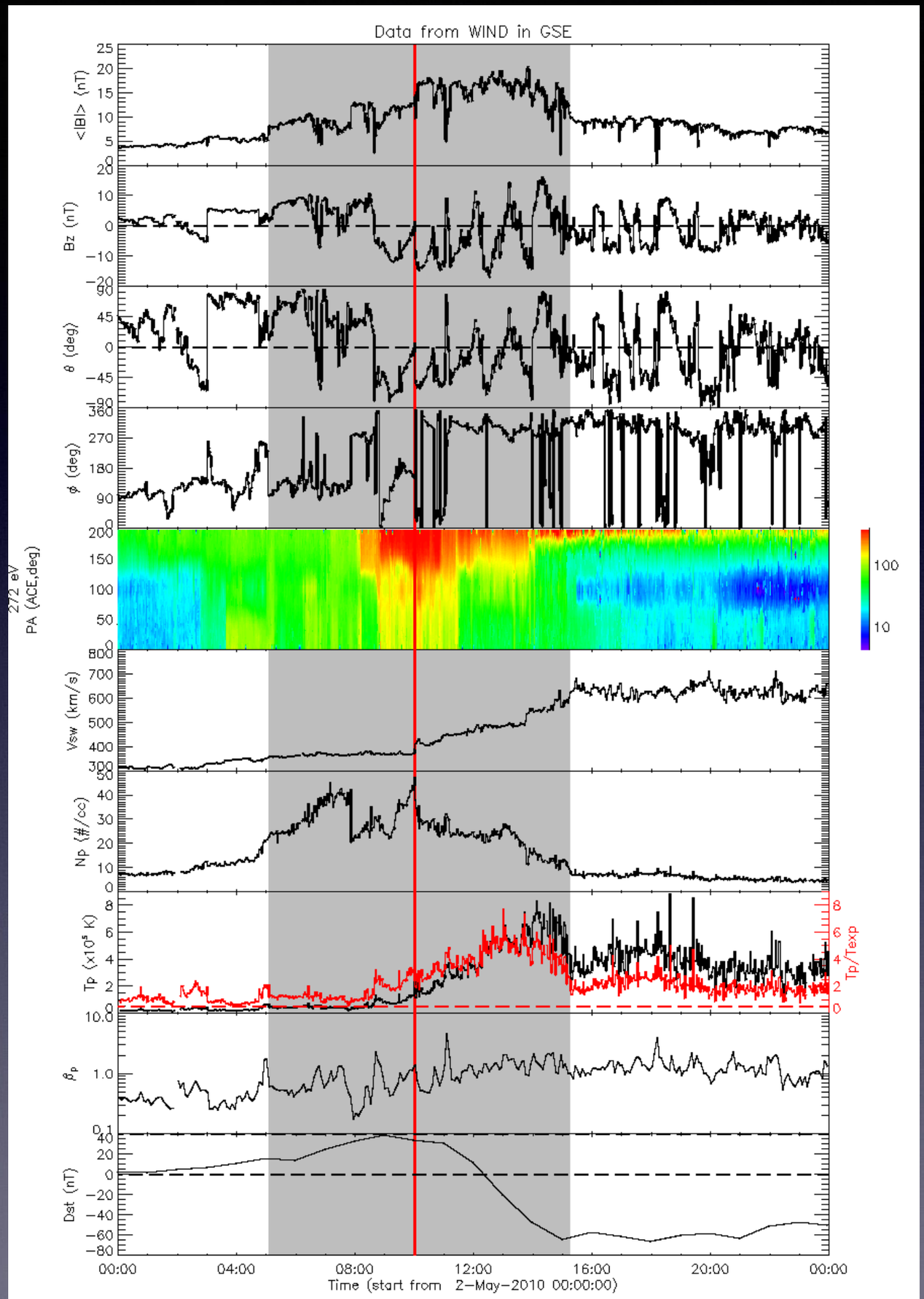
CIR

ICME



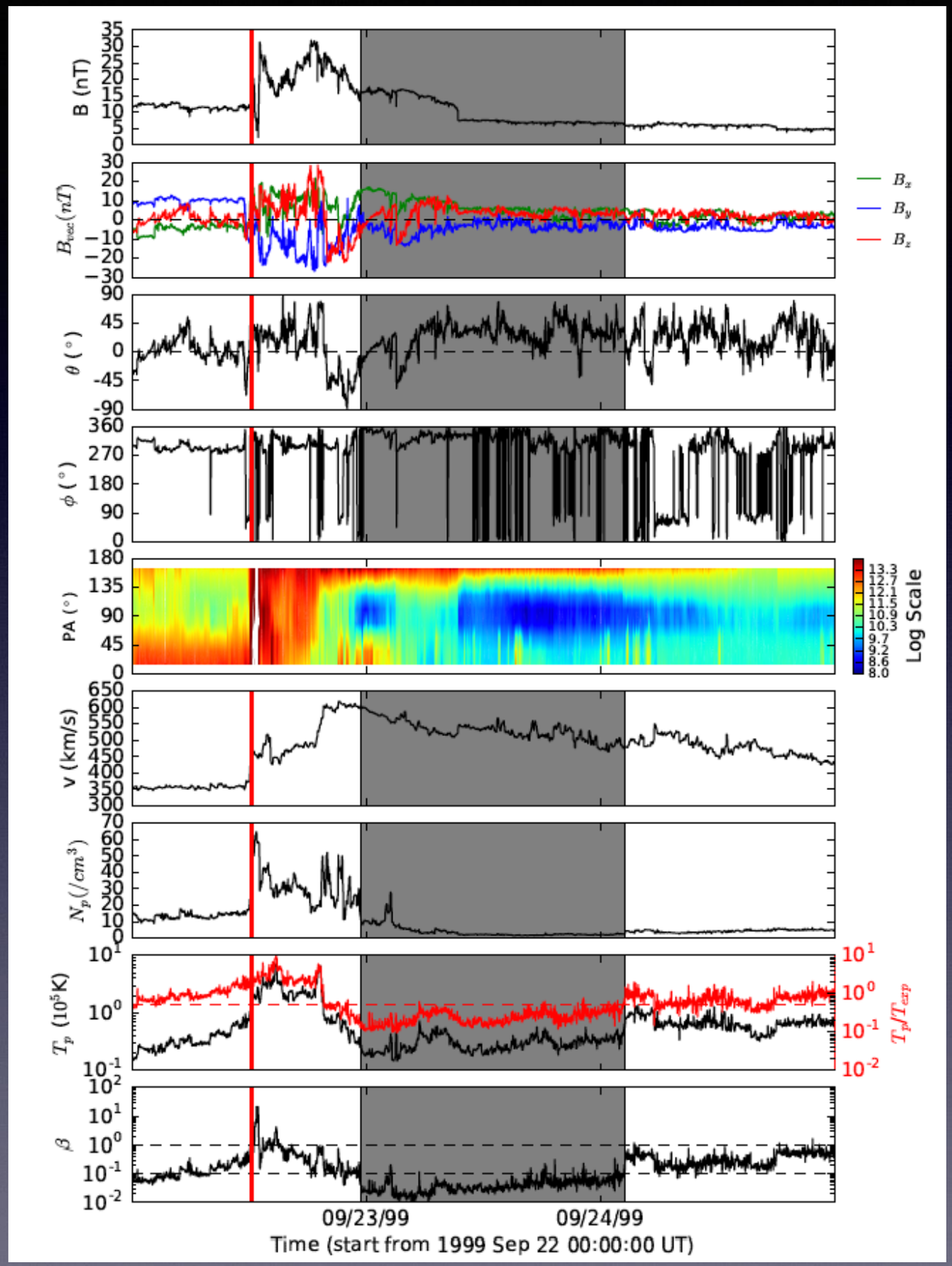
共转流相互作用区（CIR）

- 太阳风速度增加
- 压强达到峰值
- 压缩的磁场
- 压缩的密度
- 增加的温度



行星际日冕物质抛射 (ICME)

- 增强的磁场
- 平滑旋转的磁场方向
- 膨胀的速度特征
- 双向电子流
- 双向低能质子流
- 低的质子温度
- 低的等离子体Beta
- 高的Fe电离态
- Forbush Decrease
- …



地磁爆的源分析

地磁爆

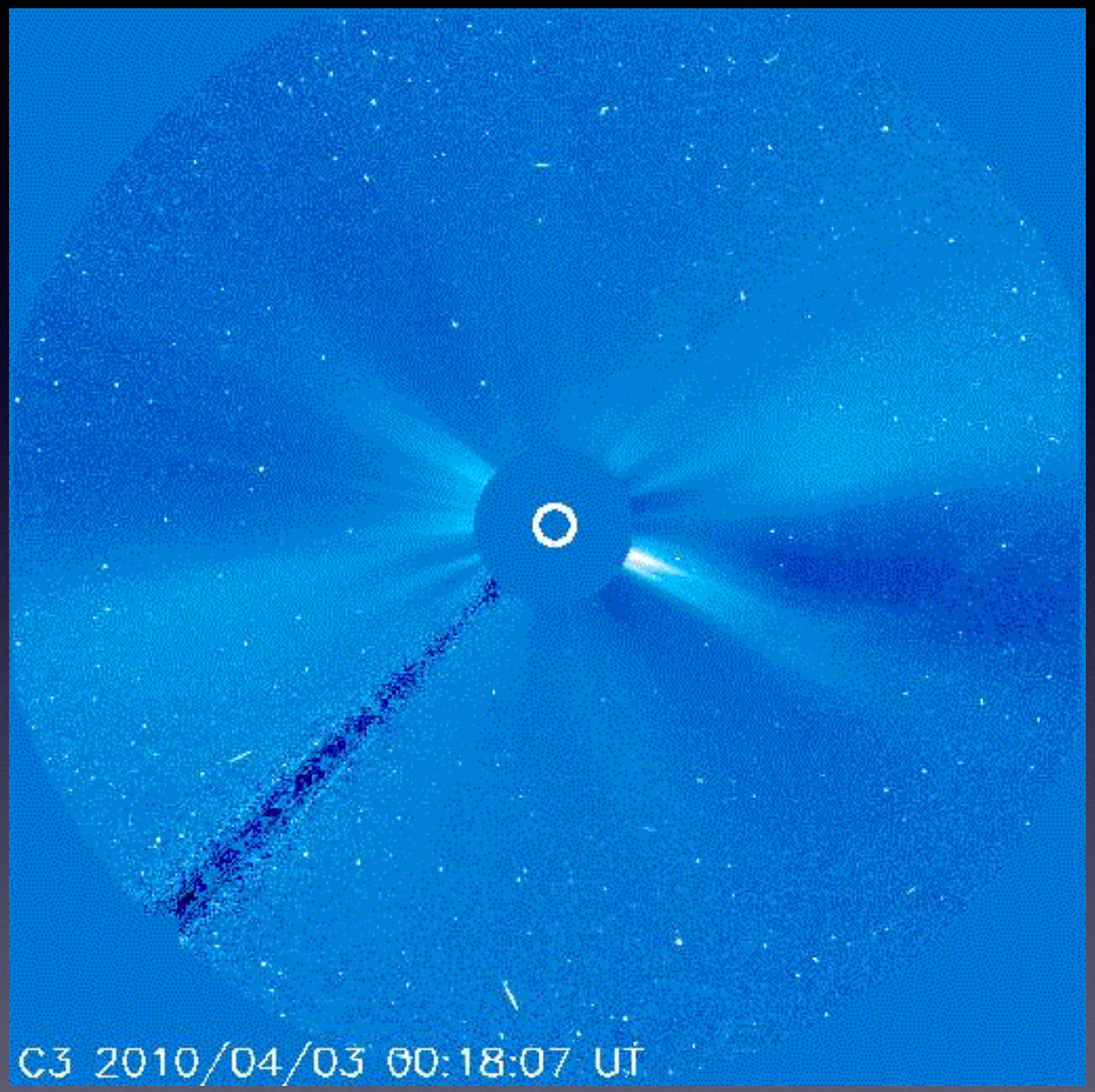
行星际源

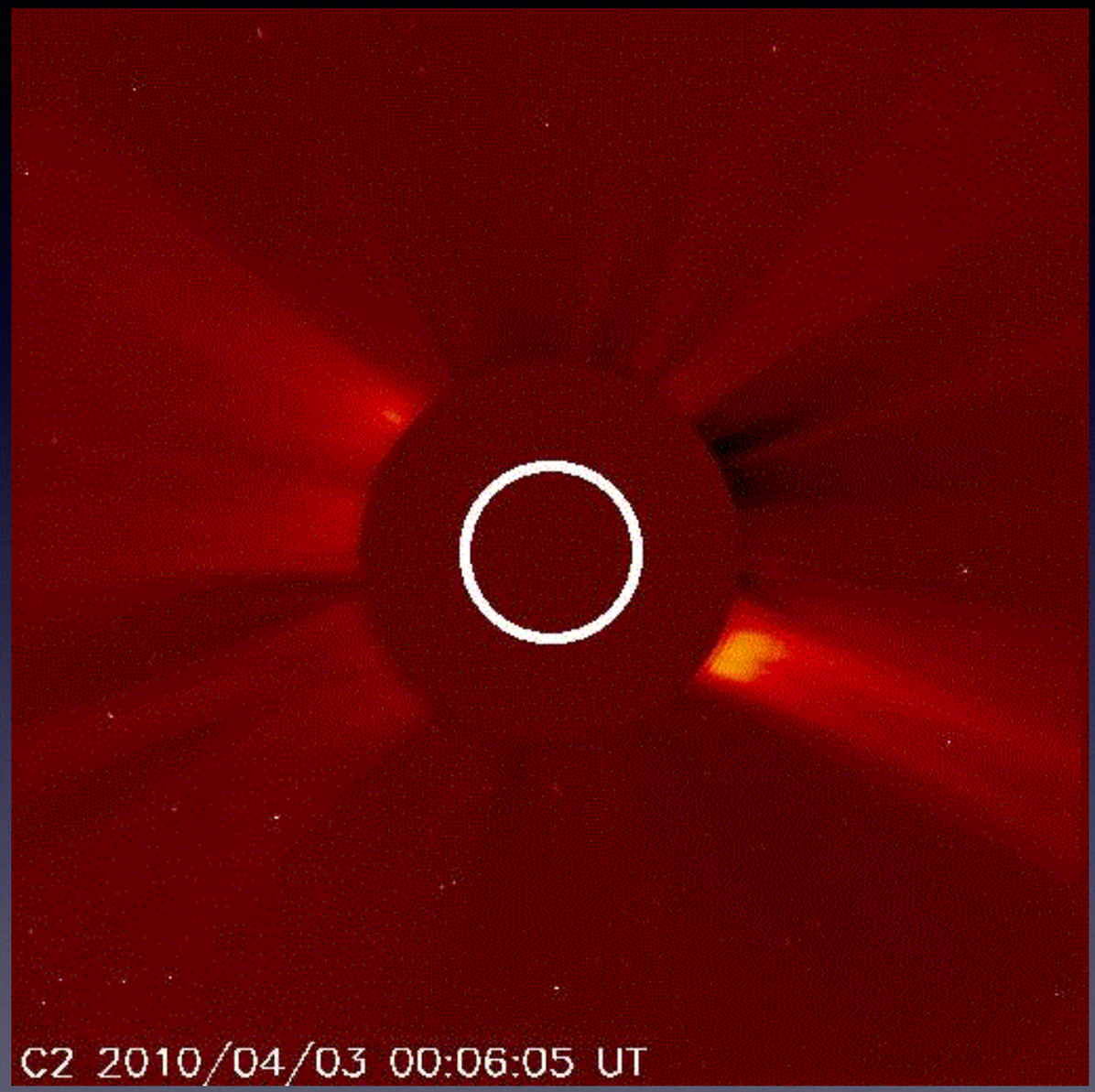
CIR

ICME

太阳源(CME)

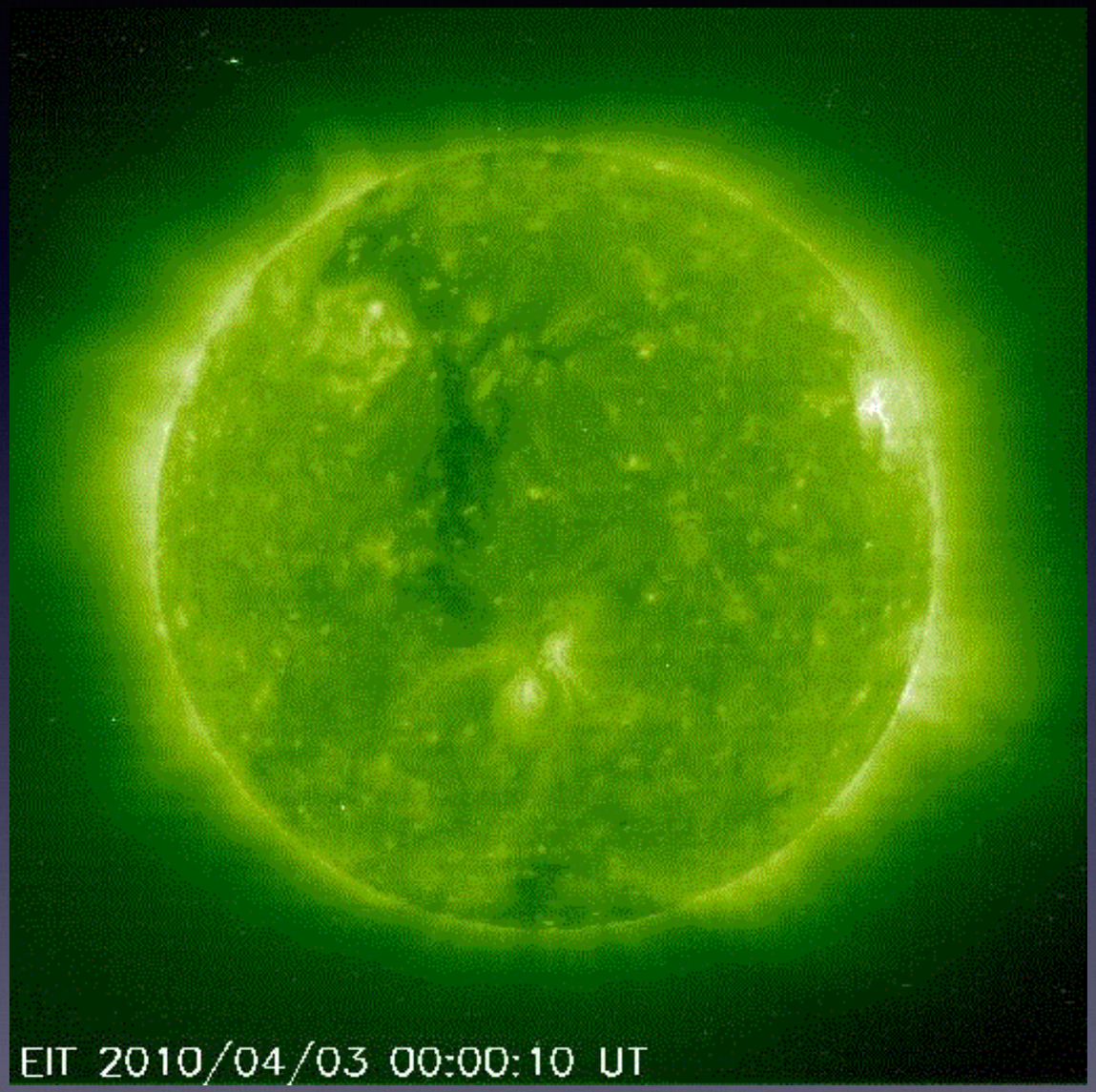
2010/04/01	06:30:05	83	46	114	115	113
2010/04/01	12:30:05	284	73	36	24	48
2010/04/01	14:30:05	216	14	223	188	256
2010/04/01	22:30:05	309	59	312	283	344
2010/04/02	04:30:05	4	12	471	432	513
2010/04/02	09:30:05	111	13	285	172	391
2010/04/02	09:54:05	262	53	164	22	302
2010/04/02	11:06:05	6	16	632	347	900
2010/04/02	11:30:05	80	55	550	505	594
2010/04/03	03:56:29	243	29	236	163	310
2010/04/03	10:33:58	Halo	360	668	677	658
2010/04/03	20:30:05	110	12	180	207	154
2010/04/03	21:30:08	82	14	301	330	274
2010/04/03	22:30:05	53	8	215	212	218
2010/04/03	23:06:52	222	91	247	214	282
2010/04/04	08:54:05	235	44	248	238	259
2010/04/04	16:06:05	303	13	283	303	262
2010/04/04	16:06:05	56	33	63	18	119
2010/04/04	17:06:05	126	24	131	126	136
2010/04/04	17:30:05	233	38	222	213	229
2010/04/05	04:54:05	121	25	249	177	324
2010/04/05	05:30:05	231	18	325	227	430





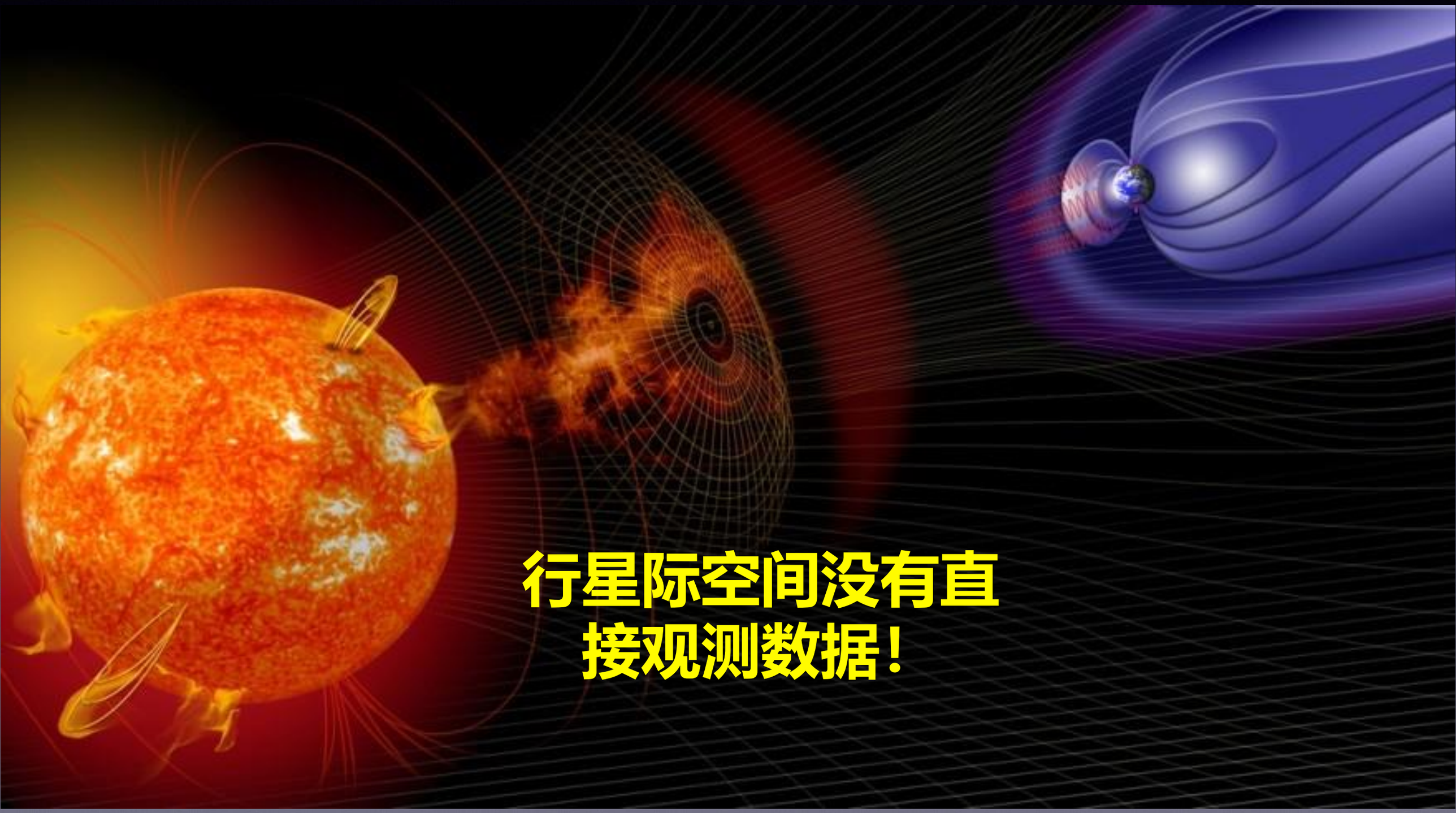
太阳源：CME

地磁爆的太阳源



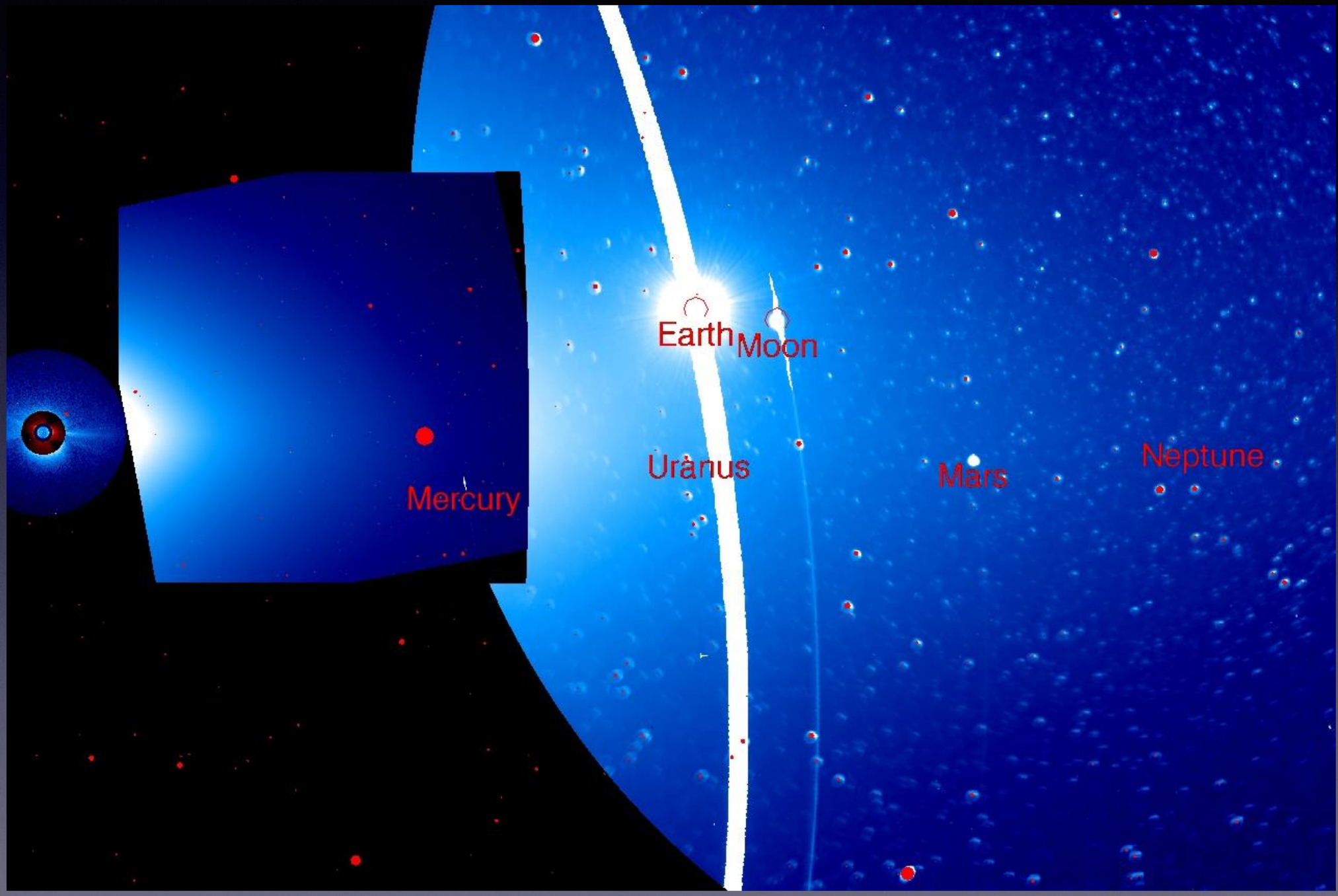
耀斑?　　日珥?　　...?

地磁爆源分析(STEREO期间)

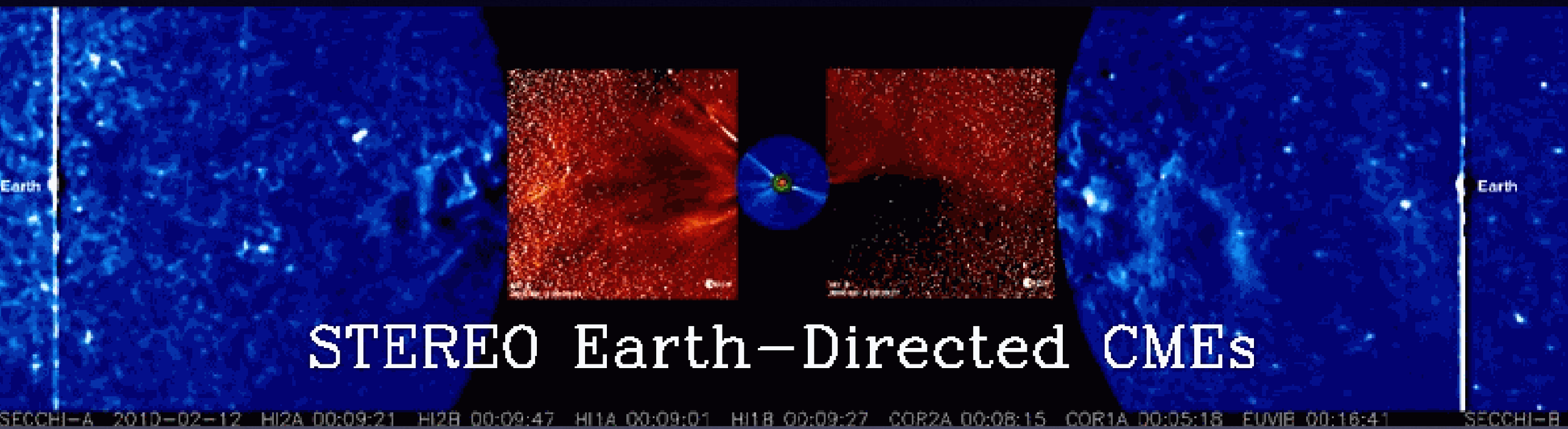


行星际空间没有直接观测数据！

地磁爆源分析(STEREO期间)

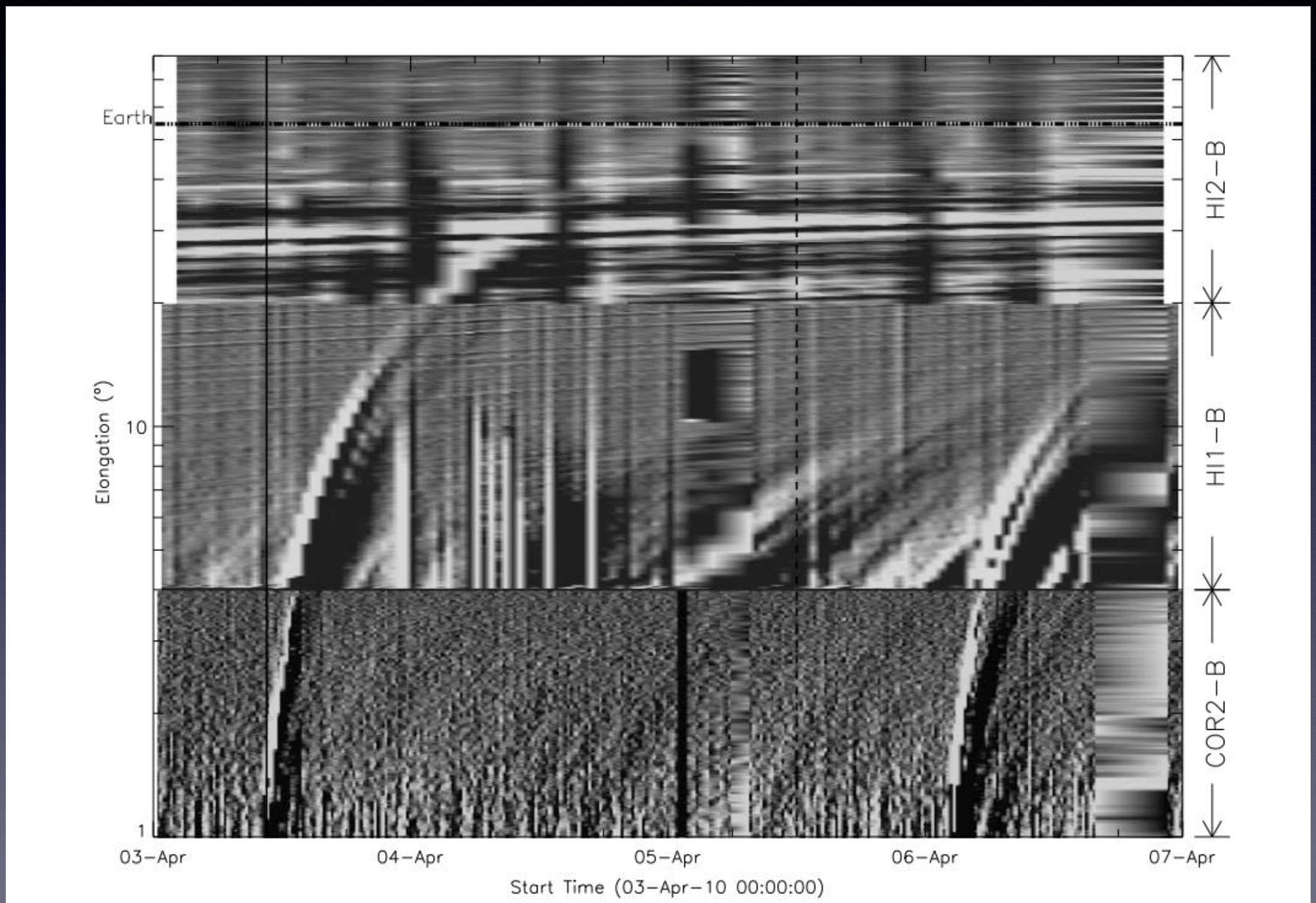


地磁爆源分析(STEREO期间)



可以监测CME从爆发到传播到达地球的全过程！

J-Map


Earth
10
1
Elongation (°)
03–Apr
04–Apr
05–Apr
06–Apr
07–Apr
Start Time (03–Apr–10 00:00:00)
HI2–B
HI1–B
COR2–B

2、常用数据处理

数据读取

- ★ 文本格式
- ★ CDF格式
- ★ FITs格式

数据预处理

- ★ 显示效果预处理
- ★ 数据产品预处理

数据显示

- ★ 一维图形显示
- ★ 二维图形显示
- ★ 三维图形显示
- ★ 其他图形显示

☑ 数据处理程序规范化　☑ 海量数据的自动处理

读取文本文件

openr：

OPENR, *Unit*, *File*

全部取出完后需要：free_lun,unit

readf：

READF, [*Prompt*,] *Unit*, Var_1, ..., Var_n

Keywords: [, AM_PM=*[string, string]*] [, DAYS_OF_WEEK=*string_array*{7 names}] [, FORMAT=*value*] [, MONTHS=*string_array*{12 names}] [, PROMPT=*string*]

reads：

READS, *Input*, Var_1, ..., Var_n [, AM_PM=*[string, string]*] [, DAYS_OF_WEEK=*string_array*{7 names}] [, FORMAT=*value*] [, MONTHS=*string_array*{12 names}]

其他常用的命令：

strsplit：

Result = STRSPLIT(*String* [, *Pattern*] [, COUNT=*variable*] [, ESCAPE=*string* | , /REGEX [, /FOLD_CASE]] [, /EXTRACT | , LENGTH=*variable*] [, /PRESERVE_NULL])

strmid：

Result = STRMID(*Expression*, *First_Character* [, *Length*] [, /REVERSE_OFFSET])

strlen：

Result = STRLEN(*Expression*)

2003/10/28/1215	2003/10/29/0615	29500	2003/10/28/1054	147	1054	2003/10/28/1110	S16E08
2003/11/02/1105	2003/11/03/0815	1570	2003/11/02/0954	360	2036		
2003/11/04/2225	2003/11/05/0600	353	2003/11/04/1954	360	3657	2003/11/04/1929	S19W83
2003/11/21/2355	2003/11/22/0230	13	2003/11/21/0026	52	494	2003/11/20/2353	N02W17
2003/12/02/1505	2003/12/02/1730	86	2003/12/02/1730	32		2003/12/02/0948	W limb
2004/04/11/1135	2004/04/11/1845	35	2004/04/11/0430	314	1645	2004/04/11/0419	S14W47
2004/07/25/1855	2004/07/26/2250	2086	2004/07/25/1530	360	1333	2004/07/25/1514	N08W33
2004/09/13/2105	2004/09/14/0005	273	2004/09/12/0036	360	1328	2004/09/12/0056	N04E42
2004/09/19/1925	2004/09/20/0100	57	2004/09/19/2224	99		2004/09/19/1712	N03W58
2004/11/01/0655	2004/11/01/0805	63	2004/11/01/0606	146	925	Backside	
2004/11/07/1910	2004/11/08/0115	495	2004/11/07/1706	360	1759	2004/11/07/1606	N09W17
2005/01/16/0210	2005/01/17/1750	5040	2005/01/15/2306	360	2861	2005/01/15/2302	N15W05
2005/05/14/0525	2005/05/15/0240	3140	2005/05/13/1722	360	1689	2005/05/13/1657	N12E11
2005/07/14/0245	2005/07/15/0345	134	2005/07/13/1430	360	1423	2005/07/13/1657	N10W80
2005/07/27/2300	2005/07/29/1715	41	2005/07/27/0454	360	1787	2005/07/27/0502	N11E90
2005/08/22/2040	2005/08/23/1045	330	2005/08/22/1730	360	2378	2005/08/22/1727	S12W60
2006/12/13/0310	2006/12/13/0925	698	2005/12/13/0254	360	1774	2005/12/13/0240	S05W23
2011/03/08/0105	2011/03/08/0800	50	2011/03/07/2000	360	2125	2011/03/07/2012	N24W59
2011/03/21/1950	2011/03/22/0135	14	2011/03/21/0236	360	1341		
2011/09/23/2255	2011/09/26/1155	35	2011/09/27/0430	73	261	2011/09/22/1101	N11E74
2012/01/23/0530	2012/01/24/1530	6310	2012/01/23/0400	360	2175	2012/01/23/0359	N28W36
2012/01/27/1905	2012/01/28/0205	796	2012/01/27/1827	360	2508	2012/01/27/1837	N27W71
2012/03/07/0510	2012/03/08/1115	6530	2012/03/07/0036	360	2684	2012/03/07/0024	N17E15
2012/03/13/1810	2012/03/08/1115	469	2012/03/13/1736	360	1884	2012/03/13/1741	N18W62
2012/05/17/0210	2012/05/17/0430	255	2012/05/17/0148	360	1582	2012/05/17/0147	N12W89
2012/05/27/0535	2012/05/27/1045	14	2012/05/27/2112	360	1966		
2012/06/16/1955	2012/06/16/2020	14	2012/06/14/1430	360	987	2012/06/14/1435	S17E14
2012/07/07/0400	2012/07/07/0745	25	2012/07/06/2312	360	1828	2012/07/06/2308	S18W50
2012/07/12/1835	2012/07/12/2225	96	2012/07/12/1649	360	885	2012/07/12/1710	S16W09
2012/07/17/1715	2012/07/18/0600	136	2012/07/17/1348	176	958	2012/07/17/1715	S17W75
2012/07/23/1545	2012/07/23/2145	12	2012/07/23/0236	360	2003	S16W86	

CDF文件

CDF(Common Data Format):

What is Common Data Format (CDF)?

- Self-describing data format for the storage of scalar and multidimensional data in a platform- and discipline-independent way
- Scientific data management package (CDF Library) allows application developers to manage these data arrays
- Transparent access to data and meta-data through Application Programming Interfaces (APIs)
- Built-in support for data compression (gZip, RLE, Huffman) and automatic data uncompression, and checksum
- Large file support (> 2G-bytes)
- CDF library includes a suite of tools that allow users to manipulate CDF files
- Provide read/write interfaces for C, FORTRAN, Java, Perl, C#/Visual Basic, IDL, MATLAB (and user-supplied software, e.g., Python, Sybase, mySQL)

http://cdf.gsfc.nasa.gov

读取CDF文件

使用CDAWEB程序包
Website：http://cdf.gsfc.nasa.gov/html/cdf_patch_for_idl.html

加载方式：
IDL> @compile_cdaweb
常用命令：
read_mycdf:
运行方式
npstr=read_mycdf(‘np’,’fitsname’)

```
IDL> npstr=read_mycdf('Np','wi_k0_swe_20140123_v02.cdf')
read_myCDF took        0.0000000 seconds to generate VVs.
IDL> help,npstr,/str
** Structure <e54c68>, 2 tags, length=13112, data length=13100, refs=1:
   NP              STRUCT    -> <Anonymous> Array[1]
   EPOCH           STRUCT    -> <Anonymous> Array[1]
IDL> help,npstr.np.dat,/str
<Expression>    FLOAT     = Array[870]
```

datastr=read_mycdf(‘’,’fitsname’,/all)

```
IDL> datastr=read_mycdf('
read_myCDF took    9.5367
IDL> help,datastr,/str
** Structure <dde5c8>, 30
   EPOCH           STRUCT
   DELTA_TIME      STRUCT
   TIME_PB5        STRUCT
   UNIT_TIME       STRUCT
   LABEL_TIME      STRUCT
   FORMAT_TIME     STRUCT
   GAP_FLAG        STRUCT
   MODE            STRUCT
   SC_POS_GSE      STRUCT
   LABEL_POS_GSE   STRUCT
   SC_POS_GSM      STRUCT
   LABEL_POS_GSM   STRUCT
   SC_POS_R        STRUCT
   DQF             STRUCT
   QF_V            STRUCT
   QF_VTH          STRUCT
   QF_NP           STRUCT
   QF_A$P          STRUCT
   V_GSE           STRUCT
   LABEL_V_GSE     STRUCT
   V_GSM           STRUCT
   LABEL_V_GSM     STRUCT
   V_GSE_P         STRUCT
   LABEL_V_POLAR   STRUCT
   UNIT_POLAR      STRUCT
   THERMAL_SPD     STRUCT
   NP              STRUCT
   ALPHA_PERCENT   STRUCT
   CARTESIAN       STRUCT
   POLAR           STRUCT
```

```
IDL> help,datastr.np,/str
** Structure <e4a138>, 47 tags, length=4808, data length=4800, refs=2:
   VARNAME         STRING    'Np'
   TITLE           STRING    'WIND> Solar Wind Parameters'
   PROJECT         STRING    'ISTP>International Solar-Terrestrial Physics'
   DISCIPLINE      STRING    'Space Physics>Interplanetary Studies'
   SOURCE_NAME     STRING    'WIND>Wind Interplanetary Plasma Laboratory'
   DESCRIPTOR      STRING    'SWE>Solar Wind Experiment'
   DATA_TYPE       STRING    'K0>Key Parameter'
   TEXT            STRING    Array[27]
   MODS            STRING    Array[5]
   ADID_REF        STRING    'NSSD0138'
   LOGICAL_FILE_ID STRING    'WI_K0_SWE_20140123_V02'
   LOGICAL_SOURCE  STRING    'WI_K0_SWE'
   LOGICAL_SOURCE_DESCRIPTION
                   STRING    'Wind Solar Wind Experiment, Key Parameters'
   PI_NAME         STRING    'K. Ogilvie'
   PI_AFFILIATION  STRING    'NASA GSFC'
   MISSION_GROUP   STRING    'Wind'
   INSTRUMENT_TYPE STRING    'Plasma and Solar Wind'
   TEXT_SUPPLEMENT_1
                   STRING    ''
   GENERATION_PROGRAM
                   STRING    'WIND_SWE_KP'
   INPUT_FILE      STRING    Array[12]
   FIELDNAM        STRING    'Proton density'
   VALIDMIN        FLOAT           0.00000
   VALIDMAX        FLOAT           1000.00
   SCALEMIN        FLOAT           0.00000
   SCALEMAX        FLOAT           200.000
   LABLAXIS        STRING    'Ion N'
   UNITS           STRING    '#/cc'
   MONOTON         STRING    ''
   VAR_TYPE        STRING    'data'
   FORMAT          STRING    'f7.1'
   FORM_PTR        STRING    ''
   LABL_PTR_1      STRING    ''
   UNIT_PTR        STRING    ''
   FILLVAL         FLOAT      -1.00000e+31
   DEPEND_0        STRING    'Epoch'
   DEPEND_1        STRING    ''
   DICT_KEY        STRING    'density>proton'
   CATDESC         STRING    'Solar Wind Proton Number Density, scalar'
   DELTA_PLUS_VAR  STRING    ''
   DELTA_MINUS_VAR STRING    ''
   AVG_TYPE        STRING    ''
   DISPLAY_TYPE    STRING    'time_series'
   VAR_NOTES       STRING    ''
   DIM_SIZES       LONG                 0
   CDFTYPE         STRING    'CDF_REAL4'
   CDFRECVARY      STRING    'VARY'
   DAT             FLOAT     Array[870]
```

直接对CDF文件操作读取数据

打开CDF文件：id=CDF_open(filename)

查询CDF文件信息：info=cdf_inquire(id)

```
IDL> id=cdf_open('wi_k0_swe_20140101_v01.cdf')
IDL> help,id,/str
ID              LONG      =     17138664
```

```
IDL> info=cdf_inquire(id)
IDL> help,info,/str
** Structure <dbc1b8>, 9 tags, length=80, data length=72, refs=1:
   NDIMS           LONG                 1
   DECODING        STRING    'HOST_DECODING'
   ENCODING        STRING    'NETWORK_ENCODING'
   MAJORITY        STRING    'COL_MAJOR'
   MAXREC          LONG               857
   NVARS           LONG                30
   NZVARS          LONG                 0
   NATTS           LONG                42
   DIM             LONG      Array[1]
```

NDIMS — The longword integer specifying the number of dimensions in the rVariables in the current CDF.

DECODING — A string describing the decoding type set in the CDF file, such as `MAC_DECODING' or `ALPHAVMSD_ENCODING'.

ENCODING

MAJORITY

MAXREC

NVARS

NZVARS

NATTS — A longword integer specifying the number of attributes in the CDF. Note that the number returned in this field includes both global and variable attributes. You can use the GET_NUMATTR keyword to theCDF_CONTROL routine to determine the number of each.

DIM — A vector where each element contains the corresponding dimension size for the rVariables in the current CDF. For 0-dimensional CDF's, this argument contains a single element (a zero).

name=CDF_VARINQ(i,2,/ZVARIABLE)

IS_ZVAR

NAME

DATATYPE

NUMELEM

RECVAR

DIMVAR

DIM

```
IDL> name=cdf_varinq(id,1)
IDL> help,name
NAME            STRUCT    = -> <Anonymous> Array[1]
IDL> help,name,/str
** Structure <d760d8>, 6 tags, length=72, data length=55, refs=1:
   IS_ZVAR          INT               0
   NAME             STRING    'Delta_time'
   DATATYPE         STRING    'CDF_REAL8'
   NUMELEM          LONG                 1
   RECVAR           STRING    'VARY'
   DIMVAR           BYTE      Array[1]
IDL> name=cdf_varinq(id,2)
IDL> help,name,/str
** Structure <e5c558>, 6 tags, length=72, data length=55, refs=1:
   IS_ZVAR          INT               0
   NAME             STRING    'Time_PB5'
   DATATYPE         STRING    'CDF_INT4'
   NUMELEM          LONG                 1
   RECVAR           STRING    'VARY'
   DIMVAR           BYTE      Array[1]
IDL> name=cdf_varinq(id,6)
IDL> help,name,/str
** Structure <e4ab08>, 6 tags, length=72, data length=55, refs=1:
   IS_ZVAR          INT               0
   NAME             STRING    'GAP_FLAG'
   DATATYPE         STRING    'CDF_INT4'
   NUMELEM          LONG                 1
   RECVAR           STRING    'VARY'
   DIMVAR           BYTE      Array[1]
```

读取数据:

cdf_varget, id, 'epoch', epoch, rec_count=info.MAXREC

```
IDL> cdf_varget,id,'Epoch',epoch,rec_count=info.maxrec
IDL> help,epoch
EPOCH           DOUBLE    = Array[3, 857]
```

```
IDL> cdf_varget,id,'Np',np,rec_count=info.maxrec
IDL> help,np
NP              FLOAT     = Array[3, 857]
```

```
IDL> cdf_varget,id,'tp',np,rec_count=info.maxrec
% CDF_VARGET: Cannot find variable: tp.
% CDF_VARGET: CDF file error. BAD_VAR_NUM: Illegal variable number specified.
% Execution halted at: $MAIN$
```

FITS文件

FITS: Flexible Image Transport System

What is FITS?

- The standard data format used in astronomy
- Stands for 'Flexible Image Transport System'
- Endorsed by NASA and the International Astronomical Union
- Much more than just another image format (such as JPEG or GIF)
- Used for the transport, analysis, and archival storage of scientific data sets
    - Multi-dimensional arrays: 1D spectra, 2D images, 3D+ data cubes
    - Tables containing rows and columns of information
    - Header keywords provide descriptive information about the data
- See also the descriptions on the Wikipedia and Library of Congress Web sites.

http://fits.gsfc.nasa.gov

读取FITS文件

使用Solar software程序包

Website：http://www.lmsal.com/solarsoft/sswdoc/sswdoc_jtop.html

通用程序：readfits

```
Result = READFITS( Filename/Fileunit,[ Header, heap, /NOSCALE, EXTEN_NO=,
                NSLICE=, /SILENT , STARTROW =, NUMROW = , HBUFFER=,
                /CHECKSUM, /COMPRESS, /FPACK, /No_Unsigned, NaNVALUE = ]
```

mreadfits

```
Result = MRDFITS( Filename/FileUnit,[Exten_no/Exten_name, Header],
                /FSCALE , /DSCALE , /UNSIGNED,
                ALIAS=strarr[2,n], /USE_COLNUM,
                /NO_TDIM, ROWS = [a,b,...], $
                /POINTER_VAR, /FIXED_VAR, EXTNUM=
                RANGE=[a,b], COLUMNS=[a,b,...]), ERROR_ACTION=x,
                COMPRESS=comp_prog, STATUS=status, /VERSION )
```

SSW中帮助查看：doc_library 或者xdoc

```
IDL> xdoc,'secchi_images'
```

secchi_images.pro

```
pro secchi_images,start_dt,end_dt,sc,instrument,wavelength,database=database,outmode=outmode,diff=di

;+
;Purpose      Generate/show SECCHI images
;
;Input
;  start_dt    Start date & time, usually in YYYY/MM/DD HH:MM:SS format
;  end_dt      End date & time, usually in YYYY/MM/DD HH:MM:SS format
;  sc          a | b
;  instrument  euvi | cor1 | cor2 | hi_1 | hi_2
;  wavelength  304 | 171 | 195 | 284
;
;Output       Images or mvi movie
;
;Keywords
;  outmode     0 - mvi movie; 1 - JPEG images
;  diff        Time difference (in minutes) between two fits, which are used in generating a
;              difference image; set 0 for direct images; 1 for running difference images regardless
;              of time difference
;  brange      Two-element array, used to specity the minimum and maximum values for rescaling images
;  isize       The size of images
;  color       Use color or not
;  outpath     The path used to store output images
;
;Calls        find_secchi_fits, ymds2datearray, mk_euvi, mk_cor1, mk_cor2, mk_hi1, mk_hi2, make_mvi
;
;History
;  2008-11-06  Created by Yuming Wang (ymwang@ustc.edu.cn)
;-
```

Go To: Line Top Bottom

Find Case sensitive

Cursor Position: line column

Dismiss

常用太阳观测卫星数据读取

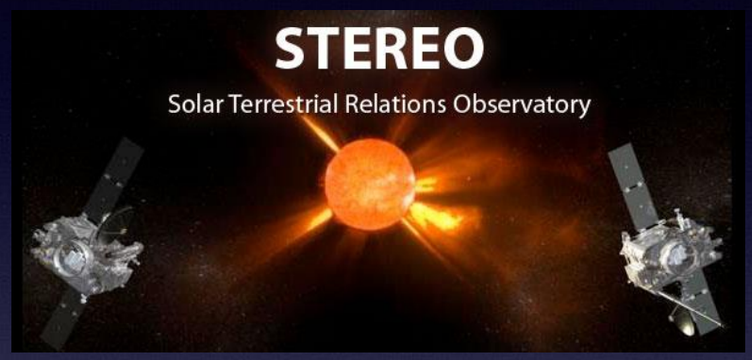
★EIT:

read_eit

eit_prep

★LASCO:

lasco_readfits



★SECCHI:

secchi_prep

★AIA & HMI:

read_sdo

★AIA: aia_prep

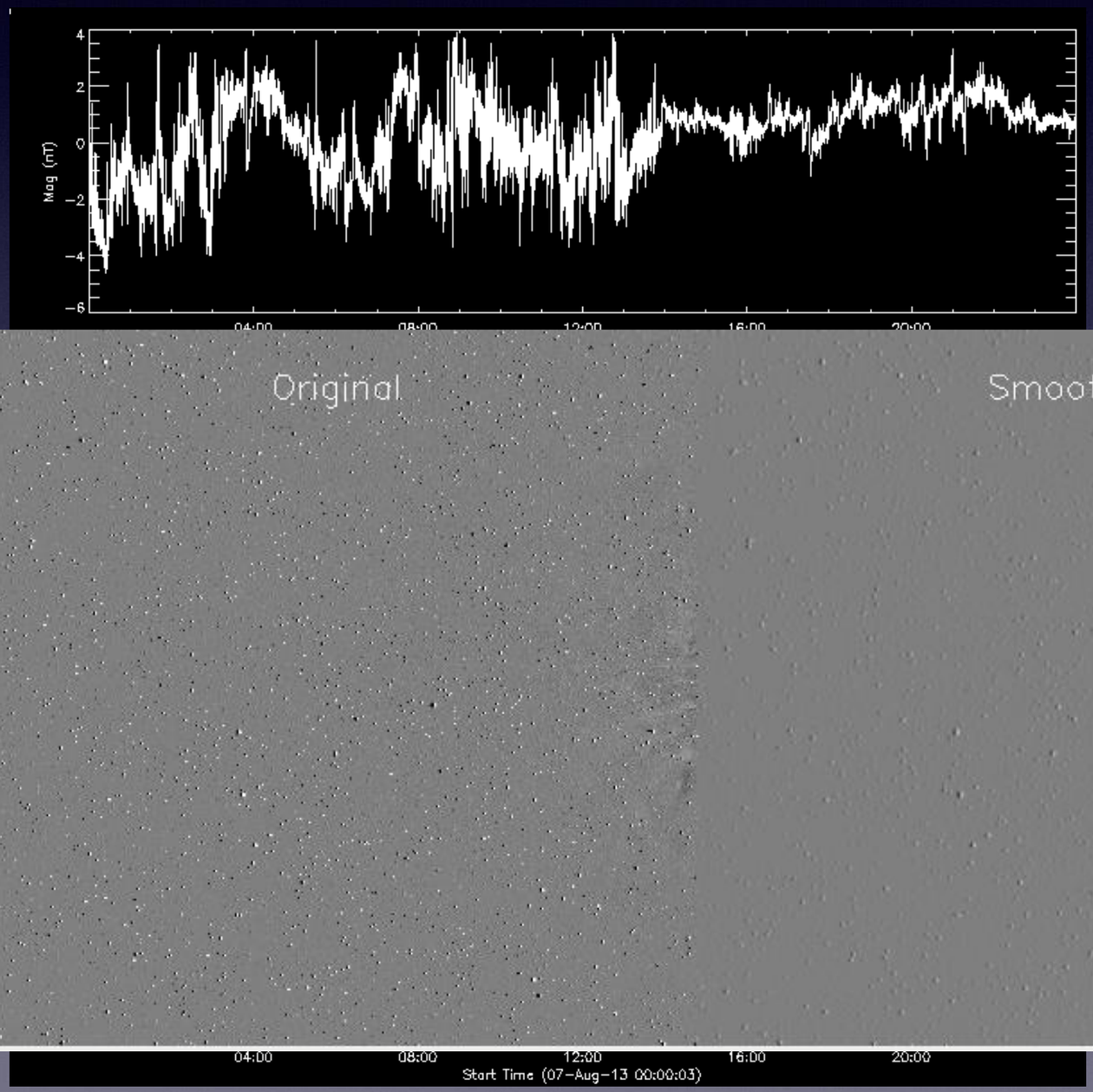
数据处理：数据平滑

smooth

Result = SMOOTH(*Array*, *Width* [, /EDGE_TRUNCATE] [, MISSING=*value*] [, /NAN])

median

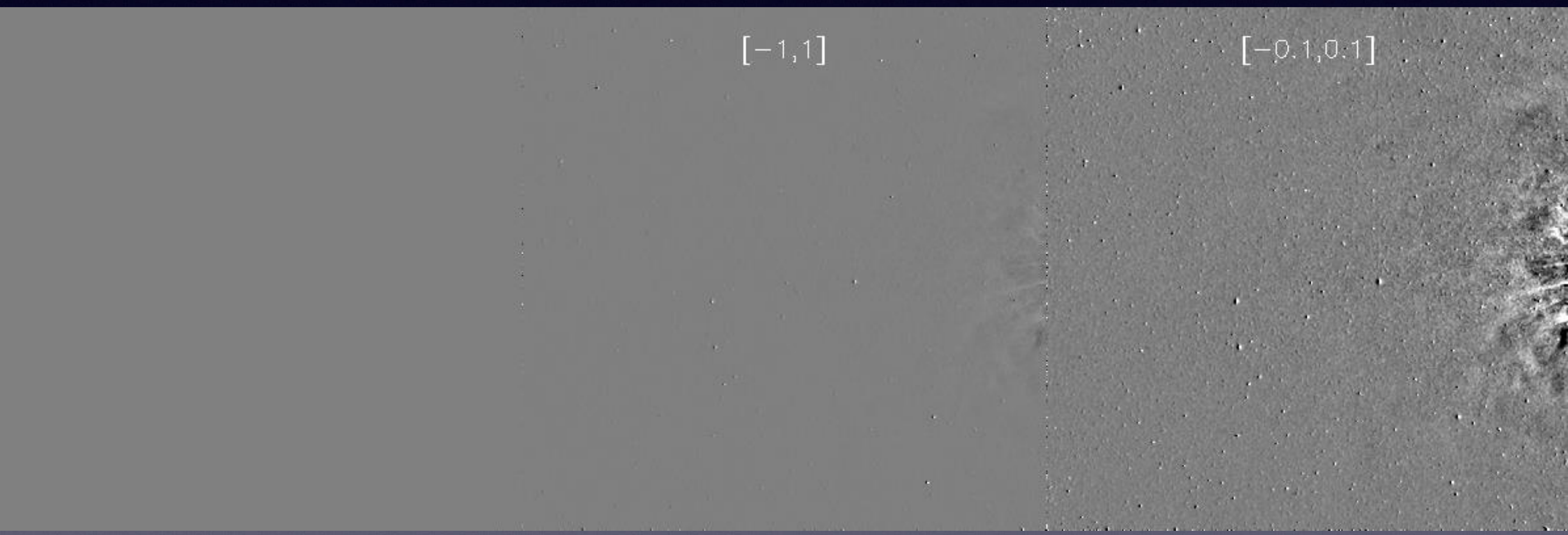
Result = MEDIAN (*Array* [, *Width*] [, /DOUBLE] [, DIMENSION=*value*] [, /EVEN])



图像对比度调节方式

bytscl:

Result = BYTSCL(*Array* [, MAX=*value*] [, MIN=*value*] [, /NAN] [, TOP=*value*])



对数方式，指数方式，其他特殊的算法

图形显示设备

★ set_plot,’设备代码’　　常用的设备代码：win, x, ps

例1：set_plot,‘ps’

!d.name当前设备名

```
IDL> print,!d.name
WIN
```

★ device,……　　设置当前图形显示设备的属性

常用的参数：

decomposed=0|1 设置显示模式为8为伪彩色或24位真彩色

retain=0|1|2 设置备份存储模式，及当前窗口被改变或者被刷新是是否备份。0表示不备份，1表示由系统备份，2表示由IDL备份。（win,x)

★ close /close_file 关闭文件（ps）

例:　dname=!d.name；获得系统的当前设置

set_plot,‘ps’

.

.

.

.

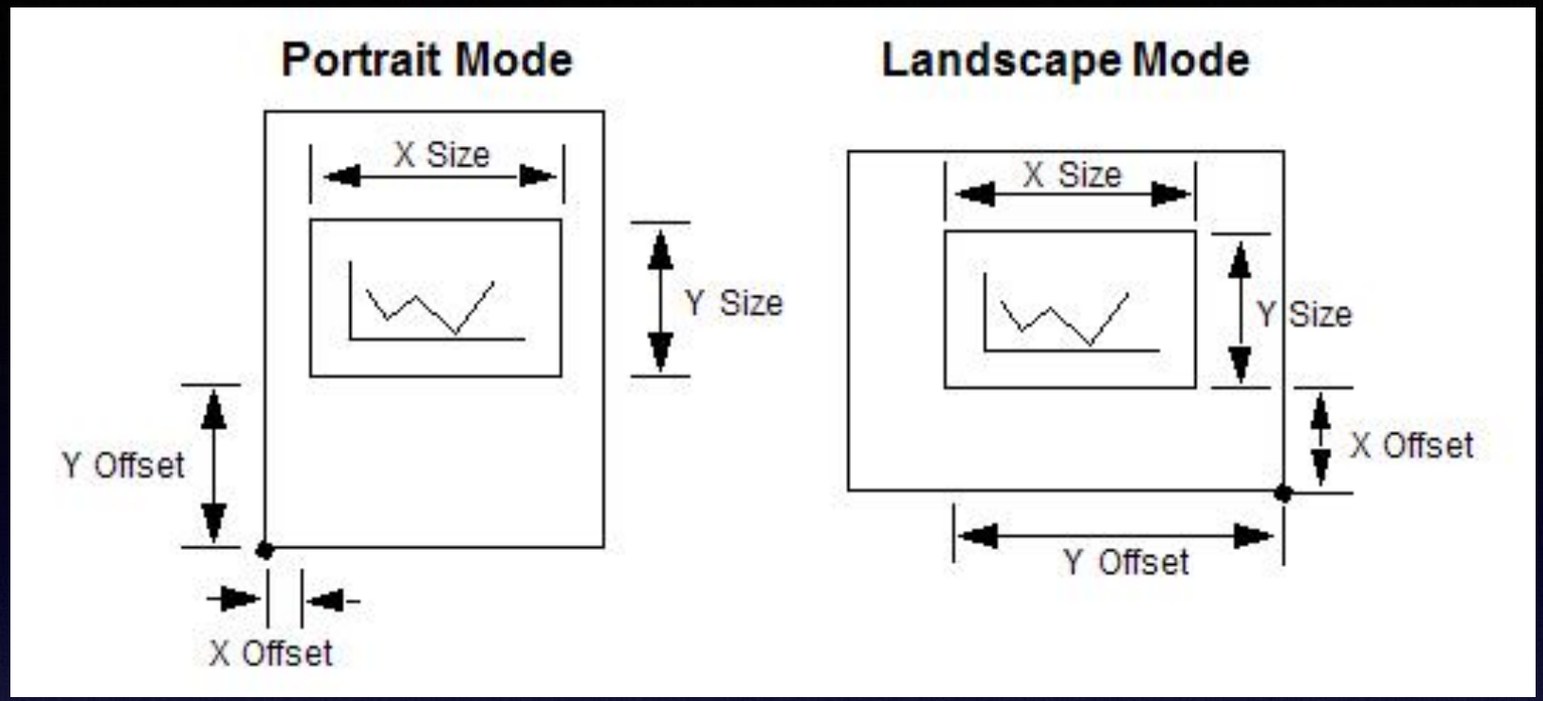
.

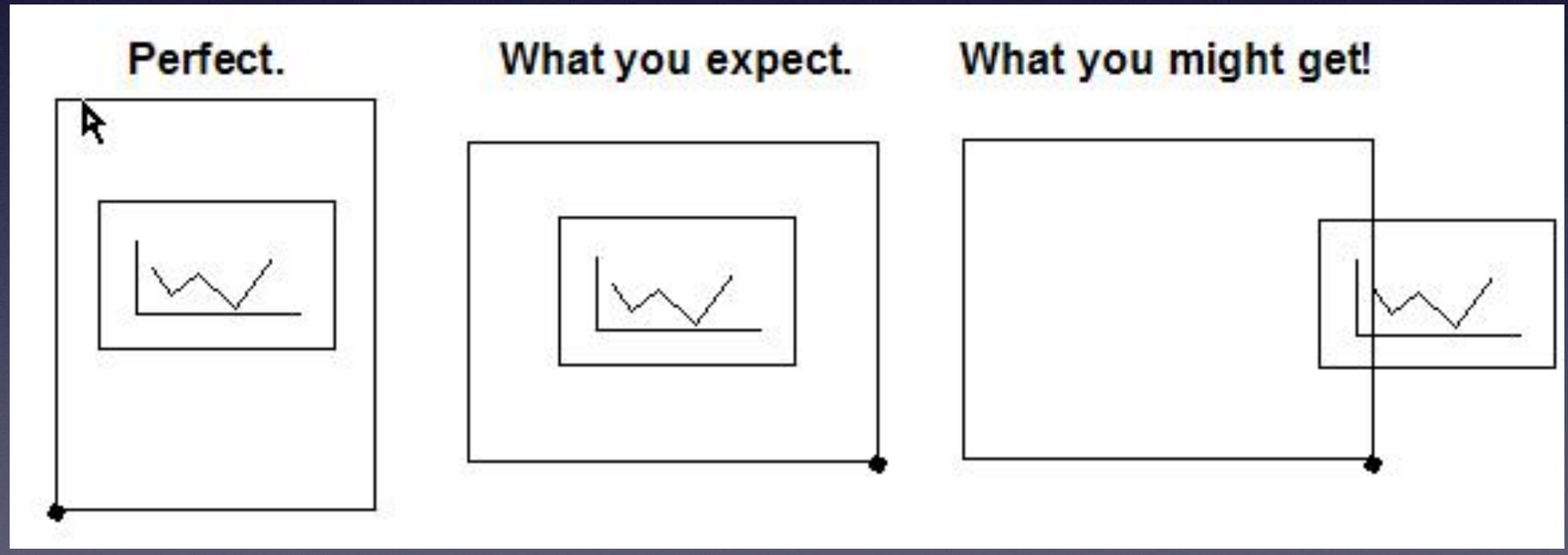
.

.

device,/close

set_plot,dname; 恢复系统设置(重要)





Set_Plot, ‘PS’
Device, XSize=xs, YSize=ys, XOffset=xoff, YOffset=yoff, /Landscape

窗口系统

1. 窗口的建立

window[,窗口索引变量][,retain=0|1|2]
[,title='标题变量'][,xsize=xsize][,ysize=ysize]
[,xpos=xpos][,ypos=ypos][,/free][,/pixmap]

建立一个指定属性的窗口

窗口索引变量:IDL可以允许同时打开128个窗口。

其中用户可以指定索引号1-31，其它索引号需要加参数free建立。当前窗口的索引号存放在系统变量!d.window中。

pixmap: 创建位于内存的不可见窗口

例: set_window.pro

```
IDL> help,!d,/str
** Structure !DEVICE, 17 tags, length=84, data length=84:
   NAME            STRING    'WIN'
   X_SIZE          LONG                640
   Y_SIZE          LONG                512
   X_VSIZE         LONG                640
   Y_VSIZE         LONG                512
   X_CH_SIZE       LONG                  7
   Y_CH_SIZE       LONG                 13
   X_PX_CM         FLOAT           47.3684
   Y_PX_CM         FLOAT           47.3684
   N_COLORS        LONG          16777216
   TABLE_SIZE      LONG               256
   FILL_DIST       LONG                 1
   WINDOW          LONG                -1
   UNIT            LONG                 0
   FLAGS           LONG            328124
   ORIGIN          LONG      Array[2]
   ZOOM            LONG      Array[2]
```

2.wset[,窗口索引号]

把窗口指引变量指定的窗口设置为当前窗口

例：wset,0

3. erase[,背景颜色变量][,color=背景颜色变量]

使用背景颜色变量指定的颜色清楚当前窗口的内容

例：erase

erase,255

4. wshow[,窗口索引变量][,0|1][,/iconic]

显示窗口索引变量指定窗口，即把该窗口置于最前

注意：wshow只是显示指定窗口，并不把该窗口设为当前窗口

例： wshow,1

5.wdelete[,窗口索引变量]

删除窗口索引变量指定的窗口，并释放其所占的内存

例：wdelete,0

!p设置

```
IDL> help,!p,/str
** Structure !PLT, 22 tags, length=296, data length=288:
   BACKGROUND      LONG                 0
   CHARSIZE        FLOAT           0.000000
   CHARTHICK       FLOAT           0.000000
   CLIP            LONG      Array[6]
   COLOR           LONG          16777215
   FONT            LONG                -1
   LINESTYLE       LONG                 0
   MULTI           LONG      Array[5]
   NOCLIP          LONG                 0
   NOERASE         LONG                 0
   NSUM            LONG                 0
   POSITION        FLOAT     Array[4]
   PSYM            LONG                 0
   REGION          FLOAT     Array[4]
   SUBTITLE        STRING    ''
   SYMSIZE         FLOAT           0.000000
   T               DOUBLE    Array[4, 4]
   T3D             LONG                 0
   THICK           FLOAT           0.000000
   TITLE           STRING    ''
   TICKLEN         FLOAT          0.0200000
   CHANNEL         LONG                 0
```

!p.multi[0]：是否擦掉以前的内容
!p.multi[1]：每行显示的个数
!p.multi[2]：每列显示的个数
!p.multi[3]：Z方向显示的个数
!p.multi[4]：画图的顺序

一维图形显示

常用命令：

Plot:

PLOT, [*X*,] *Y* [, /ISOTROPIC] [, MAX_VALUE=*value*] [, MIN_VALUE=*value*] [, NSUM=*value*] [, /POLAR] [, THICK=*value*] [, /XLOG] [, /YLOG] [, /YNOZERO]

Graphics Keywords: [, BACKGROUND=*color_index*] [, CHARSIZE=*value*] [, CHARTHICK=*integer*] [, CLIP=*[X_0, Y_0, X_1, Y_1]*] [, COLOR=*value*] [, /DATA | , /DEVICE | , /NORMAL] [, FONT=*integer*] [, LINESTYLE={0 | 1 | 2 | 3 | 4 | 5}] [, /NOCLIP] [, /NODATA] [, /NOERASE] [, POSITION=*[X_0, Y_0, X_1, Y_1]*] [, PSYM=*integer*{0 to 10}] [, SUBTITLE=*string*] [, SYMSIZE=*value*] [, /T3D] [, THICK=*value*] [, TICKLEN=*value*] [, TITLE=*string*]
[, {X | Y | Z}CHARSIZE=*value*]
[, {X | Y | Z}GRIDSTYLE=*integer*{0 to 5}]
[, {X | Y | Z}MARGIN=*[left, right]*]
[, {X | Y | Z}MINOR=*integer*]
[, {X | Y | Z}RANGE=*[min, max]*]
[, {X | Y | Z}STYLE=*value*]
[, {X | Y | Z}THICK=*value*]
[, {X | Y | Z}TICK_GET=*variable*]
[, {X | Y | Z}TICKFORMAT=*string*]
[, {X | Y | Z}TICKINTERVAL= *value*]
[, {X | Y | Z}TICKLAYOUT=*scalar*]
[, {X | Y | Z}TICKLEN=*value*]
[, {X | Y | Z}TICKNAME=*string_array*]
[, {X | Y | Z}TICKS=*integer*]
[, {X | Y | Z}TICKUNITS=*string*]
[, {X | Y | Z}TICKV=*array*]
[, {X | Y | Z}TITLE=*string*]
[, ZVALUE=*value*{0 to 1}]

Plots:

PLOTS, *X* [, *Y* [, *Z*]] [, /CONTINUE]

Graphics Keywords: [, CLIP=*[X_0, Y_0, X_1, Y_1]*] [, COLOR=*value*] [, /DATA | , /DEVICE | , /NORMAL] [, LINESTYLE={0 | 1 | 2 | 3 | 4 | 5}] [, /NOCLIP] [, PSYM=*integer*{0 to 10}] [, SYMSIZE=*value*] [, /T3D] [, THICK=*value*] [, *Z*=*value*]

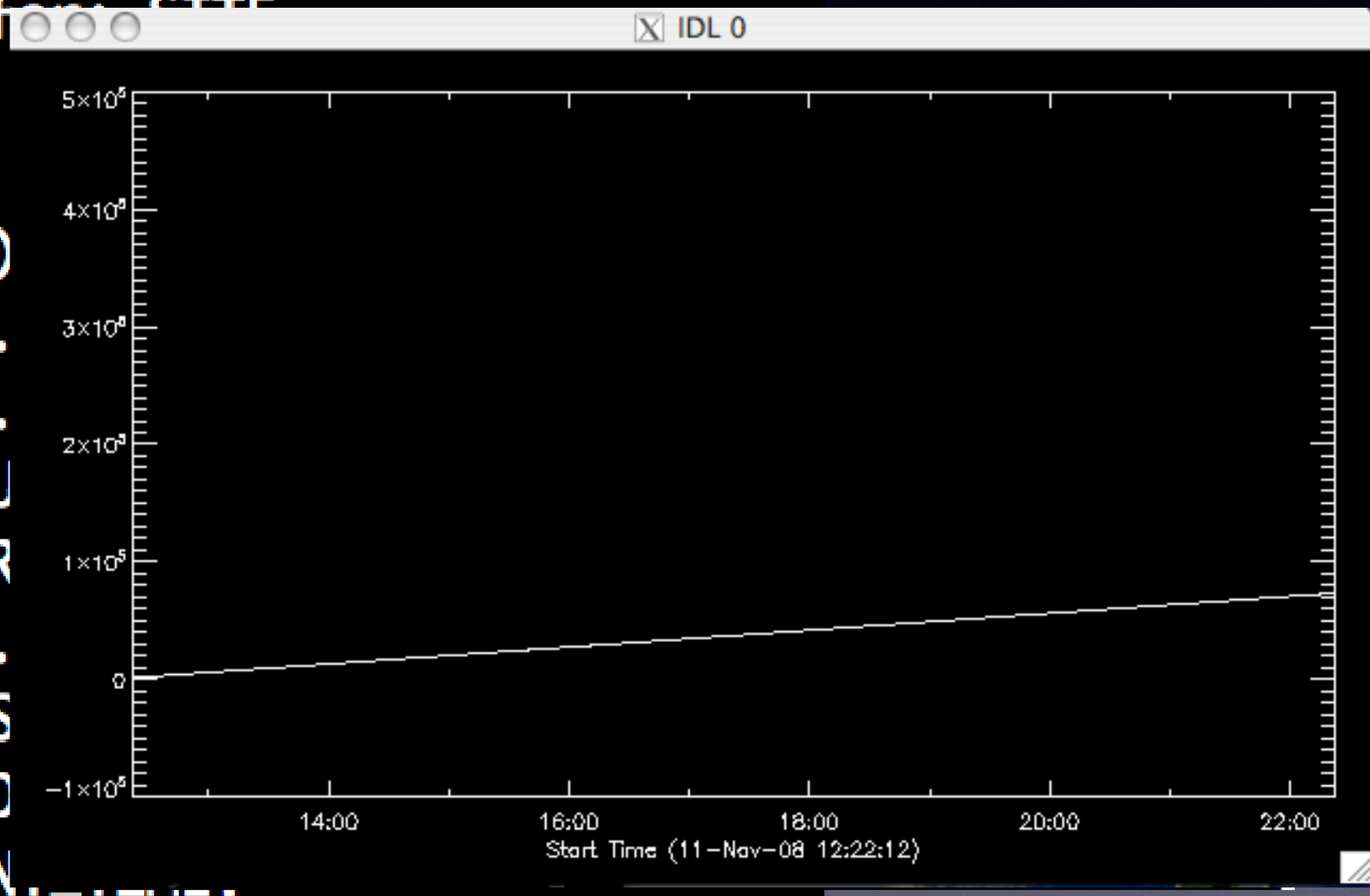
Utplot:

```
  Plot X vs Y with Universal time labels on bottom X axis.
CALLING SEQUENCE:
  UTPLOT,X,Y,BASE_TIME( or Utstring or Xst0), $
    TIMERANGE=TIMERANGE, LABELPAR=LBL, /SAV,TICK_UNIT=TICK_UNIT,NTICKS=NTICKS, $
    MINORS=MINORS, /NOLABEL, ERROR=ERROR,$
              [& ALL KEYWORDS AVAILABLE TO PLOT]
      UTPLOT, roadmap, y
      utplot, x, y, '1-sep-91'
      utplot, x, y, '1-sep-91', timerange=['1-sep-91', '2-sep-92']
      utplot, x, y, '1-sep-91', timerange=[index(0), index(i)], xstyle=1
```

Default

Utplot ($SSW/gen/idl/time)
Plot X vs Y with Universal time labels on bottom X axis
使用方法:
Utplot,x,y,basetime

```
Executing SSW IDL_STARTUP for: MDI
Executing SSW IDL_STARTUP for: CDS
Executing SSW IDL_STARTUP for: SITE
IDL> x=findgen(60*60L)*60
IDL> basetime='20008/11/11
IDL> btai=utc2tai(basetime)
% Compiled module: UTC2TAI.
% Compiled module: STR2UTC.
% Compiled module: VALID_NU
% Compiled module: BOOST_AR
% Compiled module: UTC2INT.
% Compiled module: TAG_EXIS
% Compiled module: DATE2MJD
% Compiled module: CHECK_IN.
```




```
IDL> t1='20008/11/11 12:22:12'
IDL> t2='20008/11/11 22:22:12'
IDL> utplot,x,y,basetime,timerange=[t1,t2],xstyle=1
```

关于颜色的程序库

fsc_color: 获取指定颜色的代码（88种颜色）

使用方法： plot,[1,2,3],color=fsc_color(‘red’)

例：fsc_color_example.pro

D Fanning的程序库

http://www.dfanning.com/

$ssw/packages/nrl/id/nrlgen/dfanning

Almond	Antique White	Aquamarine	Beige	Bisque	
Blue	Blue Violet	Brown	Burlywood	Charcoal	Chartreuse
Chocolate	Coral	Cornsilk	Cyan	Dark Goldenrod	Dark Gray
Dark Green	Dark Khaki	Dark Orchid	Dark Salmon	Deep Pink	Dodger Blue
Firebrick	Forest Green	Gold	Goldenrod	Gray	Green
Green Yellow	Honeydew	Hot Pink	Indian Red	Ivory	Khaki
Lavender	Lawn Green	Light Coral	Light Cyan	Light Gray	Light Salmon
Light Yellow	Lime Green	Linen	Magenta	Maroon	Medium Gray
Medium Orchid	Moccasin	Navy	Olive	Olive Drab	Orange
Orange Red	Orchid	Pale Goldenrod	Pale Green	Papaya	Peru
Pink	Plum	Powder Blue	Purple	Red	Rose
Rosy Brown	Royal Blue	Saddle Brown	Salmon	Sandy Brown	Sea Green
Seashell	Sienna	Sky Blue	Slate Gray	Snow	Spring Green
Steel Blue	Tan	Thistle	Tomato	Turquoise	Violet
Violet Red	Wheat	White	Yellow		

二维数据显示

TV

TV, *Image* [, *Position*] [, /CENTIMETERS | , /INCHES] [, /ORDER] [, TRUE={1 | 2 | 3}] [, /WORDS] [, XSIZE=*value*] [, YSIZE=*value*]

or

TV, *Image* [, *X*, *Y* [, *Channel*]] [, /CENTIMETERS | , /INCHES] [, /ORDER] [, TRUE={1 | 2 | 3}] [, /WORDS] [, XSIZE=*value*] [, YSIZE=*value*]

Graphics Keywords: [, CHANNEL=*value*] [, /DATA | , /DEVICE | , /NORMAL] [, /T3D] [, Z=*value*]

TVSCL

TVSCL, *Image* [, *Position*] [, /CENTIMETERS] [, /INCHES] [, /NAN] [, /ORDER] [, TOP=*value*] [, TRUE={1 | 2 | 3}] [, /WORDS] [, XSIZE=*value*] [, YSIZE=*value*]

or

TVSCL, *Image* [, *X*, *Y* [, *Channel*] [, /CENTIMETERS] [, /INCHES] [, /NAN] [, /ORDER] [, TOP=*value*] [, TRUE={1 | 2 | 3}] [, /WORDS] [, XSIZE=*value*] [, YSIZE=*value*]]

Graphics Keywords: [, CHANNEL=*value*] [, /DATA | , /DEVICE | , /NORMAL] [, /T3D] [, Z=*value*]

Contour：

CONTOUR, *Z* [, *X*, *Y*] [, C_ANNOTATION=*vector_of_strings*] [, C_CHARSIZE=*value*] [, C_CHARTHICK=*integer*] [, C_COLORS=*vector*] [, C_LABELS=*vector*{each element 0 or 1}] [, C_LINESTYLE=*vector*] [, C_ORIENTATION=*degrees*] [, C_SPACING=*value*] [, C_THICK=*vector*] [, /CELL_FILL | , /FILL] [, /CLOSED] [, /DOWNHILL] [, /FOLLOW] [, /IRREGULAR] [, /ISOTROPIC] [, LEVELS=*vector*] [, NLEVELS=*integer*{1 to 60}] [, MAX_VALUE=*value*] [, MIN_VALUE=*value*] [, /OVERPLOT] [{, /PATH_DATA_COORDS, PATH_FILENAME=*string*, PATH_INFO=*variable*, PATH_XY=*variable*} | , TRIANGULATION=*variable*] [, /PATH_DOUBLE] [, /XLOG] [, /YLOG] [, ZAXIS={0 | 1 | 2 | 3 | 4}]

Graphics Keywords: Accepts all graphics keywords accepted by PLOT except for: LINESTYLE, PSYM, SYMSIZE. See Graphics Keywords Accepted.

- 画数据图像
- 选取数据中符合某些条件的部分

缺点：图片文件很大

Contour,Z[,X,Y]

常用的参数：

- ★ C_annotation:设置轮廓线上的文本标注内容
- ★ Levels:设置每一层轮廓所对应的值的数组
- ★ Nlevels:设置绘制轮廓线的层数（最大值60）
- ★ 如果设置了levels,改值无效
- ★ /fill:使用填充的方式绘制轮廓
- ★ Path_xy:获得等值线的坐标
- ★ Path_info:获得等值线的路径信息

例：example_contour.pro

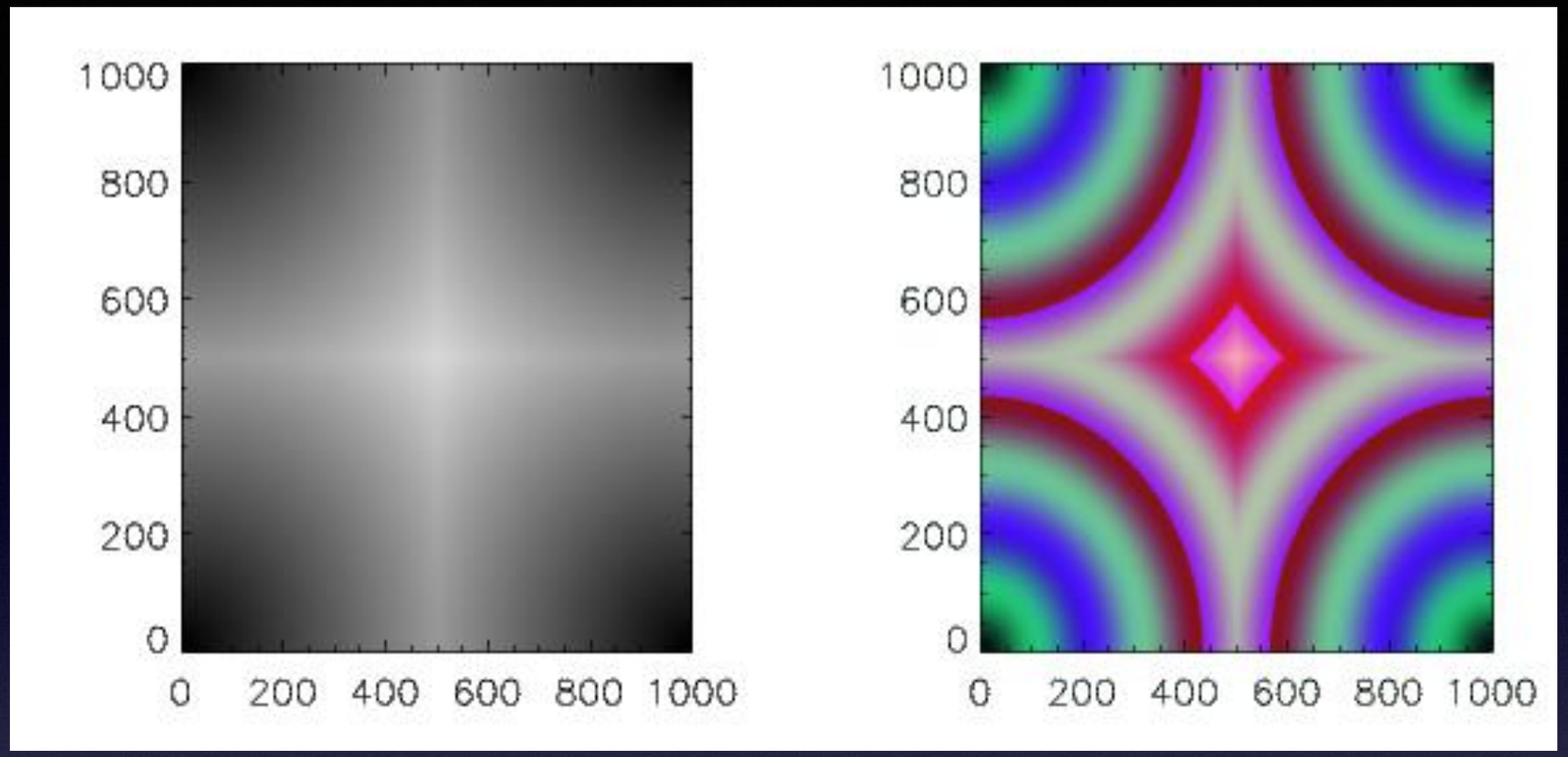
Shade_surf

SHADE_SURF, *Z* [, *X*, *Y*] [, AX=*degrees*] [, AZ=*degrees*] [, IMAGE=*variable*] [, MAX_VALUE=*value*] [, MIN_VALUE=*value*] [, PIXELS=*pixels*] [, /SAVE] [, SHADES=*array*] [, /XLOG] [, /YLOG]

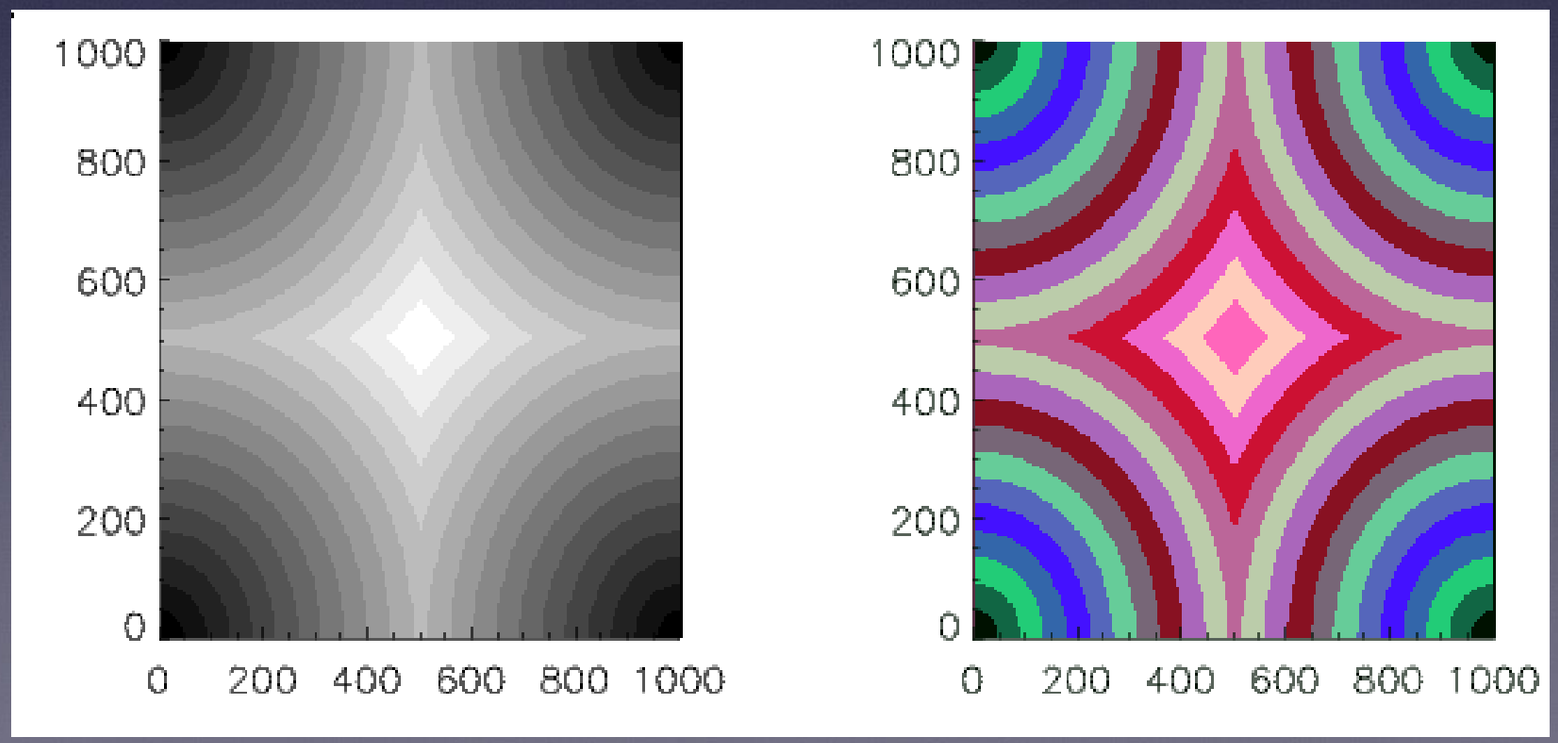
Graphics Keywords: [, CHARSIZE=*value*] [, CHARTHICK=*integer*] [, COLOR=*value*][, /DATA | , /DEVICE | , /NORMAL] [, FONT=*integer*] [, /NODATA] [, POSITION=*[X_0, Y_0, X_1, Y_1]*] [, SUBTITLE=*string*] [, /T3D]
[, THICK=*value*] [, TICKLEN=*value*] [, TITLE=*string*]
[, {X | Y | Z}CHARSIZE=*value*]
[, {X | Y | Z}GRIDSTYLE=*integer*{0 to 5}]
[, {X | Y | Z}MARGIN=*[left, right]*]
[, {X | Y | Z}MINOR=*integer*]
[, {X | Y | Z}RANGE=*[min, max]*]
[, {X | Y | Z}STYLE=*value*]
[, {X | Y | Z}THICK=*value*]
[, {X | Y | Z}TICKFORMAT=*string*]
[, {X | Y | Z}TICKINTERVAL= *value*]
[, {X | Y | Z}TICKLAYOUT=*scalar*]
[, {X | Y | Z}TICKLEN=*value*]
[, {X | Y | Z}TICKNAME=*string_array*]
[, {X | Y | Z}TICKS=*integer*]
[, {X | Y | Z}TICKUNITS=*string*]
[, {X | Y | Z}TICKV=*array*]
[, {X | Y | Z}TICK_GET=*variable*]
[, {X | Y | Z}TITLE=*string*]
[, ZVALUE=*value*{0 to 1}]

```
pro compare_contour_shadesurf
nn=1001
a=dist(nn)
x=findgen(nn)
y=findgen(nn)
image=bytscl(a)
set_plot,'ps'
device,filename='compare.ps',xs=16,ys=8,/color
!p.multi=[0,2,1]
contour,image,x,y,nlevels=255,/fill
loadct,32
contour,image,x,y,nlevels=255,/fill
loadct,0,/silent

shade_surf,a,x,y,ax=90,az=0,xstyle=1,ystyle=1,zstyle=5,shades=image
loadct,32
shade_surf,a,x,y,ax=90,az=0,xstyle=1,ystyle=1,zstyle=5,shades=image
loadct,0

device,/close
end
```

Contour



Shade_surf

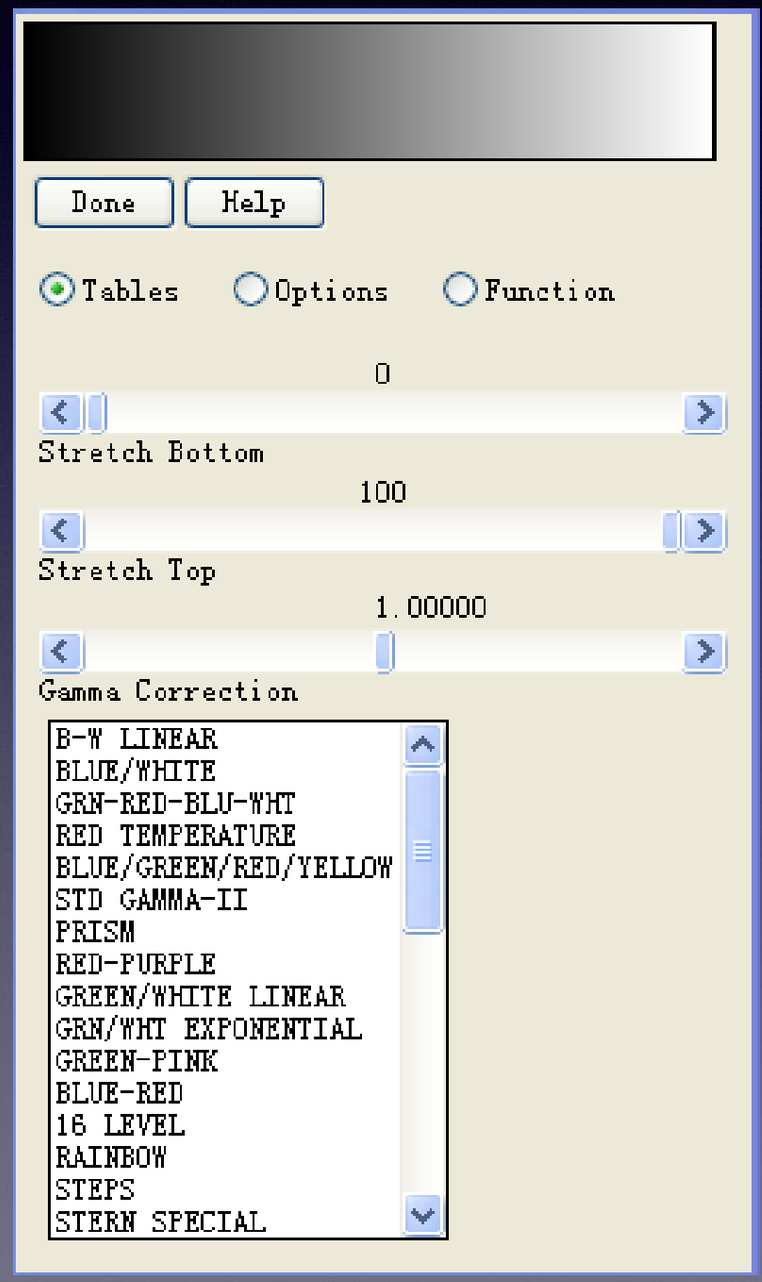


装入颜色表

loadct ,[颜色表索引|变量]

装入制定颜色表文件的制定颜色表

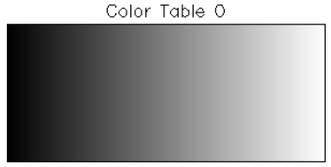
```
IDL> loadct,3
% LOADCT: Loading table RED TEMPERATURE
IDL>
IDL> loadct,3,/silent
```



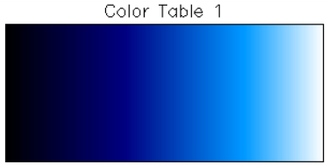
xloadct:以交互方式装入或修改颜色表文件中的指定颜色表

默认颜色表

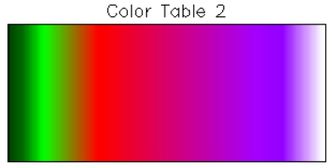
Color Table 0



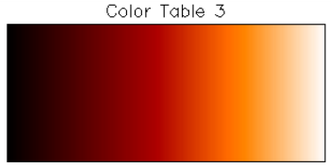
Color Table 1



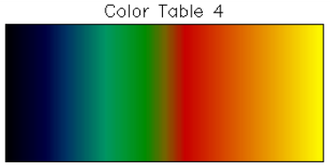
Color Table 2



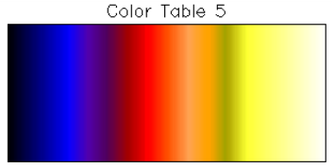
Color Table 3



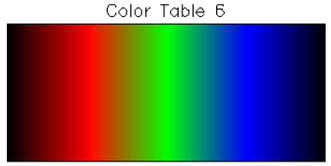
Color Table 4



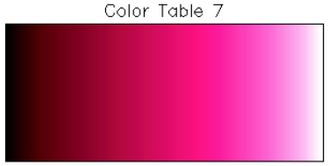
Color Table 5



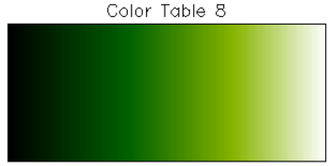
Color Table 6



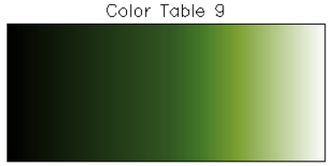
Color Table 7



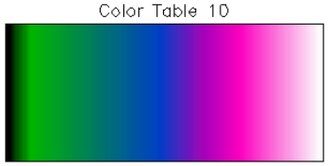
Color Table 8



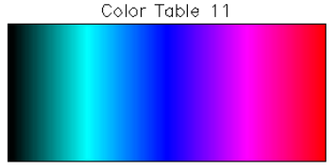
Color Table 9



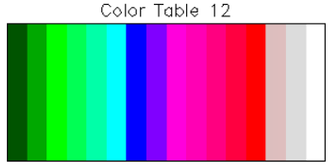
Color Table 10



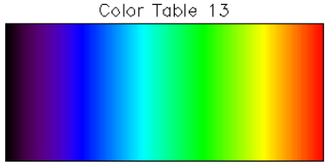
Color Table 11



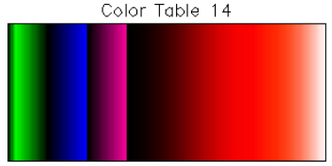
Color Table 12



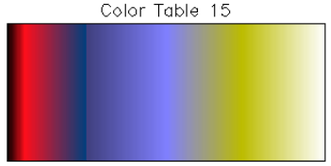
Color Table 13



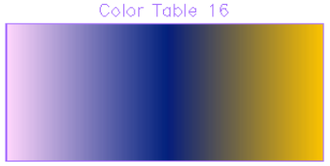
Color Table 14



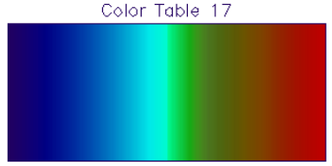
Color Table 15



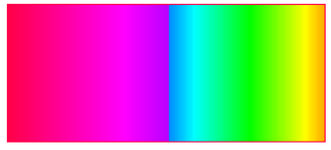
Color Table 16



Color Table 17



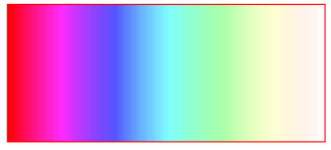
Color Table 18



Color Table 19

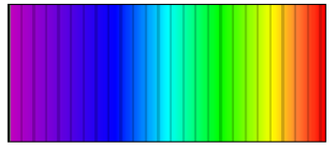
Color Table 20

Color Table 21



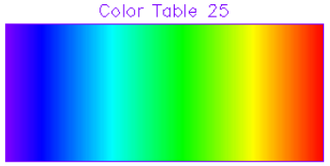
Color Table 22

Color Table 23

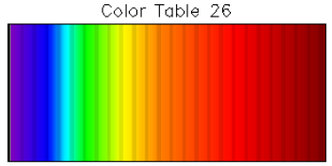


Color Table 24

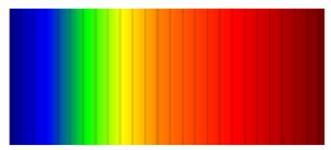
Color Table 25



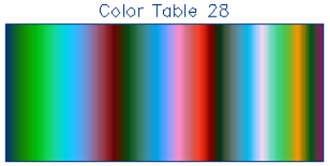
Color Table 26



Color Table 27

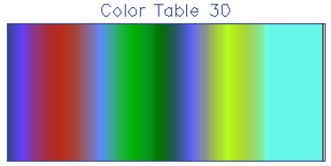


Color Table 28

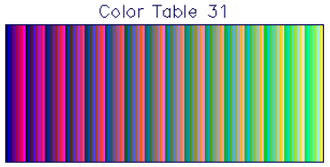


Color Table 29

Color Table 30



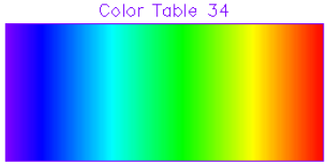
Color Table 31



Color Table 32

Color Table 33

Color Table 34

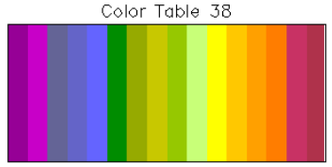


Color Table 35

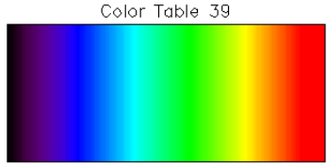
Color Table 36

Color Table 37

Color Table 38



Color Table 39



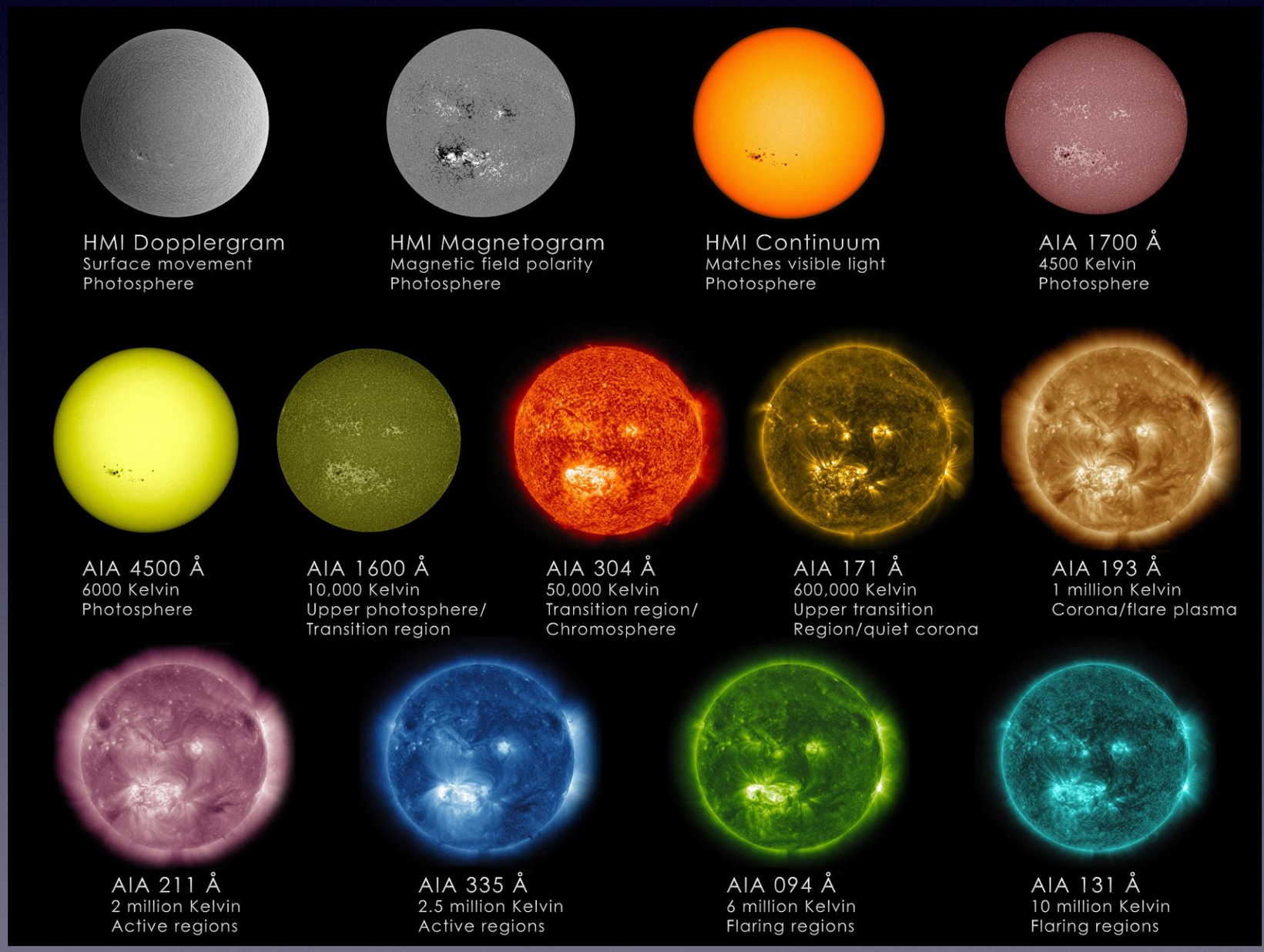
Color Table 40

20	Hue Sat Lightness 2

常用观测数据的颜色表

SDO/AIA: aia_lct

```
pro aia_lct,r,g,b,wavelnth=wavelnth,load=load
```



常用观测数据的颜色表

STEREO/SECCHI: secchi_colors

```
Project     :  STEREO - SECCHI

Name        :  SECCHI_COLORS

Purpose     :  Load appropriate SECCHI color table

Category    :  STEREO, SECCHI, Color-table

Explanation :  Restores the appropriate IDL save file containing the color
               table for the selected telescope/wavelength combination.

Syntax      :  SECCHI_COLORS, TELESCOPE, WAVELENGTH, RED, GREEN, BLUE

Examples    :  SECCHI_COLORS, 'EUVI', 195, R, G, B
               SECCHI_COLORS, 'COR1', 0,   R, G, B
               SECCHI_COLORS, 'COR2', /LOAD
```

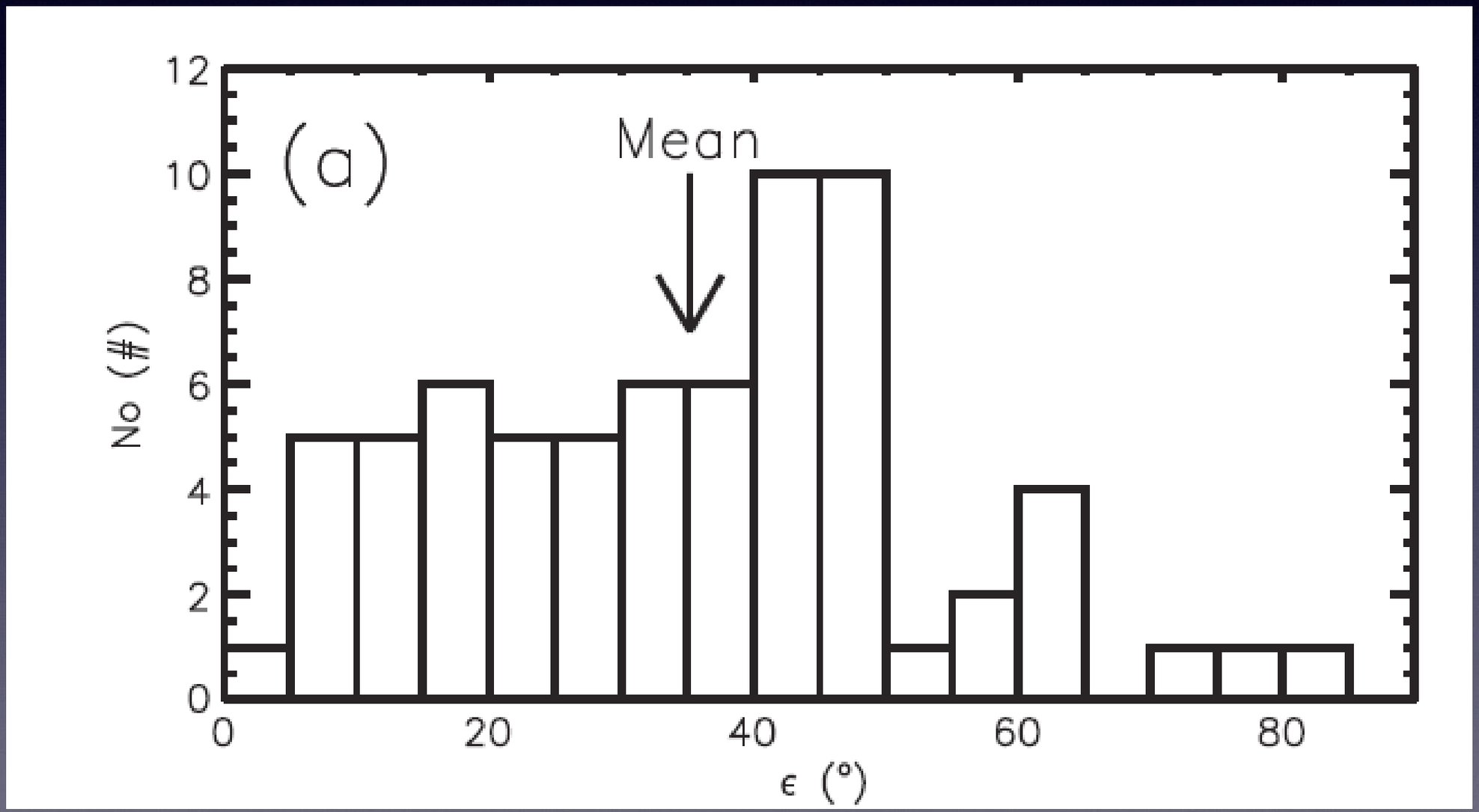
SOHO/EIT: eit_colors

```
Name: eit_colors

Purpose: load EIT color tables

Input Parameters:
   table - EIT table number or associated wavelength

Output Parameters:
   r,g,b - optional return RGB

Calling Examples:
   eit_colors,0                ; load the first EIT table (relative)
   eit_colors,41               ; load the first EIT table (absolute)
   eit_colors,284,r,g,b        ; map wavelenght-> table, return RGB
```

柱状图

histogram

Result = HISTOGRAM(*Array* [, BINSIZE=*value*] [, INPUT=*variable*] [, LOCATIONS=*variable*] [, MAX=*value*] [, MIN=*value*] [, /NAN] [, NBINS=*value*] [, OMAX=*variable*] [, OMIN=*variable*] [, /L64 | REVERSE_INDICES=*variable*])

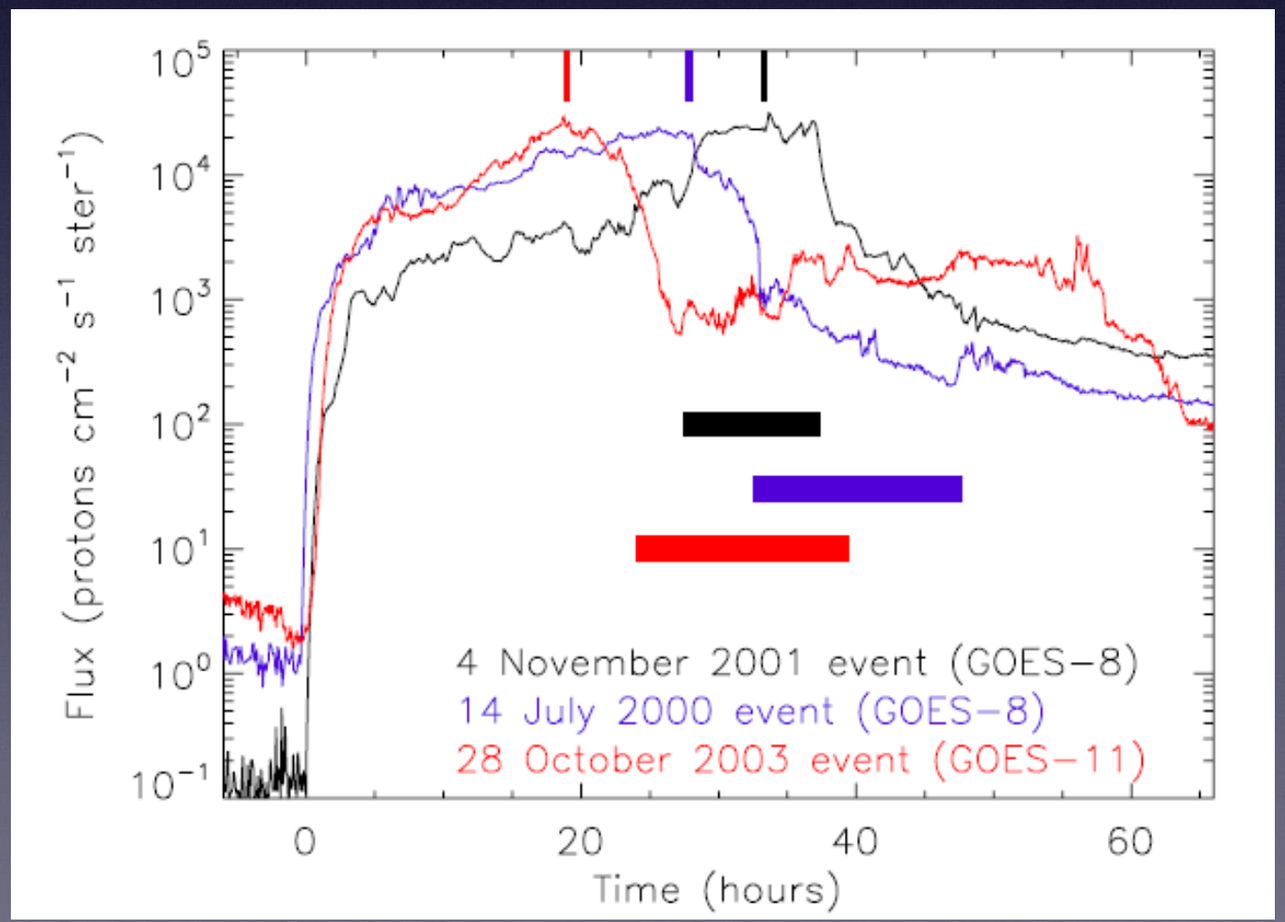


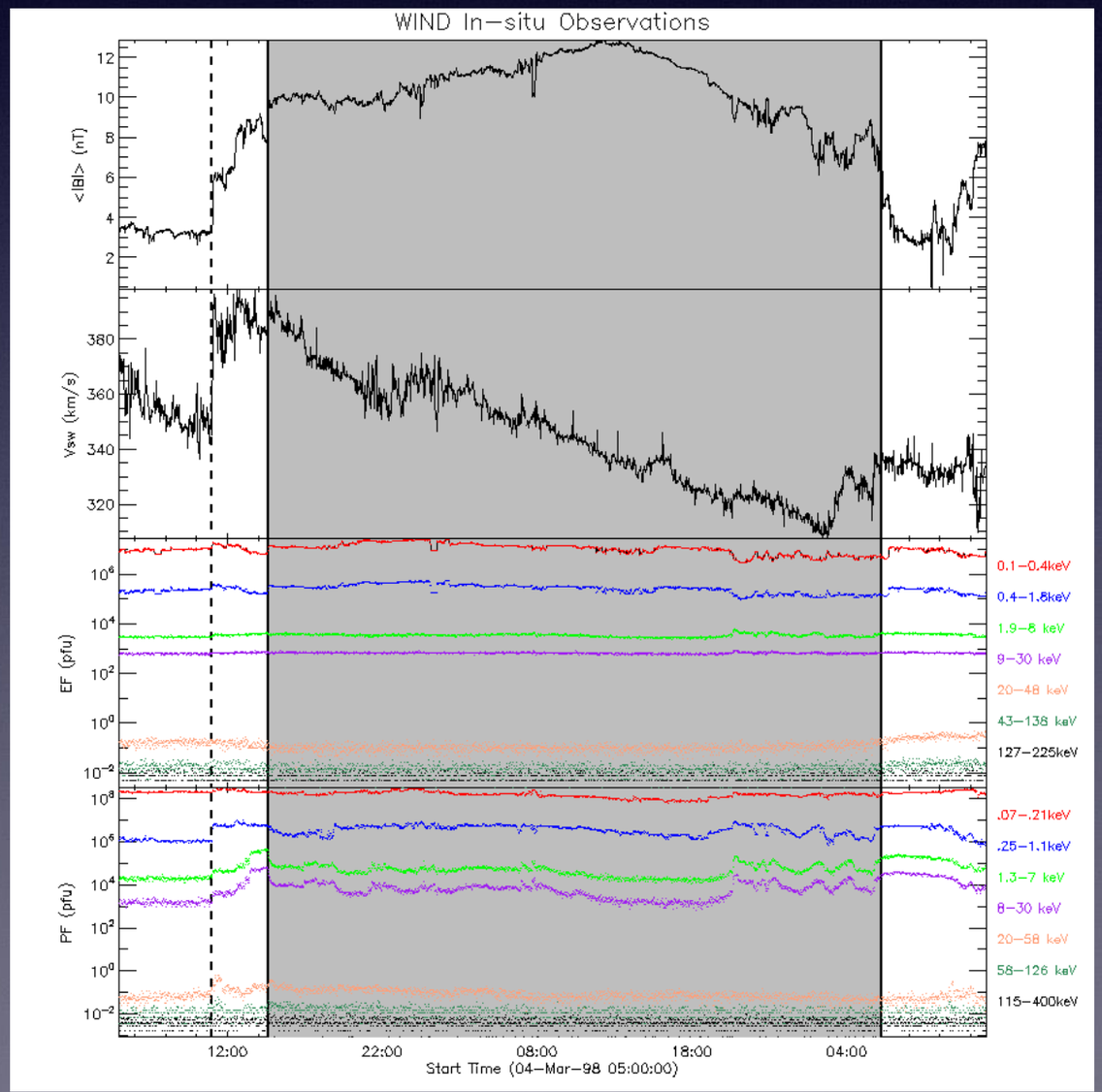
画阴影

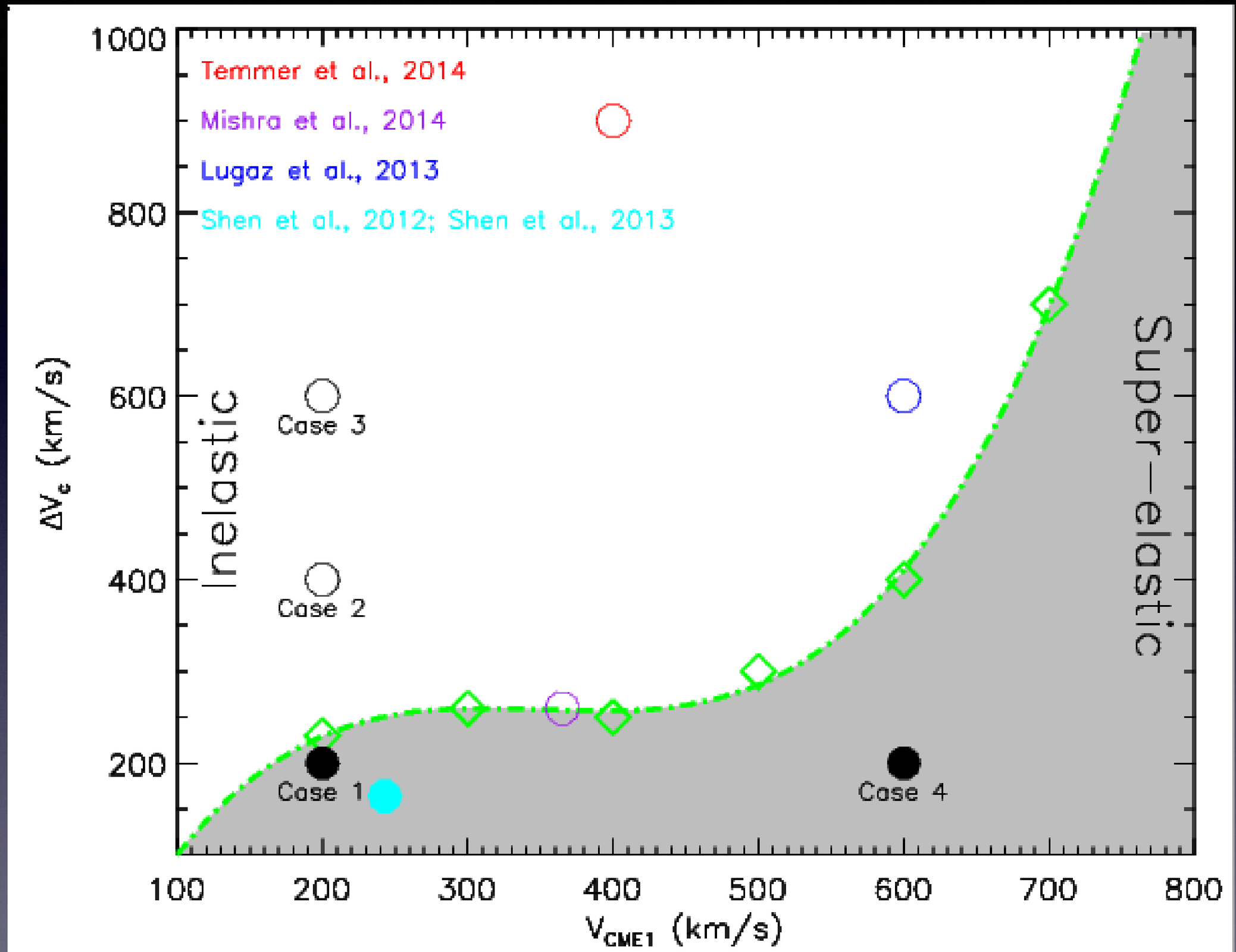
Polyfill

POLYFILL, *X* [, *Y* [, *Z*]] [, IMAGE_COORD=*array*] [, /IMAGE_INTERP] [, /LINE_FILL] [, PATTERN=*array*] [, SPACING=*centimeters*] [, TRANSPARENT=*value*]

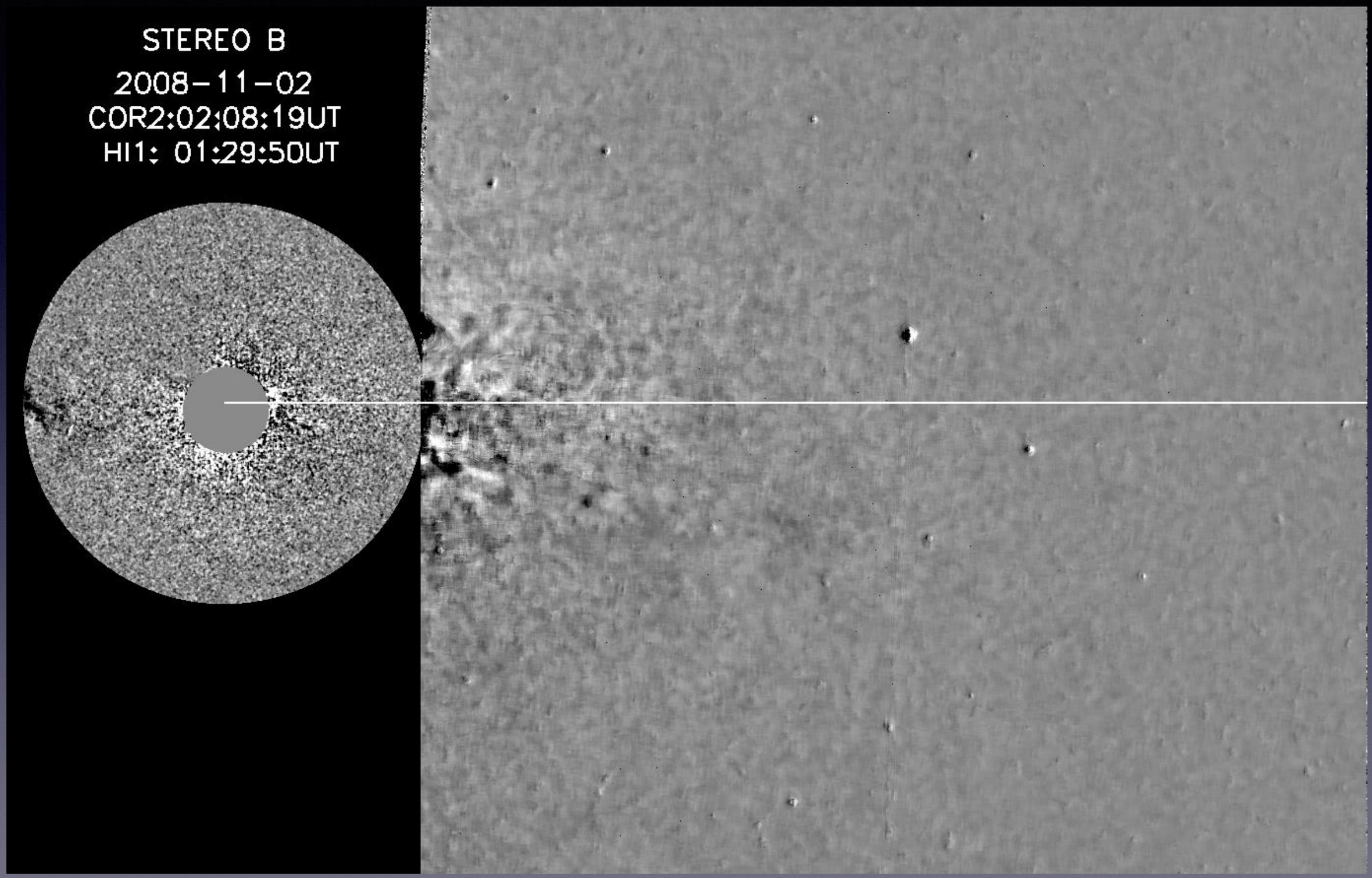
Graphics Keywords: [, CLIP=[X_0, Y_0, X_1, Y_1]] [, COLOR=*value*] [, /DATA | , /DEVICE | , /NORMAL] [, LINESTYLE={0 | 1 | 2 | 3 | 4 | 5}] [, /NOCLIP] [, ORIENTATION=*ccw_degrees_from_horiz*] [, /T3D] [, THICK=*value*] [, Z=*value*]





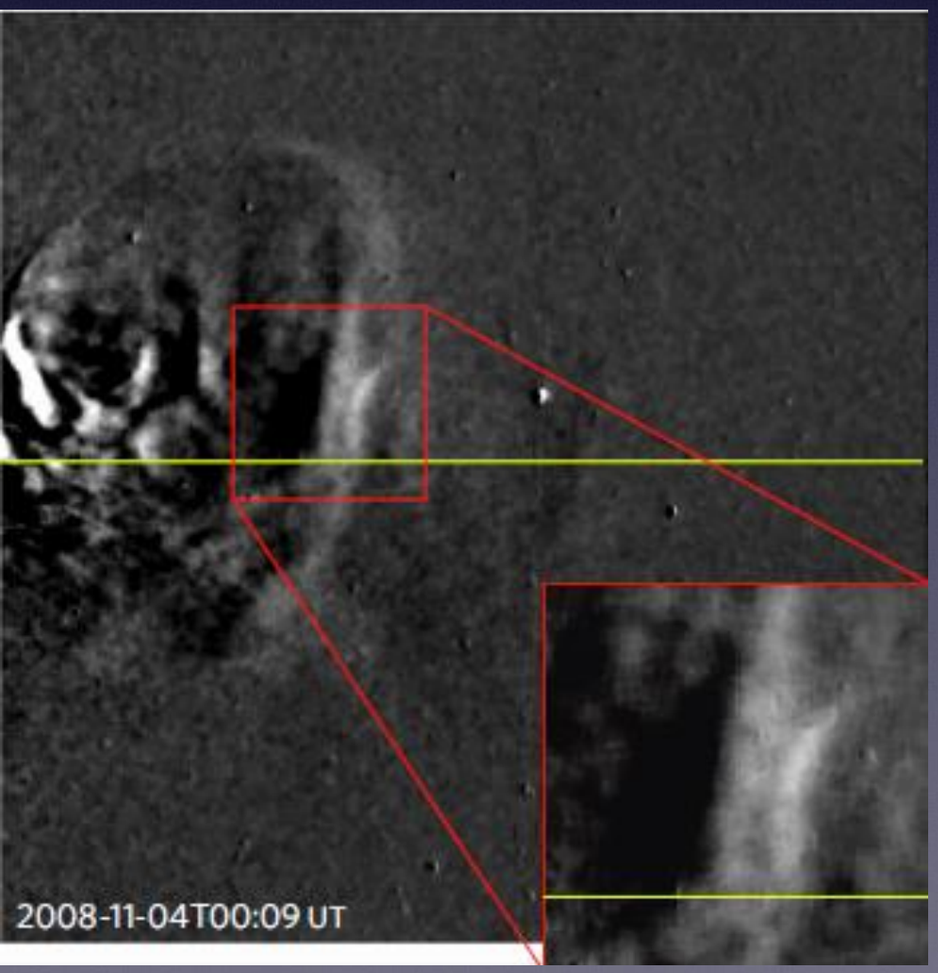

Temmer et al., 2014
Mishra et al., 2014
Lugaz et al., 2013
Shen et al., 2012; Shen et al., 2013
Inelastic
Super-elastic
Case 1
Case 2
Case 3
Case 4
ΔV_e (km/s)
V_{CME1} (km/s)
1000
800
600
400
200
100
200
300
400
500
600
700
800

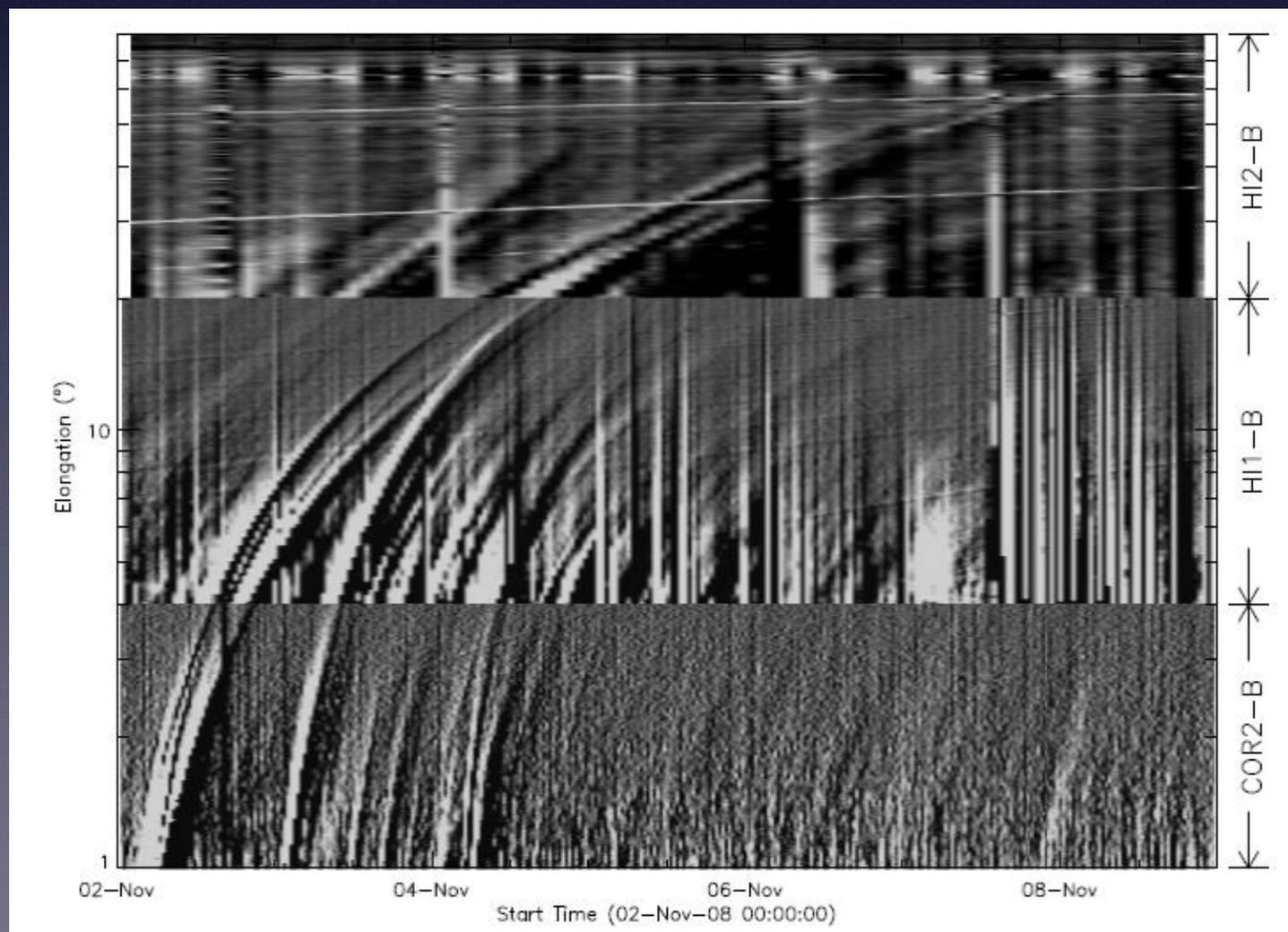
Jmap [时间高度图像]



Jmap [时间高度图像]

读取单一时刻数据 → 选取其中一条slice，得到一个一维数组 → 不同时刻的一维数据合成到一起





IDL中的不同字体和字体属性

!C:开始新的一行
!U:上标开始标志
!D: 下标开始标志
!N:恢复正常字体

矢量字体

Font 3, 3

Octal	00	01	02	03	04	05	06	07	10	11	12	13	14	15	16	17
04x		!	"	#	$	%	&	'	(	)	*	+	,	–	.	/
06x	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
10x	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
12x	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
14x	'	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
16x	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	
20x																
22x																
24x																
26x																
30x																
32x																
34x																
36x																ÿ

Font 4, 4

Octal	00	01	02	03	04	05	06	07	10	11	12	13	14	15	16	17
04x		!	"	#	$	%	&	'	(	)	*	+	,	–	.	/
06x	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
10x	@	A	B	Γ	Δ	E	Z	H	Θ	I	K	Λ	M	N	Ξ	O
12x	Π	P	Σ	T	ϒ	Φ	X	Ψ	Ω	∞	ℓ	[	\	]	^	_
14x	'	α	β	γ	δ	ϵ	ζ	η	θ	ι	κ	λ	μ	ν	ξ	o
16x	π	ρ	σ	τ	υ	ϕ	χ	ψ	ω	∞	ℓ	[	\	]	^	_

例：showfont.pro

鼠标交互

cursor

Syntax

CURSOR, *X*, *Y* [, *Wait* | [, /CHANGE | , /DOWN | , /NOWAIT | , /UP | , /WAIT]] [, /DATA | , /DEVICE, | , /NORMAL]

Wait Value	Corresponding Keyword	Action
0	NOWAIT	Return immediately.
1	WAIT	Return if a button is down.
2	CHANGE	Return if a button is pressed, released, or the pointer is moved.
3	DOWN	Return when a button down transition is detected.
4	UP	Return when a button up transition is detected.

```
IDL> plot,findgen(100)
IDL> cursor,x,y
IDL> print,x,y
      27.702125       48.038586
```

!MOUSE

A structure variable that contains the status from the last cursor read operation. !MOUSE has the following fields:

```
** Structure !MOUSE, 4 tags, length=16:
   X               LONG                511
   Y               LONG                252
   BUTTON          LONG                  4
   TIME            LONG         1428829775
```

where the meaning of the fields a are described in the following sections.

X

Contains the X location (in device coordinates) of the cursor when the mouse button was pressed.

Y

Contains the Y location (in device coordinates) of the cursor when the mouse button was pressed.

BUTTON

Contains:

- 1 (one) if the left mouse button was pressed,
- 2 (two) if the middle mouse button was pressed
- 4 (four) if the right mouse button was pressed.

TIME

Contains the number of milliseconds since a base time.

与时间有关的函数($SSW/gen/idl/time)

时间格式 utc （Coordinated Universal Time，世界协调时）
tai （International Atomic Time, 国际原子时）
具体可以查看$SSW/gen/idl/time/aaareadme.txt

TAI International Atomic Time. This is the time used by the SoHO spacecraft, and is defined as the number of standard seconds since 0h on 1 January 1958. In the CDS software, TAI time is always expressed as a double precision floating point number.

UTC Coordinated Universal Time. This is the time standard on which civil time is based. The main distinction between UTC and TAI, at least since 1 January 1972, is that occasionally a "leap second" is inserted into the UTC time to keep it in sync with the rotation of the earth. (Before 1972 the situation was more complicated.) TAI time has no leap seconds. Therefore, in order to convert between the two kinds of time, one needs to know when leap seconds were added to the UTC time. This information is maintained within the file "leap_seconds.dat" in the directory given by the environment variable TIME_CONV.

```
IDL> a=anytim2tai('2008/11/11 12:12:12')
IDL> print,a
   1.6050968e+09
IDL> b=anytim2utc('2008/11/11 12:12:12')
IDL> help,b,/str
** Structure CDS_INT_TIME, 2 tags, length=8, data length=8:
   MJD             LONG                 54781
   TIME            LONG              43932000
IDL> print,utc2tai(b)
   1.6050968e+09
IDL> bb=tai2utc(a)
IDL> help,bb,/str
** Structure CDS_INT_TIME, 2 tags, length=8, data length=8:
   MJD             LONG                 54781
   TIME            LONG              43932000
IDL> print,utc2str(bb)
2008-11-11T12:12:12.000
```

计算出2007/10/11 08:46:46后200天零8小时5分钟的时间?

计算出2007/10/11 08:46:46后2000天的时间?

计算出2007/10/11 08:46:46后200天零8小时5分钟的时间?

```
IDL> a=anytim2tai('2007/10/11 08:46:46')
IDL> a1=a+200L*24*60*60+8L*60*60+5L*60
IDL> print,utc2str(tai2utc(a1))
2008-04-28T16:51:46.000
```

```
IDL> b=anytim2utc('2007/10/11 08:46:46')
IDL> b1=b
IDL> b1.mjd=b1.mjd+200
IDL> b1.time=b1.time+8L*60*60*1000+5L*60*1000
IDL> print,utc2str(b1)
2008-04-28T16:51:46.000
```

计算出2007/10/11 08:46:46后2000天的时间?

```
IDL> a=anytim2tai('2007/10/11 08:46:46')
IDL> a1=a+2000L*24*60*60+8L*60*60+5L*60
IDL> print,utc2str(tai2utc(a1))
2013-04-02T16:51:45.000
IDL>
IDL> b=anytim2utc('2007/10/11 08:46:46')
IDL> b1=b
IDL> b1.mjd=b1.mjd+2000
IDL> b1.time=b1.time+8L*60*60*1000+5L*60*1000
IDL> print,utc2str(b1)
2013-04-02T16:51:46.000
```

Wcs（World coordinate system） $SSW/gen/idl/wcs

```
IDL>
IDL> secchi_prep,'20080426_220754_d4c2B.fts',header,data,/rotate_on,/silent
IDL>
IDL>
IDL>
IDL> wcs=fitshead2wcs(header)
IDL>
IDL>
IDL>
IDL>
IDL>
IDL> print,wcs_get_coord(wcs,[2048,1024])/header.rsun
      16.002487     -0.012753654
IDL>
IDL>
IDL> print,wcs_get_coord(wcs,[1024,1024])/header.rsun
    -0.078774878     -0.012787877
IDL>
IDL>
IDL>
IDL> print,wcs_get_pixel(wcs,[0,16.*header.rsun])
      1029.0068       2043.5581
IDL>
IDL>
IDL>
IDL>
IDL> print,wcs_get_pixel(wcs,[4,4]*header.rsun)
      1283.2569       1279.0626
```

Fitshead2wcs:从FITS header中获得wcs程序包识别的数据

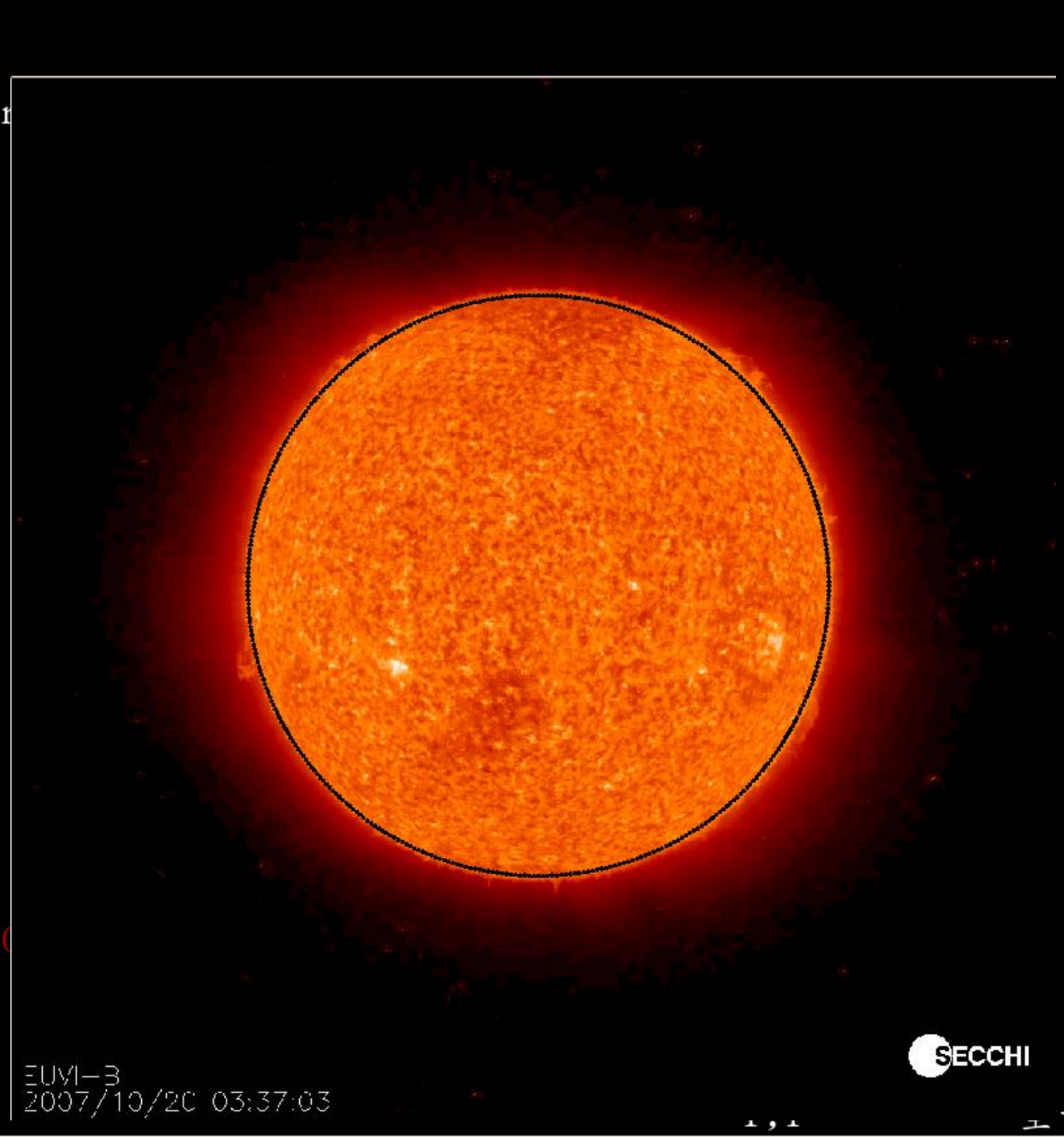
Wcs_get_coord:转换图像的像素坐标到数据坐标

Wcs_get_pixel:转换数据坐标到图像像素坐标

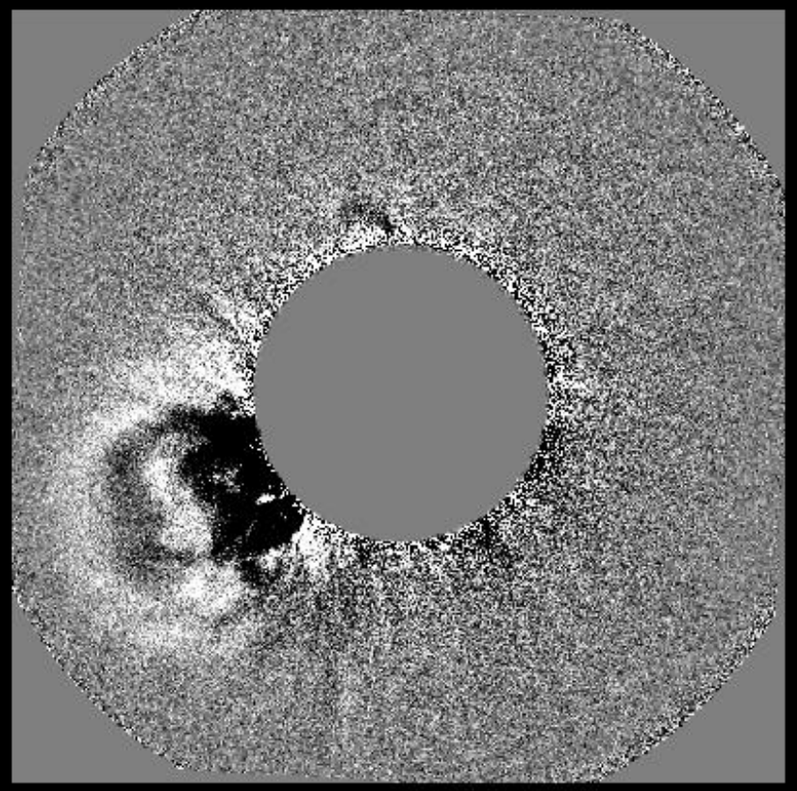
SSW中的坐标信息获取

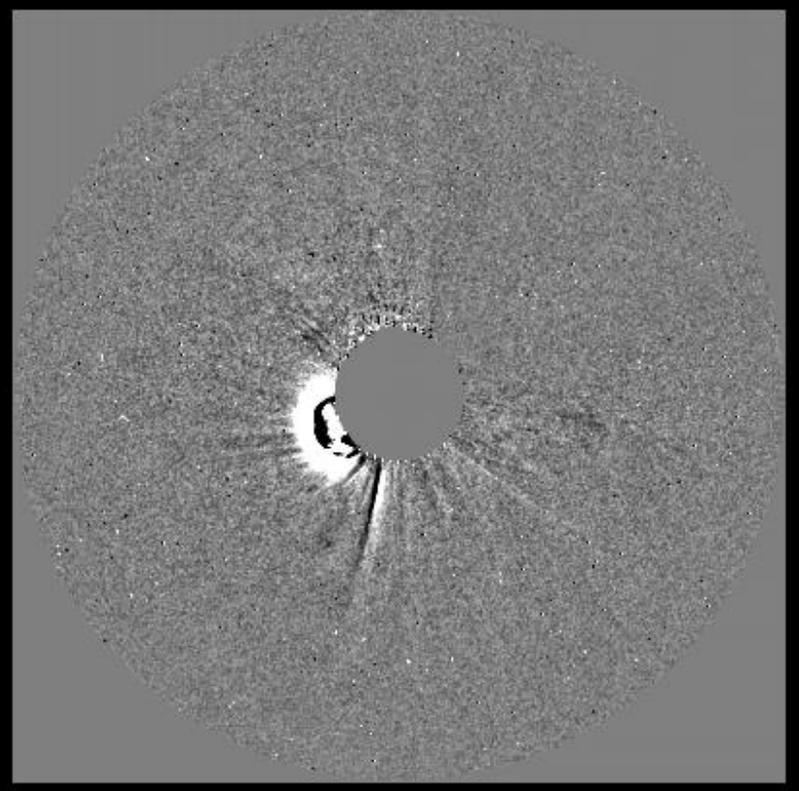
在EUVI上添加1个太阳半径的一圈

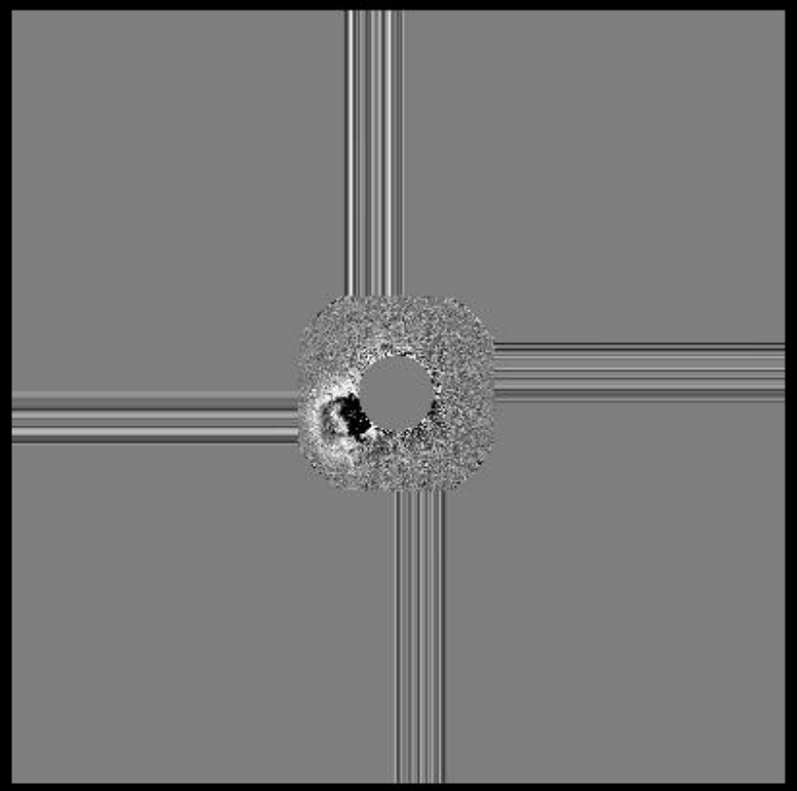
```
pro coord

image=scc_mk_image('20071020_033615_n4euB.fts',/rotate_on,/nopop)
secchi_prep,'20071020_033615_n4euB.fts',index,data,/silent,/rotate_on
window,0,xs=800,ys=800

device,retain=2
secchi_colors,index.detector,index.wavelnth,r
image=congrid(image,800,800)
tv,image

npoint=400
theta=findgen(npoint)*2*!pi/npoint
xy=fltarr(2,npoint)
xy[0,*]=sin(theta)
xy[1,*]=cos(theta)

wcs=fitshead2wcs(index)
xy=wcs_get_pixel(wcs,xy*index.rsun)
help,xy

np=16
phi = Findgen(16) * (!PI * 2 / 16.)
phi = [ phi, phi[0] ]
UserSym, Cos(phi), Sin(phi), /Fill

plots,xy/2048.,/norm,psym=8,color=0,symsize=(

end
~
"coord.pro" 29L, 609C
```

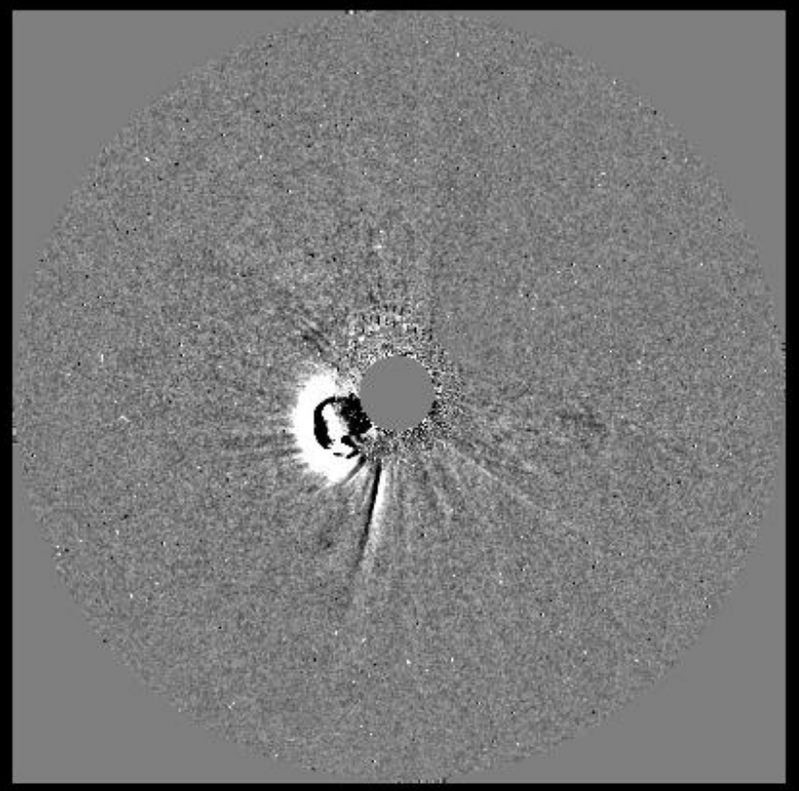


```
pro combine_example
filecor1=findfile('*c1*')
filecor2=findfile('*c2*')
print,filecor1,filecor2
secchi_prep,filecor1,cor1hdr,cor1data,/smask_on,/rotate_on,/silent,/polariz_on
secchi_prep,filecor2,cor2hdr,cor2data,/smask_on,/calimg_off,/calfac_off,/rotate_on,/silent


cor1image=bytscl((cor1data[*,*,1]-cor1data[*,*,0])*1e9,-1,1)
cor2image=bytscl(median(smooth(cor2data[*,*,1],5)-smooth(cor2data[*,*,0],5),5),-2,2)

device,retain=2,decomposed=0
window,0,xs=1000,ys=1000
tv,congrid(cor1image,400,400),50,550,xs=400,ys=400
tv,congrid(cor2image,400,400),550,550,xs=400,ys=400


wcs_cor1=fitshead2wcs(cor1hdr[1])
wcs_cor2=fitshead2wcs(cor2hdr[1])

coord=wcs_get_coord(wcs_cor2)
pixel=wcs_get_pixel(wcs_cor1,coord)
help,cor1image,pixel
new_cor1=reform(interpolate(cor1image,pixel[0,*,*],pixel[1,*,*]))

;index=where(finite(new_cor1))
;new_cor1[index]=128

tv,congrid(new_cor1,400,400),50,50

mask=GET_SMASK(cor2hdr[1])
index=where(mask eq 0 and finite(new_cor1))
help,index

cor2image[index]=new_cor1[index]

tv,congrid(cor2image,400,400),550,50


end
~
```









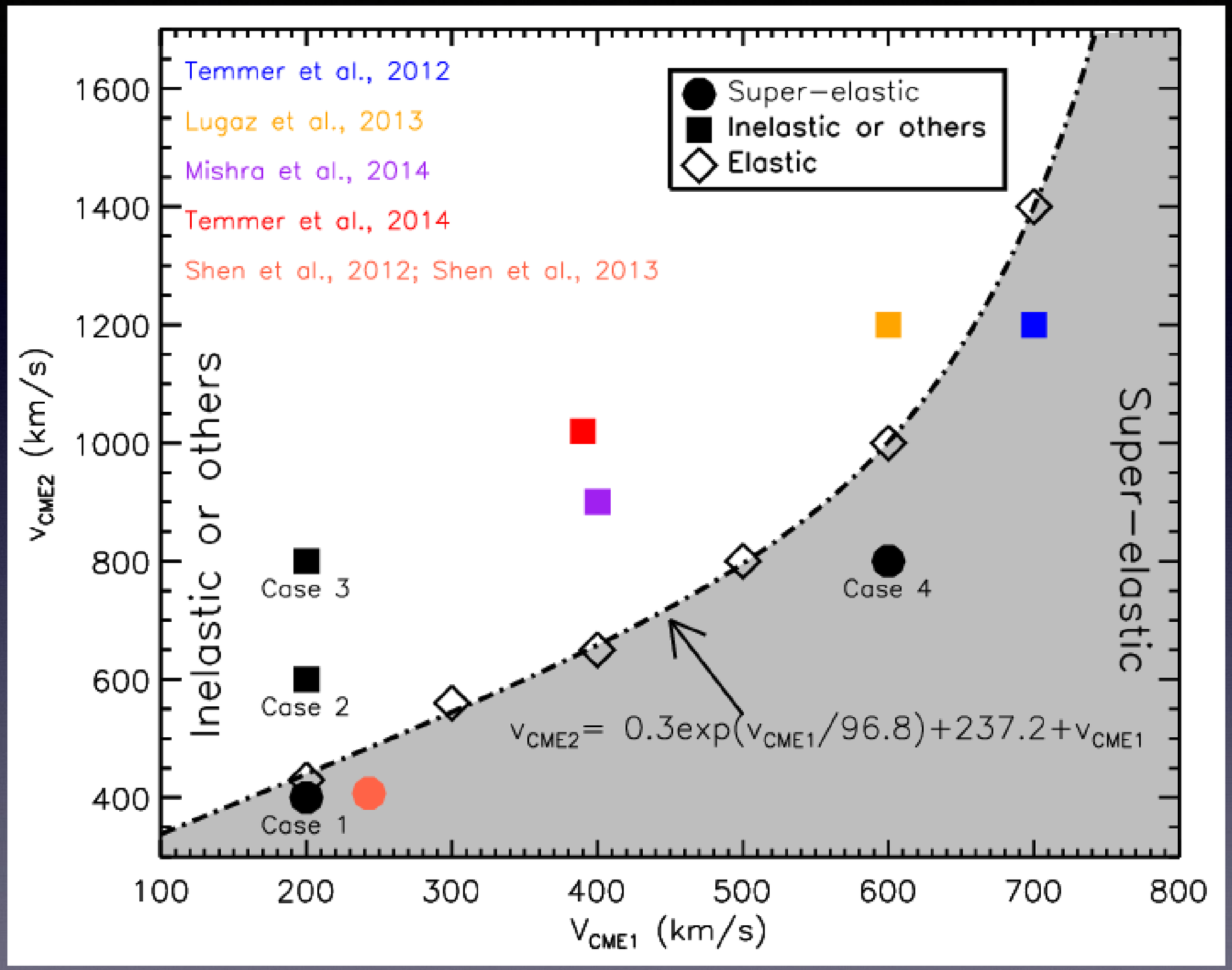
编程几个建议

1. 文件名具有意义
2. 变量名具有意义
3. 编写一些通用的子程序
4. 尽量少用循环

```
pro combine_example
filecor1=findfile('*c1*')
filecor2=findfile('*c2*')
print,filecor1,filecor2
secchi_prep,filecor1,cor1hdr,cor1data,/smask_on,/rotate_on,/silent,/polariz_on
secchi_prep,filecor2,cor2hdr,cor2data,/smask_on,/calimg_off,/calfac_off,/rotate_on,/silent


cor1image=bytscl((cor1data[*,*,1]-cor1data[*,*,0])*1e9,-1,1)
cor2image=bytscl(median(smooth(cor2data[*,*,1],5)-smooth(cor2data[*,*,0],5),5),-2,2)

device,retain=2,decomposed=0
window,0,xs=1000,ys=1000
tv,congrid(cor1image,400,400),50,550,xs=400,ys=400
tv,congrid(cor2image,400,400),550,550,xs=400,ys=400


wcs_cor1=fitshead2wcs(cor1hdr[1])
wcs_cor2=fitshead2wcs(cor2hdr[1])

coord=wcs_get_coord(wcs_cor2)
pixel=wcs_get_pixel(wcs_cor1,coord)
help,cor1image,pixel
new_cor1=reform(interpolate(cor1image,pixel[0,*,*],pixel[1,*,*]))

;index=where(finite(new_cor1))
;new_cor1[index]=128

tv,congrid(new_cor1,400,400),50,50

mask=GET_SMASK(cor2hdr[1])
index=where(mask eq 0 and finite(new_cor1))
help,index

cor2image[index]=new_cor1[index]

tv,congrid(cor2image,400,400),550,50


end
~
```

图像几个建议

1. 图像标示要清楚
2. 图中文字大小适中
3. 图像尽可能直观
4. 颜色标示要清楚


Temmer et al., 2012
Lugaz et al., 2013
Mishra et al., 2014
Temmer et al., 2014
Shen et al., 2012; Shen et al., 2013
Super−elastic
Inelastic or others
Elastic
Inelastic or others
Super−elastic
Case 1
Case 2
Case 3
Case 4
$v_{CME2}= 0.3\exp(v_{CME1}/96.8)+237.2+v_{CME1}$
V_{CME2} (km/s)
V_{CME1} (km/s)
400
600
800
1000
1200
1400
1600
100
200
300
400
500
600
700
800

Python中的卫星数据读取

读取CDF数据： spacepy

SpacePy docu

SpacePy is a package
visualization easier.
output direct from a

- Quickly
- Create p
- Perform
- Run com
- Change
- Harness

SpacePy Code

- ae9ap9 – Handle AE9/AP9 data files
- coordinates – module for coordinate transforms
- datamanager – easy access to and manipulation of data
- datamodel – easy to use general data model
- data assimilation – data assimilation module
- empiricals – module with heliospheric empirical modules
- irbempy – Python interface to irbem/ONERA library
- lanlstar – module to calculate Lstar or Lmax using artificial neural network
- omni – module to read and process NASA OMNIWEB data
- plot – Plot, various specialized plotting functions and associated utilities
- PoPPy – Point Processes in Python
- PyBats – SWMF & BATS-R-US Analysis Tools
- pycdf – Python interface to CDF files
- radbelt – Functions supporting radiation belt diffusion codes
- SeaPy – Superposed Epoch in Python
- time – Time conversion, manipulation and implementation of Ticktock class
- toolbox – Toolbox of various functions and generic utilities

pycdf – Python interface to CDF files

This package provides a Python interface to the Common Data Format (CDF) library used for many NASA missions, available at http://cdf.gsfc.nasa.gov/. It is targeted at Python 2.6+ and should work without change on either Python 2 or Python 3.

The interface is intended to be 'pythonic' rather than reproducing the C interface. To open or close a CDF and access its variables, see the `CDF` class. Accessing data within the variables is via the `Var` class. The `lib` object provides access to some routines that affect the functionality of the library in general. The `const` module contains constants useful for accessing the underlying library.

The CDF C library must be properly installed in order to use this package. The CDF distribution provides scripts meant to be called in a user's login scripts, `definitions.B` for bash and `definitions.C` for C-shell derivatives. (See the installation instructions which come with the CDF library.) These will set environment variables specifying the location of the library; pycdf will respect these variables if they are set. Otherwise it will search the standard system library path and the default installation locations for the CDF library.

If pycdf has trouble finding the library, try setting `CDF_LIB` before importing the module, e.g. if the library is in `CDF/lib` in the user's home directory:

http://pythonhosted.org/SpacePy/pycdf.html

读取FITS格式数据：astropy

FITS File handling (`astropy.io.fits`)

Introduction

The `astropy.io.fits` package provides access to FITS files. FITS (Flexible Image Transport System) is a portable file standard widely used in the astronomy community to store images and tables.

http://docs.astropy.org/en/stable/io/fits/

https://sunpy.org

sunpy

The community-developed, free and open-source solar data analysis environment for Python.

What's new in SunPy 0.9?

Current Version: 0.9.3

INSTALL SUNPY | TOUR OF SUNPY | EXAMPLE GALLERY

Jupyter Notebooks

jupyter Untitled 最后检查: 1 分钟前 (未保存改变)

Logout

File Edit View Insert Cell Kernel Widgets Help 可信的 Python 3

运行 代码

```
In [3]: print,1
Out[3]: (<function print>, 1)

In [4]: print,len([1,2,3])
Out[4]: (<function print>, 3)

In [ ]:
```

Python可视化编辑工具

Anaconda 和 Spyder