

太陽多波長フレアデータ解析研究会 NSRO-CDAW10  
初心者講習2：実習1  
太陽観測データの可視化

2010/10/25@野辺山太陽電波観測所

# 本初心者講習の流れ

1. 太陽観測データのフォーマット：講義
2. SSW-IDL 太陽用データ解析環境：講義
3. 野辺山電波ヘリオグラフ/RHESSI：講義
4. 野辺山太陽電波観測所の解析システム：実習 1
5. 観測データの読み込みから可視化：実習 1
  - データの読み込み
  - グラフ作成
  - ムービー作成
  - 2波長の位置合わせ
6. 野辺山電波ヘリオグラフの像合成：実習 2

# 作業を始める前に

- みなさん、端末に Login できましたか？
- 出来なかったら、至急 教えてください。

# 野辺山太陽電波観測所の データ解析システム:1

- どのLinuxマシン（端末・サーバ）からも、同じ構成のディレクトリーが見えるようにしてあります。
- どのマシンでも、SSW+IDLが動きます。
  - マシンの差は、画面の大きさ・CPU・メモリー量だけです。
- 端末とサーバ：使い方は同じ
  - 端末→皆さんの前にあるPC
  - サーバ→ssh等でloginして利用する。
    - ex. `ssh -X username@burst1`
    - マシン名：burst1, burst2, radio1, radio2, burst6
- 端末とサーバの使い分け
  - これから行う様な可視化やムービー作成→端末
  - 大規模データ処理(ex. 数十枚以上のNoRH像合成)→サーバ

# 野辺山太陽電波観測所の データ解析システム:2

- 解析したデータの置き場所
  - 以下のディレクトリー下に、自分のアカウント名と同じディレクトリーを作って、その下に作成したファイル等を置いてください。
    - ワークディレクトリー
      - /scr/s?? [??: 01,02,03,04,05,06,07,11,12,13]
      - 利用量の制限はありませんが、不必要なデータは消してください。
    - ホームディレクトリーは、1GBしかデータが置けません。

# 野辺山太陽電波観測所の データ解析システム:3

- SSWIDLの起動
  - ターミナル(Terminal)のウィンドウを開き、以下のコマンドを打ってください。

*prompt# idlh*

- 以下のプロンプトが出たら、起動完了です。

IDL>

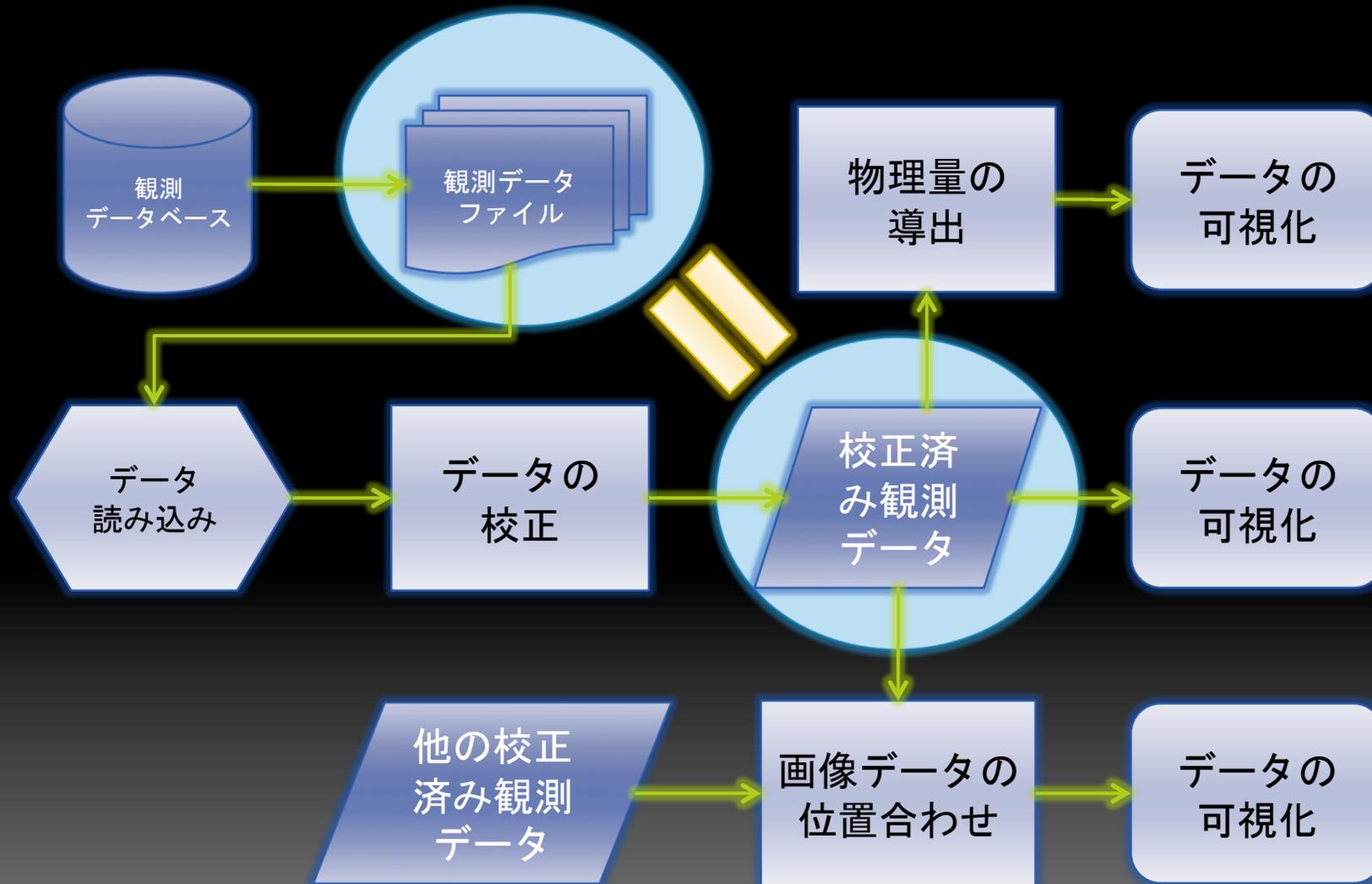
IDL> journal (プロンプトが出たら、左のコマンドを打っておいてください。)

- SSWIDL起動のコマンドは、各システム（研究機関や研究室）で異なります。
  - 京大花山天文台の一部のシステムで、idlhでSSWIDLが起動できる様ですが、それは2000年頃に野辺山のシステムを花山に移植したからです。

# データの読み込み：1

- 講習会用データのありか1
  - /scr/s07/CDAW10\_Lec/data\_single
    - NoRH: ifa100620\_013001
    - HMI/SDO:  
hmi\_20100620\_013045\_magnetogram.fits
    - AIA/SDO:  
aia\_??\_20100620\_0130??\_image\_lev1.fits

# 太陽データ解析の流れ



# データの読み込み: NoRH編1

- 以下のコマンドを実行

```
IDL> dir = '/scr/s07/CDAW10_Lec/data_single/'
```

```
IDL> nfile = 'ifa100620_013001'
```

```
IDL> norh_rd_img, dir+nfile, nind, ndat
```

これで、ndatに17GHz(電波)画像データが、nindにFITSヘッダーに入っている観測情報が入る。それぞれの中身の情報は、以下のコマンドで見ることができる。

```
IDL> help, ndat
```

```
NDAT          FLOAT    = Array[512, 512]
```

# データの読み込み: NoRH編2

- nindの中身

```
IDL> help, nind
```

```
NIND      STRUCT  =-> NORHIDXo Array[1]
```

```
IDL> help, nind,/str
```

```
** Structure NORHIDXo, 2 tags, length=116, data length=107:
```

```
  GEN      STRUCT  -> GEN_NRO Array[1]
```

```
  NORH     STRUCT  -> NORH_INDEX_REC Array[1]
```

```
IDL> help, nind.gen,/str
```

```
** Structure GEN_NRO, 4 tags, length=16, data length=12:
```

```
  TIME     LONG      5401828
```

```
  DAY      INT        11494
```

```
  TIMEJST  LONG      37801828
```

```
  DAYJST   INT        11494
```

```
IDL> help, nind.norh,/str
```

```
** Structure NORH_INDEX_REC, 15 tags, length=100, data length=95:
```

```
  OBS_FREQ  STRING   '17GHz'
```

```
  POLARIZ   STRING   'r+l'
```

```
  SHAPE     INT      Array[2]
```

```
  SEC_PER_PIX  FLOAT    4.91106
```

```
  SOLRFAC    FLOAT    1.01250
```

```
  ROLL       FLOAT    0.00000
```

```
  SOLCEN     FLOAT   Array[2]
```

```
  OLDPFI     BYTE     0
```

```
  SOLP       FLOAT   -7.65810
```

```
  PMAT       FLOAT   Array[4]
```

```
  SEC_PER_PIX_DTY  FLOAT    4.64947
```

```
  PROGNAME   STRING   'snap2d17_steer v6.2 Y. Hanaoka/H. Koshiishi'
```

```
  EFL_IN_PIX_DTY  FLOAT    2.06547
```

```
  CELLSIZE   INT      -1
```

```
  EXPDUR     FLOAT    10.0000
```

```
IDL>
```

# データの読み込み：汎用編

- AIAのファイルを読み込む

```
IDL> afile = 'aia_304_20100620_013050_image_lev1.fits'  
IDL> mreadfits, dir+afile, aind, adat
```

同様に、adatに304Å(極端紫外線)画像データが、aindにFITSヘッダーに入っている観測情報が入る。

```
IDL> help, adat  
ADAT          LONG      = Array[4096, 4096]  
IDL> help, aind  
AIND          STRUCT    = -> MS_026121220001 Array[1]  
IDL> help, aind, /str
```

# データの可視化

- まず、描写する為のWindowを開く。

```
IDL> wdef, 512
```

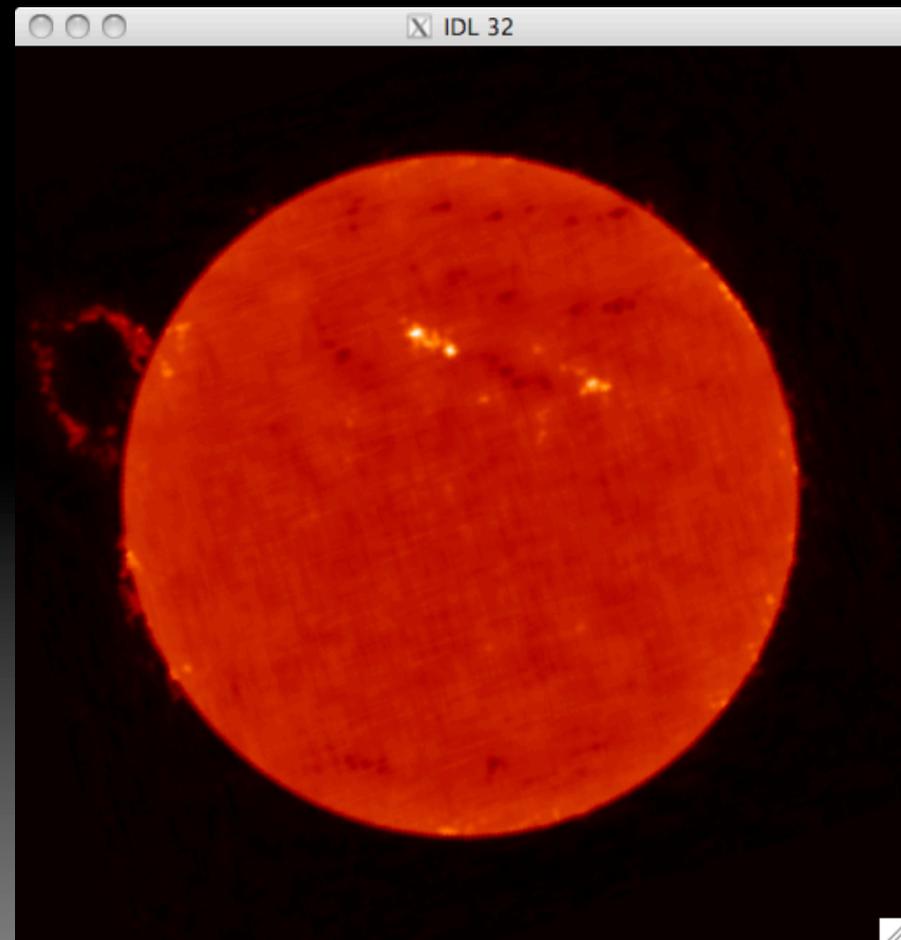
- 画像を書かせる。

```
IDL> tvscl, ndat
```

- 色を変える

```
IDL> loadct, 3
```

```
IDL> tvscl, ndat
```



# データの可視化: グラフ化

- あるY座標でのX軸方向の強度変動をグラフ化する。

```
IDL> window, 1
```

```
IDL> plot, ndat[*], 350]
```

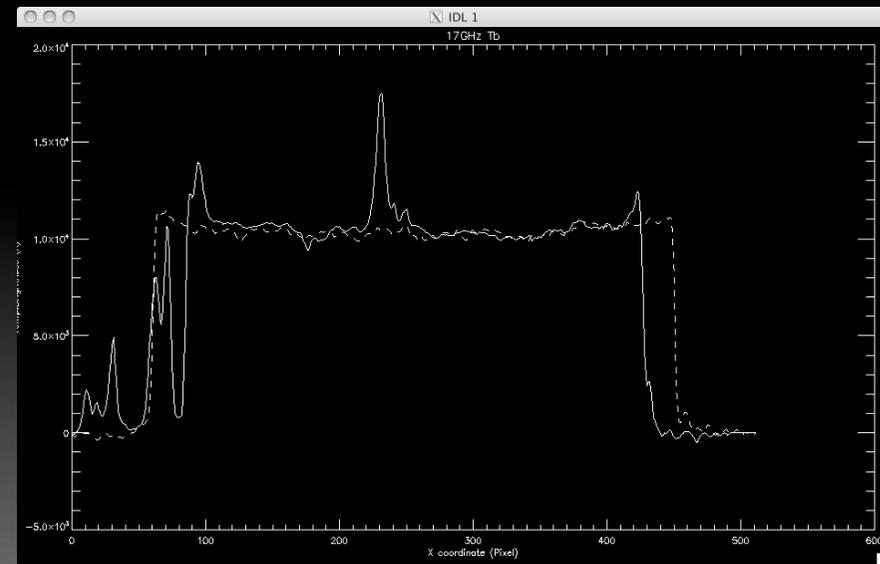
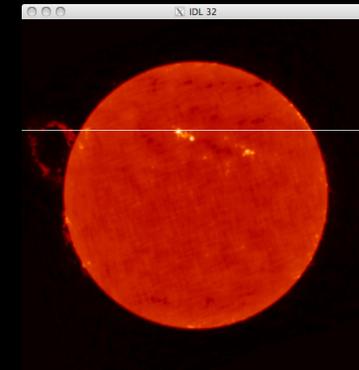
```
IDL> plot, ndat[*], 350], $
```

```
IDL> title='17GHz Tb', $
```

```
IDL> xtitle='X coordinate (Pixel)', $
```

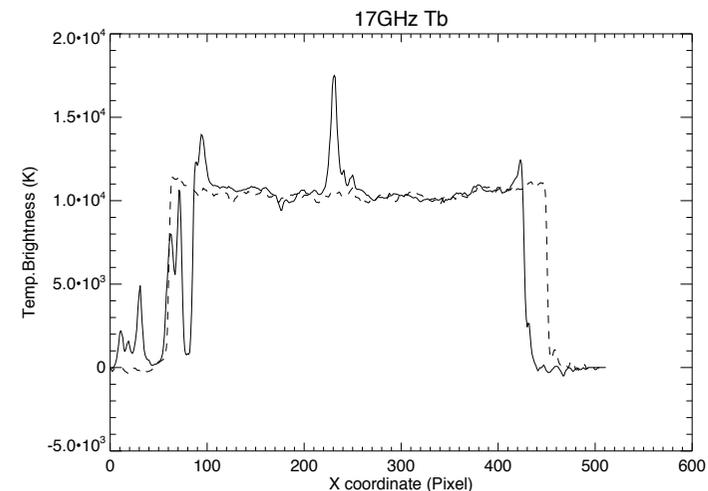
```
IDL> ytitle='Temp. Brightness (K)'
```

```
IDL> oplot, ndat[*], 256], linestyle=2
```



# データの可視化: グラフの印刷

```
IDL> set_plot,'ps'  
IDL> !p.font=0  
IDL> plot, ndat[*], 350, $  
IDL> title='17GHz Tb', $  
IDL> xtitle='X coordinate (Pixel)', $  
IDL> ytitle='Temp. Brightness (K)'  
IDL> oplot, ndat[*], 256, linestyle=2  
IDL> device, /close  
IDL> !p.font=-1  
IDL> set_plot,'x'  
IDL> $ !pr idl.ps
```



# データの可視化: 大きさ・カラー

AIAデータを例に紹介：

```
IDL> wdef, 512
```

```
IDL> loadct, 0
```

```
IDL> tvscl, adat →画像が出ない。
```

画像の大きさが、4Kx4Kで太陽が  
Window上に現れない。

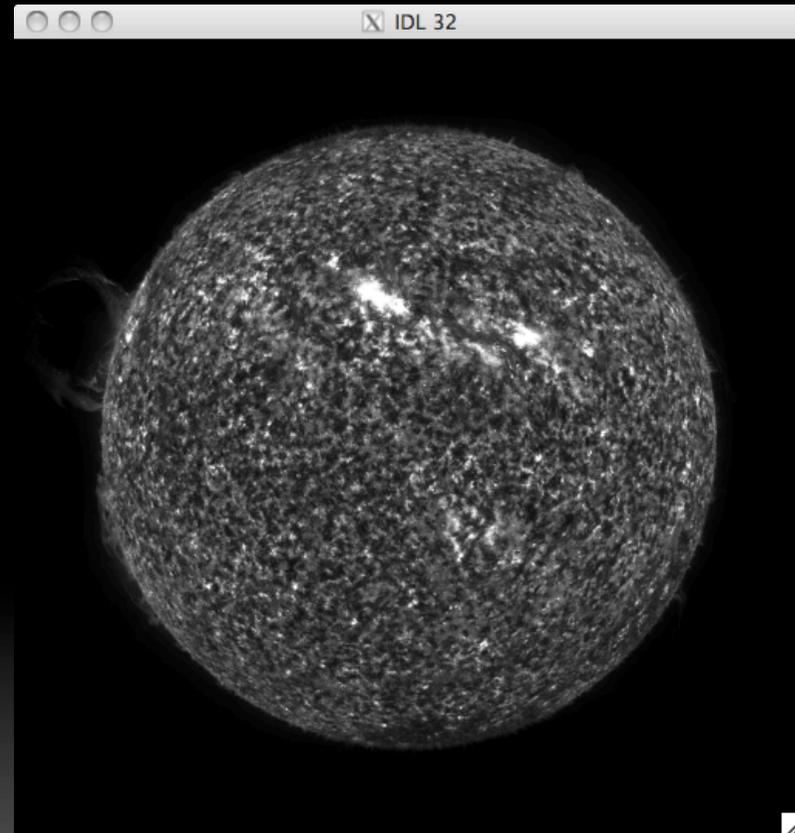
```
IDL> tvscl, rebin(adat, 512, 512)
```

画像の大きさを、512x512に縮小。  
写るけど暗い。

```
IDL> tv, rebin(bytscl(adat, $
```

```
IDL> min=0, max=4e2), 512, 512)
```

画像データのカウン트가0から  
400までに色を割り当てた。



画像をPPT等に張り込む為の画像ファイル  
(JPEG, GIF, PNG等の形式)はGIMPを使用

# 2つの波長の画像の位置合わせ

- map形式への変換

```
IDL> norh_index2map, nind, ndat, nmap
```

```
IDL> index2map, aind, adat, amap
```

- map形式の表示

```
IDL> plot_map, nmap
```

```
IDL> plot_map, amap, /log
```

- 重ね合わせ

```
IDL> plot_map, amap, /log
```

```
IDL> plot_map, nmap, /over
```

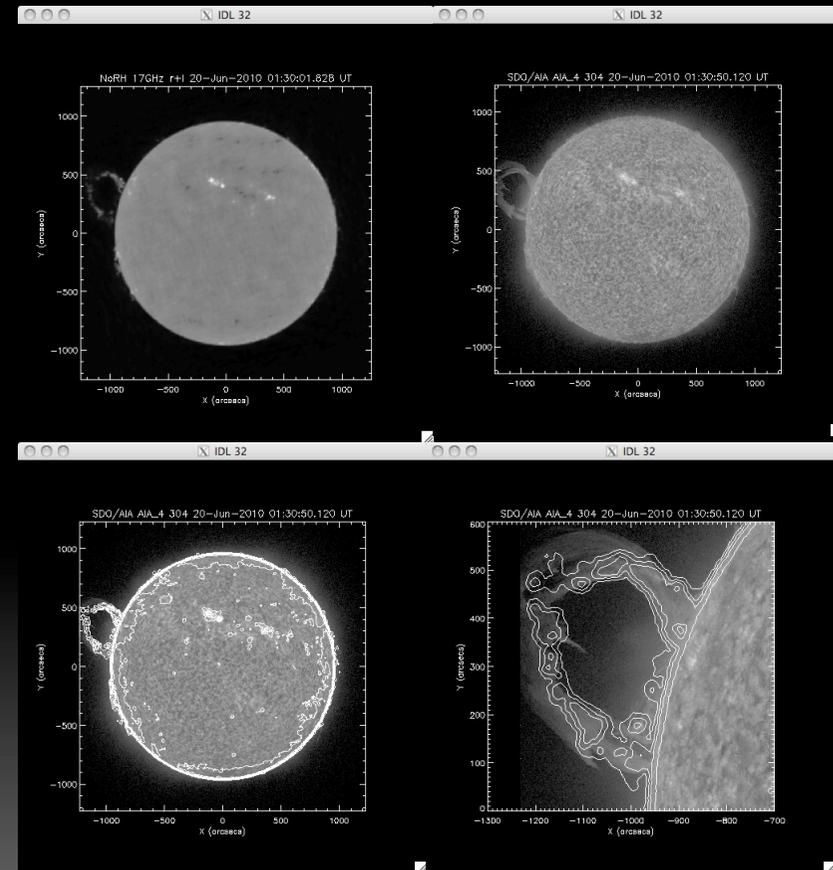
- Zoom UpとContourの調整

```
IDL> plot_map, amap, /log, $
```

```
IDL> fov=10, cen=[-1000, 300]
```

```
IDL> plot_map, nmap, /over, $
```

```
IDL> levels=[1e3, 2.5e3, 5e3, 1e4, 2e4]
```



## データの読み込み：2(多数枚の画像)

- 講習会用データのありか2
  - /scr/s07/CDAW10\_Lec/data\_movie/XRT/
    - このしたに、H0000, H0100, H0200のディレクトリーはあり、総数234枚のXRTで撮像された画像が入っている。

- ファイル名の一括取得

```
IDL> dirs = '/scr/s07/CDAW10_Lec/data_movie/XRT/'
```

```
IDL> files = findfile(dirs+'H0???'XRT*.fits')
```

```
IDL> help, files
```

```
FILES          STRING    = Array[234]
```

## データの読み込み：2(多数枚の画像)

- とにかく全部読み込んでみる。

```
IDL> read_xrt, files, xind, xdat, /force
```

- パラパラ漫画的なムービーでしてみる。

```
IDL> wdef,512
```

```
IDL> stepper, xdat, info=$
```

```
IDL> xind.date_obs+' '+'$
```

```
IDL> string(xind.naxis1, format='(I04)')+' '+'$
```

```
IDL> xind.ec_fw2_
```

→うまく繋がらないムービー。

理由1：長時間露出と短時間露出が混ざっている。

理由2：大きさが異なる画像と可視光画像が混ざっている。

# データの読み込み：2(多数枚の画像)

- データの選択

```
IDL> read_xrt, files, index, /nodata
```

(画像を読み込まず、ヘッダー情報だけ読み込む)

```
IDL> ss = where ((index.naxis1 eq 384) and $
```

(画像のX方向の大きさが384 pixelのモノだけを選べ)

```
IDL> (index.exptime lt 4.) and $
```

(露出時間が4秒以下のモノを選べ)

```
IDL> (index.ec_fw2 eq 2))
```

(X線フィルター”Ti/poly”を使って撮像したものだけ選べ)

```
IDL> read_xrt, files[ss], xind, xdat
```

(上記で選択されたファイル[ss番目のファイル]だけを読み込め)

```
IDL> help, xdat
```

```
XDAT          INT          = Array[384, 384, 113]
```

[校正したい場合は、IDL> xrt\_prep, files, ss, xind, xdat 時間がかかります。]

## データの読み込み：2(多数枚の画像)

- パラパラ漫画的なムービーでしてみる。

```
IDL> wdef, 384
```

```
IDL> stepper, xdat^0.2, info=$
```

```
IDL> xind.date_obs+' '+'$
```

```
IDL> string(xind.naxis1, format='(I04)')+' '+'$
```

```
IDL> xind.ec_fw2_
```

ましなムービーになっている。

(自動露出時間調整機構と

衛星の揺れで見づらいムービーにはなっている。)

# 時間変動のグラフの作成

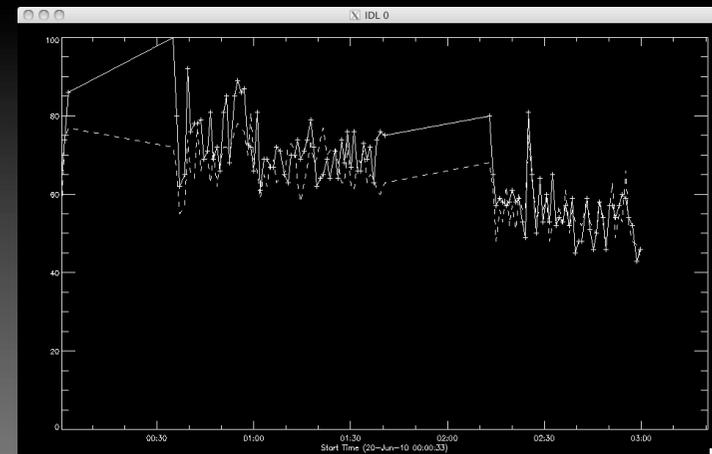
- ある点の強度変化を知りたいとき。

```
IDL> utplot, xind.date_obs, xdat[128, 128, *]
```

(NoRHの場合、IDL> utplot, index, data[x, y, \*])

```
IDL> outplot, xind.date_obs, xdat[128, 128, *], psym=1
```

```
IDL> outplot, xind.date_obs, xdat[128, 100, *], linestyle=2
```



# MPEGムービーの作り方

- 最も簡単な方法

```
IDL> index2map, xind, xdat, xmap
```

```
IDL> movie_map, xmap
```

明日、別方法を教えます。

# 作ったデータのセーブ

- 処理したデータ（変数）のセーブ

```
IDL> save, nmap, amap, filename='test.dat'
```

- データの復元

```
IDL> restore, filename='test.dat'
```

# 今日の講習はこれで終わり。

- IDLを終了させてください。  
IDL> exit

ホームディレクトリーに

```
idlsave.pro
```

というファイルが出来ているはず。  
これは、今まで打ったコマンドが記録されている。  
また、このファイルの最後に

```
end
```

という行を加えてIDL内で以下のコマンドを打つと、いままでのコマンドを自動実行してくれる。

```
IDL> .r idlsave.pro
```

# 実習1の課題

1. 様々な紫外線の画像(AIA/SDO)にNoRHの等高線を書き、その画像をgifファイルにする。
  - AIA/SDOは、304Å以外に、94Å, 131Å, 171Å, 193Å, 211Å, 335Å, 1700Åがある。
  - プロミネンスのズームアップの画像も作る事。