

小望遠鏡による基礎観測実習 (2006 年度)

平成 18 年 11 月 24 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します。

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いたいとおもいます。

2004 年度ではフィルターが夏場の高温多湿で曇ってしまい、それをそのまま使って実習を行った。2005 年度の観測実習では新たなフィルターを用いて、

観測から望遠鏡/カメラシステムのシステム効率を出す さらに、このシステム効率がわかれば画像データ 1 ピットが何等級に相当するか (conversion factor) も分かりますので、それも同時に計算してもらいます。

限界等級を算出する これは、先ほどの conversion factor から空の明るさを測定したうえで計算することになります。

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

をやりましたが、データ取得時に不備があったため信用できる値がでていません。今年はこれを作り直します。

なお、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いてください。

2 限界等級

2.1 システム効率とは

単純に言ってしまうと、

『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))

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

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))

$$n_i = \frac{N f_{conv}}{t} \quad (2)$$

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

$$\eta = \frac{n_i}{s_i} \quad (3)$$

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

$$\begin{aligned} S/N &= \frac{n_i t}{N_{\text{noise}}} \\ &= \frac{\eta s_i t}{N_{\text{noise}}} \end{aligned} \quad (4)$$

のように書ける。ここで N_{noise} はノイズ成分、 t は積分時間。ノイズ成分は入射光子のポア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、

$$N_{\text{noise}} = \sqrt{n_i t + n_{\text{sky}} t + n_{\text{dark}} t + N_{\text{read}}^2} \quad (5)$$

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

複数素子の場合に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よい。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_{noise} は

$$N_{\text{noise}} = \sqrt{n_i t + m n_{\text{sky}} t + m n_{\text{dark}} t + m N_{\text{read}}^2} \quad (6)$$

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。1
- 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 に転送してあげることになる。Windows98 マシンの beetle 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 1 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 “.FITS” 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 とかかっている (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 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 からは vw として見えているので (図 1)、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 tct2006@vw:/home/ccdimages/test.FTS test.fits
```

のようにすればいいわけである。

4 観測

4.1 観測する星の選出

http://mthamilton.ucolick.org/techdocs/standards/standards_intro.html から選びましょう。

等級は場合によっては “Allen’s Astrophysical Quantities” を参照する必要があるかもしれません。

4.2 今夜の星空

xplns というソフトがあります。golf で起動して使いましょう。

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でしょう。

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1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/ioa10 から golf にアクセスして、コピー
ただし、ioa09/ioa10 からは、golf は vw として見える。
- ioa09/ioa10 上で解析を行う

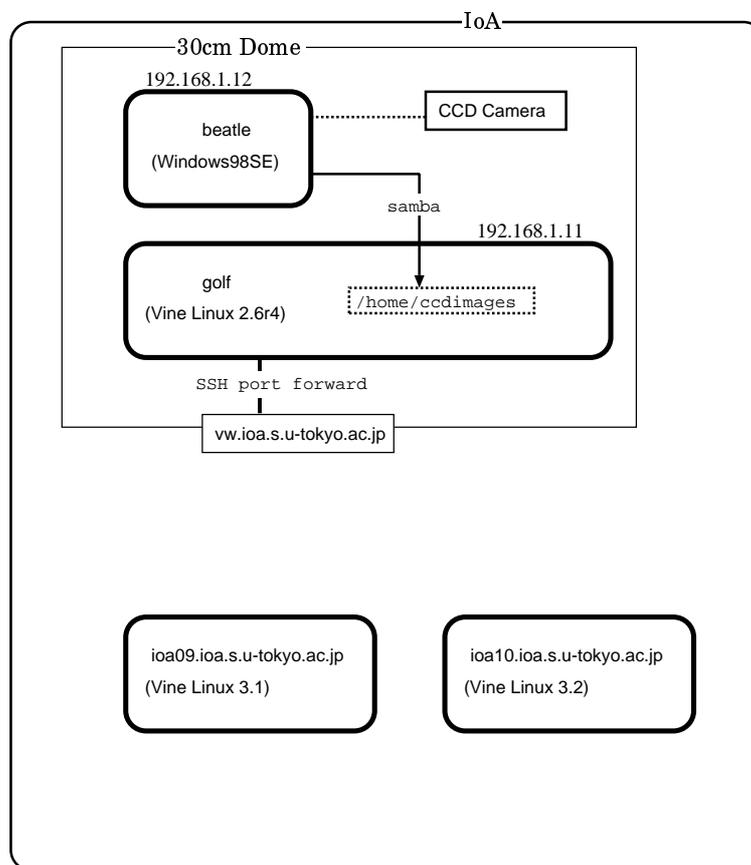


図 1: 計算機群の構成

5.2 アカウントとパスワード

golf/ioa09/ioa10 のアカウント、パスワードは :

% tct2006 / ***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 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 .

5.3.2 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。

5.3.3 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。

5.3.4 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。

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。

5.3.5 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
- ds9
を実行して画像ビューワーを立ち上げる
- 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ルリングが 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。

- imexam (filename)
画像の統計量を調べる
- 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

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。