

Suprime-Cam 2010 秋の学校版

おしながき

1. 整約

- SDFRED をインストール
- テストデータ (SDF field) を SMOKA からダウンロード
- SDFRED で処理

2. 位置較正

- WCSTools をインストール
- 画像の座標を較正してみる

3. 測光較正

- Landolt(1992,2009) 論文を探す
- 標準星のどれがどれかを一致させる
- 開口測光
- 測光零点を求める

4. 測光

- 重ね合わせた画像を SExtractor で天体検出および測光を行なう
- 較正結果を使って RA, Dec, mag のカタログを作成する。

5. その先

- まだ時間があったら、好きなデータを smoka からダウンロードして上記一連の作業を繰り返しても良い。

本文書は秋の学校 2008 版を元に Anooshiravan Roozrokh, Chi-Hung Yan and Hanindyo Kuncarayakti 各氏のコメントにより更新された後、毎回学校のたびにじわじわ更新されているものです。

えとぶん：やぎ&なかた

1 SAOimage ds9

秋の学校 09 Suprime-Cam では、saoimage ds9 を標準の FITS ブラウザとして利用することにする。今回は天文台解析システムに入っているので準備は必要ないが、ds9 は以下からダウンロードできる

<http://hea-www.harvard.edu/RD/ds9/>

なお、天文台解析システムに入っている /usr/local/bin/ds9 は 6.0 であり、ds9 の最新版は 6.2 である。ds9 は筆者が経験する限り、困ったことにバージョンによりこまごまとしたところで挙動が異なる。自分が使っているのはどのバージョンかは意識しておいた方がよい。

以下の情報は x86_64 上での Linux の ds9 6.0 に関して確認している。

なお ds9 はあくまで以下の説明等で「標準」として使う。もし使いたければ、詳しい説明は期待できないということは承知の上で使い馴染んだブラウザ、例えば ESO-SkyCat や fv などを使っても構わない。

1.1 初期設定

まずは ds9 を起動する。

```
$ ds9 &
```

上にあるプルダウンメニューから Edit-Preferences を選択。この中でいくつか設定変更を行なっておく。

- Edit (マウスの挙動)¹

1. 設定ウィンドウで “Edit” タブを選択。
2. 上のペインにある “Menu” ボタンのプルダウンで、“Pointer” から “None” に変更する。

- Scale

1. 設定ウィンドウで “Scale” タブを選択。
2. 上のペインにある “Menu” ボタンのプルダウンで、“2 ペイン目の “MinMax” から “ZScale” に変更。Suprime-Cam のデータだと、何故かこれがうまくい場合が多いので。

- Analysis (Catalog server)

1. 設定ウィンドウで “Analysis” タブを選択。
2. 下の方のペインにある Catalogs Server を “cds” から “adac” に変更。この adac は vizier.nao.ac.jp と天文台内にあるサーバなので、天文台の解析システムに使うにはフランスより遥かに近い。

- Print (Printer 設定)

1. 設定ウィンドウで “Print” タブを選択。
2. 一番上の Print To” を “Printer” から “File” に変更する。これでうっかりプリンターに出力してしまう事故を避けられる。

¹これはやらなくてもいいが、初心者はポインタを不要なところにつけてしまっって面倒なことになる場合が良くあるので、“none” に設定しておくことを推奨する

3. 下の Page Size を “Letter” から “A4” に変更する。

以上の操作を行なったら、右下の Save を押して保存する。この設定は `~/ds9.prf` として保存される。

ここで設定したのはデフォルトの挙動で、起動時の設定である。起動後に変えたい場合はメニューから直接行なう。例えば、カタログサーバを一時的に変更したい場合は、Analysis-Catalog Tool を起動して、そのメニューの Catalog Server で変更する。

1.2 操作の初歩

1.2.1 起動

ファイル名を指定して

```
$ ds9 a.fits
```

で開く。

IRAF と連携して使う場合は、fifo オプションを付ける必要がある。これは、後述するように、秋の学校ではそれぞれ違うポートを使うためである。例えば `/dev/imt2` が指定されてる場合は以下のように起動する。

```
$ ds9 -fifo /dev/imt2
```

1.2.2 マウスの操作

右ボタンドラッグ：色スケールの変更

上下がコントラスト

左右が明暗

中ボタン押し：押した位置が中心に来るように移動

左ボタン押し：無反応（となるように ‘none’ を設定しておいた）

1.2.3 ds9 ウィンドウの中段にあるボタンの一群

- file open（他のファイルを開く）
- zoom +（ズームイン）
- zoom -（ズームアウト）
- scale minmax（他のスケールに変更も可能）
- color bb（黒体放射っぽい色付け）

などなど。

2 IRAF 最低限の使い方

まず最初に。筆者は IRAF が嫌いなのでほとんど使っていない。詳細は他の方に尋ねて欲しい。
また、この章に書かれている内容の一部は国立天文台の天文データセンター解析システム特有だったりもするので、そこも要注意である。

2.1 ana** で IRAF を使い始める

IRAF を使う際にはまず、パラメータファイル (login.cl) と、各タスクのパラメータを入れておくディレクトリ (uparm/) が置いてある場所で作業を開始する必要がある。

```
--- (work directory) -- ./login.cl
                        -- uparm/*****.par
```

秋の学校 2010 Suprime-Cam では、ほとんどの解析は {work directory}/object/ か {work directory}/standard/ で行なうことにしている。そして、この {work directory} は、例えば /mfs01a/subaru00/suprime なり、/mfs04s/yagims/suprime なりになっているはずである。

この login.cl と uparm/ は “mkiraf” というコマンドで最初に作る。mkiraf は “Terminal Type” を聞いてくるが、ここでは “xgterm” を指定しておくことを推奨する。

例)

```
$ cd /mfs01a/subaru00/suprime
$ mkiraf
-- creating a new uparm directory
Terminal types: xgterm,xterm,gterm,vt640,vt100,etc.
Enter terminal type: xgterm
A new LOGIN.CL file has been created in the current directory.
You may wish to review and edit this file to change the defaults.
```

これで、/mfs01a/subaru00/suprime/login.cl と /mfs01a/subaru00/suprime/uparm/ が作成されるはずである。これ以降、IRAF を実行する際には常にこの /mfs01a/subaru00/suprime で開始することを勧める。

さて、天文データセンターの IRAF ではデフォルトの画像サイズは 800x800 である。これは Suprime-Cam のデータを扱う際には小さすぎるので、login.cl をいじっておく必要がある。

好きなテキストエディタで、以下の変更を行なう。

```
#(old)
#set      stdimage      = imt800
#(new)
set      stdimage      = imtu2
==
```

この imtu2 は /dev/imt2 を使った 2048×2048 のバッファである。IRAF は ds9 などのブラウザとの連携に /dev/imt* というデバイスを介するのだが、ここで同じサーバで同じデバイスを複数人で利用しようとすると、A さんの表示が B さんの ds9 に表示されるなどのわけのわからない事態になりうる。自分がこの秋の学校でどのデバイスを使うべきかは別途指示があるので、それに従って欲しい。

ここまで準備が出来たら IRAF の起動である。まず、xgterm を起動する。そして、出てきた xgterm の中で “cl” で起動する。

例)

```
$ xgterm &
```

```
$ cl
```

```
NOAO/IRAFNET PC-IRAF Revision 2.14.1 Mon Sep 15 10:12:05 MST 2008
```

```
This is the RELEASED version of IRAF V2.14 supporting PC systems.
```

```
...
```

```
    apropos      dbms.      lists.      plot.      stsdas.      utilities.
```

```
    color.      images.    noao.      proto.     system.
```

```
    dataio.     language.  obsolete.  softtools. tables.
```

```
ec1> cd object/
```

```
ec1>
```

この説明では起動は作業ディレクトリで行なっているが、自分のホームディレクトリで起動してから作業ディレクトリに移動したいという人もいるかもしれない。挙動を理解しているのであれば、それも OK である。

例)

```
$ cd ~/
```

```
$ mkiraf
```

```
...
```

```
$ cl
```

```
...
```

```
ec1> cd /mfs01a/subaru00/suprime/object/
```

```
ec1>
```

そして、IRAF 終了のコマンドは、logout である。

```
...
```

```
ec1> logout
```

```
$
```

2.2 タスクの使い方

前述のように私は IRAF 嫌いなので imstat(imstatistics) と imexam(imexamine) の2つだけ紹介する。

2.2.1 imstat (imstatistics)

これはある領域中のピクセルの統計量を出すタスクである。例えば、ある画像(test.fits)のピクセル値の rms を出すのは以下の通り。

```
ecl> imstat test.fits
#           IMAGE      NPIX      MEAN      STDDEV      MIN      MAX
      test.fits    19285    129183.    3.647E6   -1.009E7    1.501E8
```

出力の“STDDEV”が rms である。他の統計量も同時に測定されている。

```
ecl> imstat test.fits fields="image,mean,midpt,mode,stddev"
#           IMAGE      MEAN      MIDPT      MODE      STDDEV
      test.fits    129183.   -58460.   -59156.    3.647E6
```

この midpt というのが、「median の推定量」である。IRAF は median そのものの測定はせず、推定しているのである²。

この統計を出す際に [x1:x2,y1:y2] という表記で範囲を絞ることができる。ここで x1,x2 はそのピクセルを含み、かつ 1-origin である（左下隅のピクセルが (1,1)）。

```
ecl> imstat test.fits[1:100,1:100]
#           IMAGE      NPIX      MEAN      STDDEV      MIN      MAX
test.fits[1:100,1:100]    10000    327.6    3260.    -0.66    32767.
```

その他の使い方は help を読むこと。

```
ecl> help imstat
IMSTATISTICS (Feb01)          images.imutil          IMSTATISTICS (Feb01)

NAME
    imstatistics -- compute and print image pixel statistics

USAGE
    imstatistics images

PARAMETERS
    ...
```

2.2.2 imexam (imexamine)

画像を吟味する。今回の学校では imexam はシーイングサイズ (PSF の FWHM) を調べる際に使う。

まず、ds9 を UNIX のコマンドプロンプトから立ち上げる。この際の起動オプションは前の章で述べたので確認すること。

```
$ ds9 &
```

そして、画像を IRAF から読み込む。使うタスクは display である。

```
ecl> display test.fits 1
```

この最後の“1”というのは ds9 の中でのフレームの番号であるが、とりあえず気にせずおまじないだと思っておいて良い。ds9 に画像が表示されたら、imexam 開始である。

²多分これは計算機が鈍かった時代の遺物である

```
ecl> imexam test.fits
display frame (1:) (1):
```

マウスカーソルが ds9 の上に動き、カーソルの形が、点滅する輪っか状になるはずである。このカーソルを星っぽい天体のところに移動させて、キーボードで 'j' か 'k' を叩く。

```
j: column (x) plot
k: row (y) plot
```

すると、“irafterm” というウィンドウが新しく出てくるはずである。このウィンドウの下の方に center=*** peak=*** sigma=*** fwhm=*** bkg=* という情報が書かれている。この sigma と fwhm の単位はデフォルトではピクセルである。Suprime-Cam のピクセルスケールは 0.202 arcsec/pixel なので、例えば fwhm=5.9 と出たら、シーイングは 1.2 arcsec である。

また jk の代わりに 'r' や 'a' を使うのも良い。

```
r: radial plot
a: aperture sum
```

'r' は動径方向のプロファイル (radial profile) を書くのだが、この際ウィンドウの下側に出てくる値はラベルが何もついてないので、解釈に苦労する。この意味は勿論 help で調べても良いが、'a' と同じである。

'a' は特に irafterm 上には何も表示せず、テキストで値を打ち出してくれる。

#	COL	LINE	COORDINATES								
#	R	MAG	FLUX	SKY	PEAK	E	PA	BETA	ENCLOSED	MOFFAT	DIRECT
838.41	2046.48	838.41	2046.48								
14.70	12.22	129095.	19482.	4818.	0.25	-63	8.95		4.29	4.56	4.91

読みにくい2行のデータである。1行目が座標、2行目にいろいろな測定値がでているが、後半の “enclosed”, “moffat”, “direct” が3種の方法による FWHM の推定値である³。

imexam の簡単な help は '?' とタイプすると出てくる。

imexam を終了させるには 'q' とタイプすれば良い。

これ以上の詳細はやはり “help imexam” を読んで欲しい。

³ちなみに IRAF の FWHM 測定は星の中心がピクセルのどこにあるかにもものすごく依存しているようだということは知っておいて損はないかもしれない

3 位置較正

大雑把に言えば、FITS ヘッダの情報を追加更新して、例えば ds9 で FITS を表示させた時にも (RA,Dec) が正しく表示されるようにする操作である。

3.1 使うツール

このような FITS の x,y を RA,Dec に関連付ける標準的な機構が WCS(world coordinate system) である。

http://fits.gsfc.nasa.gov/fits_wcs.html

最近のソフトウェアはみんなこの WCS に対応している。

位置較正に使うツールとしては例えば

- WCStools (SAO)
- SCamp (Terapix)
- mscred (IRAF)

などがあるが、今回の学校では WCStools を使うことにする。

天文データセンター解析システムには WCStools はインストールされてないので、自分のホームディレクトリでコンパイルすることにする。最新版は

<http://tdc-www.harvard.edu/wcstools/>

にある。多分

<http://tdc-www.harvard.edu/software/wcstools/wcstools-3.8.1.tar.gz>

だと思うが、確認して持ってきて欲しい。

ソースコードをダウンロードしたら、コンパイルする。

```
$ tar xvzf wcstools-3.8.1.tar.gz
$ cd wcstools-3.8.1
$ make
```

これで、必要なバイナリ (bin/imwcs) ができるはずである。

3.2 較正の流れ

FITS の位置較正はおおよそ以下のような流れである。

1. 画像上で天体を探して x,y のリストを作る
2. 同じ領域で天体の (RA,dec) のリストを準備
3. 2つのリストを比較して同じ物をペアにして選び出す。
4. 選び出されたペアで (x,y) -(RA,dec) の間の変換を求め、WCS 情報として FITS に書きこむ

Step 1) 天体探し

実際には WCStools の imwcs はこの4段階すべて行なってくれたりするのだが、経験上 imwcs の天体検出はイマイチなので、ここでは SExtractor を代わりにつかう。

外から imwcs に天体(x,y) カタログを食わせる際には、ファイルのフォーマットは以下であることが求められる。

```
x y mag
```

そこで以下のような astmt.param ファイルをエディタ (emacs とか vi とか) で作成する。

```
XWIN_IMAGE
YWIN_IMAGE
MAG_AUTO
```

そして、SExtractor のパラメタファイルを以下のように作成する。

```
$ sex -d > astmt.sex
```

ここで出来た標準のパラメタファイル (astmt.sex) の中で、いくつかを編集する。

```
$ emacs astmt.sex
```

標準から変えるべきパラメタは以下である。

CATALOG_NAME	all.cat	# name of the output catalog
CATALOG_TYPE	ASCII	# NONE,ASCII,ASCII_HEAD, ASCII_SKYCAT, # ASCII_VOTABLE, FITS_1.0 or FITS_LDAC
PARAMETERS_NAME	astmt.param	# name of the file containing catalog contents
...		
DETECT_MINAREA	30	# minimum number of pixels above threshold
DETECT_THRESH	20	# <sigmas> or <threshold>,<ZP> in mag.arcsec-2
ANALYSIS_THRESH	20	# <sigmas> or <threshold>,<ZP> in mag.arcsec-2
...		
FILTER	N	# apply filter for detection (Y or N)?
...		
SATUR_LEVEL	40000.0	# level (in ADUs) at which arises saturation

パラメタは画像によっていろいろ変えるべきだが、おそらく Suprime-Cam ブロードバンドでモザイク済画像だとこんなもんでいいかと思う。少なくとも筆者が各種試した限り、なんとなくこれでうまくいっている。

ここまで準備できたら、SExtractor を実行である。all.fits が較正したい FITS ファイルだとする。

```
$ sex -c astmt.sex all.fits
```

注意すべき点は imwcs は最大でも明るい方から 200 天体しか使わないということである。従って、DETECT_THRESH と DETECT_MINAREA を調整して、暗い天体を含み過ぎないようにすることが効率の面で重要である。

上記例では FILTER は N にしてあるが、ゴミ検出がカタログに混じらないようにするために Y としても良い。この場合はフィルターファイルが必要なので、ここデータセンター環境の場合は /usr/local/share/SExtractor/2.8.6/config/ から、ファイルをコピーしておく必要がある。

```
$ cp /usr/local/share/SExtractor/2.8.6/config/default.conv .
```

Step 2) その領域の天体カタログ

一方、RA,dec のファイルだが、これは色々なソースが考えられる。ここでは安直にオンラインの USNO-B1 カatalogのサービスを使うことにする。WCStools のタグとしてはこれは ub1 になる。一般的には USNO-B1 は SAO のサービスが利用できる。環境変数を以下のように設定すれば良い。書き忘れていたが、この辺りのコマンド全て bash ベースである。

```
$ export UB1_PATH=http://tdc-www.harvard.edu/cgi-bin/scat
```

しかし、秋の学校では専用に USNO-B1 サーバを立ち上げたので、ここを使って欲しい。

```
$ export UB1_PATH=http://work45.mtk.nao.ac.jp/cgi-bin/scat
```

このサーバは天文台内からしか使えないので、学校後自機関で解析する場合は、前述のハーバードのサーバを使うか、あるいは自機関に scat のサーバがあるならそれを使うなりして欲しい。

Step 3) カatalogのマッチング

Step 4) 関係式を解いて WCS を書き込む

この3,4 は imwcs が行なってくれる（実質2も imwcs がやってくれるのだけど）。ここには各種パラメタがあるのだが、例えば試験データに関しては以下のパラメタがうまく動作した。

```
$ imwcs -v -d all.cat -n 8 -w -h 200 -c ub1 -q first all.fits
```

- v うだうだと途中経過表示せよ
- d 次のカatalogを使え
- n フィットするパラメタの自由度
- w WCS を出力に書き込め
- h 参照星の最大数
- c 使うカatalogの種別
- q i 繰り返せ
 - r 座標原点は動かせ
 - s σ クリップではずれ値は弾け
 - t 許容度は繰り返しのたびに減らせ

この辺りは全て調査したわけではないので、より良いパラメタがあるかもしれない。ともあれ、成功すれば imwcs が allw.fits を出力する。これが較正済の FITS である。

3.3 うまくいったか確かめる

まず、ds9 で all.fits を表示させる。これが BEFORE である。

```
$ ds9 all.fits &
```

右のほうの Analysis プルダウンから、Catalogs を選んで、サブメニューから GSC2.3 を選択する。ここは SDSS DR6 や 2MASS でもいい。その辺は好みである。暫く待つと、ダウンロードしてきたカタログが○で表示される。位置がずれているのがみえるはずである。

さて、次に同様に ds9 で allw.fits を表示する。これが AFTER である。

```
$ ds9 allw.fits &
```

同じカタログを表示させると、どの程度改善されたか、目でわかるはずである。

ここで、再度強調しておくが、この手順は、そもそもデータ解析を SDFRED で行なったことも含め、高精度の位置構成や weak lens などには全く向いてない手法である。その辺りを承知の上で使うこと、いいね。

4 測光較正

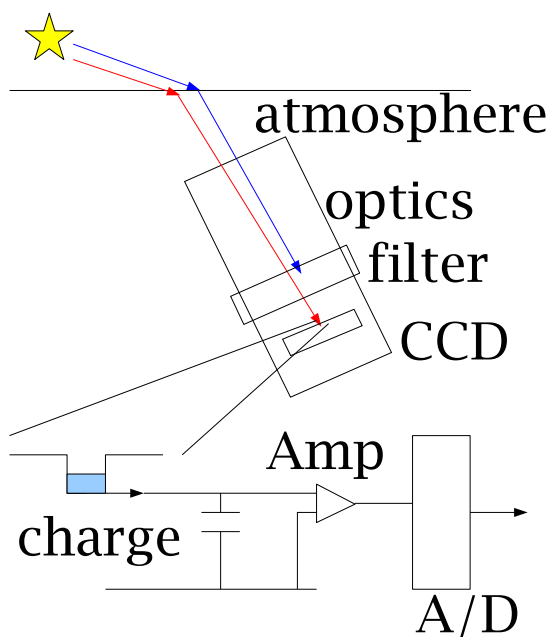
なんか自分で書いた英文を和訳する不毛な作業に段々飽きてきた。そもそもこれ元は和文を英訳したのを編集校正したものだったはず。何で今更逆翻訳してんだろ、とか思ったりもするけど、そうも言ってもらえないので続ける。

CCD データ整約の重要な部分は、CCD の出力を物理的な光度に焼き直すことである。SDFRED は出力画像中の相対的な光度／輝度分布は誤差の範囲で一応保証するが、さて、それではピクセル値が物理的な明るさにどう対応するのかは全くフォローしてくれない。このピクセル値と物理的な値との間の関係を求めるのが測光較正である。

4.1 モデル

まずは光子の伝播を以下のようにモデル化する。

天体で出た光子 (object) はバックグラウンドの光子 (sky) と混じり、その一部は大気や光学系で散乱されたり、吸収されたりして、最終的に CCD 中である確率で光電効果で電荷に化ける。object+sky が最終的に電荷に化ける確率を “absolute flat” と呼ぶことにする。CCD 中では量子効果というか熱の効果で暗電流と呼ばれる電荷が湧いて出る。この電荷は全部まとめて読み出され、電圧に変換された後 A/D 変換でデジタルの値になる。この変換係数の逆数がゲインと呼ばれ、A/D 変換の際に印加される底上げを bias と呼ぶ。



まとめると、出力のカウント値 [ADU: Analog-Digital conversion Unit] は

$$count = (((Object + Sky) \times t_e \times flat) + (dark \times t_d)) \times CF + bias,$$

となる。 t_e は露出時間、 t_d は暗電流をためた時間である。あと、細かい話だが上記の式では bias 電圧も ADU に変換して書かれている。

この式を解くと

$$Object + Sky = \frac{((count - bias) \times gain - dark \times t_d)}{flat \times t_e}$$

となる。

- (t_d, t_e) は時計で測定
- bias はオーバースキャンで推定
- dark は一般的にはダーク撮像から推定するが Suprime-Cam では無視できる。
- flat(relative) は flat 画像から推定できる。

さて、この時点でも gain と absolute flat の flat(relative) に対する係数は求まっていない。この係数はシステムの効率 (スループット) や、空や光学系の状況などに依存する量である。この flat*gain の不定性の係数を k ととりあえず書いて、

$$\begin{aligned} \text{dark}' &= \text{dark}/\text{gain}, \\ k * \text{flat}' &= \text{flat}/\text{gain}, \end{aligned}$$

と定義すると、上の式はこうなる。

$$\text{Object} + \text{Sky} = \frac{((\text{count} - \text{bias}) - \text{dark}' \times t_d)}{k \times \text{flat}' \times t_e}$$

この式の dark' と flat' は観測データから作ることが出来るので、 $k * (\text{Object} + \text{Sky})$ は、とりあえず観測データから求められる。Sky と Object は空間的な構造から分離できるとすれば、最終的に $k * \text{Object}$ は求まる。

ここまでは SDFRED のやっていることであり、画像全体の不定係数 k を除いて相対的な光度/面輝度は求まっている。もし、出力画像の 1 カウントが何等に相当するか (等級零点) が求まれば、flux の絶対較正は終了である。

この係数 k を求めるために、測光標準天体 (星である場合が多いので通常標準星と呼ぶ) を使うのであった。目的天体以外に、観測しているフィルターでの flux (同じことだが等級) が既にわかっている天体 (標準星) を含む天域を観測する。このようなデータを「標準星データ」と呼ぶ。以下 Standard と書く。

さて、これがあると、システムの効率 (スループット) や、空や光学系の状況などが目的天体と標準星天体を撮った時で変化してないという仮定の上で

$$\begin{aligned} k * \text{Target} &= ((\text{count}_1 - \text{bias}) - \text{dark}' * t_{d1}) / (\text{flat}' * t_{e1}) - \text{sky}_1 \\ k * \text{Standard} &= ((\text{count}_2 - \text{bias}) - \text{dark}' * t_{d2}) / (\text{flat}' * t_{e2}) - \text{sky}_2 \end{aligned}$$

となる。ここで "Standard" の物理的な値がわかっているならば、2 式目から k を解くことが出来て、target(obs) を物理量 (Target) に変化することが出来る。

$$\begin{aligned} k * \text{Target} &= \text{target}(\text{obs}) \\ k * \text{Standard} &= \text{standard}(\text{obs}) \\ \text{Target} &= \text{Standard} * \text{target}(\text{obs}) / \text{standard}(\text{obs}) \end{aligned}$$

可視のブロードバンド撮像の場合、これらの式は「等級」で書かれることが多い。

$$\begin{aligned} \text{mag}(\text{Target}) &= \text{mag}(\text{Standard}) + \text{mag}(\text{target}(\text{obs})) - \text{mag}(\text{standard}(\text{obs})) \\ &= \text{mag}(\text{target}(\text{obs})) + (\text{mag}(\text{Standard}) - \text{mag}(\text{standard}(\text{obs}))) \end{aligned}$$

この式の中の $(\text{mag}(\text{Standard}) - \text{mag}(\text{standard}(\text{obs})))$ が等級原点 (ZP) と呼ばれることが多い。

$$\text{mag}(\text{Target}) = \text{mag}(\text{target}(\text{obs})) + ZP.$$

$$ZP = \text{mag}(\text{catalog}) - 2.5 \log(((\text{count} - \text{bias}) - \text{dark}' * t_{d2}) / (\text{flat}' * t_{e2}) - \text{sky})$$

であり、ZP は 1 count/s が何等に相当するかの等級である。なお、ここまでの流れを良く思い出せばわかるように ZP は flat の規格化定数によって変わるという点には十分注意が必要である。つまり、標準星と目標天体は同じ flat を使わなければならない。

4.2 SDFRED で解析した Suprime-Cam データの測光較正

SDFRED で Suprime-Cam のデータを解析すると、各チップの flat の規格化は各チップに対応する flat によって行なわれ、更にその上で、チップ間の規格化が *.mos ファイルで行なわれる。

ここで問題になるのは標準天体の解析の際には、必ずしもそのデータはチップ間の (再) 規格化が行なわれないという点である。最終的な画像の ZP を出すには、この点を考慮しなければならない。

具体例を挙げる。標準天体が *_fio.fits chip で観測されたとする。このチップから計算される零点を ZP0 とすれば、それは fio での $-2.5 \log(\text{count}/\text{sec})$ である。一方、最終的な *.mos が以下のようなだったとする。

```
bAspgfTo_RH090523object038_chihiro.fits 0.000000 0.000000 0.000000 1.000000
bAspgfTo_RH090523object038_clarisse.fits 2074.952263 1.570599 -0.000075 1.016956
bAspgfTo_RH090523object038_fio.fits 4183.941985 1.120248 -0.000017 1.080026
...
```

この *.mos から作られる最終的な FITS 画像は chihiro で規格化されている。従って、この chihiro と fio の相対比、上記例で言えば、1.080 に相当する係数だけ補正する必要があるのである。

この例では ZP0 は fio での 1 ADU/sec である。合成済画像の中では、fio の 1 ADU/sec は $1/1.080026 = 0.925904$ ADU となっている。従って ZP0 の等級は最終画像では 0.925904 ADU/sec である。この分を補正した結果、正しい零点としては以下でなければならない。

$$ZP = ZP0 - 2.5 \log(1.080026)$$

4.3 大気吸収補正

さて、ここまでの議論は標準天体は理想的に目標天体と同じ状態で観測されたとしてきた。しかし実際にはこのような理想的な場合ばかりではない。

現実的に問題になるのは、まずは大気の吸収と散乱である。この大気の効果は観測所の海拔高度 (例えば高度 600km ほどまで行けば、大気はない)、観測天域の仰角高度 (日没は赤い)、そして天候状況に左右される。

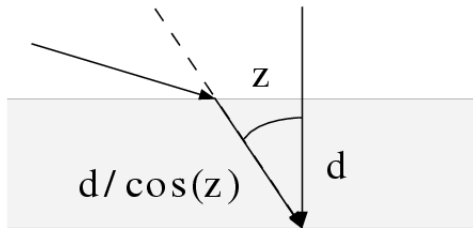
もし天候が良くない場合は、測光の絶対較正はできない。諦める。そんな夜は「非測光夜」と呼ばれる。あるデータが非測光夜のものなのか、測光の出来る状態 (測光夜) のものなのかは、観測ログがないと判断出来ない場合も多いので要注意である。まあすばるの場合、殊にリモート観測などの場合は、観測ログにも書かれてなかったりするのだが、アーカイブ利用の場合はログを見られる可能性はかなり絶望的である。ここは「警戒する」しかない。

一方、測光夜の場合、大気による吸収散乱の量は、観測波長と大気の厚みで決まるとして良い⁴。この吸収散乱の量を等級 Δm で書くことにすると、

$$\Delta m = a(\lambda) * [airmass]$$

$$airmass = \sec(z) - 0.0018167(\sec(z) - 1) - 0.002875(\sec(z) - 1)^2 - 0.0008083(\sec(z) - 1)^3 \dots$$

(Hardie 1962). となる。この主要な項は、天頂からの離角 (天頂角 z) の正接 (secant) である $\sec(z)=1/\cos(z)$ 。



この係数 $a(\lambda)$ を求めるには、2つの異なった z での零点が求まれば良い。このため、通常は1晩で1つのフィルターに対し、標準天体は少なくとも2つの異なる高度で観測される。これは同じ天体を時間を変えて2回撮る場合と、2つの標準天体を撮る場合とがある。 a が求まれば、上記式から任意の z に対する Δm を推定できる。

なお、この方法が標準的かつ正当派ではあるのだが、 $a(\lambda)$ を求めることは必ずしも較正に必須ではない。例えば、もし目標天体と標準天体が同じ高度で観測できているなら、この高度依存性はキャンセルできるからである。この一番極端な場合は、目標天域の中に標準天体がいる場合である。例えば SDSS i の較正の場合はこちらの手法が使える場合も多い。

4.4 フィルターの違い (color term)

測光較正の際に気にしなければならないもう一つの要素は標準フィルターシステムと観測の際に用いた光学系と観測装置の波長感度特性の違いである。測光標準天体の等級は標準フィルターシステムで与えられているのだが、これは必ずしも他の望遠鏡などで観測に使われるものと同じとは限らない。

この事が測光較正にどのように影響するかを考えるために、単純な例として、透過波長域が僅かにずれている場合を想定してみる。観測で使われるフィルター (A) が標準フィルター (B) にくらべ、やや長波長側にずれていたとする。この場合、赤い天体は filter A で観測した場合、filter B で観測するよりも多くの flux を受け取る。一方、青い天体では、filter A では filter B で観測するよりも flux が少なくなる。従って、標準フィルターシステムで同じ等級の天体でも filter A では違う明るさとして観測されてしまう事となる。この部分を天体の色で補正する必要がある。

最初の近似としては、この関係は色、例えば (B-V) なり (B-R) なりの線形の関数として近似できると考えられている。つまり

$$\Delta m = b * [color]$$

と書けるとする。色が異なる2つの標準天体があれば、この係数 b を求めることが出来る。

⁴敢えて書くまでもないが、観測所の海拔高度は変化しないと思っている

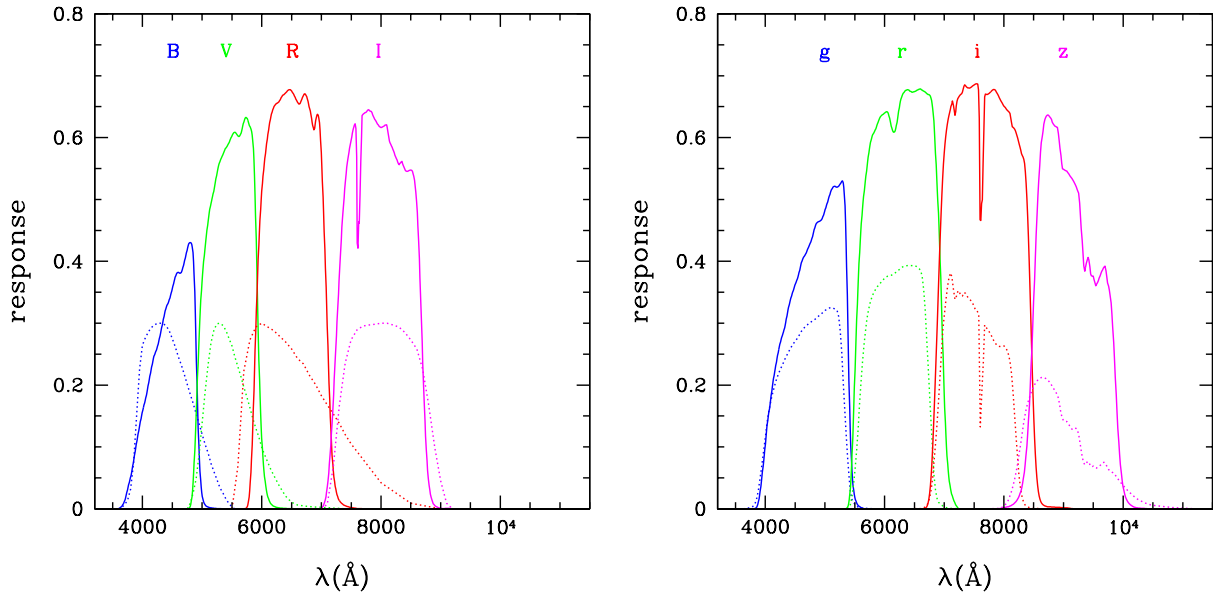


図 1: Standard filter system(破線) と Suprime-Cam filters(実線)

この場合、標準天体を2つのフィルターで観測するか、標準天体の等級が標準システムで2つ以上のバンドで与えられていれば、観測から求めた色かカタログでの色を用いることが出来る。

例えば、2つの標準天体 (std1, std2) を Suprime-Cam の R-band で観測し、その標準天体はカタログに B, R の標準システムでの等級が与えられてたとする。std1, std2 それぞれから、R バンドの零点を $ZP = \text{mag}(\text{catalog}) - \text{mag}(\text{observation})$ として求めることが出来る。

この時、color、この場合は (B-R) が 0 の時の零点を ZP_0 として、一次近似で

$$ZP1 = ZP_0 + b(\text{color}_{std1})$$

$$ZP2 = ZP_0 + b(\text{color}_{std2})$$

と書ける。この2つの式を解くことで係数 b が求まる。重要なのはこの式での color は標準システムの color だということである。

さて、この結果を目標天体に対して使う場合には、目標天体の標準システムでの色が必要となるのだが、これは標準天体を2つのバンドで測光し、そのそれぞれについて、同じ標準システムの color で上記のように color term を求めた後、連立方程式を解けば原理的には求められる。

$$R = R_{inst} + ZP(R)_0 + b_R(B - R)$$

$$B = B_{inst} + ZP(B)_0 + b_B(B - R)$$

もっとも b が十分小さい場合にはこのような作業はすっ飛ばして、(B-R) に典型的な目的天体の色を入れ、B-R の違いに起因する変化は誤差だと思って処理してしまうことも多い。

と、ここまでは一次近似として色に対して線形で近似できる場合の説明だったが、現実にはこの関数は必ずしも線形近似できるわけではない。特に Suprime-Cam の場合、B, R は Johnson Kron-Cousins との差が大変なことになっている。以下は星のスペクトルライブラリを使って求めた色依存性である。

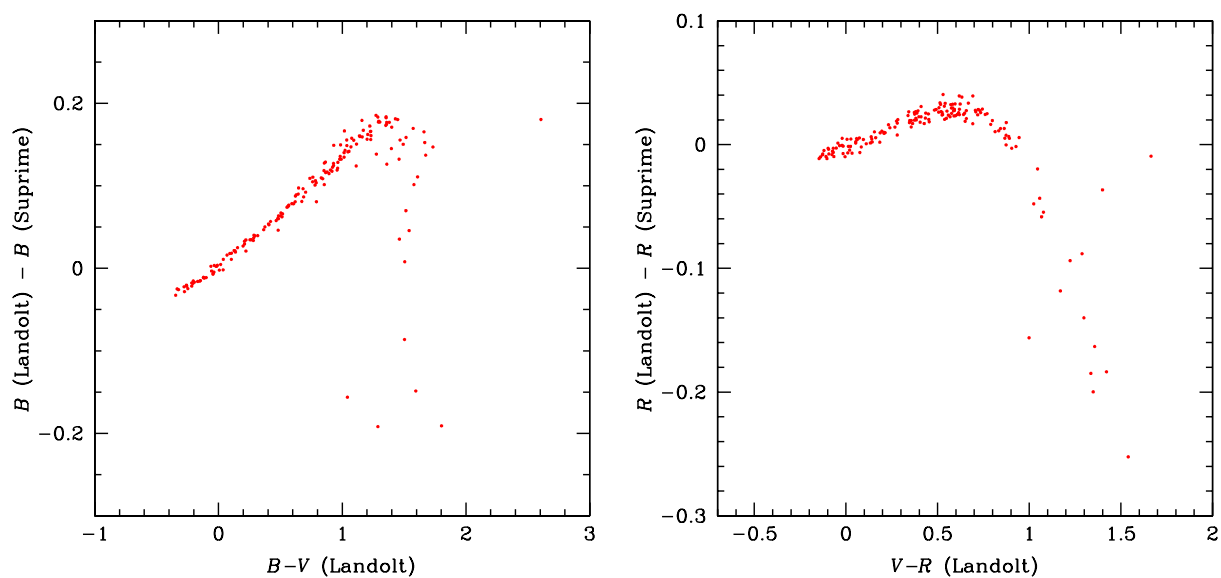


図 2: Landolt の測光システムと、Suprime-Cam の測光システムで星のモデルスペクトルを観測した場合の等級差を、Landolt での色のグラフとして書いた図

このようなカラーバンドの変換は、他の標準システムで行なわれた研究と比較する際には必須である。これは観測結果のみならず理論モデルと比較する際にも必要になりうる。一方で、このような標準システムへの変換を行わず、「装置での AB 等級」として結果を出せば十分な研究もまた多い。自分の研究が何を目標しているかをきちんと踏まえた上で、必要な較正を行なうことが必要である。

5 Landolt(1992,2009) の標準星を使って測光較正する

この章では、結構広く使われている Landolt(1992,2009) の赤道付近の選択領域 (SA) の標準星を使った R-band の較正の手順を簡単に示してみる。以下では SA104 の場合が書かれている。

5.1 カタログの準備

Landolt(1992) のカタログは CDS から取得できる⁵。アジアからオンラインカタログを探す場合は国立天文台の <http://dbc.nao.ac.jp/> から探すのが良い。

この Landolt(1992) のオリジナルのカタログは広く使われているのだが、位置精度が低かったりと問題があるので、改訂された他のカタログを使うことをお勧めしたい。本来は同著者による改訂版である Landolt(2009) を使うのが良い。しかし、これは CDS 等のカタログアーカイブにはまだ入っていないようなので、ADS 経由で AJ@IOP の論文を探し、そこから machine readable の Table 2 を拾わなければならない。所属機関等が AJ を購読していればこれで可能である。

一方、以前筆者が見つけて良く利用していたのは以下の2つである。

`ftp://ftp.lowell.edu/pub/bas/starcats/landolt.all`

`http://web.pd.astro.it/blanc/landolt/landolt_usno.1.dat`

今後は Landolt(2009) の利用をお勧めしたいが、Landolt(2009) の利用が難しい場合はこれらでも代替できる。

さて、カタログには色々な情報が書かれているが、星の名前、RA、Dec、等級、そしてエラーが重要な情報である。R-band の較正には V と V-R が必要となる。これら必要なカラムを以下のように抜き出すことにしよう。どうやってこのように抜き出すかは各自工夫して欲しい。

#name	RA(2000)	Dec(2000)	V	V-R	e(V)	e(V-R)
104-306	12:41:03.59	-00:37:12.4	9.370000	0.832000	0.006680	0.002010
104-423	12:41:36.01	-00:31:11.3	15.602000	0.262000	0.024750	0.036060
104-428	12:41:41.33	-00:26:25.8	12.630000	0.534000	0.003800	0.002400
104-L1	12:41:49.45	-00:21:00.7	14.608000	0.374000	0.000710	0.011310
104-430	12:41:50.31	-00:25:52.3	13.858000	0.364000	0.005160	0.004650
104-325	12:42:02.27	-00:41:35.4	15.581000	0.345000	0.016260	0.025460
...						

このようにカタログを整形するには、cut,paste,join,awk などの標準コマンドを使ってもいいし、perl なり ruby なりを使っても、あるいは自分でプログラム書いてもいい。ともあれ、整形して SA104.cat というファイル名で作成しておく。

抜き出したカタログを後々 ds9 で使うため、region ファイルにも変換しておく。

```
$ awk 'BEGIN{printf("fk5\n");}  
{printf("CIRCLE %s %s 10\" # text={ %s }\n",$2,$3,$1)}' SA104.cat > SA104.reg
```

⁵一般にカタログを論文から目で見ても打ち込んだりすると間違えるので、オンラインで利用できる物を探す努力をすべきである

5.2 Suprime-Cam データの測光

次に Suprime-Cam のデータで星の明るさを測ることになるのだが、以下、簡単だが精度はそんなに高くない方法を示す。以下の例では歪みの補正まで済んだ標準天体視野 (SA104) を “gfTo_RH090523object021_*” だとして話を進めている。

まず、バックグラウンドを引き去る。この時、メッシュの大きさは多少大きめ、例えば 256 などにしておく。

```
$ ls gfTo_RH090523object021_*.fits > standard_skysb.lis
$ skysb.csh standard_skysb.lis 256
```

これで、スカイが引かれた画像が一式出来上がる (sgfTo_*.fits)。

次に、ここがトリッキーなのだが、目標天体と標準星天体ではチップ間の相対感度は変わっていないという仮定のもとで、目標天体の “object/all.mos” の最初の 10 行 (1shot 10chip 分) を standard.mos として切り出す。

```
bAspgfTo_RH090523object038_chihiro.fits 0.000000 0.000000 0.000000 1.000000
bAspgfTo_RH090523object038_clarisse.fits 2074.952263 1.570599 -0.000075 1.016956
bAspgfTo_RH090523object038_fio.fits 4183.941985 1.120248 -0.000017 1.080026
...
```

このファイル名を作っておいたスカイが引かれた画像名に編集する。

```
sgfTo_RH090523object021_chihiro.fits 0.000000 0.000000 0.000000 1.000000
sgfTo_RH090523object021_clarisse.fits 2074.952263 1.570599 -0.000075 1.016956
sgfTo_RH090523object021_fio.fits 4183.941985 1.120248 -0.000017 1.080026
...
```

この編集済の standard.mos を使って、標準天体 10chip 分を繋いでしまう。

```
$ imcio2a -dist_add -dtype=FITSFLOAT -pixignr=0 standard.mos standard.fits
```

この出力の “standard.fits” では、相対的な flux は *.mos の 5 カラム目が 1.00000 になっている 1 行目に規格化されており、このチップ (chihiro) はコピーしてきた元の目的天体と同じなので、前の章で述べたチップ間の相対 flux の補正の問題はこの手順を踏む限り気にしなくて良くなる。

さて、この出力された standard.fits で測光を行なう。どの星が標準星なのかは Landolt 論文の finding chart を見つつ、前節で作成した *.reg を利用するとわかりやすいであろう。なお、この時点で standard.fits は位置較正されていないので、10-20 arcsec ほど位置はずれていると思われるが、それでも Landolt 標準星は十分明るいので同定は容易なはずである。IRAF でも ds9+IRAF でも SExtractor でも、あるいは更に他のプログラムでも好きなもので測光すれば良い。いくつかの方法は次節以降で述べる。

注意すべきことは、ここで求めるのは APERTURE FLUX(開口 flux) あるいは APERTURE MAGNITUDE (開口等級) であるという点である。これにはいくつか理由があるのだが、まず Suprime-Cam では普通に Landolt 標準星を撮ると 1 秒以下で中心がサチってしまうため、大幅にピンボケした状態で撮っている。このため PSF の形は普通の中心集中した綺麗な形ではなく、広がっていて、場合によってはドーナツ状になっていたりするのである。このため、星の測光に良く使われる PSF MAGNITUDE などを使うと大幅に道を誤ってしまう。更に Landolt のカタログ自体が直径 $\phi = 14$ arcsec の開口等級を与えているため、それと近い測定を行なうのが良いと言う面もある。つまり、暗い天体が脇にいた場合、その天体を含まないような測光ではなく、それも含めて測光する方が Landolt の測光力

タログを使う場合は正しい、という理解である。ちなみにこの 14 arcsec という直径は Suprime-Cam では 69.3 pix である。このため、バックグラウンドを引く際に、星広がりの裾などを引き過ぎてしまわぬように十分大きなメッシュを取ったのである。

なお、本当はこの測光の際に誤差も求めて最終的な零点推定に使うべきなのだが、以下の例ではここはすっかり忘れて書いてしまっていた。ごめんなさい。将来の改訂バージョンでは直したいと思います⁶。なお、以下3つの方法が書いてあるが、どれか1つを行えば良い。

5.2.1 IRAF を使う場合

“imexam” に慣れていれば話は簡単である。stdimage を imtu2 に設定した後、IRAF を起動し、ds9 を “-fifo /dev/imt2” オプションつけて起動する。

```
$ ds9 -fifo /dev/imt2 standard.fits &
```

imexam の動径方向パラメタを変更する。このためのタスクが epar である。ここでは半径が必要なので 34.6 pix を入れる。

```
cl> rimexam
...
(radius =                34.6) Object radius
(buffer =                  5.) Background buffer width
(width =                   10.) Background width
```

この epar はカーソルの動きや編集の仕方がわかりにくいのだが、デフォルトは vi と同じである。カーソルを→で radius まで持っていき、そこで 34.6 と入力、改行、一つ飛ばして width で 10. と入力、改行、終わった後 :wq と入力、改行、で反映出来る。なお、この挙動を emacs にする方法があるらしいのだが、筆者は成功していない⁷。なお、この buffer と width が適切なのかどうかは筆者は吟味してない。

rimexam のパラメタを編集したら、imexam である。カーソルが輪っかになったら、標準星に大体合わせて、a キーを押す。以下、104-470, 104-455, 104-457 について試した例。

#	COL	LINE	COORDINATES								
#	R	MAG	FLUX	SKY	PEAK	E	PA	BETA	ENCLOSED	MOFFAT	DIRECT
	792.84	4509.21	792.84	4509.21							
	14.08	10.14	881494.	5.238	34391.	0.08	48	16.0	4.31	4.61	4.69
	3043.48	6159.65	3043.48	6159.65							
	13.03	10.93	425237.	3.189	18054.	0.13	38	5.89	4.05	4.70	4.34
	2881.50	4816.81	2881.50	4816.81							
	12.72	11.72	204484.	1.323	9174.	0.10	44	6.36	3.91	4.66	4.24

必要なのは2行目3カラム目の FLUX の値である。これをコピー&ペーストして以下のようなファイル (suprime.cat) を作る。なお、読みにくいが PEAK (2行目5カラム目) の値が 45000 を越えている場合、サチっている可能性があるので、そのようなデータはこの後の解析に入れないようにする。

⁶と書いてから、ここは延々改訂されてないわけだが

⁷emacs のつもりで C-x C-c と押したりすると、C-x は無視され C-c で強制終了になる。C-x C-s と押すと、スクリーン停止になったりしてややこしい

```
104-470 881494.
104-455 425237.
104-457 204484.
...
```

5.2.2 SExtractor を使う場合

(筆者のように) IRAF が嫌いな場合は SExtractor を使っても良い。この場合、出力パラメタファイル (photo.param) を以下のように設定する。

```
NUMBER
X_IMAGE
Y_IMAGE
FLUX_APER(1)
FLUXERR_APER(1)
FLAGS
```

位置較正の場合は XWIN_IMAGE, YWIN_IMAGE を使ったが、今回は星は中心集中してない変な形状なので、単純重心 X_IMAGE, Y_IMAGE の方が良い。もう一つのパラメタファイル (photo.sex) は

```
$ sex -d > photo.sex
```

で、デフォルトを吐いた後、例えばこんな感じに設定する。

```
CATALOG_TYPE      ASCII
PARAMETERS_NAME    photo.param  # name of the file containing catalog contents
DETECT_MINAREA     10           # minimum number of pixels above threshold
DETECT_THRESH      10           # <sigmas> or <threshold>,<ZP> in mag.arcsec-2
FILTER              N
SATUR_LEVEL        45000.0      # level (in ADUs) at which arises saturation
PHOT_APERTURES     69.3         # MAG_APER aperture diameter(s) in pixels
BACK_SIZE          256          # Background mesh: <size> or <width>,<height>
```

他の部分はデフォルトのままで良い。

これで SExtractor を実行すると、

```
$ sex -c photo.sex standard.fits
```

test.cat が作られる。この内容は以下のような感じである。

```
1  7780.392    38.628    7060.612    242.0873    0
2  1932.326    45.783    1459.556    242.0706    0
3  6390.656    48.530    14626.65    242.068     0
4   356.454    58.379    2858.758    242.0591    0
...
```

ここで、検出した中からサチった星を捨てておくのが良い。SExtractor の場合、サチったピクセルを含む検出には 4 のビットのフラグが立つ。つまり、フラグの値を 8 で割ったあまりが 4 より大きかったら、それはサチっていると SExtractor が判断した天体である。このような検出だった場合はデータを捨てる。例えば以下のようなエントリである。

758	5414.925	3715.250	2418351	237.1149	4
1266	6189.725	6207.583	2605862	237.1225	6

また、視野端等で aperture が取れなかった場合は 16 が立つ。これも捨てる。

815	34.003	3996.474	2038.01	236.6959	16
-----	--------	----------	---------	----------	----

さて、このカタログ中から標準星を見つけなければならない。Landolt カタログでやったのと同じように ds9 用の region ファイルを作る方法をここでは紹介する。以下のようにまず test.reg を作る。ここで、上述のフラグ判断を入れると良い。以下では手抜きをしてフラグが 0 でなければ駄目としている。

```
$ awk 'BEGIN{printf("image\n");}
{if ($6==0) {c="yellow"} else {c="red"} printf("CIRCLE %s %s 10\" # text={ %s }
color=%s\n",$2,$3,$1,c)}' test.cat > test.reg
```

この test.reg と SA104.reg を ds9 上に読み込めば、Landolt 標準星（但しこれは位置がずれているが）が SExtractor の出力の何番に対応しているかが分かる。

これで、以下のような supprime.cat を手作業で作ることができる。

```
104-470 902396.6
104-455 437396.9
104-457 209862.5
...
```

なお、SExtractor はチップ間の IGNRVAL を理解してくれないので、そこは人間の目で見えて除去する必要がある。テストデータの例で言えば、104-461 が該当する。

5.2.3 With WCStools(sky2xy) and IRAF apphot

ここまでどうしても星の同定に手作業が入っていたのだが、前の章で示した方法で standard.fits の位置較正を行なって、standardw.fits を作ってしまえば、標準星の位置は WCStools の sky2xy を使うことで、x,y で求めることができるようになる。

まずは standardw.fits を作っておく。次に、測光したい標準星の位置の正確なカタログ (coordinates.dat) を以下のように準備する。最初の 3 つのカラムは RA, Dec, Equinox である必要がある。

```
12:41:03.59 -00:37:12.4 J2000 104-306
12:41:36.01 -00:31:11.3 J2000 104-423
12:41:41.33 -00:26:25.8 J2000 104-428
...
```

ここで sky2xy を実行する。

```
$ sky2xy standardw.fits @coordinates.dat

12:41:03.59 -00:37:12.4 J2000 -> 11085.424 2277.007 (off image)
12:41:36.01 -00:31:11.3 J2000 -> 8686.673 4078.410
12:41:41.33 -00:26:25.8 J2000 -> 8298.872 5494.662
12:41:49.45 -00:21:00.7 J2000 -> 7704.100 7108.161
...
```

これで、変換後にどの座標に来るかわかる。この出力結果は 1 行目が coordinates.dat の 1 行目に対応するので、元の coordinates.dat と組み合わせて、FITS 上の位置のカatalog (positions.dat) を作成する例えば筆者は以下のようなコマンドを利用する。

```
$ sky2xy standardw.fits @coordinates.dat | paste - coordinates.dat |  
grep -v off | awk '{print $5,$6,$10}' > position.dat  
$ cat position.dat  
  
8686.673 4078.410 104-423  
8298.872 5494.662 104-428  
7704.100 7108.161 104-L1  
...
```

ここまで出来れば、開口測光を自動化することもできるようになる。例えば IRAF の noao.digiphot.apphot.phot を使う例が以下である。再度書くが筆者は IRAF は良く分からないのでこの部分の質問は受け付けられない。

```
cl> digiphot  
      apphot.  daophot.  photcal.  ptools.  
di> apphot  
      aptest      findpars@    pconvert      polymark      psort  
      center      fitspf       pdump         polypars@     qphot  
      centerpars@ fitsky       pexamine      polyphot      radprof  
      daofind      fitskypars@  phot          prenumber     wphot  
      datapars@    pcalc       photpars@     pselect  
ap> epar phot  
...  
image      =      standardw.fits  The input image(s)  
(coords =      position.dat) The input coordinate files(s) (default: image.co  
(interac=      no) Interactive mode ?  
...  
ap> fitskypars  
...  
  
(salgori=      constant) Sky fitting algorithm  
...  
ap> phot  
The input image(s) (standardw.fits):  
  
Centering algorithm (centroid) (CR or value): none  
      New centering algorithm: none  
Sky fitting algorithm (constant) (CR or value):  
      Sky fitting algorithm: constant  
File/list of aperture radii in scale units (3.) (CR or value): 34.6  
      Aperture radius 1: 34.6 scale units 34.6 pixels  
Standard deviation of background in counts (INDEF) (CR or value):
```

```

New standard deviation of background: INDEF counts
Minimum good data value (INDEF) (CR or value):
New minimum good data value: INDEF counts
Maximum good data value (INDEF) (CR or value): 50000
New maximum good data value: 50000. counts

```

この結果、standardw.fits.mag.1 が作成される。
これは例えば以下のようなファイルである。

```

...
#N IMAGE          XINIT      YINIT      ID      COORDS          LID      \
#U imagename      pixels     pixels     ##      filename       ##      \
#F %-23s          %-10.3f    %-10.3f    %-6d    %-23s          %-6d
#
#N XCENTER        YCENTER    XSHIFT    YSHIFT    XERR      YERR          CIER CERROR \
#U pixels         pixels     pixels     pixels     pixels     pixels       ##  cerrors \
#F %-14.3f        %-11.3f    %-8.3f    %-8.3f    %-8.3f    %-15.3f      %-5d %-9s
#
#N MSKY           STDEV        SKEW        NSKY      NSREJ        SIER SERROR \
#U counts         counts       counts       npix      npix         ##  serrors \
#F %-18.7g        %-15.7g     %-15.7g     %-7d      %-9d         %-5d %-9s
#
#N ITIME          XAIRMASS    IFILTER          OTIME          \
#U timeunit       number      name            timeunit       \
#F %-18.7g        %-15.7g     %-23s          %-23s
#
#N RAPERT         SUM          AREA         FLUX         MAG      MERR      PIER PERROR \
#U scale          counts       pixels       counts       mag      mag      ##  perrors \
#F %-12.2f        %-14.7g     %-11.7g     %-14.7g     %-7.3f  %-6.3f  %-5d %-9s
#
standardw.fits    8690.500  4078.596  1      position.dat    1      \
8690.500  4078.596  0.000  0.000  INDEF  INDEF          0      NoError \
0.          INDEF          INDEF          0      0          0      NoError \
1.          INDEF          INDEF          INDEF          \
34.60      264651.      3761.67    264651.      11.443  0.002  0      NoError
standardw.fits    8302.380  5493.040  2      position.dat    2      \
8302.380  5493.040  0.000  0.000  INDEF  INDEF          0      NoError \
0.          INDEF          INDEF          0      0          0      NoError \
1.          INDEF          INDEF          INDEF          \
34.60      4907146.      3760.498  4907146.      INDEF  INDEF  305  BadPixels
...

```

これを編集して supprime.cat を作成できる、はずである。この時も、サチっていないか確認が必要であるが、最後の PERROR が NoError であるかどうかで確認できる。

5.3 図を描いて零点を求める

ともあれ、SA104.cat と supprime.cat が準備できたとする。

```
$ cat SA104.cat

#name    RA(2000) Dec(2000) V      V-R    e(V)      e(V-R)
104-306  12:41:03.59 -00:37:12.4  9.370000 0.832000 0.006680 0.002010
104-423  12:41:36.01 -00:31:11.3  15.602000 0.262000 0.024750 0.036060
104-428  12:41:41.33 -00:26:25.8  12.630000 0.534000 0.003800 0.002400
104-L1   12:41:49.45 -00:21:00.7  14.608000 0.374000 0.000710 0.011310
104-430  12:41:50.31 -00:25:52.3  13.858000 0.364000 0.005160 0.004650
104-325  12:42:02.27 -00:41:35.4  15.581000 0.345000 0.016260 0.025460
...
$ cat supprime.cat

104-470  881494.
104-455  425237.
104-457  204484.
...
```

これらは、1 カラム目が一致していれば、UNIX の “sort” と “join” を使って簡単に結合できる。

```
$ sort SA104.cat > SA104.sorted
$ sort supprime.cat | join SA104.sorted - > joined.cat
```

この結果、以下ようになる。

```
$ cat joined.cat

104-325 12:42:02.27 -00:41:35.4 15.581000 0.345000 0.016260 0.025460 259311.
104-330 12:42:11.45 -00:40:40.5 15.296000 0.369000 0.010330 0.012390 363746.
104-336 12:42:24.67 -00:39:57.8 14.404000 0.461000 0.006680 0.005880 901022.
...
```

R-band の測光原点は

$$ZP = V - (V - R) + 2.5 \log_{10}(\text{count}_{\text{supprime}}) - 2.5 \log_{10}(\text{EXPTIME})$$

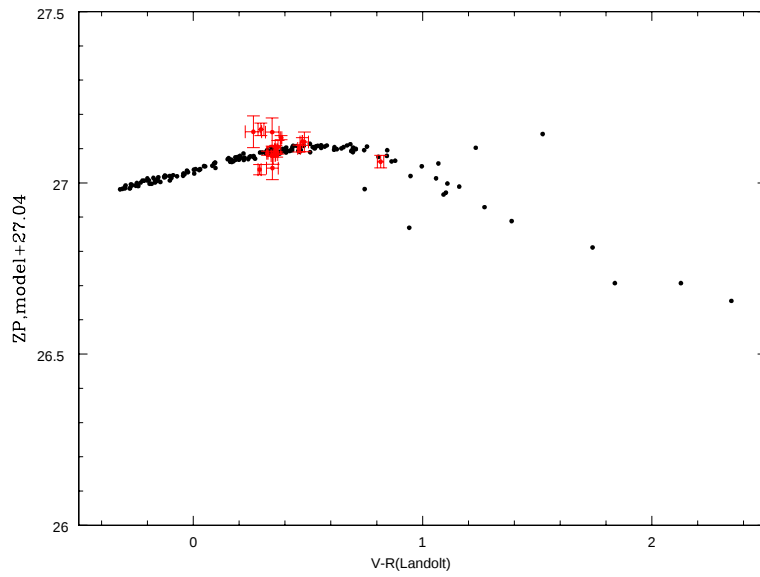
と計算できる。露出時間 (EXPTIME) が 5 秒の場合、awk を使って

```
$ awk '{e=5;printf("%s %s %f %f\n",$1,$5,$4-$5,
    $4-$5+2.5*log($8)/log(10)-2.5*log(e)/log(10))}' joined.cat

104-325 0.345000 15.236000 27.023127
104-330 0.369000 14.927000 27.081571
104-336 0.461000 13.943000 27.082413
...
```

と計算できる。この出力は、左から星の名前、V-R, R, 零点である。

星のスペクトルモデルから計算された値とこの結果を重ねると以下の図である。観測点にはカタログと測光値のエラーも付けてみた。



ここから、 $ZP=27.04$ と求められる。

なお、過去の実績から Suprime-Cam(2009/01-) では、R-band の零点は、天気の良い時に chihiro で、およそ $\sim 27.2\text{-}27.3$ mag/sec である。従って上記標準星は、多少大気の透過率が悪い時に撮られたのだろうと推測できる。