小望遠鏡による基礎観測実習 (2009 年度)

平成 21 年 12 月 23 日

1　この実習の目的

東大天文センターの 30cm カセグレン望遠鏡を用い、CCD カメラによる基本的な天体観測を行います。望遠鏡の基本操作法、赤経・赤緯に基づいた天体観測の基本事項、および CCD 観測の基本技術の一通りを習得することを目指す。

毎年、具体的なテーマを一つ決めて、CCD カメラや望遠鏡の改良を行い、感度の向上をはかる。思った感度向上が得られているかを観測を通して検証することにより、天体観測において最も重要な量の一つである「限界等級」について理解を深めてもらたい。

今年度は、

観測から、昨年度の望遠鏡のミラー清掃後、システム効率がどの程度になっているかを実際の観測で確認する

限界等級を算出する

実際に観測を行って、その限界等級が得られるかを調べる

行う。

特に、一昨年度の測定ではシステム効率が 5%だったはずが、昨年度の主鏡清掃後はなぜか 4%という更に低い値がでているので、これがただしかったかを確認するのが重要な任務になるだろう。

望遠鏡等のマニュアル/資料などは

`http://www.ioa.s.u-tokyo.ac.jp/~kmotohara/30cm/`

にありますので、事前に目を通しておくように。

2　限界等級

2.1　システム効率とは

単純に言ってしまえば、

『望遠鏡に入射した星からの光子のうち、CCD で電荷となって読出されたものの割合』

。

ということで、

天体から望遠鏡に入射した光子数 (s_i (個/s))

$$s_i = \pi \left(\frac{D}{2}\right)^2 \frac{\Delta\lambda F_\lambda}{h\nu} \tag{1}$$

ここで、

D :望遠鏡の口径
F_λ :天体からのフラックス
$\Delta\lambda$:フィルターの波長範囲

と、

CCD で生じた電荷の個数 (n_i (e^-/s))

$$n_i = \frac{N f_{conv}}{t} \tag{2}$$

ここで、

N　　:画像上でのカウント
f_{conv}:コンバージョンファクター
t　　:積分時間

とすると、システム効率 η は

$$\eta = \frac{n_i}{s_i} \tag{3}$$

で求まる。

2.2　限界等級 (単素子による検出の場合)

それをふまえた上で、限界等級はどのように定義されるかであるが、一般的には S/N という量で測る。すなわち、ノイズに対して信号がどの程度来ているかを評価し、それがある一定の値を超えれば検出できた、ということにする。通常光赤外では $S/N = 5$ をこえれば検出できた、とすることが多い。

で、その S/N は

$$\begin{aligned} S/N &= \frac{n_i t}{N_{\mathrm{noise}}} \\ &= \frac{\eta\, s_i t}{N_{\mathrm{noise}}} \end{aligned} \tag{4}$$

のように書ける。ここで $N_{\rm noise}$ はノイズ成分、t は積分時間。ノイズ成分は入射光子ののポアッソンノイズと検出器からの読出しノイズでほぼ占められており、

$$N_{\rm noise} = \sqrt{n_i t + n_{\rm sky} t + n_{\rm dark} t + N_{\rm read}^2} \tag{5}$$

と書ける。ここで、$n_{\rm sky}({\rm e}^-/{\rm s/pix})$ は単素子あたりの検出された背景放射の光子数、$n_{\rm dark}({\rm e}^-/{\rm s/pix})$ は検出器の単素子あたりの暗電流、$N_{\rm read}({\rm e}^-{\rm r.m.s./pix})$ は検出器からの単素子あたりの読出しノイズ。

ここまでは、検出器が単一素子の場合を考えた。それでは、CCD のように複数の素子に星像が結像する場合はどうなるのか?

2.3 限界等級 (複数素子による検出の場合)

複数素子の場合は複数の素子からのノイズを考慮する必要があるが、それらは統計的に足しあわせればいい。すなわち、m 個のピクセルに広がった像を検出する場合、ノイズ成分 $N_{\rm noise}$ は

$$N_{\rm noise} = \sqrt{n_i t + m n_{\rm sky} t + m n_{\rm dark} t + m N_{\rm read}^2} \tag{6}$$

となる。

3 CCD カメラ

3.1 CCD について

- CCD の動作原理は `http://www.kusastro.kyoto-u.ac.jp/~iwamuro/LECTURE/OBS/detector.html` を参照。
- データについて具体的には西浦版『可視光域データ・リダクション法』を参照。

3.2 実習用 CCD カメラの使い方

起動は、

- まずは制御箱の電源を入れる
- デスクトップ `=> Camera => Hpc50216.exe` でソフトウェア起動
- その後冷却が安定するまで 30 分ほど待つ

です。その他の主要な手順は;

- メニューから `Expose => Expose Control` で各種設定
  - 積分時間
  - ダーク/バイアス引き、フラットフィールドをするかどうか (今回はしない)
- メニューから `Expose => Full Frame Exposure (Ctrl-e)` で露出
- メニューから `Initialilze => Configure` で設定
  - バイアス、ダーク、フラットフレームの指定 (今回はしなくていい)
  - 画像保存先のパスの指定
- メニューから `File => Save (Ctrl-s)` で画像保存
  - この時、 unsigned-short で保存するように。

3.3 CCD カメラのデータの扱い

3.3.1 ドーム内 PC への転送

取得したデータは、Linux マシンである golf に転送していじることになる。Windows98 マシンの beatle の デスクトップ `=> ccdimages` フォルダは、 golf の `/home/ccdimages` になっているので (図 2 参照)、ここにコピーすればいい。

この `/home/ccdimages` から自分の作業領域にコピーしてきて解析をする必要があるのだが、いくつか注意点を。

- IRAF で画像を扱うのだが、この際に拡張子が ".FTS" ではちゃんと認識してくれない。".fits" に変更する必要がある。

- 出力される画像のピクセルタイプが unsigned short になっているのだが、画像のヘッダにはなぜか signed short とかかれている (らしい)。なので、ヘッダを unsigned short と書き直してあげる必要がある。

上記二つの作業をするためには、たとえば test.FTS という画像があった場合には IRAF 上で

```
cp test.FTS test.fits
chpixtype test.fits mod-test.fits ushort
```

のようにすればいい。

3.3.2 実験室 PC への転送

さらに、ioa09, ioa10, ioa11 で扱うためには golf からリモートコピーをすることになる。golf は ioa09/10/11 からは vw として見えているので (図 2)、たとえば `/home/ccdimages/test.FTS` というファイルを持ってきたい場合はコマンドラインから

```
scp tct2008@vw:/home/ccdimages/test.FTS test.fits
```

のようにすればいいわけである。そのほか応用編として

- `/home/ccdimages/hogehoge` というディレクトリ以下すべてを現在のディレクトリに転送したい場合は

  ```
  scp -r tct2008@vw:/home/ccdimages/hogehoge .
  ```

- `/home/ccdimages/test.FTS` をホームディレクトリ直下にある `obs` というにコピーしたい場合は

  ```
  scp tct2008@vw:/home/ccdimages/test.FTS ~/obs/
  ```

- `/home/ccdimages/test1.FTS`, ... `/home/ccdimages/test100.FTS` と “test” で始まる FITS ファイルをホームディレクトリ直下にある `obs` というにコピーしたい場合は

  ```
  scp tct2008@vw:/home/ccdimages/test*.FTS ~/obs/
  ```

4 観測

4.1 今夜の星空

xplns を使う。

4.2 観測する星の選出

4.2.1 コンバージョンファクター測定用

xplns を使って探す。表示される等級は V 等級。

4.2.2 限界等級測定用

11 等級よりくらいものなので、オンラインのカタログを用いる。操作が GUI で直感的な aladin sky atlas（ `http://aladin.u-strasbg.fr/` ）を使う。
なおカタログの等級は必ずしも V 等級ではないので、必要であれば V 等級を何らかの方法で推定する必要がある。

4.3 キャリブレーションデータの取得

4.3.1 バイアスとダーク

CCD カメラマニュアル参照。

4.3.2 フラット

望遠鏡鏡筒前面をに白い紙を置き、懐中電灯で照らして取得。数枚とる必要があるだろう。

5　データ解析

5.1　用いる計算機群の構成

使う計算機群は図 2 のようになっている。データの流れは

- 取得データは beatle 上から golf にコピー (Windows 上で行う)
- それを、ioa09/10/11 から golf にアクセスして、コピー
  ただし、ioa09/10/11 からは、golf は vw として見える。
- ioa09/10/11 上で解析を行う

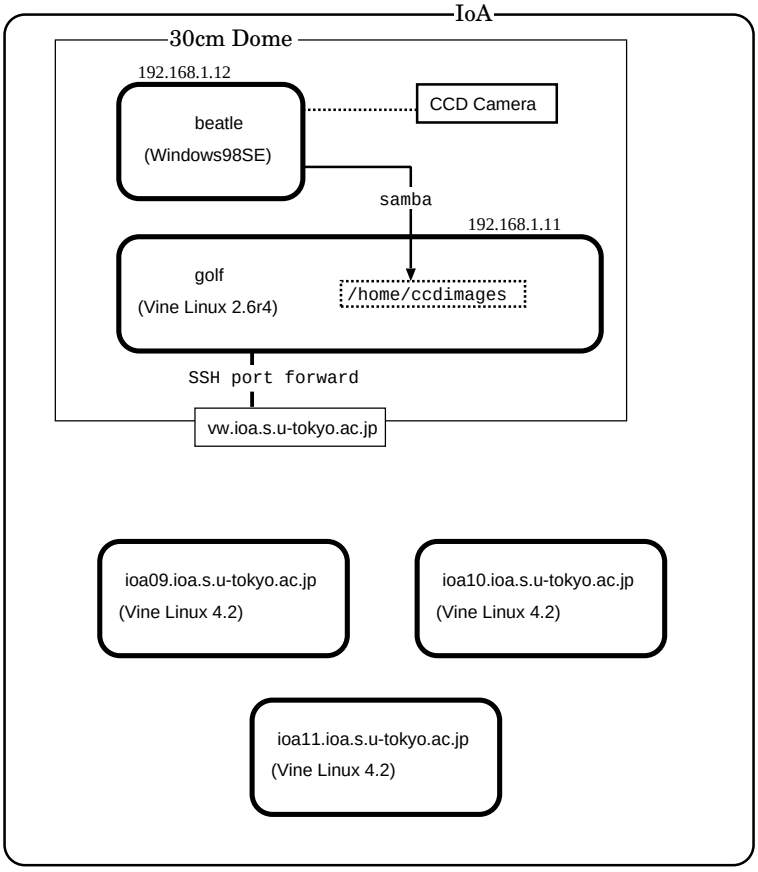

図 1: 計算機群の構成

5.2　アカウントとパスワード

golf/ioa09/10/11 のアカウント、パスワードは：

`tct2009 / ***実習時に教えます***`

5.3 UNIX コマンドたちを使いましょう

5.3.1 ディレクトリに関する基本的な約束ごとなど

- 現在のディレクトリは
  `.`
- 一つ上のディレクトリは
  `..`
- ということは、もう一つ上のディレクトリは
  `../..`
- ホームディレクトリは `~/`

と表される。
現在のディレクトリを知りたい場合は
`pwd`
と打つ。

なお、一般的なディレクトリ構造は下図のようになっている。

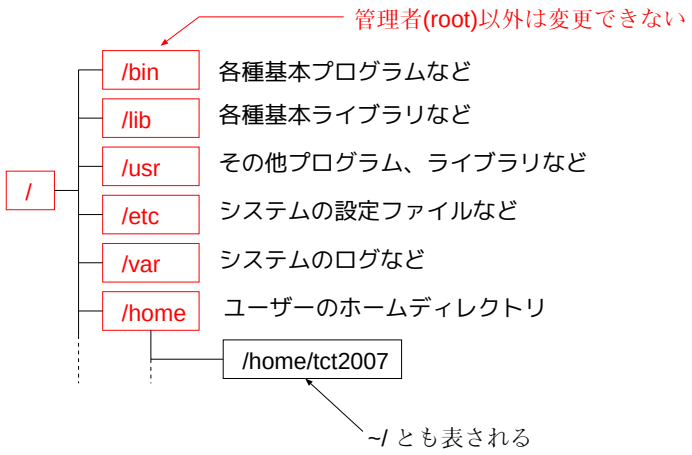

図 2: UNIX 系 OS のディレクトリ構造の一例

5.3.2 正規表現

文字列の集合を表現する約束ごと。いろいろあるが、とりあえず役に立ちそうなものを以下に示す。

- `*` ：任意の文字集合
  たとえば、あるディレクトリで `tmp` で始まるファイルだけを表示したい場合は

```
ls tmp*
```

とする。あるいは、.FTS という拡張子をもつものだったら

```
ls *.FTS
```

とする。

- ？ ：任意の一文字
  *が複数文字を代用するのに対して、一文字だけを表す。
  たとえば、拡張子が 2 文字のファイルだけを表示する場合は

  ```
  ls *.??
  ```

  とする。ls *.* とすると . が入っているファイルすべてが表示されてしまう。

- [ ] ：括弧内に含まれる一文字
  たとえば、test のあとに数字一文字がついた.FTS をホームディレクトリにコピーしたい場合は

  ```
  cp  test[0123456789].FTS ~/
  ```

  とする。これは代わりに

  ```
  cp  test[0-9].FTS ~/
  ```

  と書くことも可能。
  二桁の数字がつくファイルの場合には

  ```
  cp  test[0-9][0-9].FTS ~/
  ```

  とすればいい。

5.3.3 ssh/scp

他のワークステーションにログインする。コピーする

- foo.ioa.s.u-tokyo.ac.jp に bar というユーザー名でログインしたいときは
  `ssh foo.ioa.s.u-tokyo.ac.jp -l bar`
- foo.ioa.s.u-tokyo.ac.jp に bar というユーザー名のホーム下にある aho.txt というファイルをカレントディレクトリにコピーしたいとき
  `scp bar@foo.ioa.s.u-tokyo.ac.jp:aho.txt .`

5.3.4 less, more, cat

ファイルの中身を見たいときに使う。less, more は端末に出せるところまで表示して、続きはスペースキーなどで見ていくことが可能。cat はファイル全部を一気に端末に流し出す。

5.3.5 テキストファイルの編集

テキストファイルの編集をおこなうメジャーなものは vi と emacs が挙げられる。他にも色々あるが。

`emacs aho.txt` のように編集したいファイルを指定して起動。

基本的にマウスではなく、いろんなキー操作で編集するので、詳しいことはマニュアル本を見よ。

5.3.6 gawk

gawk はファイルテーブルの操作や、様々な計算が簡単にできる。

- たとえば

  ```
  1 4   20.31
  2 50  40.1
  3 5   60.3
  ```

  というテーブルが aho.txt という形であって、各行で 2 番目と 3 番目の要素を足し合わせて多テーブルに変換したい場合

  `gawk '{print($1,$2+$3)}' aho.txt`

  とすればいい。さらに、その結果を boke.txt に書き込みたいのなら

  `gawk '{print($1,$2+$3)}' aho.txt > boke.txt`

  とする。

- 電卓のようにしても使える。

  $$10 \times 2 \times \frac{5}{2 \times \sin(5)} \times \exp(10.4/300)$$

  だと

  `gawk 'BEGIN{print(10*2*5/(2*sin(5))*exp(10.4/300))}'`

  とすればいい。

  変数を定義して使うこともできる。$a = 10\sqrt{5}$, $c = 3 \times 10^{10}$ として

  $$100 * a/c$$

  を計算したいのであれば、

  `gawk 'BEGIN{a=10*sqrt(5);c=3e10;print(100*a/c)}'`

  とする。

  常用対数関数がないので要注意である。

  `gawk 'BEGIN{print(log(100))}'`

  は $\ln(100)$ を表示する。

他にもいろんなことができる。詳しくはマニュアルとかを見よ。

5.3.7 リダイレクト、パイプ

あるコマンドの出力を、ファイルに流し込みたいときはリダイレクト。" > "を使う。さっきの
`gawk '{print($1,$2+$3)}' aho.txt > boke.txt`
とか。

さらにその出力をべつのコマンドに流し込みたいときはパイプ " | "を使う。さっきの
`gawk '{print($1,$2+$3)}' aho.txt | gawk '{print($2-$1)}'`
のようにする。

5.4 IRAF の操作

- まず、作業するディレクトリで
  `mkiraf`
  を実行。このとき、terminal type を聞いてくるので、xterm と答える。

- `xterm &`
  を実行して xterminal を立ちあげる。

- `ds9 &`
  を実行して画像ビュワーを立ち上げる

- 先ほど立ち上げた xterm 内部で `cl`
  で IRAF を立ち上げる。

  以下、IRAF の様々なコマンドたち。() で囲った引数は必須、[] はオプション。オプションは一部しか書いていないので、詳しいことは

  `epar (command)` で引数のチェック
  `help (command)` でヘルプ

  などでチェック。

  主なコマンドは以下のようなものがある。詳しくは参考リンクを見るか、聞いてください。

  - `display (filename) (frame) [fi+] [zr-] [zs-] [z1=xxxxx] [z2=xxxxx]`
    画像を ds9 の (frame) に表示する。

    `fi+`で全画面表示。

    `zr- zs-`とすると自動スケーリングが off になって、`z1=xxxx z2=xxxx` で表示スケールを指定する必要がある。

  - `imstat (filename)`
    画像の統計量を調べる

  - `imhead (filename)`
    画像のヘッダ情報を表示する。
    全ヘッダ情報は

imhead (filename) longhea+
で見れる。

- imcombine (filelist) (output)
(filelist) で指定したファイルをすべてたし合わせて平均する (オプションで和にもできる)。(filelist) は
  - filename1,filename2,filename3,...
    のようにコンマで書き連ねるか、
  - ファイル名リストを羅列した filename.list のようなファイルを用意して
    @filename.list
    のように指定してもいい。
- imarith (filename1) (+-*/) (filename2) (output)
(filename1) と (filename2) の加減乗除 (+-*/) を行う。
- imexam [filename]
(filename) で指定したファイルの性質をインタラクティブに調べる。ds9 上にカーソルを持って行き、そこでキーボードを操作することによって実現する。たとえば、
  - a カーソル位置にあるピークの統計を調べる。FWHM、フラックス、バックグラウンドレベルなど。
  - e カーソル位置のコントアを描く。
  - z カーソル位置周辺のピクセルの値の一覧を出す。
  - m カーソル位置周辺のピクセルの統計値を計算する
  - q 終了

  等

- IRAF で測光
apphot パッケージを使う。まずは IRAF を立ち上げて

```
noao
digi
apphot
```

とする。そのあと各種パラメータの設定

- epar phot
を実行
- photpar にカーソルを移動して、:e と打つ
  - apertur を適当な値 (アパーチャー直径が FWHM の倍以上になるように)
  - zmag は求めておいた値 (あるいはそのままでもい)
  - 終わったら、:q と打つ
- fitskyp にカーソルを移動して、:e と打つ
  - annulus を適当な値 (直径が FWHM の 3 倍以上になるように)
  - 終わったら、:q と打つ

- datapar にカーソルを移動して、:e と打つ
  - itime に露出時間の値 (秒) を入れる
  - 終わったら、:q と打つ
- 終わったら、:q と打つ

次に、測光を行う。

phot（画像ファイル名）

で ds9 に画像が表示されるので、測光したい星にカーソルを持っていてスペースキーを押す。
複数の星を測定することも可能。
終わったら q を 2 回打つ。
結果は (画像ファイル名).mag.[1-9] というファイルに保存される。中身は以下のようになっている。（#で始まる行はコメント行）

```
test.fits            2082.000  2449.000  2     nullfile              0      \
   2081.515   2448.083   -0.485  -0.917  0.575   0.777          0    NoError   \
   0.001193246    0.01377304     0.0068786      911    29       0    NoError   \
   1.             INDEF          INDEF                 INDEF                  \
   3.00     4.924091      28.69609   4.88985       23.277 0.491 0    NoError
```

このなかで、最後の行が重要で

- (アパーチャー半径)
- (アパーチャー内のカウント [スカイ込み])
- (アパーチャー面積)
- (スカイを引いたアパーチャー内のフラックスカウント)
- (等級に変換した値 [上の zmag を用いている])
- (等級誤差)

となっている。

5.5 解析の実際

西浦版『可視光域データ・リダクション法』を参照。
実際に行うことは、

- データのバイアス引き、ダーク引き、フラットフィールドを行う
- imexam で星からのフラックス量を算出。
- それを電子数に換算して n_i を算出
- それと計算から出る s_i を比較してシステム効率と conversion factor を出す

という流れになる。

6　実際に測定すること

6.1　システム効率

- 適当な明るさの星を適当な積分時間で撮影し、検出されたカウントを測定する。
- その星から望遠鏡に入射した光子の数を計算する
- 上記２つを比較してシステム効率を出す。

6.2　等級原点

等級原点 $Z_{\rm mag}$ は、あるカメラシステムで 1 秒間の露出で 1 カウントの信号を生じさせる天体の等級のこと。たとえば、8 等級の星を 5 秒露出したときに 100 カウントの信号が検出されたとすると、このカメラシステムでの等級原点は

$$\begin{aligned} Z_{\rm mag} &= 8 + 2.5 * \log \frac{100}{5} \\ &\sim 11.3 \end{aligned}$$

となる。これを、上記のデータより算出する。

6.3　空の背景光

星の写っていない領域に入射している光の量を [photons/s/arcsec^2]、および [mag/arcsec^2] で算出する

6.4　限界等級

- 適当に星を選び、適当な積分時間で露出を行い、その等級と等級誤差を実測する。
- その等級をカタログ値と比較する。
- 等級誤差を、予想される値と比較する