Luku 7

Data-analyysi, softa, arkistot

Webbisivuja:
http://surfwww.mssl.ucl.ac.uk/sswdoc/ssw_index.html (Solarsoft)
http://www.lmsal.com/solarsoft/ (Solarsoft ’headquarters’)
http://surfwww.mssl.ucl.ac.uk/surf/guides.html (Analysis guides)
http://orpheus.nascom.nasa.gov/ zarro/idl/maps.html (’map objects’ by Zarro)

Data-arkistot (haku + download):
http://cdaw.gsfc.nasa.gov/CME_list/readme.html (LASCO CME cataloque)
http://lasco-www.nrl.navy.mil/cgi-bin/halocme_parse (LASCO halo CME emails)
http://umbra.nascom.nasa.gov/eit/eit-catalog.html (EIT cataloque search)
http://soi.stanford.edu/sssc/progs/mdi/calib.html (MDI cataloque search)
http://sohowww.estec.esa.nl/data/soho_images_form.html (SOHO images search)
http://mesola.obspm.fr/ (BASS2000 - French data)
http://www.lmsal.com/SXT/homepage.html (DARTS - Japanese data)
http://www.bbso.njit.edu/arm/latest/ (BBSO ARM)
http://surfwww.mssl.ucl.ac.uk/surf/ (SURF-database at MSSL, England)

SolarSoft http://www.lmsal.com/solarsoft/

S.L.Freeland, freeland@sxt1.lmsal.com, Last Revision: 14-October-1999

Latest version of this document is available here

Related Documents

- **SolarSoft Concepts - Coordinated analysis concepts and tutorials**
- **SolarSoft Installation** (UNIX / PC-FreeBSD, PC-Linux)
- **SolarSoft Under Windows**
- **SolarSoft Upgrades**
- **SolarSoft Setup - Running SSW IDL**
- **SolarSoft DataBase description**

What is SolarSoft?

The SolarSoft system is a set of integrated software libraries, data bases, and system utilities which provide a common programming and data analysis environment for Solar Physics. The SolarSoftWare (SSW) system is built from Yohkoh, SOHO, SDAC and *Astronomy* libraries and draws upon contributions from many members of those projects. It is primarily an *IDL* based system, although some instrument teams integrate executables written in other languages. The SSW environment provides a consistent look and feel at widely distributed co-investigator institutions to facilitate data exchange and to stimulate coordinated analysis. Commonalities and overlap in solar data and analysis goals are exploited to permit application of fundamental utilities to the data from many different solar instruments. The use of common libraries, utilities, techniques and interfaces minimzies the learning curve for investigators who are analyzing new solar data sets, correlating results from multiple experiments or performing research away from their home institution.

Some of the primary goals of the SSW are...

Provide a large reuse SW library

The software library represents an evolutionary system tracing back to SMM, through Yohkoh and SOHO, TRACE, and eventually will incorporate SXI, HESSI, and other solar observatories. Many common "solar physics" , file i/o, system, IDL structure manipulation, data display, etc. tasks have already been addressed by others (many others in some cases!)

Very small sample of existing SSW capabilities...

- Time series analysis, time conversions, **UTPLOT** *(millennium safe)*
- Spectral fitting
- Image and Image cube (movies) display
- IDL data manipulation (structure, string, array, mathematics...)
- File I/O (generic binary, ascii), FITS
- Solar (limb fitting, grid overlay, coordinate tranformations...)
- WWW related (html conversion, file conversions, FORM handling, movie making,...)
- WWW Client<->***SolarSoft IDL server*** interface permits execution of SSW/IDL utilities over the Web
  Examples include SXT dynamic WWW movie maker and WWW GOES X-Ray light curve display

WWW index which allows searching for SSW routines by category and/or string pattern is available.

Provide a system which is largely hardware-system and site independent

Installation and setup utilities simplify tranportation to systems. Fundamental shared software is written in hardware-independent and site-configuration independent form. The SSW setup utilities support local configuration files to get sites up and running

Promote the use of certain standards which facilitate coordinated data analysis

Through use of suggested standards, many powerful routines are immediately available for the analysis of new data sets.
The ***SSW** suggested Keywords and Tags* descibes pointing and time standards.
Application for overlay, alignment, scaling, rotation, etc of any SSW instruments via Dominic Zarro's *Mapping* software

Promote an evolutionary environment

Feedback and contributions from the many SSW institutes and users are strongly encouraged. Utilties to simplify remote SSW upgrades and accept user contribution are designed to make the system evolutionary instead of stagnant and divergent.

Provide access to supporting ancillary data bases

Access to supporting data sets from ground-based and satellite-based solar observatories are provided in consistent formats to facilitate coordinated analysis

Provide a file-format independent analysis environment

To accomodate the largest number of solar data sets, file I/O is isolated and analysis applications are designed to operate on objects which are independent of file type.

Provide integrated access to other IDL *packages*

When IDL libraries are identified which provide a useful extension to SSW capabilities, we will attempt to integrate them into the SSW system. For example, the *Chianti Package, K. Dere et al.* is now fully integrated into the SSW distribution and analysis environment for UV/EUV emission line analysis.

A "typical" SSW installation is looks something like...

```
                        SSW
                        |
-------------------------------------------------------------------------------------------
|        |        |                                |                        |                        |
SITE     GEN      SOHO                             YOHKOH              (...missions...)     (...packages..
         (SSW)    |                                |
                  ---------------------            ------------------
```

SOLAR & STELLAR PHYSICS GROUP

mullard space science laboratory

Homepage
Public Outreach
Research
Teaching
People
Seminars/Meetings
Solar Missions
Solar UK Research Facility
Publications
Cosmic Radiation
Grid Activities
PhD page
Links
Contact Information
Group Photos
MSSL ONLY

Solar UK Research Facility

Welcome to the Solar UK Research Facility.

The SURF archive holds data from:

- Yohkoh
- SOHO–CDS
- TRACE
- ReSIK [Direct FTP]

You can submit your data request here. Just click it!

Composite of Solar Images
from the last 24 hours

Data Policy | SURF Contacts | SolarSoft | Solar Links | Analysis Guides | Solar Images

Nobeyama Radioheliograph

English / Japanese

- **Introduction**
- **Data Use Policy**
- **Gallery, Major Results**
- **Images & Movies**
  - **Today's Sun Image at Japan noon, Latest Image (Small), (Large)**
  - **Highlights of Recent Observations**
  - **Correlation Plots (Light Curves)**
  - **Daily Images & Movies**
  - **Event Images & Movies (strong), (weak), (GOES M or X)**
  - **Limb Events**
  - **10min Images & Movies (full size), (half size)**
  - **Monthly Images**
  - **Synoptic Chart**
  - **Number of Flares Observed by NoRH**
- **Bibliography**
- **Archive, Softwares, and Manuals**
- **Nobeyama Solar Radio Observatory**
- **Acknowledgement**

Last updated
E-mail: service@solar.nro.nao.ac.jp

Keyword Standards for Events File Primary Headers

This file will contain examples of keywords found in typical FITS events files. Comments preceed some keywords, and others are hotlinked to the appropriate section of OGIP FITS working group documents. Comments are italicized. There are often several examples for some of the keywords to give a feel for possible values as some keywords have been interpreted by developers in different ways.

Please refer to OGIP_93_003 for full information about this type of file.

Events file definition from OGIP_93_003.

```
SIMPLE  =                    T / file does conform to FITS standard
BITPIX  =                   32 / number of bits per data pixel
NAXIS   =                    0 / number of data axes
EXTEND  =                    T / FITS dataset may contain extensions
```

These comments are automatically generated by FITSIO.

```
COMMENT   FITS (Flexible Image Transport System) format defined in Astronomy and
COMMENT   Astrophysics Supplement Series v44/p363, v44/p371, v73/p359, v73/p365.
COMMENT   Contact the NASA Science Office of Standards and Technology for the
COMMENT   FITS Definition document #100 and other FITS information.
```

DATE is also generated automatically by FITSIO among others.

```
DATE    = ' 5/10/94'           /FITS creation date (DD/MM/YY)
```

Content provides quicklook "HDUCLASS-like" information in the primary header.

```
CONTENT = 'BASIC   '           / file contains time intervals and events
CONTENT = 'EVENT LIST'         / this is an event list file
CONTENT = 'RAW     '           / TARGET EVENTS
REVISION=                    1 / revision number of processed data
```

The ORIGIN keyword is the original processing point. Subsequent processing and reduction is then kept in the REVISION and/or HISTORY and/or CREATOR keywords.

```
ORIGIN  = 'ESA/SSD '           / processing site
ORIGIN  = 'HEASARC/GSFC'       / origin of FITS file
ORIGIN  = 'MPE Garching, FRG'  / origin of processed data
ORIGIN  = 'USRSDC  '           / origin of processed data

OBSERVER= 'ESA/ESTEC'          / Principle Investigator
OBSERVER= 'MPE, ROSAT-TEAM'    / PI name
OBSERVER= 'NONE    '           / Principal Investigator
OBSERVER= 'PSPC CAL TEAM'      / PI name

OBS_ID  = '41      '           / Observation ID 00-64
OBS_ID  = 'CA999998P-3.N1'     / observation ID
OBS_MODE= 'POINTING'           / Observation mode
OBS_MODE= 'SCAN    '           / obs mode: POINTING,SLEW, OR SCAN
OBS_MODE= 'SLEW    '           / obs mode: POINTING,SLEW, OR SCAN

OBJECT  = '2S2251-178'         / Name of observed object
OBJECT  = 'ALUMINUM 1.47 keV calibration' /name of object
OBJECT  = 'CRAB    '           / Name of observed object

TELESCOP= 'ASCA    '           / Telescope (mission) name
Other telescope values for CGRO, ROSAT, EXOSAT, HEAO 1, XTE
INSTRUME= 'GIS2    '           / Instrument name
INSTRUME= 'LECS    '           / instrument name
INSTRUME= 'PSPCB   '           / instrument name

EQUINOX =               2000.0 / equinox
RADECSYS= 'FK4     '           / Stellar reference frame in use
RADECSYS= 'FK5     '           / World Coordinate System

RA_NOM  =         8.429000E+01 / nominal RA (deg)
```

IDL Map Software for Analyzing Solar Images

D. M. Zarro (L-3Com/GSFC)

CONTENTS

---

1. INTRODUCTION

An IDL map is a structure that contains two-dimensional (2-d) image data with accompanying pixel coordinate and spatial scale information. The latter parameters are defined as*properties* of the map and are unique for each image source. Defined in this manner, an arbitrary image can be manipulated or transformed in a manner that is independent of the image source.

This software note describes how to create and use maps for processing solar images. We will use sample images obtained from instruments onboard *SOHO* and *Yohkoh* missions. Typical processing tasks include: roll-correction, stretching, translation, solar rotation-compensation, and image coalignment. Although we discuss solar examples, the techniques described below are applicable to any two-dimensional dataset. IDL mapping software is incorporated into the SolarSoft system - a set of integrated software libraries, databases, and system utilities which provide a "common" programming and data analysis environment for Solar Physics.

1.1 Creating a Map

The low-level function for creating a map is **make_map**. In this example, a 256x256 array is created with **findgen** and converted to a map variable:

```
IDL> image=findgen(256,256)
IDL> map=make_map(image)
IDL> help,/st,map

** Structure <4031f908>, 8 tags, length=40056, refs=1:
   DATA            FLOAT     Array[100, 100]
   XC              FLOAT           0.00000
   YC              FLOAT           0.00000
   DX              FLOAT           1.00000
   DY              FLOAT           1.00000
   TIME            STRING    '11-Mar-1998 11:40:44.000'
```

The resulting map is an anonymous structure with the following tag definitions:

- *DATA:* the 2-d data values.

- *XC, YC:* the cartesian coordinates of center of the image. Since coordinate information was not specified, the origin is used as the default value.
- *DX, DY:* the spacing between between pixels in the cartesian X and Y directions, respectively. Since spatial scale was not specified, the spacings default to unity.
- *TIME:* a reference time for the image. It could be the start or mean time of the image. The default time is the current Universal Time (UT) when the map is created.

The above tag definitions are defined also as keywords in **make_map**. For example, a map with a pixel spacing of 2 units in the x-direction and 3 units in the y-direction is created by:

```
IDL> image=findgen(256,256)
IDL> map=make_map(image,dx=2,dy=3)
IDL> help,/st,map
** Structure <403cd0e8>, 9 tags, length=40072, refs=1:
   DATA            FLOAT     Array[100, 100]
   XC              FLOAT           0.00000
   YC              FLOAT           0.00000
   DX              FLOAT           2.00000
   DY              FLOAT           3.00000
   TIME            STRING    '18-Mar-1998 15:23:16.000'
```

Note that the basic map structure definition is kept intentionally simple, with data and coordinate parameters being the main defining properties. The units of the coordinate system are also arbitrary. Additional properties are added two ways:

1. At the definition stage. For example, to include a property *UNITS* that specifies coordinate units:

```
IDL> nmap=make_map(map,units='arcsecs')
IDL> help,nmap,/st
** Structure <403cd0e8>, 9 tags, length=40072, refs=1:
   DATA            FLOAT     Array[100, 100]
   XC              FLOAT           0.00000
   YC              FLOAT           0.00000
   DX              FLOAT           2.00000
   DY              FLOAT           3.00000
   TIME            STRING    '18-Mar-1998 15:23:16.000'
   UNITS           STRING    'arcsecs'
```

2. After the definition stage. Following the same example, but using the function **add_prop** :

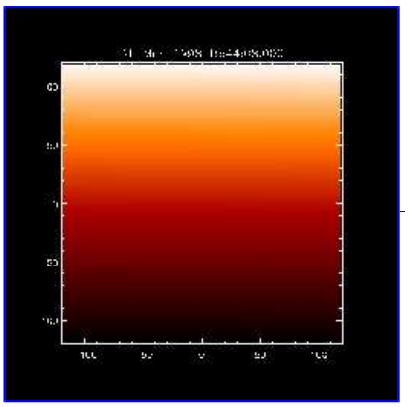
```
IDL> add_prop,map,units='arcsecs'
```

produces the same result. Multiple properties are added as multiple keywords, with the restriction that each property name is unique.

1.2 Plotting a Map

Maps are plotted using the procedure **plot_map** . Thus, using the 2-d image created earlier,

```
IDL> plot_map,map
```



Aktiivinen Aurinko

Auringon aktiivisuus (itse asiassa ’rauhallista Aurinkoa’ ei ole olemassakaan) näkyy

- liekkimäisinä spikuloina, soihtumaisina pluumeina, kirkastumina (plaget ja fakulat)
- sammalikkona (moss), ryppyinä (crinkles)
- auringonpilkkuina ja aktiivisina alueina
- pistemäisinä kirkastumina (bright points), radiokirkastumina napa-alueilla
- filamentteina, prominenssipurkauksina
- flare-purkauksina (roihut, soihdut, leimut)
- koronan massapurkauksina (coronal mass ejections, ’CME’)
- shokkiaaltoina (blast waves, piston-driven waves; Moreton waves, EIT-waves, SXT-waves)
- radiosäteilynä plasmataajuudella (tyyppi II, III, ja IV)
- tummentumina (EIT & SXT dimmings)
- koronan aukkoina, aurinkotuulena
- koronan kuumentumisena

Ja näihin kaikkiin liittyvät magneettikentän muutokset!