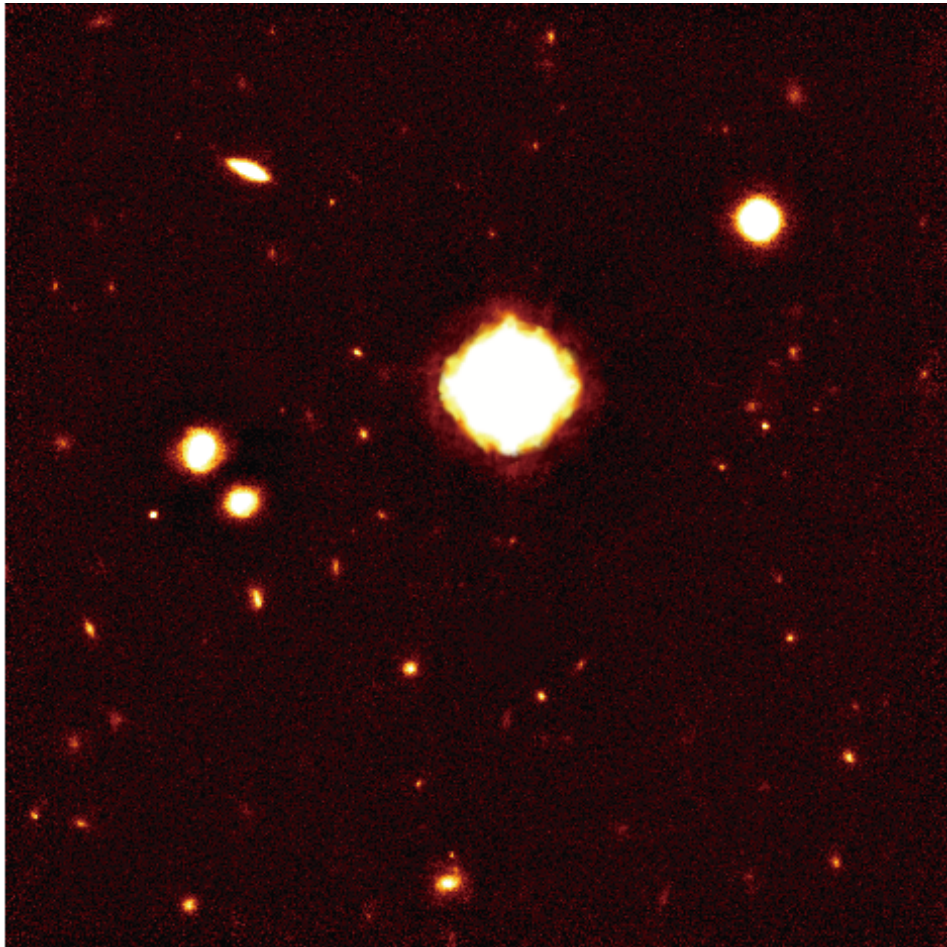


# すばる IRCS 撮像データ解析マニュアル (ver.2.0)



2008年7月3日  
美濃和 陽典 (国立天文台ハワイ観測所)

# 目次

1	はじめに	3
2	近赤外線撮像について	4
2.1	特徴	4
2.2	近赤外線撮像でのノイズの種類と S/N	6
3	IRCS 撮像について	6
3.1	データ	6
3.2	撮像観測の流れ	7
3.3	撮像データ解析の流れ	9
4	IRCS 撮像データ解析	9
4.1	サンプルデータ	9
4.2	解析環境の準備	10
4.3	データの確認	11
4.4	ダークフレームの作成	15
4.5	キャリフラットフレームの作成	15
4.6	バッドピクセルマスクの作成	16
4.7	感度むらの補正	19
4.8	スカイバックグラウンドの引き算と重ね合わせ	21
4.9	標準星の解析	28
5	限界等級	29
5.1	ゼロ等級の見積もり	30
5.2	ノイズの見積もり	33
5.3	限界等級の見積もり	34
6	最後に	38

# 1 はじめに

本マニュアルは、すばる望遠鏡の近赤外撮像分光装置 IRCS で取得された撮像データの解析方法をまとめたものです。IRCS を用いた近赤外線での撮像データの解析は、基本的には可視光の CCD での撮像とほとんど同じです。しかし、近赤外線での大気や望遠鏡からの背景光が、可視光のものとは比べて大きい、検出器が異なるという点で違いがあります。このマニュアルでは、始めに、これらの違いによる近赤外線撮像観測の特徴について説明します。

次に、IRCS の撮像データの IRAF を使った解析方法を説明していきます。IRAF の使い方等の基本的な部分は、このマニュアルでは省略します。天文情報処理研究会発行の IRAF クックブック等を参照してください。解析は、IRCS 撮像データを解析するための IRAF スクリプト (以降、`ircs_imgred` と呼ぶ) を使って行います。このマニュアルは、2006 年度のすばる解析講習のために作ったものを改訂したものです。`ircs_imgred` の各スクリプトの中身は元の解析講習のマニュアルにもう少し詳しく書かれています。

分かりにくい点や間違っている点のご指摘、その他、要望、コメント等がありましたら下記までお寄せください。

2008 年 7 月 3 日

美濃和 陽典

Subaru telescope,

650 North A'ohoku Place, Hilo, HI 96720, USA

電話: +1-808-934-5905, Fax: +1-808-934-5099

E-mail: minoways at subaru.naoj.org

## 2 近赤外線撮像について

### 2.1 特徴

波長 1 ミクロン以上の赤外線では、ほぼ全ての波長で地上からの観測が可能な可視光とは異なり、地球大気の吸収により地上に到達しない波長域があるため、観測できるのは透過率の高い波長帯（大気の窓）に限られます（図）。近赤外線観測で用いられるフィルターはこの大気の窓に合わせて作られています。吸収の深さは、大気中の水蒸気量に比例しているため、高地になるほど大気吸収の影響は少なくなります。

赤外線では、可視光に比べて地球大気や望遠鏡からの背景放射が大きくなります。そのため、可視光に比べて観測が難しく、背景光から来るポアソンノイズにより、感度も悪くなります。図では、可視光から近赤外線までの背景放射の強さを表しています。波長 2.3 ミクロンより短波長では、OH 夜光が主な背景光の要因になっています。波長 2.3 ミクロンより長波長では、地球大気と望遠鏡からの放射が支配的になり、背景光が急激に増加します。

赤外線の背景光は、観測高度や大気の状態により、その明るさや視野内での明るさの傾斜パターンが徐々に変化していきます。そのため、背景光を取り除くには、観測した画像そのものを使う必要があります。しかし、観測した画像には天体も写っているため、視野を少しずつずらして同じ場所に天体が写らないようにして何点か観測し、それらの画像をメジアンで重ね合わせて背景光だけの画像を作ります。この様に、視野を少しずつずらす事をディザリングと言います（図）。ディザリングは、背景光を取り除くのに必要なだけでなく、検出器上のバッドピクセルを避けるためにも必要です。

近赤外線の検出には、InSb(インジウムアンチモン) や、HgCdTe(水銀カドミウムテルル) でできたアレイ検出器が用いられます。可視光で使われている CCD は波長 1 ミクロン以上に感度が無いため、赤外線観測には使えません。赤外線の検出器の特徴としては、CCD の様に電荷転送をせず、各ピクセルの電圧を独立に読み出す、DRO(Direct Read Out) と呼ばれる方式をとっているため、可視光撮像で明るい星を観測した際に見られるようなブルーミングを起こしにくい点があります。また、積分中にピクセルの情報を壊すこと無く複数回読み出す非破壊読み出し (Non-destructive readout;NDR) ができるという点も特徴的です（図）。非破壊読み出しを行うことで、読み出しノイズを削減することができます (NDR の回数が  $n$  回であったとするとノイズは  $1/\sqrt{n}$  となります)。

通常の赤外線の撮像観測では、背景光が大きいため、検出器を飽和させないために、1 回の読み出しで可能な積分時間が短くなってしまいます。特に、熱放射が強い L (3.5 ミクロン)、M (4.8 ミクロン) バンドでは、積分可能な時間が 1 秒以下になる場合もあります。そのため、積分時間を稼ぐために「COADD」という方法が用いられます。これは、短い積分を複数回行い、メモリー上で足し合わせ、1 フレームとして読み出すというものです。この方法は、L,M バンドの観測以外でも、明るい天体を飽和させない様に短い積分を重ねる場合に使うことができます。

赤外線検出器により集められた電荷は、読み出し回路で増幅され、デジタル量に変換 (A/D 変換) されます。解析するデータの信号は、デジタルに変換されたもので、ADU という単位で表されます。



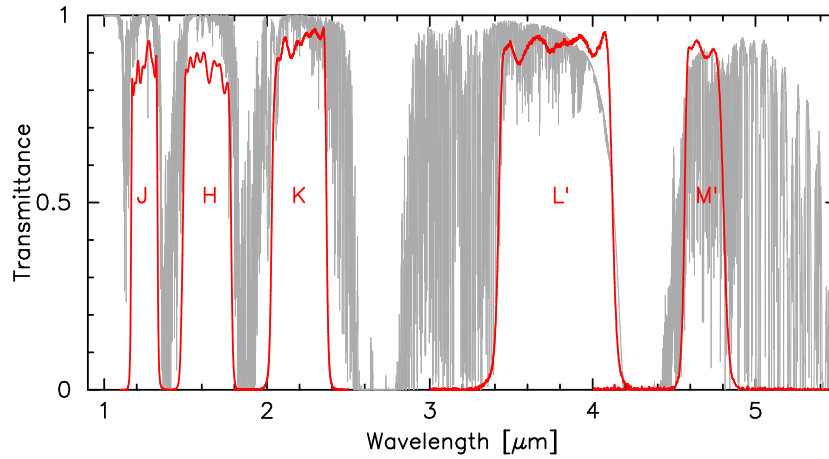


図1 マウナケア山頂での大気の透過率（灰色線）と、近赤外線波長域のブロードバンドフィルターの透過率（赤線）。

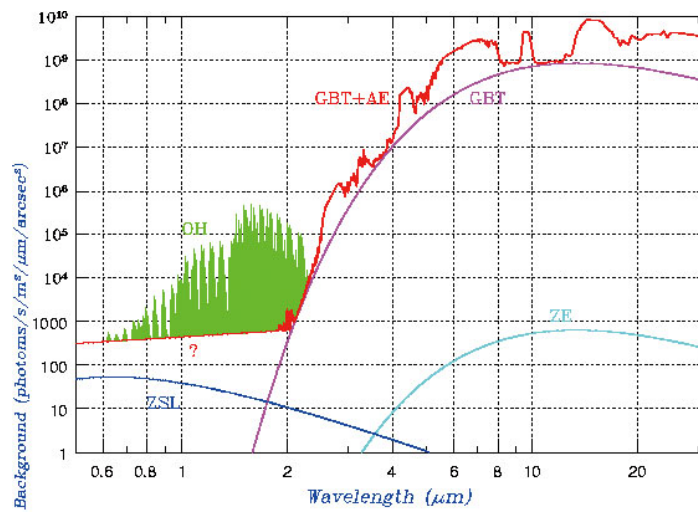


図2 可視光から近赤外線までの背景放射。波長が長いほど背景光が強い。OH: OH夜光、AE: 大気の熱放射(  $\sim 273\text{K}$  )、GBT: 望遠鏡の熱放射(  $273\text{K}$  )、ZSL: 黄道面付近のダストによる太陽光の散乱、ZE: 黄道面付近のダストからの熱放射( 画像は京都大学 岩室史英氏の観測天文学の講義ノートより引用、URL: <http://www.kusastro.kyoto-u.ac.jp/iwamuro/LECTURE/OBS/> )。

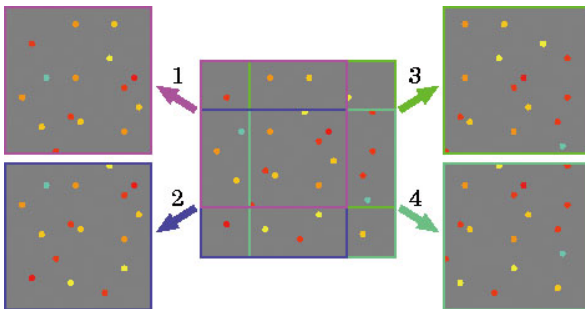


図3 ディザリングの概念図。ディザリングにより取得された異なる場所での画像を、天体の位置を合わせて重ね合わせることで、S/Nの良い画像が得られます。また、位置を合わせずにメジアンで重ね合わせることで、天体の寄与を消した背景光だけの画像を作ることができます( 画像は京都大学 岩室史英氏の観測天文学の講義ノートより引用 )。

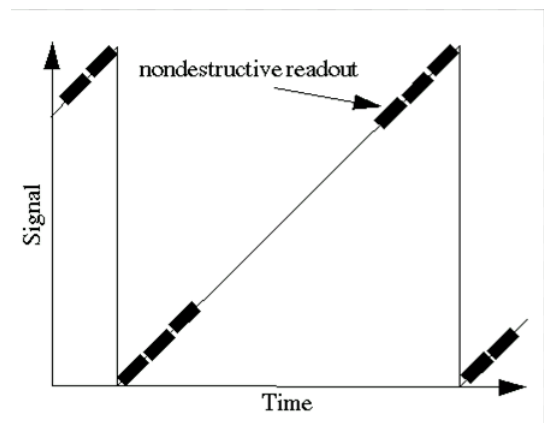


図4 非破壊読み出しの概念図。NDRの数が  $n$  回の際には、積分開始時と積分終了時に  $n$  回ずつ読み出しています。

## 2.2 近赤外線撮像でのノイズの種類と S/N

天体観測におけるノイズは、通常以下のものが要因となります。

- 空 + 望遠鏡の背景放射 : SKY [counts pixel<sup>-1</sup> sec<sup>-1</sup>]
- ダークカレント : DARK [counts pixel<sup>-1</sup> sec<sup>-1</sup>]
- 読み出しノイズ : R [electron pixel<sup>-1</sup>/read]
- 天体からのフラックス : OBJ [counts/sec]

半径  $r$  [pixel] に広がるフラックス  $OBJ$  [counts/sec] の天体を積分時間  $t$  [sec] で観測した場合の S/N は、以下ようになります。

$$S/N = \frac{OBJ \times t \times g}{\sqrt{(OBJ + (SKY + DARK) \times \pi r^2) \times g \times t + \pi r^2 R^2}}$$

ここで、 $g$  [electrons/ADU] は、1ADU が電子何個に相当するかを表すコンバージョンファクターです。

近赤外線撮像観測の場合、通常は背景放射がノイズの主で、上式の分母の  $\sqrt{\quad}$  内の  $SKY$  以外の項は無視できます。その際、S/N は積分時間の  $1/2$  乗に比例します。このような場合を、バックグラウンドリミットと言います。ただし、J バンドなどの短い波長で、短時間の積分を行った場合、背景放射のノイズよりも、読み出しノイズが勝る場合があります。この場合は、S/N は積分時間に比例します。

## 3 IRCS 撮像について

### 3.1 データ

IRCS (Infrared Camera and Spectrograph) は近赤外線波長域において、撮像、分光観測を行う装置です。大気揺らぎによる星像の乱れを除去する補償光学システムとともに設置されており、0.1 秒角を切る高空間分解能での観測にも対応しています。IRCS の撮像モードは、0.9-5 $\mu$ m に感度がある 1024 $\times$ 1024 の Aladdin III array という InSb 検出器を使用しています。ピクセルスケールは、0.023 秒角/ピクセルの高解像モードと、0.058 秒角/ピクセルの低解像モードがあり、それぞれ 23 $\times$ 23 平方秒角、1 $\times$ 1 平方分角の視野の撮像ができます。使用できるフィルターは、z,J,H,K',K,L',M' のブロードバンドフィルターと、いくつかの狭帯域フィルターがあります。

IRCS は、2005 年 7 月まですばる望遠鏡のカセグレン焦点に取りつけられていましたが、それ以降は補償光学システムのバージョンアップに伴い、ナスミス焦点へ移りました。この移動により、ピクセルスケールが若干変わっており、0.020 秒角/ピクセル、0.052 秒角/ピクセルとなっているため、新しいデータを解析するには注意が必要です。詳細は、すばる望遠鏡の IRCS のウェブページ <http://www.naoj.org/Observing/Instruments/IRCS/parameters.html> で確認してください。

IRCS の低解像度モードのデータでは、明るい星の北側と西側 (PA=0° の場合) にゴーストが出ます (図)。これは、光を補償光学に送る可視光と、IRCS に送る赤外光に分けるためのビームスプリッタとなる平行平面基板によるゴーストと、集光光線が平行平面板を通過することで起きる非点収差を打ち消すためのコンペンサータとなる平行平面板によるゴーストです。以下で解析するサンプルデータにもこのゴーストが出てきますので、注意してください。高解像度モードでは、視野が狭

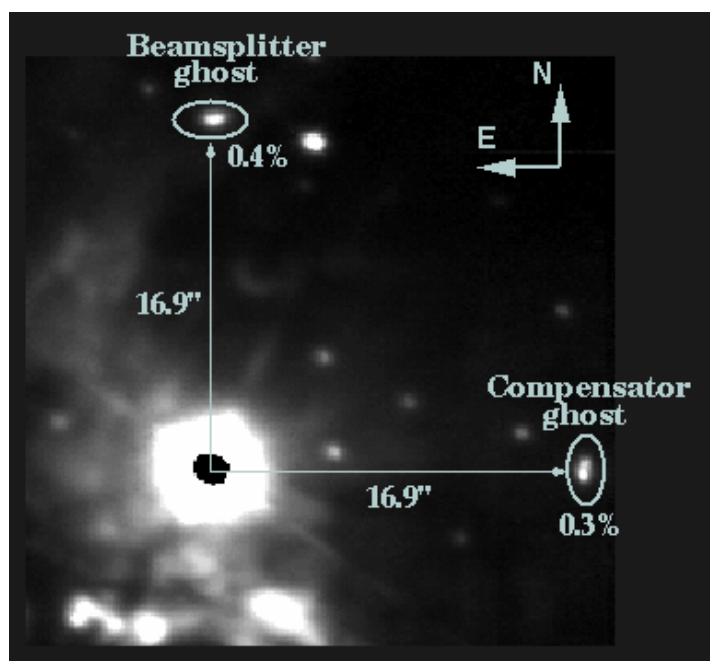


図5 IRCSの低解像度モードで主に見られる、明るい星の周りのゴースト。

くなるので、明るい星を視野中心に置いている場合には、このようなゴーストは見られません。

### 3.2 撮像観測の流れ

IRCSの撮像観測は、一般的に以下のような手順で行われます。

1. 望遠鏡を天体に向け、追尾状態にする。
2. 観測装置のフィルター、ピクセルスケールを変える。
3. 積分時間を決める。

近赤外線の撮像観測の場合、通常はバックグラウンドリミット（読み出しノイズより背景光のポアソンノイズが卓越する）になる様に、検出器の線形性が保たれる範囲でできるだけ長くする。積分時間を長くしすぎると、入射した光子の数に対する、検出器のデジタル信号の増加が非線型になってしまう。IRCSの場合には、背景光が4000ADU前後になるようにしていれば線形性は保たれています（図）。

4. 視野をすこしずつ動かしながら積分を繰り返す（ディザリング）。

ディザリングの1セットは、背景光の時間変動の影響が大きくなりえない程度の時間間隔に収まるようにします。この間隔は、観測波長が長くなるほど短くなります。IRCSの場合、1画像につき数秒～数十秒の積分を5点または9点撮って1セットとしています（図）。

5. 観測した視野の近くの測光標準星<sup>\*1</sup>撮像をする。

標準星のairmass（図）はできるだけ観測した天体と同じになるように選びます。ただし、マウナケア山頂では、近赤外線でのairmassの違いによる減光は0.1等/airmass以下とい

\*1 測光標準星とは明るさが精度良く決められている星です。近赤外線では、UKIRTのFanit standard ([http://www.jach.hawaii.edu/UKIRT/astrometry/calib/phot\\_cal/faint\\_stds.html](http://www.jach.hawaii.edu/UKIRT/astrometry/calib/phot_cal/faint_stds.html)) や、Persson et al. (1998, AJ, 116, 2475)の標準星カタログが有名です。

う観測結果があり\*2、誤差が0.1等級以下になるような高精度の測光をを求めるのでなければ、0.2-0.3程度の airmass の差は問題ないと筆者は考えています。

6. 観測モードを変える時は2に、天体を変える時は1に戻る。
7. 観測終了後の明け方にその日に観測したデータ用のダークフレーム、フラットフレームを取得する。

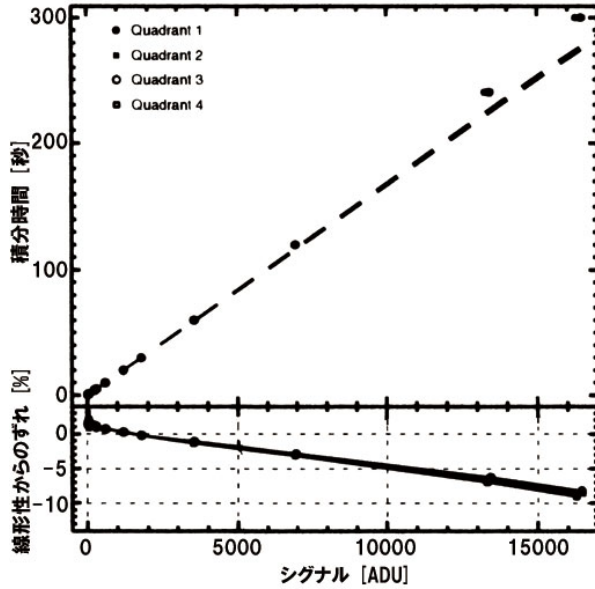


図6 IRCS のカメラ側検出器の線形性 (Terada et al. 2004, Proc. SPIE, 5942, 1542)。4000ADU 前後であれば、線形性からのずれは3%以下であり、積分時間に対しほぼ線形にシグナルが増えていきます。それ以上のシグナルでは、線形性からのずれが顕著になるため、1フレームでの背景光のシグナルが4000ADU前後になるように積分時間を調整して観測をする必要があります。明るい星を観測する場合には、背景光よりも天体からのシグナルが強くなるため、天体のピークが4000ADU前後になるように積分時間を調整します。

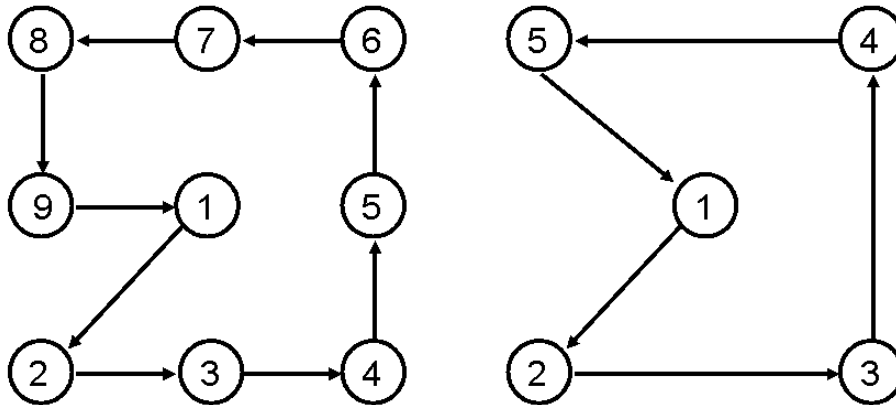


図7 IRCS の撮像観測で使われるディザリングパターン。IRCS では通常1セット5点が9点のディザリングを行い観測しています。

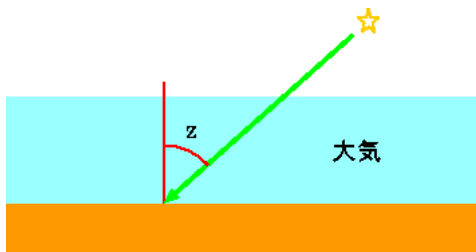


図8 Airmass の概念図。Airmass は  $\sec(z)=1/\cos(z)$  として定義されており、天頂で  $\text{airmass}=1$  となります。airmass が大きいと、地球大気を通過する距離が長くなるので、星からの光は減光されます。観測は airmass が1付近で行うのが望ましいが、近赤外線の観測では1.5を超えても観測している場合もある。Airmass が2を超えるような観測はほとんどありません(画像は京都大学 岩室史英氏の観測天文学の講義ノートより引用)。

\*2 <http://www.jach.hawaii.edu/UKIRT/astrometry/utls/exts.html>

### 3.3 撮像データ解析の流れ

一般的に撮像観測で取得した生のデータには、天体からのシグナル以外にも様々なノイズが乗っています。例えば、ある時間  $t$  に撮られた生データのある座標  $(x,y)$  にあるピクセルのカウント値  $RAW(x,y,t)$  は、原理的には以下のように構成されています。

$$RAW(x,y,t) = f(x,y) \{OBJ(x,y,t) + SKY(x,y,t)\} + DARK(x,y)$$

ここで、 $OBJ(x,y)$  は天体からのシグナル、 $DARK(x,y)$  は検出器の暗電流 (ダーク)、 $f(x,y,t)$  は検出器の感度むら、 $SKY(x,y,t)$  は空からの放射、及び望遠鏡、観測装置の熱放射を表しています。 $OBJ$  と  $SKY$  は時間の関数になっていますが、これは時間経過に伴うシーイング、大気の透過率、背景光の量の変化を表しています。近赤外線の観測の場合、特に  $SKY$  の時間変動が可視光に比べて大きいことに注意する必要があります。IRCS 撮像データの場合、ダーク、フラットについては一晩の間に大きな変化はありません。

撮像データの解析でまず始めにやることは、(1). ダークの引き算、(2). 感度むらの補正、(3). スカイバックグラウンドの引き算を行い、天体からのシグナルだけの画像を作ることです。この後、全てのフレームの位置合せをし、重ね合わせて最終的な画像を作成します。

次節では、IRCS 撮像データの解析手順を、以下のような流れで解説していきます。

- 1.
2. データの確認
3. ダークフレームの作成
4. キャルフラットフレームの作成
5. バッドピクセルマスクの作成
6. スカイフラットフレームの作成
7. 感度むらの補正
8. スカイフレームの作成、差引
9. バッドピクセルマスクの補間
10. オフセットの測定
11. 重ね合わせ
12. 測光標準星の解析 (1~9 の手順と同じ)
13. 限界等級の見積もり

## 4 IRCS 撮像データ解析

### 4.1 サンプルデータ

IRCS 撮像データ解析の練習として、遠方銀河の撮像データを解析してみましょう。このデータは、すばるディープフィールドと呼ばれる北銀極方向のプランクフィールドを、IRCS と補償光学 (AO36) を使って  $K'$  バンドで観測したものです (Minowa et al. 2005, ApJ, 629, 29)。この観測では、約 27 時間の長時間撮像をしていますが、今回はその一部の約 3 時間分のデータを解析します。

解析するサンプルデータは、sample1.tar.gz にあります。まず始めに、sample1.tar.gz を以下のように解析を行うディレクトリに展開してください。sample1/ ディレクトリの下に IRCA\*.fits というファイルができます。

```
$ cd [解析を行うディレクトリ]
$ tar xvzf sample1.tar.gz
$ ls sample1
```

## 4.2 解析環境の準備

本マニュアルでは、主に IRAF を使ったデータ解析方法を解説します。解析の作業を始めるにあたり、IRAF を使えるように準備をする必要があります。ここでは、IRAF ver. 2.14 を使うことを前提に説明していきます\*<sup>3</sup>。

以下では、IRCS 撮像解析用の IRAF スクリプト (ircs\_imgred) を使って解析する方法を説明します。IRAF スクリプトのインストールは”ircs\_imgred[mmddy].tar.gz”を展開したディレクトリにある”setup\_ircs.pl”という perl のスクリプトを使って行います。これによりスクリプト (ircs\_imgred/clscript) が使えるようになり、IRAF の設定ファイルである login.cl を IRCS 撮像データ用に書き換えます。

```
$ cd [解析を行うディレクトリ]
$ tar xvzf ircs_imgred[mmddy].tar.gz
$ ln -s ircs_imgred[mmddy] ircs_imgred
$ cd ircs_imgred
$ perl setup_ircs.pl
-- creating a new uparm directory
Terminal types: xgterm,xterm,gterm,vt640,vt100,etc.
Enter terminal type: 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.
Done
```

以降、IRAF を起動する際は login.cl ファイルがある ircs\_imgred ディレクトリで行ってください。

```
$ cd ircs_imgred --- login.cl のあるディレクトリへ移動
$ cl
c1> cd [解析ディレクトリへ移動]
```

次に、fits 画像を表示させるソフトを起動します。ここでは、ds9 というソフトを使うことにします。IRAF が起動しているターミナルとは別のターミナルで ds9 & と打てばソフトが立ち上がります。IRAF が起動している xgterm 上でも!ds9 & と打てば同じようにソフトが立ち上がり\*<sup>4</sup>以下の作業では、必ず ds9 を立ち上げて行ってください。

---

\*<sup>3</sup> IRAF のバージョンが古い場合、本マニュアルで使用するスクリプトが使えない可能性があります。

\*<sup>4</sup> IRAF が起動しているターミナルで UNIX のコマンドを使いたいときにはコマンド名の前に! を付けると実行でき

以下の説明では、IRAF を起動した xgterm 上での作業を、

```
cl>
```

として表し、IRAF ではない通常のターミナル上での作業を、

```
$
```

として表します。

その他、以下の解析手順では perl のスクリプトを使用します。このスクリプトが使えるように、以下の例を参考に perl スクリプトが置いてあるディレクトリにパスを通しておいてください。

```
# bash の場合 --- ~/.bashrc か ~/.bash_profile に以下の行を追加
export PATH=$PATH:[ircs_imgred を展開したディレクトリの full path]/plscript
# tcsh の場合 --- ~/.cshrc に以下の行を追加
set path ([ircs_imgred を展開したディレクトリの full path]/plscript $path)
```

この perl スクリプトでは、SExtractor という天体検出ソフトウェア (Bertin & Arnouts 1996, A&AS, 117, 393) を使っています。解析を始める前に、このソフトもインストールしておいてください。このソフトウェアは、[http://terapix.iap.fr/rubrique.php?id\\_rubrique=91/](http://terapix.iap.fr/rubrique.php?id_rubrique=91/) からダウンロードできます。

### 4.3 データの確認

通常、どのファイルが何のデータに対応するかは、観測時に電子ログや手書きのログを取って記録しておきますが、アーカイブデータ等を使う場合には、この記録が不十分な場合があります。このような場合には、ファイルのヘッダを見れば、何のデータをどのようなモードで取ったのかをある程度確認できます。fits ファイルのヘッダは以下のようになっています。これは imhead というコマンドで見ることができます。

```
cl> imhead raw/IRCA00091688.fits l+ | page
raw/IRCA00091688.fits[1024,1024] [int]: SDF
No bad pixels, min=0., max=0. (old)
Line storage mode, physdim [1024,1024], length of user area 7412 s.u.
Created Tue 00:00:00 01-Jan-1980, Last modified Tue 00:00:00 01-Jan-1980
Pixel file "raw/IRCA00091688.fits" [ok]
FRAME-ID=                / Frame Id
I_ARCH =                  0 / ARCHIVED? 0:No 1:Yes
I_HDRVER=                 1.22 / IRCS HEADER VERSION
...
```

ヘッダには、1 行辺りにキーワード、値、その説明が書いてあります。全てのファイルから、自分の知りたいヘッダ情報だけを抽出するには、hselect という IRAF コマンドが便利です。

---

ます。ただし、cd や ls などの基本的なコマンドは!無しでも動きます。

```
cl> hselect sample1/IRCA*.fits > file_header.lst
fields to be extracted: $I,title,EXP1TIME,COADDS,NDR,FILTER02,I_MDFMST,I_DTHPOS
boolean expression governing selection: yes
```

この例では、\*はワイルドカードですので、IRCA??????.fits という名前の全ての fits ファイルのヘッダ情報を読んで「file\_header.lst」に書き出しています。抜き出したヘッダ情報は、ファイル名 (\$I)、オブジェクト名 (title)、積分時間 (EXP1TIME)<sup>\*5</sup>、COADD の回数 (COADDS)、非破壊読出しの回数 (NDR)、フィルター (FILTER02)<sup>\*6</sup>、ピクセルスケール (LMDFMST)、ディザリング位置 (LDTHPOS) です。ただし、ここであげたヘッダキーワードは IRCS 独特のものもあり、他の観測装置では必ずしも同じでは無いことに注意してください。出力結果の 'file\_header.lst' には、1 行毎にそれぞれの fits データの情報が、hselect コマンドの 'fields to be extracted' で指定した順に表示されています。

これらの情報を見て、自分の解析したいオブジェクト、標準星、それに対応するダーク、キャリブフラットのフレームはどれかを確認します。この時、観測ログ等で確認して、フィールド確認のために取られたフレームや、天体導入に失敗したフレームなどの余分なデータの行を消しておくことで便利です。

今回の場合は、オブジェクトが 'SDF'、標準星が 'P330-E' という名前のものを解析します。これらのオブジェクト名に該当するヘッダ情報を見ると、例えば以下のようになっています。

```
raw/IRCA00091688.fits  SDF      120.0000    1   8   K   58MAS    1
raw/IRCA00091860.fits  P330-E    3.0000     1  11   K   58MAS    1
```

これは、オブジェクトの 1 フレームが、120 秒積分で COADD=1, NDR=8、標準星が、3 秒積分で COADD=1, NDR=11 で撮られている事を意味します。最後の 3 列の情報はどちらも同じで、K バンド、58mas モードで撮ったディザリングの 1 番目のデータであるという事を意味しています。

まず始めに、解析したいオブジェクトの生データのリストを作成しておきます。このリストは、例えばヘッダ情報のファイルから grep コマンドを使って以下のようにして作成できます。

```
$ grep SDF file_header.lst | awk '{print $1}' > object_raw.lst
```

この段階で、リストの中のいくつかの画像を見てみて、背景光のカウント値、または目標とする天体のピークカウント値が、カウント値の上昇が非線形になる領域に達していないかどうかを確認してください (図参照)。もしカウント値が大きすぎるフレームがある場合は、正しい測光ができなくなるため、そのフレームは捨てなければなりません。サンプルデータの場合、観測対象は遠方銀河で非常に暗いため、背景光のカウント値が十分小さい (4000ADU 前後) であることを確認してください。具体的には、imexam コマンドで画像を表示させ、ds9 のウィンドウ上の適当な場所で 'z' と打つと、その周りのカウント値が xgterm 上に表示されます。また、'm' と打つと、その周りのピクセルの平均値、メジアン値等が表示されます。以下は、raw/IRCA00091689.fits を imexam コマンドで ds9 上に表示させ、座標 (500,280) 付近で 'm' と打った出力結果です。

<sup>\*5</sup> 積分時間のキーワードは EXP1TIME と EXPTIME がありますが、EXPTIME=EXP1TIME×COADDS となっています。

<sup>\*6</sup> IRCS にはフィルターホイールが 3 個ありますが、JHKLM のブロードバンドフィルターは 2 番目のフィルターホイール (FILTER02) に入っています。ナローバンドフィルターや、z バンドフィルターは 1、3 番目のフィルターホイールに入っているため、場合によっては FILTER01, FILTER03 の情報も確認してください。



#	SECTION	NPIX	MEAN	MEDIAN	STDDEV	MIN	MAX
	[503:507,278:282]	25	35261.	35308.	576.	34176.	36276.

メジアン値が 35308 となっていますが、IRCS は非破壊読み出し (NDR) を複数回行っているため、その回数分だけ割らないといけません。また、1 フレームの積分時間が非常に短い時には COADD を行っている場合があるので、その回数でも割らなければいけません。このフレームの COADD 数は 1、NDR 数は 8 なので、平均値を 8 で割ると、背景光のシグナルは 4400ADU 程度であることが分かります。これぐらいのカウント値であれば、シグナルの線形性が保たれていると判断できます (図参照)。全てのフレームについて、ラフに背景光のメジアン値を知りたい場合には `imstat` コマンドを使ってできます。

```
cl> imstat @object_raw.lst fields=image,midpt nclip=50 usigma=3 lsigma=3
```

次に、すべての画像を目で見ておかしい画像が無いかを確認します。おかしい画像の一例としては、画像上のダークのパターンが明らかに他のものと異っているものがあります (図参照)。この現象は、IRCS では観測開始直後や積分を長時間とめた後のセットで起りやすく、ディザリングの 1 番目のフレームや、まれに 2 番目以降のフレームで現れることもあります。その他、観測中のトラブルで、望遠鏡の追尾がうまくできていなく、星が動いてしまった様なフレームや、極端にシーイングが悪かったり、極端に透過率が悪かったりするフレームがある場合があります。これらの悪いフレームは、S/N を落とす原因になるので、この段階でリストから除いておきます。

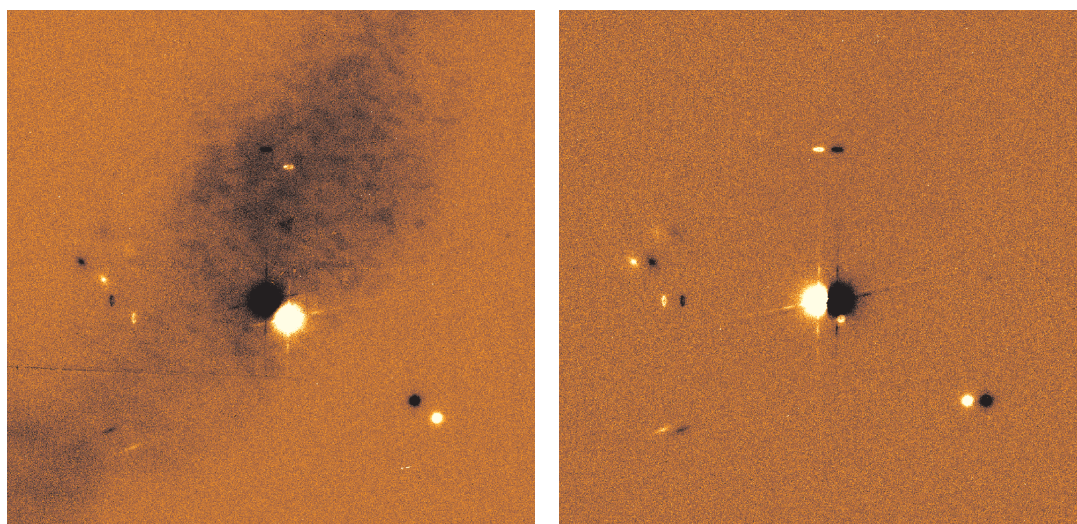
このような悪い画像があるかどうかは、ディザリングにより位置を少しずらした 2 枚の画像を引き算、割り算することで、確認できます。例えば、1 つのディザリングのセットの中に、1~9 までの画像があった場合、1-2, 2-3, ..., 8-9, 9-1, 1/2, 2/3, ..., 8/9, 9/1 という画像の割算、引き算を行った結果を表示し、悪いフレームが無いかを確認した上であればリストから除きます。このような作業は、IRCS 解析スクリプトの `frcheck` というコマンドを使ってできます。コマンドを実行する前に必ず `ds9` を立ち上げておいてください。具体的な使い方は以下の通りです。

```
cl> frcheck object_raw.lst mklist=yes calc=subtract
```

`ds9` 上に引き算 (`calc=divide` とした場合は割り算) をした結果が表示されます。ファイル名は、“IRCA00091688-IRCA00091689.fits” という様な名前になっていますが、これは IRCA00091688 と IRCA00091689 を引き算 (または割り算) した結果という意味になります。この段階では IRAF の `imexem` コマンドが実行されているので、`ds9` 上で 'n' と打ち、画像を進めて結果を確認してください。すべて確認し終わったら 'q' と打ち、`ds9` から IRAF のターミナルに戻ります。その際に、以下のように表示されるので、悪いフレームがある場合は該当する行に移動し、'dd' と打ってその行を消してください (ここでは、vi エディタが立ち上がっています)。

```
sample1/IRCA00091688.fits
sample1/IRCA00091689.fits
sample1/IRCA00091690.fits
sample1/IRCA00091691.fits
sample1/IRCA00091692.fits
...
```

その後、':wq' と打ち編集を終了してください。このコマンドにより、悪いフレームを取り除いたリストが、もとのリストの名前の後ろに'-b' がついて出力されます（この例では、'object\_raw.lst-b' という名前になります）。もしリストの中に悪いフレームが一枚も無かった場合も、もとのリストをそのままコピーして、'-b' がついたリストが作成されます。サンプルデータでは、IRCA00091688, IRCA00091760 がダークパターンが他とは異なる画像ですので、これらのフレームをリストから削除しておきます。参考のために、実際に悪いフレームと正常なフレームを引き算した結果がどのようなになるかの例を図に挙げます。



(a) 悪いフレームと正常なフレームの引き算

(b) 正常なフレーム同士の引き算

図 9 引き算による悪いフレームの確認の例。

サンプルデータでは、オブジェクトの観測は、9点ディザリングのセットを複数繰り返して行っています。スカイフレームの作成時には、ディザリングの1セット単位のリストがあると便利なので、ヘッダのディザリング位置 (LDTHPOS) の情報を見て、そのようなリストを作ります。ここでは、ircs\_imgred パッケージにある `mksubset` コマンドを使ってディザリングセット毎のリストを作ることにします。

```
cl> mksubset object_raw.lst-b object_raw
```

サンプルデータでは、このコマンドを実行すると `object_raw[01-10].lst` という10個のファイルができます。

例えば、`object_raw01.lst` というファイルを開くと、9個の `fits` ファイルの名前が書かれています。試しに、この画像リストを `ds9` で見てみましょう。

```
cl> !ds9 &
```

```
cl> imexam @object_raw01.lst
```

このコマンドで画像が表示されますが、`ds9` 上で'n' と打つとリスト上の次のファイルが、'p' と打つと前のファイルを見ることができます。9枚の画像を見てみると、図の様な9点のディザリングをしている事が分かると思います。

## 4.4 ダークフレームの作成

ダーク画像は、通常観測後に観測した天体と同じ積分時間、COADD 数、NDR 数で 10 枚程度取得します。これらの画像から、以下のようにして 1 枚のダークフレームを作成します。

具体的な手順は、以下の通りです。

1. オブジェクトフレームの積分時間、NDR 数( サンプルデータでは積分時間 120 秒、NDR=8 ) に合ったダークフレームのリストを作成します。

```
$ grep DARK file_header.lst | awk '{if($3==120 && $5==8) print $1}'  
> object_dark.lst
```

2. 前後に取得したフレームとの引き算、割り算をすることで、悪いフレームが無いかを確認し、あればリストから除きます( サンプルデータでは悪いフレームはありません)。ここでは、前節で紹介した `frcheck` というスクリプトを使用します。

```
cl> frcheck object_dark.lst mklist=yes calc=subtract
```

3. 全てのダーク画像 (`object_dark.lst-b`) を、メジアンで重ね合わせる。重ね合わせの際には、各フレームのピクセルのメジアン値が等しくなるように、重ね合わせの前にそれぞれのフレームに一定値を足し引きする作業 (`zeroing`) と、重ね合わせ時に、 $\pm 3\sigma$  を超えるものをサンプルから外す作業 (`clipping`) を行っています。この作業は、`mkdark` という `ircs_imgred` のスクリプトで行います。

```
cl> mkdark @object_dark.lst-b object_dark.fits
```

オブジェクトフレームの生データには、原理的にはダーク分が乗っていますが、IRCS 撮像データはダークカウントが非常に小さく、また最終的にはスカイを引く段階で引算することになるため、ここで作ったダークフレームは、オブジェクトフレームの解析に直接は使いません。ダークフレームは、後節で説明するバッドピクセルマスクの作成時に使用します。

## 4.5 キャルフラットフレームの作成

フラットフレームとは、検出器に一様な光を当て、検出器が読み取った各ピクセルでのカウント値をその平均値で割ったものです。もし、検出器に感度むらが無ければフラットフレームの全てのピクセルの値は 1 になります。実際には、感度の良い所は 1 より大きくなり、悪いところは 1 より小さくなります。そのため、このフラットフレームで生データを割ることで、検出器の感度むらの補正をすることができます。

IRCS の場合は、人工的な光源 (キャル) を用いて検出器を一様に照射した画像から作ります (キャルフラット)。通常は、観測終了後に人工光源を点灯し、観測装置のセッティング (フィルター、ピクセルスケール等) を天体観測時と同じにして撮像します。このフレーム (ON) には、一様光以外にも、ダーク、望遠鏡、観測装置からの熱放射等も乗っているため、ON のフレームを撮った後、人工光源を消したフレーム (OFF) も撮ります。OFF のフレームはダーク + 熱放射のフ

フレームになっているため、ON - OFF をすることで一様光だけのフレームを作ることができます。サンプルデータでの具体的な解析手順は以下の通りです。

1. 'file\_header.lst' から、キャルフラットに該当するファイル名を抜き出す。キャルフラットの ON, OFF 画像は、それぞれ "IMAGE\_Kp\_ON" と "IMAGE\_Kp\_OFF" というオブジェクト名がついています。

```
$ grep IMAGE_Kp_ON file_header.lst | awk '{print $1}' > calflat_on.lst
$ grep IMAGE_Kp_OFF file_header.lst | awk '{print $1}' > calflat_off.lst
```

2. ON, OFF の各フレームにおかしなフレームが無いか確認し、あればリストから除きます (サンプルデータでは、IRCA00091869, IRCA00091869 を除きます)。

```
cl> frcheck calflat_on.lst mklist=yes calc=subtract
cl> frcheck calflat_off.lst mklist=yes calc=subtract
```

3. OFF の各フレームをメジアンで重ね合わせたものを、ON の各フレームから引き、ON-OFF の一様光だけのフレームを作ります。ON-OFF の各フレームを、それぞれのピクセルの平均値で割り、メジアンで重ね合わせ、平均値を 1 で規格化したキャルフラットフレームを作成します。この作業は、ircs\_imgred の `mkcalflat` というコマンドを使って行います。

```
cl> mkcalflat @calflat_on.lst-b @calflat_off.lst-b calflat.fits mask=none
```

## 4.6 バッドピクセルマスクの作成

一般的に、赤外線検出器には、CCD に比べてバッドピクセルが多く存在します。バッドピクセルには、主に以下の 2 種類があります。

- 感度が悪すぎるピクセル
- ダークカレントが高すぎるピクセル

これらのバッドピクセルは、解析の過程で取り除かなければいけません。バッドピクセルを取り除くのに、IRAF の `fixpix` を使用します。このコマンドは、バッドピクセルのリストにあるピクセルを、周りのピクセルを使って補間するというものです。このコマンドでは、バッドピクセルのある位置のピクセルのカウント値を 1 (または正の整数値) とし、それ以外を 0 とした画像を使用します。このようにして作った画像をバッドピクセルマスクといいます。以下に、具体的なバッドピクセルマスクの作り方を説明します。バッドピクセルマスクは、一度作ってしまえば頻繁に作り直す必要はなく、検出器の状態に変更がなければ IRCS の他の観測データの解析にも使うことができます。

バッドピクセルマスクは、前節までに作ったダークフレームとキャルフラットフレームから作ることができます。まず始めに、ダークフレームのピクセルカウント値のヒストグラム (図) を見て、ダークカレントが高すぎるピクセルとみなす閾値を決めます。サンプルデータでは、2500ADU 以

上をバッドピクセルとみなすことにします。次に、キャリフラットフレームのピクセルカウント値のヒストグラム (図) を見て、感度が低すぎるピクセルとみなす閾値を決めます。サンプルデータでは、0.3 以下をバッドピクセルとみなすことにします。この閾値を用いて、ダークカレントが高すぎるピクセル、または感度の低すぎるピクセルを 1 として、それ以外のピクセルを 0 とするフレームを作成し、これをバッドピクセルマスクとします。この作業は、ircs\_imgred の `mkbpmask` というコマンドを使って行います。

```
cl> mkbpmask object_dark.fits calflat.fits ircsing_bpmask.fits
Showing histogram of object_dark.fits
Input threshold for high dark current pix mask:
2500
Showing histogram of calflat.fits
Input threshold for low sensitivity pix mask:
0.3
```

バッドピクセルマスクの完成例を図に示します。

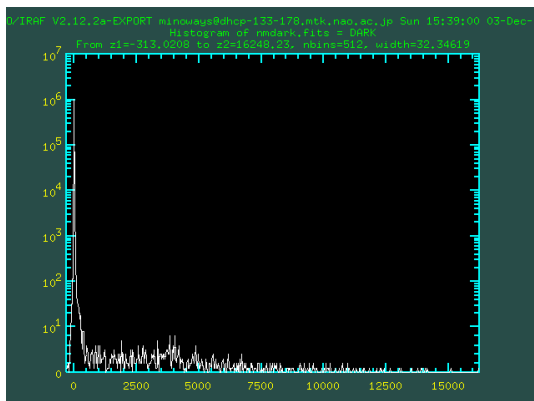


図 10 ダークフレームのピクセルカウント値のヒストグラム。横軸がカウント値 (ADU)、縦軸がそのカウント値をもつピクセル数である。

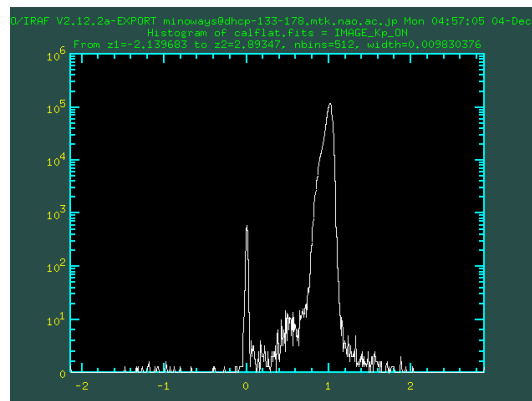


図 11 フラットのピクセルカウント値のヒストグラム。横軸が感度、縦軸がその感度をもつピクセル数である。

#### 4.6.1 スカイフラットの作成

スカイフラットは、ディザリングをして観測天体が写ったフレームからダークを引いたものを、重ね合わせ、平均値で割算して作ります。天体のデータは、ディザリングをして取得しているため、フレーム毎に天体が写る位置が異なるり、オフセットさせずにメジアンで重ね合わせれば天体からの寄与は消え、スカイバックグラウンドに検出器の感度むらをかけたイメージを作ることができます。スカイバックグラウンドは、複数枚重ね合わせたイメージでは一定になっていると考えられるので、このイメージを平均値で割れば感度むらの情報だけが残ったフラットフレームを得ることができます。しかし、重ね合わせる枚数が少ないと、S/N が悪くなってしまうため、スカイフラットを作る場合は十分な枚数が必要です。

基本的には、ダークを引いたオブジェクトフレームのピクセルカウントの平均値で割り、メジアンで重ね合わせれば良いのですが、明るい天体の周りでは天体がディザリング幅以上に広がっている場合があり、メジアンで重ね合わせても天体からのシグナルの影響をうまく取り除くことができ

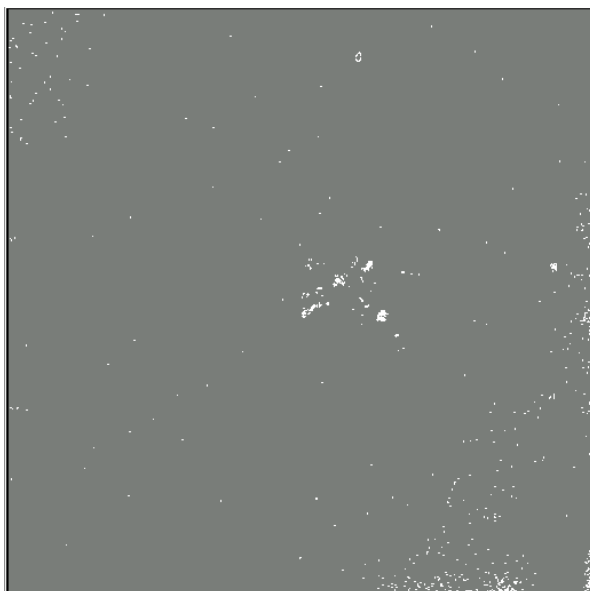


図 12 IRCS 検出器上のバッドピクセルの位置を示した、バッドピクセルマスク。

ません。そのため、明るい天体にはマスクをかけ、メジアンで重ね合わせます。

スカイフラットを作る手順は以下の通りです。

1. オブジェクトフレームの生データを、各々のピクセルカウントの平均値で割る。
2. 平均値で割り規格化したオブジェクトフレームをメジアンで重ね合わせ、暫定的なフラットフレームを作る。
3. オブジェクトフレームの生データを暫定フラットで割る。
4. フラットで割ったフレームをバッドピクセルを補間する (で作ったバッドピクセルマスクを使用)。
5. バッドピクセルを補間した各フレーム内の明るい天体にマスクをかける領域を決める (図参照)。
6. オブジェクトフレームの生データ対し、上で作ったマスクをかけてピクセル値の平均値を計算し、その値で割り算する。
7. 平均値で割ったフレームに、マスクをかけてメジアンで重ね合わせ、最終的なスカイフラットを作る。

この作業は、`ircs_imgred` の `mkskyflat` を使って行います。

```
# オブジェクトマスクのリストと保存先ディレクトリを作成
$ mkdir objmask
$ sed -e "s/sample1/objmask/" object_raw.lst-b > object_mask.lst
cl> epar mkskyflat
```

I R A F

Image Reduction and Analysis Facility

PACKAGE = `ircs_imgred`

TASK = `mkskyflat`

```

objlist = @object_raw.lst-b List of raw object frames with @mark
output = object_skyflat.fits Output sky flat frame
masklist= @object_mask.lst Output object mask list with @mark
(bpmask = yes) Fix bad pixels?
(bpfile = ircsing_bpmask.fits) Name of bad pixel mask frame
(iternum= 50) Number of clipping iterations
(nclip = 3) N-sigma rejection limit
(conv = block 3 3) Convolution kernel(objmask)
(hsig = 2.) Sigma threshold above sky (objmask)
(lsig = 10) Sigma threshold below sky (objmask)
(minpixe= 50) Minimum number of pixels in detected objects(obj
(list1 = )
(list2 = )
(mode = ql)
# :g で実行

```

mkskyflat では、マスク領域を決めるのに noao の nproto というパッケージに入っている objmask というコマンドを使っています。このコマンドでは、適当なフィルター関数 (上の例では conv=block 3 3) で convolution しなませた画像内で、ある閾値を超えるピクセルが、ある面積以上連続して繋がっている場合、その領域を 1、それ以外を 0 とするマスクを作ります。サンプルデータでは、閾値  $2\sigma$  を超えるピクセルが 50 ピクセル以上繋がっているものをマスクする様にすると良いでしょう。これらのパラメータは、解析したい画像セット毎に最適になるように調整してください。オブジェクトマスクを確認するには、ircs\_imgred の check\_objmask を使うと便利です。

```

cl> check_objmask object_raw.lst-b object_mask.lst
add sample1/IRCA00091689.fits,objmask = objmask/IRCA00091689.fits[pl]
sample1/IRCA00091689.fits updated

### sample1/IRCA00091689.fits with object mask objmask/IRCA00091689.fits
### Press [return]

z1=30784.63 z2=39642.23
z1=30784.63 z2=39642.23
...

```

## 4.7 感度むらの補正

感度むら補正は、オブジェクトフレームをフラットフレームで割って行います。ここまでの作業で、キャルフラットとスカイフラットの両方を作成しましたが、IRCS 撮像データの場合、経験上スカイフラットで割算した方が感度むらの補正がうまく行える事が分かっています。図は、キャル



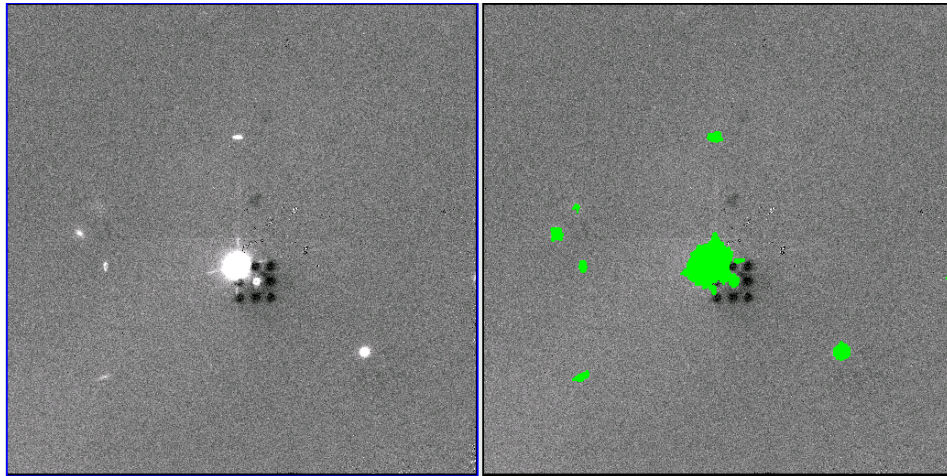


図 13 オブジェクトマスクをかけたときの例。左側がマスクをかける前の画像で、右側が `objmasks` コマンドにより作ったマスクを上重ねたものです。

フラットとスカイフラットで割算した結果を比較していますが、キャルフラットの方はフリンジのような模様が見えています。この理由は、キャルフラットとスカイフラットの撮り方の違いにあります。キャルフラットフレームは、カセグレン焦点 (現在はナスミス焦点) 付近に置かれた人工光源を使うため、主鏡、副鏡を通過して来ている光源から作られています。そのため、検出器への光の入りかたが実際の観測の条件とは異なっています。また、キャルフラットは 0.5 秒程度の積分で取得されるため、実際の観測の積分時間とは大きく異なる場合が多いです。真中のフリンジ状のパターンは主にこの違いからくるものではないかと考えています。さらに、キャルフラットを撮るときは、望遠鏡は天頂方向を向いていますが、天体を観測するときは、望遠鏡は当然観測天体の方向に向きます。キャルフラットフレームで割ったことで表れるフリンジ状のパターンは、このような望遠鏡の姿勢の違いにも影響があるのではないかと考えています。スカイフラットの方は天体のフレームから作るため、このような現象は起きていません。そのため、スカイフラットを使って感度むらを補正します。

サンプルデータにはあてはまりませんが、天体が視野全体に広がっているような場合はスカイフラットを作ることができません。通常、そのような天体の場合、別にスカイだけの画像を撮りますが、何らかの事情でそのような画像が用意できない場合には、キャルフラットを使います。キャルフラットを使った時に現れるパターンは一定のパターンですので、感度むら補正の段階で消えなくとも、スカイ引きの段階である程度きれいに消えてしまうことが多いです。

感度むら補正は、オブジェクトフレームをスカイフラットフレームで割り算するだけですが、この段階でオブジェクトフレームを COADDS, NDR の数で割って置きます。また、各フレームをそれぞれの積分時間で割っておき、ピクセル値が単位時間あたりの値になるようにしておきます。この作業は、`ircs_imgred` の `flatfield` コマンドを使って行います。

```
# 感度むら補正をしたフレームのリストと保存先ディレクトリを作成
$ mkdir ff
$ sed -e "s/sample1/ff/" object_raw.lst-b > object_ff.lst
cl> epar flatfield
```

I R A F



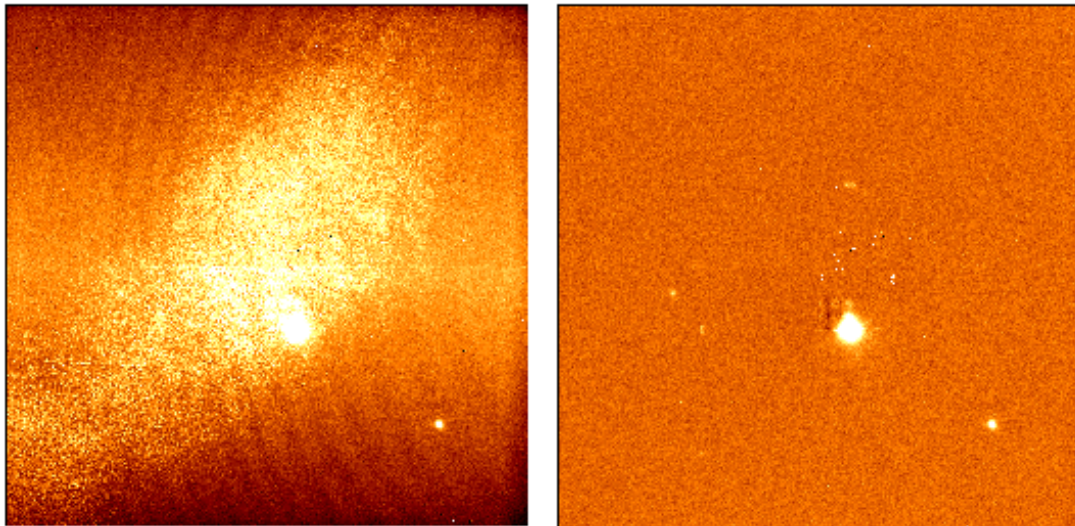
## Image Reduction and Analysis Facility

```
PACKAGE = ircs_imgred
```

```
TASK = flatfield
```

```
rawlist = @object_raw.lst-b Input raw data list with @mark  
outlist = @object_ff.lst Output data list with @mark  
flatimg = object_skyflat.fits Flat image name  
(subdark= no) Subtract dark frame?  
(darkimg= ) Dark frame normalized by COADDS*NDR  
(bpfix = yes) Fix badpixel?  
(bpmask = ircsimg_bpmask.fits) Badpixel mask  
(list1 = )  
(list2 = )  
(list3 = )  
(mode = ql)
```

```
# :g で実行
```



(a) キャルフラットを用いた場合

(b) スカイフラットを用いた場合

図 14 キャルフラットを用いた場合とセルフフラットを用いた場合の違い。キャルフラットを用いた場合には、真中のフリンジ状のパターン、縦方向に伸びた弧状のパターンが残ってしまいます。

### 4.8 スカイバックグラウンドの引き算と重ね合わせ

スカイフレームは、感度むら補正をし、単位時間あたりに直した天体の画像から作ります。赤外線イメージでは、背景放射が強いため、取得したデータのスカイバックグラウンドは空の状態の変化に合わせて時間変化します。そのため、ある1つのフレームでのスカイはなるべく近い時間に取得したフレームから作る必要があります。しかし、あまり近い時間に取得したものだけに限定してしまうと、スカイバックグラウンドのS/Nが悪くなってしまいます。IRCSの撮像データの場合

合、通常は 5~9 位置でディザリングをしているので、1 回のディザリングセットに 1 つずつスカイフレームを作っていきます。スカイフレームを作るには、各ディザリングセットごとのフレームをメジアンで重ね合わせていきます。重ね合わせの際には、スカイフラット作成の節でやったように、明るい天体をマスクしていきます。ただし、スカイフラットの作成時と異なり、重ね合わせる枚数が少ないため、マスクをかけてしまうと、その領域での統計値を出すサンプル数が少なくなってしまい、でき上がったマスクの領域は周りとは若干異なるノイズを持ってしまいます。この影響を避けるために、明るい天体の周りを、周りのピクセルのカウント値で補間するという方法を行います。

具体的な手順は以下の通りです。ここでは、まず始めにオブジェクトマスクを使い暫定的なスカイ引きをしたフレームを、位置合わせをして重ね合わせたフレームから、ピクセル補間をする明るい天体の領域を決定します。

1. スカイフラットの作成時に作ったオブジェクトマスクを使い、暫定的なスカイ引きをしたフレームを作る。スカイフレームは、ディザリングセット毎に重ね合わせたフレームを、各々のフレームのピクセルカウントのメジアンに合うようにスケールします。この作業は、`ircs_imgred` の `maskskysub` コマンドを使って行います。

# 暫定的なスカイ引きをしたフレームのリストと保存先ディレクトリを作成

```
$ sed -e "s/ff/sstmp/" object_ff.lst > object_sstmp.lst
```

```
$ mkdir sstmp
```

```
cl> epar maskskysub
```

Image Reduction and Analysis Facility

```
PACKAGE = ircs_imgred
```

```
TASK = maskskysub
```

```
fplist =      @object_ff.lst  List of flat fielded frames with @mark
sslist =     @object_sstmp.lst List of sky subtracted frames with @mark
masklist=    @object_mask.lst List of object mask with @mark
(ndthpos=           9) Number of dithering position
(iteranu=          50) Max iteration number for statistic
(nsigma =           3) Sigma rejection limit
(list1 =            )
(list2 =            )
(list3 =            )
(mode =            ql)
#:g で実行
```

2. 暫定的なスカイ引きをした各フレームの、基準フレームからの位置のオフセット量を求める。基準フレームとしては、ディザリングパターンの中心である 1 ポジション目のフレームを選ぶと良いでしょう (ここでは、IRCA00091697 を基準フレームとします)。オフセットの測定は、基準フレームとオフセット量を測定したいフレーム間の相互相関係数のピーク値

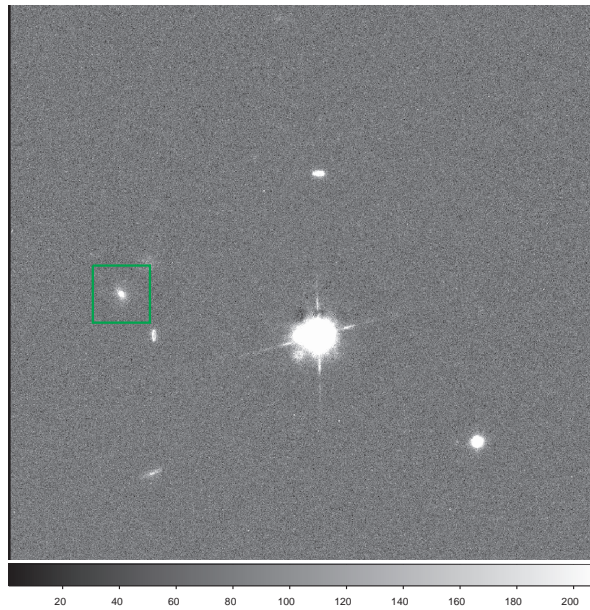


図 15 オフセット量を決めるために相互相関を取る領域。このフレーム内で、サチっていない最も明るい天体の周辺のみ使用。

から求めます。この作業は、`ircs_imgred` の `calcshift` コマンドを使って行います。二つのフレームの相互相関を取る際に、各フレーム内に明るい天体がたくさん受かっていたら画像全体を使えますが、サンプルデータの場合、一つのフレーム内には明るい天体はサチュレーションした星だけで、後は暗めの銀河がいくつか受かっているだけなので、全体の相互相関は取らず、一つの銀河の周り (図 2) だけで相互相関を取るようにします。サンプルデータは 1 ポジション目の座標を中心に  $4 \text{ arcsec} \times 4 \text{ arcsec}$  (約  $70\text{pix} \times 70\text{pix}$ ) の領域内でディザリングをしているので、その約 1.5 倍の  $100\text{pix} \times 100\text{pix}$  の領域を使うことにします。

```
c1> epar calcshift
```

I R A F

Image Reduction and Analysis Facility

```
PACKAGE = ircs_imgred
```

```
TASK = calcshift
```

```
objlist = @object_sstmp.lst List of input images with @mark
output = object_offset.dat Name of output offset data
(referen= sstmp/IRCA00091697.fits ) Reference image
(region = [144:243,441:540]) Image region for cross-correlation
(crmargi= 0) Margin to strip down (crossdriz)
(crtaper= 1) Edge region to taper (crossdriz)
(crboxsi= 100.) Box size where to search for peak (shiftfind)
(mode = ql)
```

3. 暫定的なスカイ引きをしたフレームを位置合わせをして平均値を取るようにして重ね合わせ

る。重ね合わせの際、スカイレベルのメジアン値が0になるようにオフセットさせ、積分時間で重みを付けるようにします。重ね合わせは `ircs.imgred` の `imshiftcomb` を使って行います。このコマンドでは、サブピクセル単位で位置合わせを行い重ね合わせをします。

```
cl> epar imshiftcomb
```

I R A F

Image Reduction and Analysis Facility

```
PACKAGE = ircs_imgred
```

```
TASK = imshiftcomb
```

```
input   = @object_sstmp.lst  List of input images with @mark
output  = tmp_objcombine.fits Output combined image
offset  = object_offset.dat  Input image offsets
(expw   =                      no) Create exposure map?
(expmap =                      ) Name of exposure map
(imgsize=                      1536) Print image size
(list1  =                      )
(list2  =                      )
(mode   =                      ql)
#:g で実行。
```

このコマンドでは、1024×1024 の元画像を大きめのサイズ (`imgsize` で定義) の中心に置き、IRAF の `imshift` コマンドでサブピクセルシフトをします。もしディザリング幅が非常に大きい場合には、`imgsize` の値を調整して、シフトをしても元画像がフレーム内からはみ出さないようにしてください。

4. 重ね合わせたフレームから、明るい天体を検出し、周りのピクセルで補間する領域 (中心座標と円の半径) を決める。補間する領域の中心座標は、前のステップで測定したオフセット量を用いて、重ね合わせ前の各々のフレームに合うようにシフトさせておきます。補間領域の決定には、`objaperture.pl` という perl スクリプトを使用します。

```
$ objaperture.pl tmp_objcombine.fits obj_aperture.dat
```

このコマンドでは、SExtractor を使って天体を検出しています。ここでは検出パラメータはソフト内のデフォルト値を使っていますが、補間する領域を確認して、後ほど調整することもできます。補間領域は、`ircs.imgred` の `showap` というコマンドを使って確認できます (図 4 参照)。

```
cl> showap tmp_objcombine.fits obj_aperture.dat
```

`objaperture.pl` コマンドを一度実行すると、`objeaperture.pl.sex` というファイルが作られています。パラメータの調整が必要な場合は、このファイルを編集してください。パラ

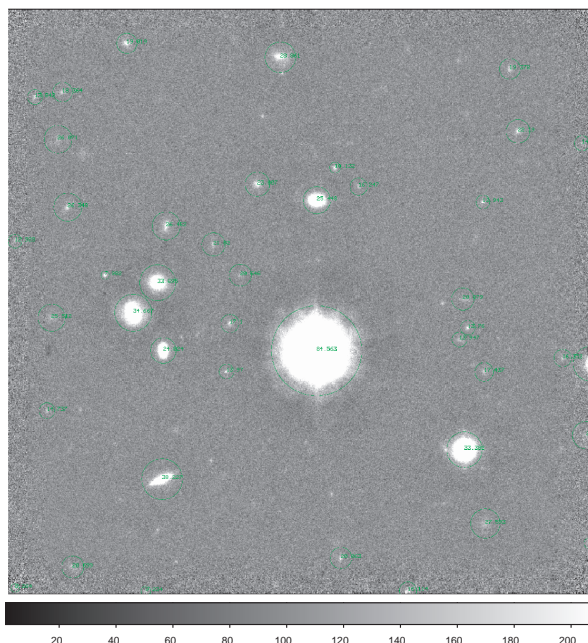


図 16 `objaperture.pl` で決めた、周りのピクセルで補間する明るい天体のある領域。

メータの詳細は SExtractor のマニュアルを参照してください。パラメータを編集後は、以下の様に再度 `objaperture.pl` を実行し、補間領域の決定を行ってください。

```
$ objaperture.pl tmp_objcombine.fits obj_aperture.dat -c objaperture_pl.sex
```

次に、`shift_aperture.pl` というコマンドを使って、重ね合わせ前の各々のフレームに合うように、補間領域をの座標をシフトさせます。シフト量は、暫定的な重ね合わせの前に測定したオフセット値 (`offset.dat`) を使って決めます。

```
$ mkdir objarea
$ sed -e "s/ff/objarea/" -e "s/fits/dat/" object_ff.lst > object_area.lst
$ shift_aperture.pl obj_aperture.dat object_offset.dat object_area.lst
```

5. 感度むら補正をしたフレームについて、上で求めた明るい天体の周辺領域を周りのピクセル値で補間する。この作業は `ircs_imgred` の `eraseobj` コマンドを使って行います。

```
# 感度むら補正をしたフレームから、明るい天体を消したフレームのリストと保存先ディレクトリを作成
$ mkdir ff-obj
    $ sed -e "s/ff/ff-obj/" object_ff.lst > object_ff-obj.lst
cl> epar eraseobje
```

I R A F

Image Reduction and Analysis Facility

PACKAGE = `ircs_imgred`

TASK = `eraseobj`

```

inlist =      @object_ff.lst  Input file list name with @mark
masklist=    @object_area.lst  Mask coordinate data file list with @mark
outlist =    @object_ff-obj.lst  Output file list name with @mark
(minsize=          1) Minimum mask aperute size
(list1 =          )
(list2 =          )
(list3 =          )
(list4 =          )
(mode =          ql)
# :g で実行

```

6. 明るい天体の領域を補間したフレームから、各ディザリングセット毎にスカイフレームを作り、最終的なスカイ引きを行う。スカイフレームは、各々のフレームのピクセルカウントのメジアンに合うようにスケーリングします。この作業は、ircs\_imgred の eraseskysub コマンドを使って行います。

```

# スカイ引きをしたフレームのリストと保存先ディレクトリを作成
$ sed -e "s/ff/ss/" object_ff.lst > object_ss.lst
$ mkdir ss
cl> epar eraseskysub

```

I R A F

Image Reduction and Analysis Facility

PACKAGE = ircs\_imgred

TASK = eraseskysub

```

fflst =      @object_ff.lst  Flat fielded image list name with @mark
ffobjlst=    @object_ff-obj.lst  Object subtracted ff image list name with @mark
sslst =      @object_ss.lst  Sky subtracted image list name with @mark
(ndthpos=          9) Number of dithering position
(iteranu=          50) Max iteration number for statistic
(nsigma =          3) Sigma rejection limit
(list1 =          )
(list2 =          )
(list3 =          )
(mode =          ql)
# :g で実行

```

7. スカイ引きをしたフレームの基準フレーム (ディザリング 1 ポジション目) からのオフセット量を求めます。暫定的なスカイ引きフレームでオフセット量を求めた場合と同様に、

calcshift コマンドを使います。

```
cl> epar calcshift
```

I R A F

Image Reduction and Analysis Facility

```
PACKAGE = ircs_imgred
```

```
TASK = calcshift
```

```
objlist = @object_ss.lst List of input images with @mark
output = object_offset_final.dat Name of output offset data
(referen= ss/IRCA00091697.fits ) Reference image
(region = [144:243,441:540]) Image region for cross-correlation
(crmargi= 0) Margin to strip down (crossdriz)
(crtaper= 1) Edge region to taper (crossdriz)
(crboxsi= 100.) Box size where to search for peak (shiftfind)
(mode = ql)
```

#:g で実行。

8. スカイ引きをしたフレームの平均値を取り重ね合わせ最終的な画像を作ります (図参照)。重ね合せの際、スカイレベルが0になるようにオフセットさせ、積分時間で重みを付けるようにします。暫定的な重ね合わせをした場合と同様に、この作業は imshiftcomb コマンドを使って行います。ここで、expw = yes として、最終画像の積分時間マップも同時に作成しておきます。このマップは、S/N を見積もるときの目安になり、SExtractor 等で自動天体検出をかける際の重みとしても使うことができます。

```
cl> epar imshiftcomb
```

I R A F

Image Reduction and Analysis Facility

```
PACKAGE = ircs_imgred
```

```
TASK = imshiftcomb
```

```
input = @object_ss.lst List of input images with @mark
output = object_final.fits Output combined image
offset = object_offset_final.dat Input image offsets
(expw = yes) Create exposure map?
(expmap = object_expmap.fits ) Name of exposure map
(imgsize= 1536) Print image size
(list1 = )
(list2 = )
(mode = ql)
```

#:g で実行。



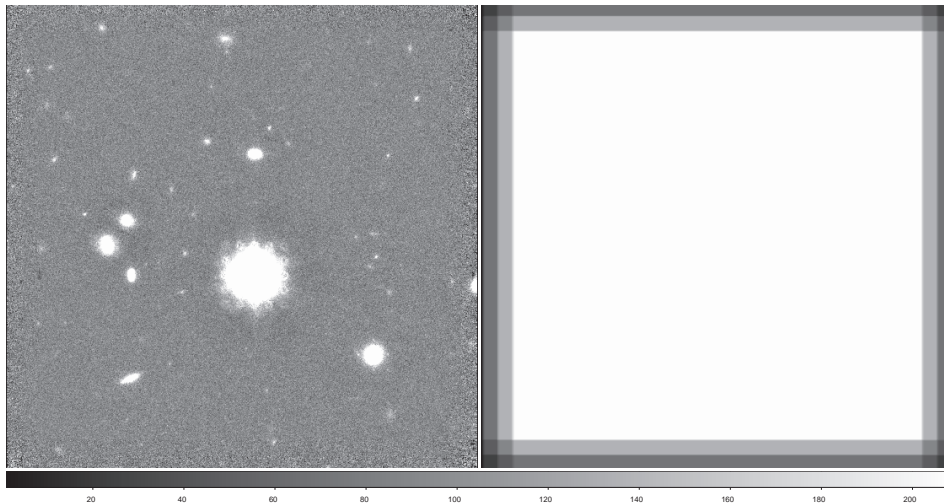


図 17 オブジェクトの最終画像 (左) と積分時間マップ (右)。

## 4.9 標準星の解析

標準星 (P330-E) のフレームも、基本的にはオブジェクトフレームで行ってきた解析方法と同様にできます。

1. 生データリストの作成、画像の確認。

```
$ grep P330-E file_header.lst | awk '{print $1}' > stdstar_raw.lst
cl> frcheck stdstar_raw.lst mklist=yes calc=subtract
# IRCA00091860 をリストから除く
```

2. スカイフラットの作成

```
$ sed -e "s/sample1/objmask/" stdstar_raw.lst-b > stdstar_mask.lst
cl> unlearn mkskyflat
cl> mkskyflat.bpmask = yes
cl> mkskyflat.bpfile = "ircsimg_bpmask.fits"
cl> mkskyflat.hsig = 2
cl> mkskyflat @stdstar_raw.lst-b stdstar_skyflat.fits @stdstar_mask.lst
```

3. 感度むら補正

```
$ sed -e "s/sample1/ff/" stdstar_raw.lst-b > stdstar_ff.lst
cl> unlearn flatfield
cl> flatfield.subdark = no
cl> flatfield.bpfix = yes
cl> flatfield.bpmask = "ircsimg_bpmask.fits"
cl> flatfield @stdstar_raw.lst-b @stdstar_ff.lst stdstar_skyflat.fits
```

4. スカイ引き



```
$ sed -e "s/ff/ss/" stdstar_ff.lst > stdstar_ss.lst
cl> unlearn maskskysub
cl> maskskysub.ndthpos = 5
cl> maskskysub @stdstar_ff.lst @stdstar_ss.lst @stdstar_mask.lst
```

## 5. オフセット量の測定

```
cl> unlearn calcshift
cl> calcshift.referen = "ss/IRCA00091861.fits"
cl> calcshift.region = "[1:1024,1:1024]"
cl> calcshift.crmargin = 50
cl> calcshift.crtapersz = 50
cl> calcshift.crboxsi = 500
cl> calcshift @stdstar_ss.lst stdstar_offset.dat
```

## 6. 重ね合わせ

```
cl> unlearn imshiftcomb
cl> imshiftcomb @stdstar_ss.lst stdstar_final.fits stdstar_offset.dat
```

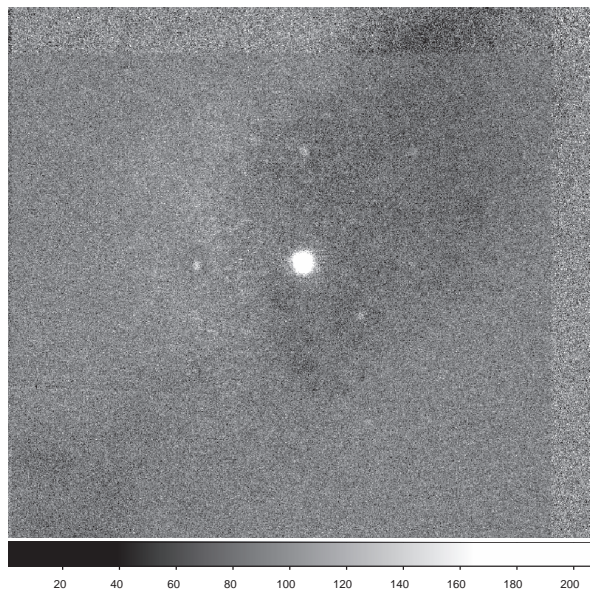


図 18 標準星の最終画像。

## 5 限界等級

リダクションが済んだら、解析した画像上で、どれくらい暗い天体が見えるのかを確認します。画像上でどの程度暗いところまで見えているかは、限界等級という値で表されます。限界等級は、一般的にはある面積の中でのスカイバックグラウンドの揺らぎ ( $1\sigma$ ) の 5 倍 ( $5\sigma$  限界等級) 程度で定義されます。揺らぎを測る面積は、広がった銀河の様な天体の観測の場合には 1 平方秒角とする場合が多いです。点源の限界等級は、シーイングが良く点源の半値幅が 1 秒角を切る様な場合に

は、1 平方秒角よりも小さい面積で測った揺らぎを使って表されることもあります。

限界等級を見積もるには、画像上の 1[ADU/sec] が何等級に相当するか（ゼロ等級）、スカイバックグラウンドの揺らぎ（ノイズ）を測る必要があります。

## 5.1 ゼロ等級の見積もり

ゼロ等級の見積もりには、オブジェクトを観測した日と同じ日に、近い airmass で観測した標準星のデータを使います。今回のサンプルデータの標準星は、Persson et al. 1998, AJ, 116, 2475 にある P330-E という星です。論文にあるカタログを見てみると、この天体の  $K_s$  バンド\*7の見かけの等級は、 $m_{K_s} = 11.429 \pm 0.006$  だとわかります。この星が、画像上で何 [ADU/sec] になっているかを求め、そこからゼロ等級を計算します。そのために、リダクションが済んだ標準星の測光を行います。

測光には様々な方法がありますが、ここでは IRAF の APPHOT を使った方法を紹介します。APPHOT は、アパーチャー測光という方法で測光を行うソフトです。アパーチャー測光とは、図のように、ある半径の円形のアパーチャー内に含まれるフラックスを、ある内径、幅の円環内で決めたスカイレベルを引いて求めるという手法です。APPHOT を使って測光を行う前に、天体を測るためのアパーチャーの半径、スカイレベルを決めるための円環の内径と幅を決めます。以下に、具体的な APPHOT での標準星の測光の仕方を紹介します。

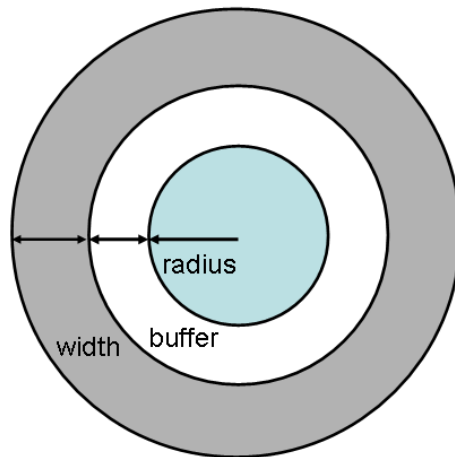


図 19 APPHOT で測光する際に使用する、円形アパーチャー、及び補間をするための周りのピクセルをサンプルする円環の取り方の例。半径 radius ピクセル内の領域を、buffer ピクセルだけ大きい内径を持つ、幅 width ピクセルの円環内のピクセルで補間します。

1. 測光する画像を予め ds9 上に表示させる

```
cl> display stdstar_final.fits 1
```

2. アパーチャー測光をするためのパラメータを設定する。

- 測光したい天体の重心検出のためのパラメータ設定 (centerpars)

\*7 観測データは  $K'$  バンドで撮ったものですが、このバンドの透過率カーブは  $K_s$  とほとんど同じで、等級の差は 0.01 等級以下であるため、ここでは同じであるとみなします。

APPHOT では、interactive に測光を行う場合、ds9 上で天体を選びます。選んだ天体の重心を検出するためのパラメータが centerpars です。ここでは、cbox=5 とし、選んだピクセルから 5×5 ピクセル以内で重心を探すようにしています。

```
cl> unlearn centerpars
cl> centerpars.calgori = "centroid"
cl> centerpars.cbox = 5
cl> centerpars.cthresh = 5
cl> centerpars.minsnra = 5
```

- スカイレベルを決めるためのパラメータ設定 (fitskypars)

fitskypars では、スカイレベルを決めるための円環の内径 (annulus) と幅 (dannulus) の取り方と、その中のピクセルカウントの代表値をどんなアルゴリズムで求めるか (salgorithm) を決めます。ここでは、画像を見ておおまかに円環の内半径 (35pix) と幅 (10pix) を決めていきます。内径と幅は、後から測光結果とともに表示される天体の動径方向のプロファイル (図 3) を見て、変えた方が良ければ、変えて再度測光を行います。スカイレベルを決めるアルゴリズムは、ここではメジアンとしています。

```
cl> unlearn fitskypars
cl> fitskypars.salgori = "median"
cl> fitskypars.annulus = 35
cl> fitskypars.dannulus = 10
```

- アパーチャー測光のためのパラメータ設定 (photpars)

photpars では、アパーチャー半径 (aperture) を決めます。この段階では、アパーチャー半径は、fitskypars で決めた円環の内径、幅と同様に、画像を見て適当に決めます (ここでは 30pix としています)。後から動径方向のプロファイル (図 3) を見て、変えた方が良ければ、変えて再度測光を行います。フラックスを等級に直すためのゼロ等級も指定することができますが、この時点では分からないので、適当な値 (デフォルト値) を入力しておき、測光結果で出てくる等級は無視して、フラックスの結果だけを使うようにします。

```
cl> unlearn photpars
cl> photpars.aperture = 30
```

### 3. 測光を実行する。

```
cl> unlearn phot
cl> phot.output = "stdstar_final_phot.dat"
cl> phot.interac = yes
cl> phot.radplot = yes
cl> phot stdstar_final.fits
```

phot を実行した後、ds9 上に移り、測光したい天体の中心付近で [スペース] を押します。そうすると、IRAF のターミナル上に以下のように測光結果が簡易的に表示されます。

```
Warning: Graphics overlay not available for display device.
stdstar_final.fits    632.63    693.71   -0.03782    12.766  ok
```

結果が表示されても、別の天体を同じパラメータで測光するような場合には、その天体の上にカーソルを合わせてスペースを押すと、続けて測光ができます。phot を終了させるには、ds9 上で”q” と打ち、ターミナル上で再度”q” と打ちます。

(radplot) を yes にしておく、と、図 3 の様に、測光する天体の動径方向のプロファイルと、指定したアパーチャー、スカイを決めるための円環の位置が表示されます。このプロファイルを見て、指定したアパーチャーサイズ等を変更する必要があるかどうかを判断します。

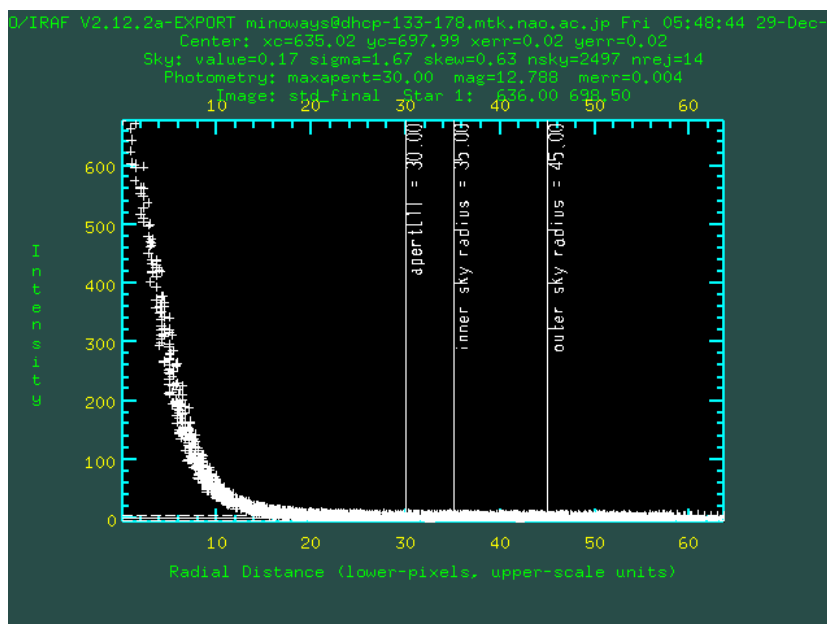


図 20 APPHOT による測光結果とともに表示される、天体の動径方向のプロファイル。アパーチャーの半径、スカイを決める円環の内径、と外 径が、天体のプロファイル上に表示されます。

phot による測光結果の詳細は、パラメータで指定した “stdstar\_final\_phot.dat” というファイルに、設定したパラメータとともに書出されています。以下は、測光結果の部分だけ抜き出してきたものです。

```
#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      SSKREW      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
#
tdstar_final      632.000      693.000      1      nullfile      0      \
632.628      693.709      0.627      0.709      0.044      0.044      0      NoError      \
-0.03782059      1.937127      -0.3873699      2492      20      0      NoError      \
1.      INDEF      INDEF      INDEF      \
30.00      78170.34      2827.482      78277.28      12.766      0.004      0      NoError

```

この結果から、標準星のフラックスは  $F=78277.28$  [ADU/sec] である事が分かります。測ったフラックスを  $F$  [ADU/sec]、カタログ等級が  $m$  等級であるとする、以下の式により観測した画像上での 1 [ADU/sec] に相当するゼロ等級 ( $m_0$ ) が分かります。

$$m_0 = m + 2.5 \log(F)$$

この式に、実際の値である  $F = 78277.28$  [ADU/sec]、 $m = 11.429$  を代入すると、ゼロ等級は、 $m_0 = 23.66$  と求まります。

## 5.2 ノイズの見積もり

近赤外線撮像の場合、多くの場合はバックグラウンドリミット (節参照) なので、ノイズはスカイバックグラウンドのピクセル (または、ある面積の領域) のカウント値のばらつきから求めます。ピクセル間のカウント値のばらつきは、IRAF の `imstat` コマンドで見積もることができます。今回のサンプルデータの様に、視野内に広がった天体が無ければ、画像全体で rejection をかけて、`imstat` コマンドを実行すれば良いですが、広がった天体があるような場合には、天体からのフラックスの影響が及んでいない背景光だけの領域を選び、そこだけで `imstat` コマンドを実行します。今回は、リダクションの途中で明るい天体の寄与を消したフレーム (`ff-obj`) を作っていますので、そのフレームからスカイを引き、位置合せをして重ね合わせると、明るい天体からの寄与を消した画像ができます。

```

$ mkdir ss-obj
$ sed -e "s/ff/ss-obj/" object_ff.lst > object_ss-obj.lst
cl> unlearn eraseskysub
cl> eraseskysub @object_ff-obj.lst @object_ff-obj.lst @object_ss-obj.lst
cl> unlearn imshiftcomb
cl> imshiftcomb @object_ss-obj.lst object_background.fits object_offset_final.dat

```

この画像から、`imstat` コマンドでバックグラウンドノイズを見積もります。以下、`imstat` コマンドの実行例です。

```
cl> unlearn imstat
```

```
cl> imstat object_background.fits nclip=50
```

#	IMAGE	NPIX	MEAN	STDDEV	MIN	MAX
	object_background.fits	1190695	0.009196	0.0163	-0.03971	0.0581

この結果から、ピクセル間のカウント値のばらつき (STDDEV) は、 $\sigma = 0.0163$  であることが分かる。

半径  $r$  秒角のアパーチャーでのノイズ ( $\sigma_r$ ) は、節の S/N の式で表したように、ピクセル間のカウントのばらつき ( $\sigma$ ) から、以下の式を使って見積もる事ができます。

$$\sigma_r = \sigma \times \sqrt{\pi(r/p)^2}$$

ここで、 $p$  はピクセルスケール [arcsec/pix] を表しています。

ただし、実際にはアパーチャー半径  $r$  とノイズの関係は線形ではなく、ノイズ  $\sigma_r$  はアパーチャー半径  $r$  の 2 次の関数で増えていく傾向になる場合が多いです (図参照)。このようになる原因は、バックグラウンドの揺らぎがピクセル毎に独立ではなく、大きなスケールで相関を持っているからではないかと考えています。大きなアパーチャー (半径  $R$  秒角) での限界等級を見積もりたい場合には、ノイズ  $\sigma_R$  は imstat コマンドを使って計算したピクセル間の揺らぎから求めるのではなく、半径  $R$  秒角のアパーチャー内のフラックスを、天体の無い背景光だけの領域でいくつか測り、そのフラックスの分散を半径  $R$  秒角のアパーチャー内でのノイズとします。ただし、アパーチャー半径が小さい場合には、図にあるように、ほぼ線形であるとみなせるため、ピクセル間の揺らぎから求めたノイズでも問題ありません。

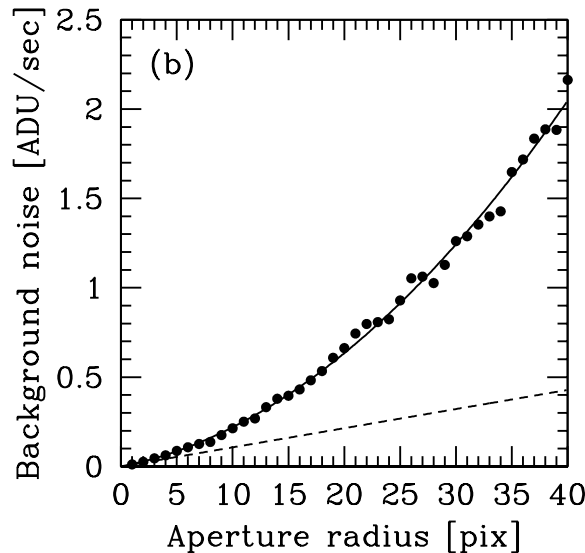


図 21 アパーチャー半径とノイズの関係の例 (Minowa et al. 2005, ApJ, 629, 29)。原理的には、アパーチャー半径とノイズの関係は線形になるが (破線)、実際には 2 次の関数で増えていく (黒丸)。

### 5.3 限界等級の見積もり

点源の検出限界 (何等級までの点源を  $5\sigma$  で検出できるか) を知りたい場合には、画像内にある点源の等級と、半径  $r$  秒角のアパーチャーに含まれる点源のフラックス  $F(r)$  (encircled energy)

を、その半径でのノイズで割った  $S/N(r)$  から求めることができます。

$$S/N(r) = F(r) / \left( \sigma \sqrt{\pi r^2 / p^2} \right)$$

例えば、点源の等級が  $m_{psf}$ 、 $S/N$  が半径  $r_{max}$  で最大になるとした場合、点源の  $5\sigma$  検出限界等級  $m_{lim}$  は、以下のように計算できます。

$$m_{lim} = m_{psf} + 2.5 \log(S/N(r_{max})/5)$$

サンプルデータの場合、図 (a) に緑丸で囲った天体が点源です。この点源を使って、実際に限界等級を求めてみます。まず始めに、点源の encircled energy  $F(r)$  を求めます。これは、APPHOT を使って、アパーチャー半径を変えて測光していくと求められます。ここでは、ircs\_imgred にある、`growth` というコマンドを使います。このコマンドは、指定した範囲内でアパーチャー半径を変えて測光した結果をファイルに書出します。使い方は以下の通りです。

```
cl> epar growth
PACKAGE = ircs_imgred
      TASK = growth

in      =      object_final  Input image file name
out     =      growth_objpsf.dat  Output growthcurve data
(rmin   =      1.) Minimum Radius of growth curve
(rmax   =      15.) Maximum Radius of growth curve
(rbin   =      1.) Resolution of growth curve
(salgor =      median) Sky fitting algorithm
(annu   =      15.) Inner Radius of sky annulus
(dannu  =      5.) Width of sky annulus
(sky    =      0.) Constant sky value (/pixel)
(mode   =      ql)
# :g で実行
```

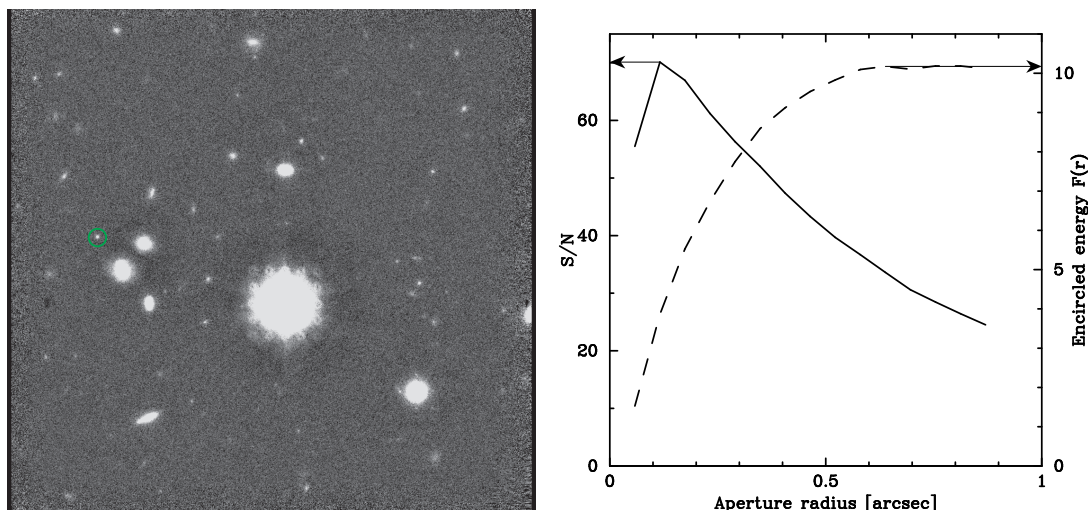
コマンドを実行後、ds9 上に画像が表示されるので、点源の位置にカーソルを合わせて、“a” と打ち、“q” で抜けます。その後、その天体についての encircled energy が “growth\_objpsf.dat” というファイルに書出されます。ここで設定するパラメータは、測光するアパーチャー半径の最小値と最大値、アパーチャー半径を変えるステップ幅、スカイレベルを決めるアルゴリズムと、円環の内径と幅です。これらのパラメータは、測る天体に合わせて変えてください。サンプルデータにある点源では、上記のパラメータでうまくいきます。

ここでできた、encircled energy のデータ ( growth\_objpsf.dat ) から、アパーチャー半径による  $S/N$  の変化を計算します。“growth\_objpsf.dat” は、1 列目に、アパーチャー半径 [pixel]、2 列目に、その半径に含まれるフラックス [ADU/sec] のデータが入っています。以下のコマンドにより、1 列目にピクルスケール  $p = 0.058$  arcsec/pix をかけて単位を秒角に、2 列目を  $\sigma_r = \sigma \sqrt{\pi r^2}$  で割って  $S/N$  に直します。

```
$ awk '{print ($1*0.058), ($2/(0.0163*sqrt(3.1416)*$1))}' growth_objpsf.dat
> snr_objpsf.dat
```



図 (b) は、`growth` コマンドにより求めた点源の encircled energy と、アパーチャー半径による S/N の変化を表しています。この図と `growth_psf.dat` から、この点源の総フラックスは、約 10.2 [ADU/sec] であると分かり、ゼロ等級 ( $m_0 = 23.66$ ) を使って見かけの等級は約 21.14 等だと計算できます。また、`snr_psf.dat` から、この点源の S/N が最大となるアパーチャー半径は  $r = 0.116$  秒角で、その S/N は約 66.8 であると分かるため、点源の  $5\sigma$  限界等級は、 $21.14 + 2.5 \log(66.8/5) \simeq 23.95$  であると計算できます。



(a) サンプルデータ内にある点源の位置

(b) 点源の encircled energy と、S/N

図 22 点源での限界等級の見積もり

次に、どれ位の表面輝度の天体が  $5\sigma$  で検出できるか (1 平方秒角当りの限界等級;  $\mu_{lim}$  [mag/arcsec<sup>2</sup>]) を見積もって見ます。一般的には、表面輝度の限界等級は面積 1 平方秒角内のアパーチャー内のフラックスのばらつき  $\sigma_{1\text{arcsec}^2}$  を用いて以下のように表せます。

$$\mu_{lim} [\text{mag/arcsec}^2] = m_0 - 2.5 \log(5\sigma_{1\text{arcsec}^2})$$

ここで、 $\sigma_{1\text{arcsec}^2}$  は半径 9.7 ピクセルという大きいアパーチャー内のフラックスのばらつきであるため、図にあるように、ピクセル間のカウント値のばらつきから求めたものよりも大きくなります。そのため、 $\sigma_{1\text{arcsec}^2}$  はピクセル間のカウント値のばらつきから求めるのではなく、実際にオブジェクトの最終画像内で、面積 1 平方秒角 (半径 9.7 ピクセル) のランダムなアパーチャー内のフラックスを測定し、その分布を正規分布と仮定してガウシアンフィットをして求めます。この作業は、`ircs_imgred` の `calcnoise` を使って行います。オブジェクトの最終画像 “`object_final.fits`” で、半径 9.7 ピクセルの円形アパーチャー内のフラックスのばらつきを求めます。region パラメータは画像のどの範囲内でフラックスのばらつきを測るかを定めるパラメータです。ここでは、重ね合わせの時に同時に作成した “`object_expmmap.fits`” を見て、積分時間が十分な領域を選んで見ます。maxaper はランダムに配置するアパーチャーの最大数です。測定したフラックスは、IRAF の `histogram` コマンドを使ってプロットされますが、そのプロットを見て、必要があればヒストグラムのパラメータ (`nhistbin`, `hist_z1`, `hist_z2`) を調整し、再度 `calcnoise` コマンドを実行してください。

```
cl> epar calcnoise
```

I R A F



Image Reduction and Analysis Facility

```
PACKAGE = ircs_imgred
TASK = calcnoise
```

```
image = object_final Image file name.
rapertur= 9.7 Aperture radius
region = [71:1025,71:1025] Image region for noise estimation
(maxaper= 500) Maximum number of apertures
(nhistbi= 30) Number of bins in histogram
(hist_z1= INDEF) Min bin value in histogram
(hist_z2= 2.) Max bin value in histogram
(list1 = )
(list2 = )
(mode = ql)
# :g で実行
```

```
### Result
#1 sigma[adu] in aperture radius 9.70[pix]
0.59
```

図に、1 平方秒角のアパーチャー内のフラックスのヒストグラムと、ガウシアンフィットの結果を表しています。これにより、1 平方秒角内のフラックスのばらつき  $\sigma_{1\text{arcsec}^2}$  は、0.59 [ADU] 程度だと分かります。この  $\sigma_{1\text{arcsec}^2}$  の値から、上式を用いて表面輝度の限界等級  $\mu_{lim}$  は 22.5[等/平方秒角] となります。

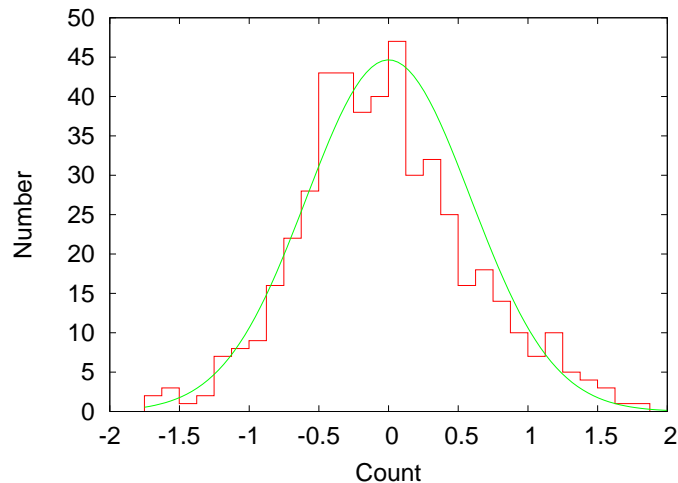


図 23 1 平方秒角のアパーチャーでのノイズの見積り。ランダムなアパーチャー内のフラックスの頻度分布と、ガウシアンフィットをした結果を表しています。分布のプラス側は天体からの寄与があるため、ガウシアンフィットはマイナス側だけで行っています。

## 6 最後に

本マニュアルで使ったスクリプトはご自由にお使ください。また、スクリプトの中身が分かる方は、自分の解析に合わせて書き換えてもらっても構いません。ただし、スクリプトのバグを発見した際には、ぜひ著者までご連絡ください。また、テキスト全体を通して、説明の分かりにくい点、間違った記述などあれば、是非、著者までフィードバックをかけてください。