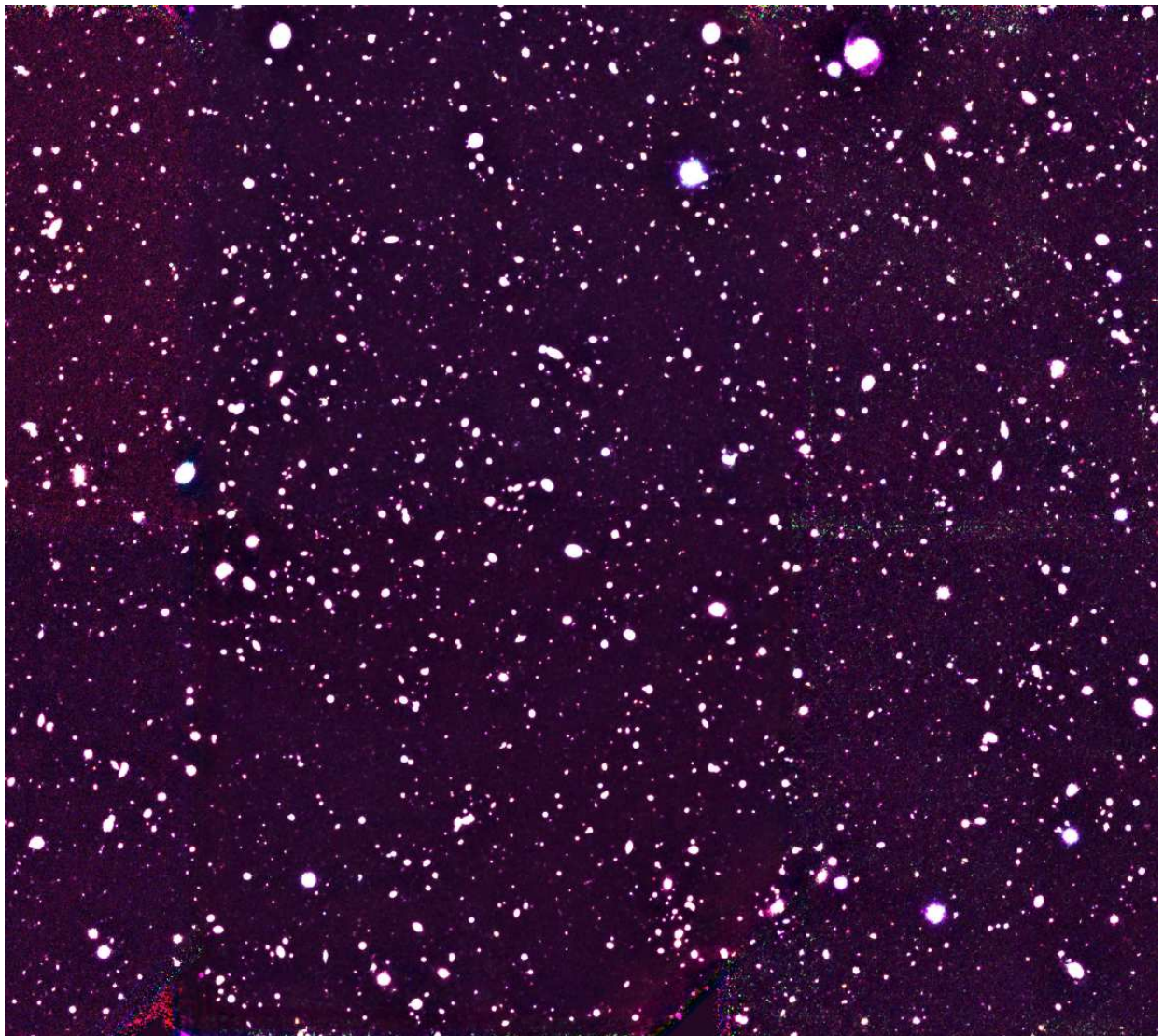


MOIRCS撮像データ処理講習用テキスト
ver 1.0



鍛冶澤 賢 (国立天文台・東北大学)
2009年5月改訂 田中 壺 (国立天文台ハワイ観測所)

目次

1	はじめに	3
2	近赤外撮像観測	3
3	MOIRCS	5
4	MCSRED とは	7
5	MCSRED の取得方法	7
6	MCSRED の使い方	7
6.1	インストール、準備	7
6.1.1	IRAF の準備	8
6.1.2	IRAF v2.12.1 の場合	9
6.1.3	環境変数の設定	10
6.1.4	SExtractor	10
6.1.5	MCSRED の始め方	10
6.2	一括実行タスク: mcsall	12
6.2.1	解析実習に用いる観測データ	12
6.2.2	入力リストの準備 - listprep	15
6.2.3	画像のチェック	15
6.2.4	画像チェックタスク - imcheck	17
6.2.5	有効カウント、COADD	18
6.2.6	タスクの実行	19
6.3	画像のモザイク	24
6.3.1	モザイクタスク dmosimg	24
6.3.2	天体検出 gsexcat	27
6.3.3	位置合わせと足し合わせ gmkgtrimages	28
6.3.4	最終合成データについて	32
6.4	標準星データ	33
6.4.1	部分読みについて	33
6.4.2	標準星データの処理	33
6.4.3	部分読み出しデータの処理	38
6.5	mcsall の後始末	41
6.6	MCSRED 使用上の注意	42
7	mcsall では行われぬ工程	43
7.1	NODDING 観測のデータ処理	43
7.2	一度足し合わせた画像から天体マスクを作る	44
7.3	フリッジパターンの処理	45

A	mcsallで使われる各タスクについて	48
A.1	mcs_mkflat	48
A.2	sbselisky	49
A.3	qmsepskysb	50
A.4	quadcor	51
A.5	mcsgeocorr	52
A.6	gsxtcat	52
A.7	gmkgtrimages	53
B	mcsallの各工程で出力されるファイル	55

1 はじめに

このテキストはすばる望遠鏡の近赤外撮像分光装置 MOIRCS の撮像データ処理の方法について、初学者向けに説明したものです。鍛冶澤 賢氏 (東北大学) により 2007 年すばる秋の学校での講習用に作成されたものを、田中 壱 (ハワイ観測所) がその後の変更を含めて (ちょっとだけ) 補足修正したものです。特に MOIRCS 撮像データ処理のために用意されたパッケージ MCSRED の使い方を中心に説明しています。MCSRED は IRAF 上で動作しますが、IRAF の詳しい説明についてはこのテキストでは割愛します。また、近赤外撮像観測及びデータ解析に関する一般的な説明についても割愛します (次節で紹介している文献を参照してください)。

本テキストに関するご意見、コメント等ございましたら下記までお寄せ下さい。

田中 壱
650 North A'ohoku Place, Hilo, HI 96720, U.S.A.
国立天文台ハワイ観測所
E-mail: ichi "at-mark" subaru.naoj.org

2 近赤外撮像観測

一般的な近赤外撮像データの特徴や処理の流れについては、すばる IRCS 撮像データ解析マニュアルが非常に参考になります¹。近赤外撮像観測の特徴を手取り早く掴みたい方はこちらの 2-5 章を参照してください。

近赤外撮像データ処理はおおよその流れは可視光 CCD 観測のデータ解析とそれほど違いはありません。CCD による撮像データ解析一般については Suprime-cam データ解析マニュアル¹ も参考になるでしょう。可視近赤外の撮像データ解析について未経験であったり、ほとんど知識がないという方はこちらの方も見ておかれると便利です。

撮像観測における可視光と近赤外の主な違いには

- 近赤外域では空、背景光が非常に明るく、時間変動も激しい。
- 検出器の振る舞いが異なる

といった点があります。特に背景光が非常に明るいことにより、検出器の飽和を防ぐため 1 回の積分時間が可視光に比べて短かめに制限され、結果として処理するフレーム数が多くなる傾向があります。また、一枚のフレームをパッと見ただけでは極端に明るいもの以外ほとんどの天体が空に埋もれてしまって何も (空しか) 見えないということがよくあります (図 1。ある程度処理を進めると図 2 のように天体が写っていることがはっきりわかります)。

¹ <http://www.naoj.org/Observing/DataReduction/index.html> にリンクがあります。

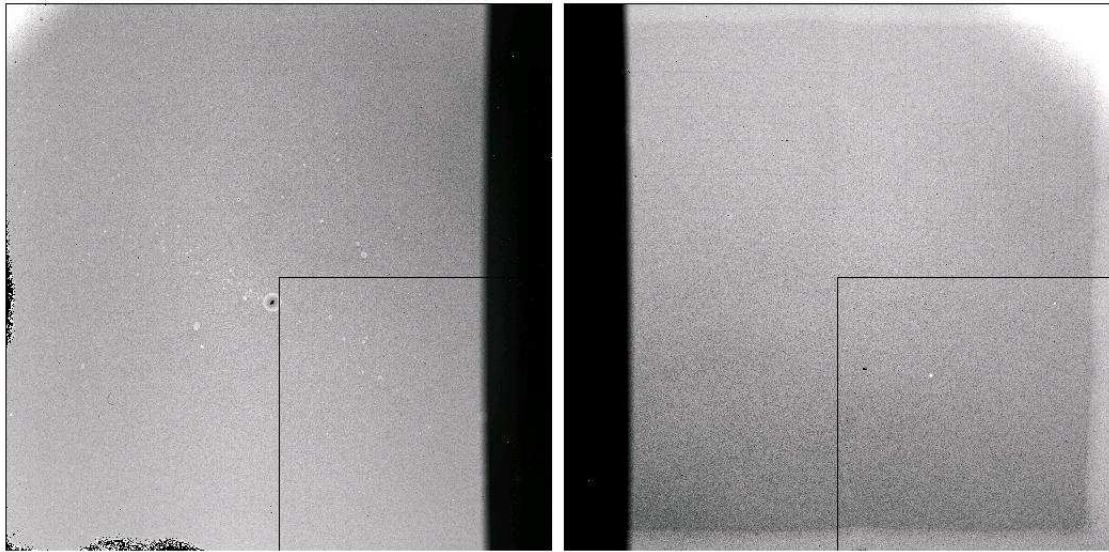


図 1: MOIRCS の生データ。左が channel-1、右が channel-2。

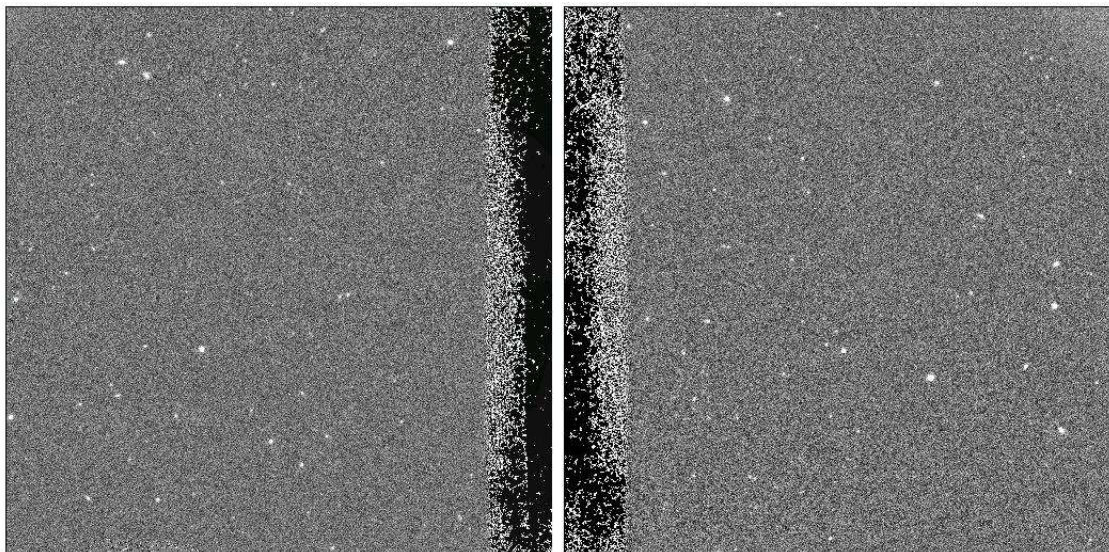


図 2: 図 1 の生データをフラットフィールド、スカイ引きを行ったデータ。

撮像データ処理のおおよその流れは

1. ダーク引き
2. 感度ムラの補正 (フラットフィールド)
3. 背景光 (スカイ) 引き
4. 光学歪み補正
5. 位置合わせ
6. 足し合わせ

のようになります。

可視光や近赤外の撮像観測では通常積分するごとにちょっとずつ位置をずらし(ディザ、あ

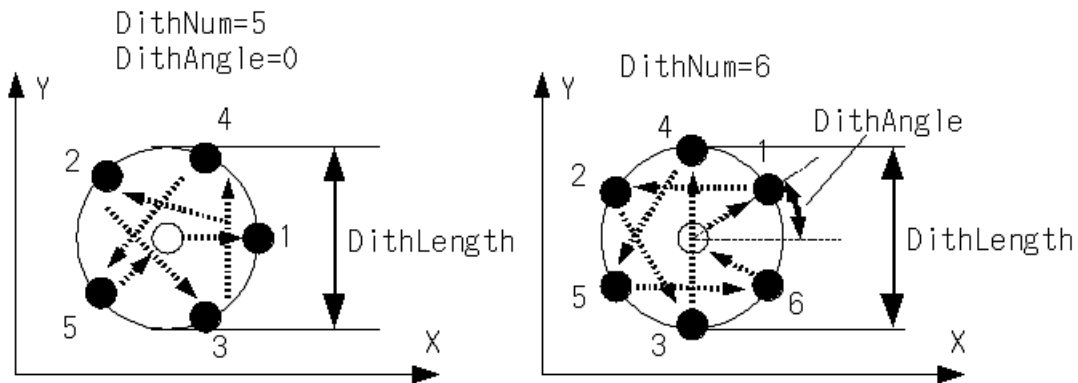


図 3: MOIRCS のディザパターンの例。

るいはディザリングと呼びます) ながら観測を行います (図 3)。これは検出器の bad pixel が天球上での一定の位置の情報を損失させることを回避したり、同じ天体を検出器上の複数の位置に置くことでフラットフィールドなどのピクセルごとの系統的な誤差を軽減するためです。このため複数回に渡って積分したデータを足し合わせる前に視野内の天体が同じ場所にくるように位置合わせを行う必要があります。またディザを使って取得されたデータでは、複数のフレームを位置合わせせずにメジアンをとることによって天体の影響のほとんどない空だけの画像 (メジアンスカイ) を作成することができ、これをスカイ引きに用いることができます。近赤外データでは可視光に比べて背景光が明るく、またその時間変動が激しい傾向があるので、スカイをどれだけきれいさっぱり引けるかという点はその後の解析の精度を決める重要な要素になることがよくあります。

光学歪み補正とは望遠鏡と観測機器の光学系による歪みを取り除く作業です。光学系による歪みが大きく残ったままだとディザを使って観測したフレーム同士を視野全体に渡ってきちっと位置合わせできなくなったりします。また弱い重力レンズの解析などでは天体の微妙な形状の測定においてこの歪みが解析に悪影響を及ぼしたりする場合があります。

3 MOIRCS

MOIRCS (Multi-Object InfraRed Camera and Spectrograph) はすばる望遠鏡のカセグレン焦点で使われる近赤外広視野撮像分光装置です。MOIRCS の詳細についてはすばる望遠鏡 webpage (<http://subarutelescope.org/Observing/Instruments/MOIRCS/index.html>) を参照してください。ここでは撮像モードのデータについて簡単に説明します。

MOIRCS は 2 つの HAWAII-2 というアレイ検出器を使って $4 \times 7 \text{ arcmin}^2$ の視野をカバーしています (図 4 右、以後このテキストでは 2 つの検出器を channel-1、channel-2 などと呼びます)。各々の HAWAII-2 アレイは $2048 \text{ pix} \times 2048 \text{ pix}$ のフォーマットで、ピクセルスケールは $0.117 \text{ arcsec/pixel}$ です。2 つの検出器の互いに接している方では約 300 pix の幅に渡って光が入ってこない領域があります (図 4 左、この領域は分光モードの観測で使われます)。

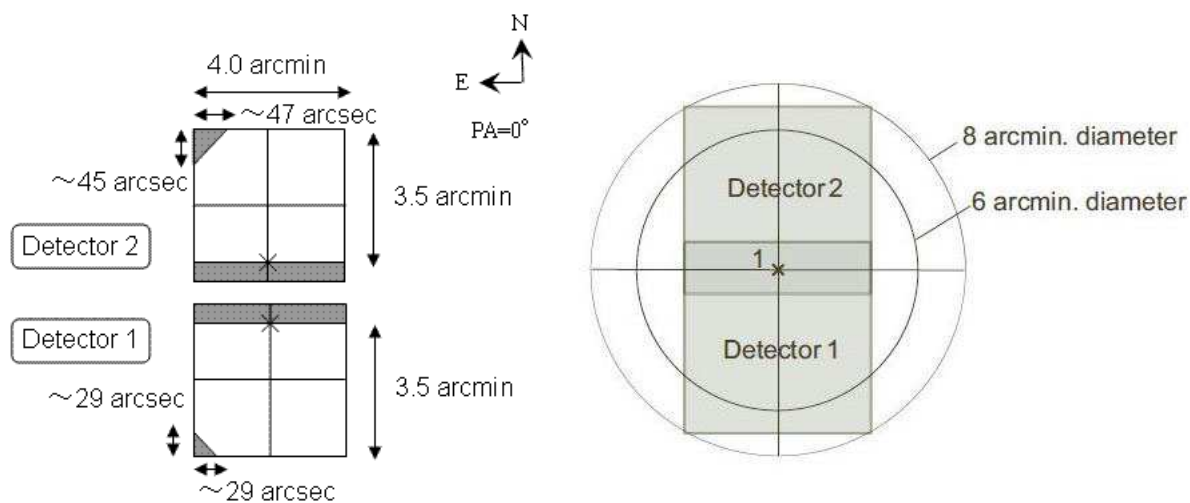


図 4: 2つの検出器による MOIRCS の視野。左側の上と下の角はけられによってデータとしては使えない領域。Detector 1 の上部と Detector 2 の下部の帯状の領域も光が当たっておらず撮像観測では使用されない。

1回積分が行われる度に各検出器から1枚ずつ、計2枚の FITS ファイルが生成されます。MOIRCS の生データの FITS ファイルは MCSA000?????.fits (?????はフレーム ID ナンバーが入ります) という名前がついていますが、channel-1 (図 4 における Detector 1) の検出器に対応する FITS ファイルは常に奇数のフレーム ID ナンバー (つまり下一桁が 1,3,5,7,9)、channel-2 (Detector 2) の検出器の FITS ファイルは常に偶数のフレーム ID ナンバー (channel-1 のナンバーに 1 を足した数) を持ちます。例えば、MCSA00014043.fits と MCSA00014044.fits がそれぞれ同時に観測された channel-1 と channel-2 の検出器のデータで、その次の積分のデータは MCSA00014045.fits と MCSA00014046.fits といった具合です。実際の FITS データの X、Y 軸の方向は図 4 を時計回りの方向に横倒しした向きになっています (図 1 参照)。

4 MCSRED とは

MCSRED は国立天文台・ハワイ観測所の田中 壺が開発を続けている、MOIRCS 撮像データ解析用ソフトウェアです。2009年5月現在においても引き続き開発中の段階であり、今後も頻繁に改良・変更等が行われる可能性がありますので、MCSRED webpage (<http://www.naoj.org/staff/ichi/MCSRED/mcsred.html>) をある程度の頻度でチェックされることを推奨します。

5 MCSRED の取得方法

MCSRED は MCSRED webpage: <http://www.naoj.org/staff/ichi/MCSRED/mcsred.html> から最新版をダウンロードできます。

以下このテキストでは、MCSRED 2009年5月版を前提として使い方を説明します。

6 MCSRED の使い方

上記のWEBサイトから tar ファイルをダウンロードし解凍すると、その中に README.pdf 等の名前の解説があります。これを通常は参照します。また、MOIRCS での観測の際に MCSRED を使いたい観測者に対しては、リクエストに応じて Support Scientist (SS) から簡単な説明とデモがなされます。

以下は MCSRED についての完全な記述ではなく、どのように MCSRED を使っていけばよいかを簡単に説明したものです。

歪み補正及び2つの検出器のモザイクパターンは MOIRCS の昇温-冷却が行われる度に変わります。歪み補正データベースのバージョンが処理・解析しようとしている観測データに合ったものかどうかについては常に注意を払う必要があります。

6.1 インストール、準備

MCSRED は IRAF ver 2.12.2 の使用を前提としています (最新版は、ver. 2.14 でも一通りの動作確認はなされています)。それ以前のバージョンの IRAF では、MCSRED で用いているいくつかの IRAF タスクがなかったり、タスクのパラメータが異なっていることによって、MCSRED が適切に動作しないので注意が必要です。また、SExtractor パッケージも必要です (<http://terapix.iap.fr/> からダウンロードできます)。

まず、MCSRED webpage からダウンロードした tar ファイルを任意のディレクトリに展開します。

```
$ tar xvzf mcsred20090425.tar.gz
```


6.1.1 IRAF の準備

MCSRED は IRAF 上で動作します (IRAF で動く複数のタスクからなるパッケージです)。また、本講習では MCSRED のタスク以外のちょっとした処理やデータのチェックなどにも IRAF を用います。

IRAF (Image Reduction and Analysis Facility) は NOAO (アメリカ光学赤外線天文学観測所) から配付されている天文データの解析を行うための汎用ソフトです。IRAF homepage: <http://iraf.noao.edu/> からダウンロードして使うことができます。

IRAF を使うために、まず `mkiraf` というコマンドを実行します。任意のディレクトリで “`mkiraf`” と入力します。

```
$ mkiraf
- creating a new uparm directory
Terminal types: xgterm,xterm,gterm,vt640,vt100,etc.
Enter terminal type:
```

とターミナルタイプを尋ねてきますので、ここでは `xgterm` を選択します。

```
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.
```

となって、コマンドを実行したディレクトリに `login.cl` という IRAF のログインファイルが新しく作られます。このファイルには IRAF の各種の設定についての情報が入っています。MCSRED を使うためには、この `login.cl` に次のような追加・変更を行います。

以下の 2 文を追加 (`login.cl` の一番最後の `keep` と書いてある行のすぐ上にでも書き加えておくとよいでしょう) :

```
task $mcsred = (tar ファイルを展開したディレクトリのフルパス)/MCSRED/mcsred.cl
set dir_mcsred = "(tar ファイルを展開したディレクトリのフルパス)/MCSRED/"
```

以下の 1 行の変更 (“`# Uncomment and edit to change the defaults.`” の行の少し下に変更する文があります。変更した行の頭の `#` も忘れずに消しておきます。) :

```
#set stdimage = imt800      —>      set stdimage = imt2048
```

IRAF を起動するには `xgterm &` と入力して新しくターミナルを開き、その開いたターミナルで `login.cl` のあるディレクトリに入って (`login.cl` を作ったディレクトリでそのまま `xgterm &` と打ってターミナルを開けば、そのターミナルはすでに `login.cl` のディレクトリに入った状態になっているはずです) `cl` と打ち込みます。

```
$ cl
```

NOAO PC-IRAF Revision 2.12.2-EXPORT Sun Jan 25 16:09:03 MST 2004

This is the EXPORT version of PC-IRAF V2.12 supporting most PC systems.

Welcome to IRAF. To list the available commands, type ? or ??. To get detailed information about a command, type 'help command'. To run a command or load a package, type its name. Type 'bye' to exit a package, or 'logout' to get out of the CL. Type 'news' to find out what is new in the version of the system you are using. The following commands or packages are currently defined:

apropos	images.	mscred.	plot.	stdasd.	utilities.
dataio.	language.	noao.	proto.	system.	
dbms.	lists.	obsolete.	softools.	tables.	

cl>

これでIRAFが立ち上がりました (login.clがないディレクトリでclコマンドを実行してもIRAFがうまく起動しないので注意が必要です)。

MCSREDのいくつかのタスクはIRAFタスクのパラメータがデフォルトの値になっていることを前提としています。上で説明したように新しくlogin.clを作ってIRAFを起動したばかりの状態では各タスクのパラメータはデフォルト設定の状態なので、そのままMCSREDを使用できます。すでにlogin.clを作ったあとにIRAFを使用していて、色々なタスクのパラメータを書き換えている場合には、IRAFに入った状態で

cl> unlearn iraf

のようにunlearnを実行することでIRAFの全タスクのパラメータをデフォルト値に戻すことができます。現在の各タスクのパラメータ設定を別途残しておきたい場合には、IRAFのログインディレクトリ (login.clがあるディレクトリ) にあるuparm/以下のファイルでunlearnを実行する前に別の場所にバックアップしておく必要があります。

IRAFを終了させるにはlogoutコマンドを使います。

cl> logout

6.1.2 IRAF v2.12.1 の場合

(このマニュアルが作成された2007年度の講習の際使用した解析環境ではIRAF v2.12.1がインストールされていました。以下は参考までの記述です。)

IRAF v2.12.1では、MCSREDが前提としてv2.12.2との違いで、ソースのCLスクリプトを一箇所修正しておく必要があります。

MCSRED のソースの CL スクリプトファイルは MCSRED の tar ファイルを展開したディレクトリの下に MCSRED/ディレクトリにあります。

```
$ cd (tar ファイルを展開したディレクトリ)/MCSRED/  
$ ls
```

とすると、****.cl という拡張子のついた CL スクリプトファイルがずらっとでてきます。そのなかの `basename.cl` というファイルの中で、`"strldx"` というタスクを使っている部分がありますが、これは IRAF v2.12.1 では導入されていないので使えません。もし IRAF v2.12.1 で使用する場合には、MCSRED ソースディレクトリの下に `CONTRIB/ディレクトリ` がありますが、この中にある `basename_v2.12.1.cl` を `basename.cl` に上書きして使用してください。

また、この講習資料 7.2 において解説している `invmask.cl` についても、IRAF v2.12.1 で使うためには 1 箇所修正しておく必要があります。

6.1.3 環境変数の設定

MCSRED を動作させるには、前もって環境変数 `MCSRED_DIR` を定義しておく必要があります。

例：bash の場合

ホームディレクトリにある `.bash_profile` か `.bashrc` のファイルに以下の 1 文を書き加えます。

```
export MCSRED_DIR="(full path to the extracted directory)/MCSRED"
```

tcsh の場合

ホームディレクトリの `.cshrc` に以下の 1 文を加えます。

```
setenv MCSRED_DIR (full path to the extracted directory)/MCSRED
```

6.1.4 SExtractor

MCSRED の主要なタスクを使うためには SExtractor がインストールされている必要があります。SExtractor は光赤外分野で広く用いられている天体検出及び測光を行うためのソフトウェアです。もしまだインストールされていない場合には

http://terapix.iap.fr/rubrique.php?id_rubrique=91
からダウンロードしてインストールします。

6.1.5 MCSRED の始め方

`.bashrc`、`.cshrc` などを再読み込みした後、`cl` コマンドで IRAF を起動します。

```
(例えば) source ~/.cshrc
(login.cl のあるディレクトリで)
$ cl
cl>
```

それから mcsred と打ち込むことによって MCSRED パッケージが読み込まれます。

```
cl> mcsred

basename      fringesub      mcs_mksflat    moscorcalc     randaperr
checkfrsb     gmkgtrimages  mcsall         nodata         sbselfsky
checkwdither  gsextcat      mcsdisplay     plsatellite    tiltskycor
chkgmpdata   imcheck       mcsgeocorr     prmask         tsubanomaly
cleanall      imreflection  mcsimstat      psfestimate
cutpr        invmask       mkcorrawdata   qchkfrsb
dmosimg      kage          mkdistmask     qmsepskysb
findms       listprep      mkdome         quadcor

mc>
```

MCSRED がロードされると “mc>” のプロンプトが表示されます (version 2.14 以降では、mcsred> となる)。その上に表示されているのは MCSRED に入っているタスクのリストです。うまくいかない場合には login.cl の設定などをもう 1 度チェックしてみます。

ds9 等の画像表示ツールも立ち上げておきます。

```
(例えば) mc> !ds9 &
```

以上で MCSRED を使うためのおおまかな準備は完了です。

6.2 一括実行タスク: mcsall

mcsall はひととおりの撮像データ処理を自動で一括して行うタスクです。mcsall はいわゆる「ディープ・フィールド」データ用になっており、非常にサイズの大きな天体が写っている場合にはスカイ引きなどの作業でうまくいかない可能性があります。ここでいう「非常に大きな」とは観測時に行うディザの幅に対してのものです。例えば視野の1/4~半分を占めるような天体がある場合には mcsall は適切に動作しません。そのようなサイズの大きな天体を観測する場合には、観測時に GETOBJECT コマンドにおいて SKYNOD オプションを使ったりします。SKYNOD オプションを使用した観測データの処理・解析については別途専用のタスクが用意されています (7.1 節参照)。

また、mcsall では1フレームごとにディザした (ずらして観測した) データになっていることが前提になっています。通常 MOIRCS の撮像観測では1フレームごとにディザして観測していますが、特別な理由でディザせずに同じ場所で連続して何フレームか観測しているような場合には、入力リストを編集して同じ位置で取られたデータが連続しない様にファイルの順番を換える必要があります。

さらに、観測中に雲の通過などで大気の透明度やシーイングサイズが急激に変わってしまった場合は、それらの画像を予め除いておかないと良好な結果が得られない場合があります。特にアーカイブを用いたデータの解析では、注意が必要です。

mcsall では次のような処理を行います (() 内は各工程に対応するタスク名) :

- 1) 天体マスク作成 (mcs_mkflat)
- 2) フラットフィールド (imarith)
- 3) メジアンスカイ引き (sbselfsky)
- 4) スカイの引け残りを低次関数 fitting を使って処理 (qmsepskysb)
- 5) 象限間の溝の補正 (quadcor)
- 6) 歪み補正 (mcsgeocorr)
- 7) 各フレームにおける天体検出 (gsextcat)
- 8) 位置合わせ及び足し合わせ (gmkgtrimages)

この他に入力 FITS ファイルリスト作成用タスク (listprep、6.2.2 節)、2つの検出器のデータをモザイクするタスク (dmosimg、6.3.1 節) などが mcsall とは別にあります。以下では mcsall の使い方を簡単に説明します。mcsall で使われている各タスクの説明については Appendix A を参照してください。

6.2.1 解析実習に用いる観測データ

ここでは2005年12月9日に観測された HDF-N (Hubble Deep Field North) というブランクフィールドの MOIRCS 撮像データの処理を例に、解説を進めます。ブランクフィールドとは特別な天体があるわけではなく、宇宙の一般的な性質を表していると考えられる視野のことです。今回の場合、遠方の (暗い) 銀河を調べるために極端に明るい天体がない領域として選択された視野になっていて、mcsall を使って処理を行うのに適したデータとなっています。このデータはハワイ大学の観測者によって取得されたものですが、すばる望遠鏡の観測データは通常観測から1年半後には公開されて、だれでも手に入れて

使うことができます。

データの取得については、講習会参加の方は講師から指示を受けてください。すばる望遠鏡アーカイブ (SMOKA) から 2005 年 12 月 9 日のデータを一括して取得して来る事も可能です。なお、ここでは便宜上、/mb24h/kajiswms/GOODSN/MOIRCS_UH/FITS/の下に生データがあるとします。

```
$ ls /mb24h/kajiswms/GOODSN/MOIRCS_UH/FITS/
```

と ls コマンドを実行すると、MCSA000?????.fits という名前の MOIRCS の生データの FITS ファイルがずらっと出力されます。

このディレクトリには 12 月 9 日に観測されたすべてのデータが置いてあって、HDF-North 以外のいくつかの天体のデータや明るさの基準を決めるための標準星のデータも含まれています。どの天体をどのような設定でどれだけ観測したデータなのかという情報は各 FITS ファイルのヘッダと呼ばれるファイルの頭の方の部分に書き込まれています。IRAF にはこれらの情報を取り出すためのいくつかのタスクが用意されています。例えば、

```
mc> imheader /mb24h/kajiswms/GOODSN/MOIRCS_UH/FITS/MCSA00014027.fits
l+
```

と入力すると、指定した FITS データのヘッダ部分がまとめて出力されます。また、hselect タスクを用いることで知りたい情報だけをまとめて取り出すこともできます。

```
mc> hselect /mb24h/kajiswms/GOODSN/MOIRCS_UH/FITS/*.fits FRAMEID,
OBJECT,FILTER01,EXPTIME yes
```

```
MCSA00013675    DARK    J        13.000
MCSA00013676    DARK    K_CONT   13.000
MCSA00013685    DARK    J        13.000
MCSA00013686    DARK    K_CONT   13.000
MCSA00013687    DARK    J        13.000
MCSA00013688    DARK    K_CONT   13.000
MCSA00013689    DARK    J        13.000
MCSA00013690    DARK    K_CONT   13.000
.....
```

hselect は最初の引数の “/mb24h/kajiswms/GOODSN/MOIRCS_UH/FITS/*.fits” で指定された FITS ファイルのヘッダから、次の “FRAMEID,OBJECT,FILTER01,EXPTIME” で指定されたキーワードの情報を読み取り、出力します。FRAMEID がフレーム番号のことで一番左に出力されていて、以下 OBJECT, FILTER01, EXPTIME がそれぞれ観測した天体、観測に用いたフィルター、積分時間を表しています。

3 節で述べたように、奇数のフレーム番号が channel-1 の検出器のデータ、偶数番号が channel-2 の検出器のデータになっています。上の例ですと、channel-1 では J バンドフィル

ターを、channel-2の方ではK_CONTフィルターを使ってダークを13秒間の積分でとったデータであることがわかります（ダークフレームの観測ではどのフィルターで観測しても関係ないのでこのようになっていますが、通常为天体の観測ではchannel-1とchannel-2で常に同じフィルターを使います）。

このような情報から使いたいデータを選びだすことができます。情報を見たい度にこのタスクを実行するのは面倒なので、この結果をファイルに保存しておきます。これから自分が作業するのに使うディレクトリに移動し、

```
mc> hselect /mb24h/kajiswms/GOODSN/MOIRCS_UH/FITS/*.fits FRAMEID,  
OBJECT,FILTER01,EXPTIME yes > header.txt
```

と打つと、今度は結果がheader.txtというファイルに書き込まれます。このファイルを

```
mc> less header.txt
```

などとやって中身を見てみると、先ほど画面上に出力されていた結果が入っているはずで
す。ずっと下の方まで見ていくと

MCSA00014016	LOCKMANHOLE	KS	100.000
MCSA00014017	HDFN1	KS	100.000
MCSA00014018	HDFN1	KS	100.000
MCSA00014019	HDFN1	KS	100.000
MCSA00014020	HDFN1	KS	100.000
MCSA00014021	HDFN1	KS	100.000
MCSA00014022	HDFN1	KS	100.000
MCSA00014023	HDFN1	KS	100.000
MCSA00014024	HDFN1	KS	100.000
MCSA00014025	HDFN1	KS	100.000
MCSA00014026	HDFN1	KS	100.000
MCSA00014027	HDFN1	KS	100.000
MCSA00014028	HDFN1	KS	100.000
MCSA00014029	HDFN1	KS	100.000
MCSA00014030	HDFN1	KS	100.000
MCSA00014031	HDFN1	KS	100.000
MCSA00014032	HDFN1	KS	100.000
MCSA00014033	HDFN1	KS	100.000
MCSA00014034	HDFN1	KS	100.000
MCSA00014035	HDFN2	KS	100.000
MCSA00014036	HDFN2	KS	100.000
MCSA00014037	HDFN2	KS	100.000
MCSA00014038	HDFN2	KS	100.000
.....			

という部分がでてきます。この HDFN1、HDFN2 の部分が Hubble Deep Field North (HDF-N) のデータであることを表しています。数字の 1 と 2 は二通りの視野をとったのでその視野ごとに数字が振ってあります。フレーム ID 番号は観測された順番（つまり時系列）になっていますので、

HDFN1 を 100 秒積分で 9 回観測 → HDFN2 の視野に移って同様に 9 回観測 → HDFN1 に戻って 9 回観測 → また HDFN2 へ → またまた HDFN1 を観測

の順番で、それぞれ HDFN1 が 9×3 回、HDFN2 が 9×2 回観測されていることを確認してみてください。データがすべて K バンド（正確には K_s バンド）で観測されたものであることも確かめられると思います。ここでは、これらのデータの処理を行ってみます。

6.2.2 入力リストの準備 – listprep

mcsall は 2 つの検出器のデータをそれぞれ独立に処理します。それに伴って入力する FITS ファイルリストもそれぞれの channel に対応したリストになります。この入力ファイルリストをタスク listprep を使って作ります。

listprep はこれから処理しようとしている FITS ファイルが置かれているディレクトリ、作成されるリストのルート名、一括処理したいデータのうちの最初の ID ナンバー、同じく最後の ID ナンバーをそれぞれ指定して実行すると、channel-1 と channel-2 それぞれに対応する FITS ファイルリストを作成します。例えば、前節で見たデータのうち HDFN1 の視野の最初の 9 枚分を一括処理する場合、

```
mc> listprep /mb24h/kajiswms/GOODSN/MOIRCS_UH/FITS hdfn1 14017 14034
```

と打ち込むと、

```
hdfn1_1.lst — channel-1 の FITS ファイルリスト  
hdfn1_2.lst — channel-2 の FITS ファイルリスト
```

という 2 つのファイルが出来上がります。hdfn1_1.lst には奇数番号の FITS ファイル、hdfn1_2.lst には偶数番号の FITS ファイル名が書き込まれています。標準出力にターゲット名、露出、フィルタ等の情報が表示されますので、間違えたファイルが混じりこんでいないかをここで確認しておきましょう。

6.2.3 画像のチェック

mcsall は入力パラメータとして大雑把なシーイングサイズ（点源が大気ゆらぎでぼけるサイズ）を入力する必要があります。また、先ほど述べた様に、mcsall で処理をするデータに悪天候などで質の悪いデータがあった場合、取り除いておく必要もあります。

まず、実行する前に処理しようとしている観測データの点源サイズを評価します。大気のゆらぎ等で点源が広がった形態を記述する関数を Point Spread Function (PSF) と言いま

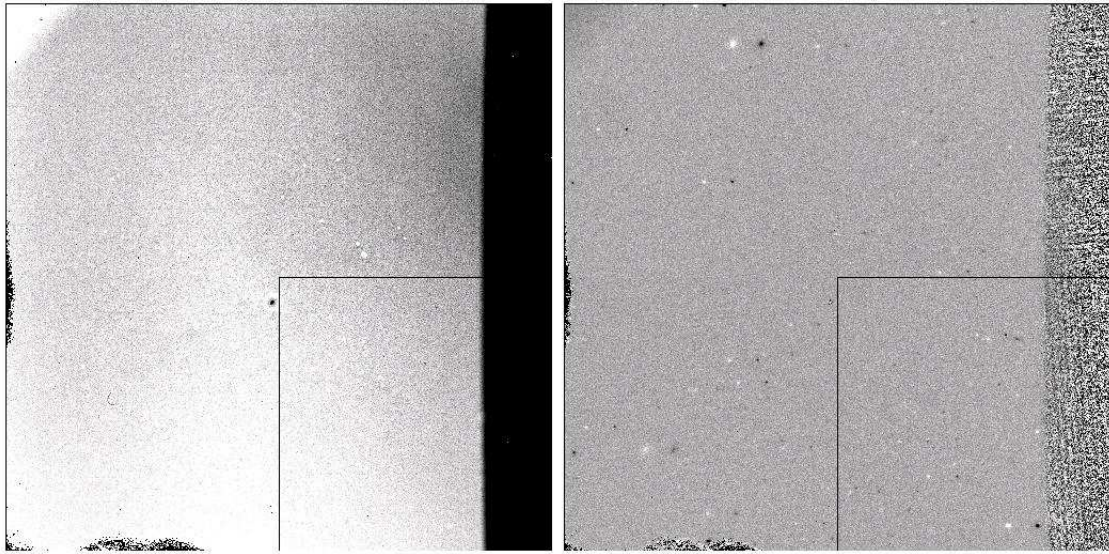


図 5: 生データ (左、MCSA00014017.fits) と生データを直後に取られた生データで割った画像 (右、MCSA00014017.fits/MCSA00014019.fits)

すが、一般には Gaussian 関数と Moffat 関数と呼ばれる形式が使われます。PSF のサイズの指標としてよく用いられるのは単純に FWHM(full-width at half-maximum: 半値全幅) で、IRAF のコマンドで簡単に調べられます。観測現場では、「PSF を見る」と言うと、ほぼ「星の FWHM を計る」と同じ意味で使われます。

では、PSF を大雑把でいいので調べてみます。観測に参加している場合には、大抵 PSF をチェックしながら観測を進めたり、その PSF の情報を観測ログにメモしたりしているので、おおよその値はすでに見当がついている場合が多いと思います。今回のようにアーカイブから取得したような場合には、まずはデータの様子を調べてみる必要があります。

2 節で述べたように生データそのままでは空に埋もれて天体の様子がはっきりわからないので (図 5)、ここでは手っ取り早くデータの様子を見るために、ある生データをそのすぐ後にとられた生データで割ってみます。

まずは生データを見てみます。例えば

```
mc> display MCSA00014017.fits 1 zr- zs- z1=7000 z2=18000
```

次に生データをすぐ後のデータで割り算 (あるいは引き算) をして、様子を見てみます。

```
mc> imarith MCSA00014017.fits / MCSA00014019.fits ql14017.fits
mc> display ql14017.fits 1 zr- zs- z1=0.95 z2=1.05
```

天体が見えるようになったら、サイズを測ってみます。

```
mc> imexamine
```

と入力すると、ds9 などの画像表示ソフト上にカーソルが出ますので、カーソルを天体の上

まで持っていき'a'を入力します。

```
mc> imexamine
# COL LINE COORDINATES
# R MAG FLUX SKY PEAK E PA BETA ENCLOSED MOFFAT DIRECT
1546.00 808.41 1546.00 808.41
17.57 31.75 2.702 1.019 0.07123 INDEF 57 3.09 6.27 5.69 5.92
```

ここで、ENCLOSED, MOFFAT, DIRECT といったあたりが天体の FWHM の推定値を示しています (3つは推定法が違う: 単位はピクセル)。見えているいくつかの天体に対して試してみます。

```
mc> imexamine
# COL LINE COORDINATES
# R MAG FLUX SKY PEAK E PA BETA ENCLOSED MOFFAT DIRECT
1546.00 808.41 1546.00 808.41
17.57 31.75 2.702 1.019 0.07123 INDEF 57 3.09 6.27 5.69 5.92
1364.85 280.42 1364.85 280.42
20.71 31.40 3.714 1.018 0.046 0.41 -4 1.44 7.89 7.09 6.95
1771.63 446.80 1771.63 446.80
18.78 30.61 7.669 1.017 0.1179 0.22 88 4.10 6.49 6.33 6.25
524.76 1381.21 524.76 1381.21
21.89 30.88 6.022 1.018 0.06351 0.21 28 2.32 7.32 7.84 7.30
126.43 1578.40 126.43 1578.40
18.10 30.80 6.451 1.017 0.1055 0.15 -22 4.23 6.28 6.47 6.06
631.54 1897.00 631.54 1897.00
28.80 29.17 29.04 1.018 0.1134 0.06 70 3.00 13.33 9.61 9.61
```

こうして測った天体のうち、小さなサイズを持つものが星又は星に近い天体と考えられます。例えばこの例では 図6 で黄色の丸で囲まれた2つ天体は星です。このようにサイズの小さい天体の FWHM のおおよその値を PSF の目安として覚えておきます。

imexamine は画像表示ソフト上で'q'を入力すると終了します。

6.2.4 画像チェックタスク – imcheck

処理しようとする画像の各々について、前節の様な処理をしてシーイングサイズを調べたり、天体の写り方がおかしいデータがないかを見るのは、良好な解析結果を得るためには不可欠なチェックです。MCSREDには、ここで紹介した様な画像の相互差し引きと imexamine の工程を自動で行うタスク、imcheck が実装されています。このタスクは listprep の結果を入力とし、リスト内の隣り合う画像同士を引き算し imexam を開始します。imexamine を q で抜けると、次の隣接画像セットで同じことをします。

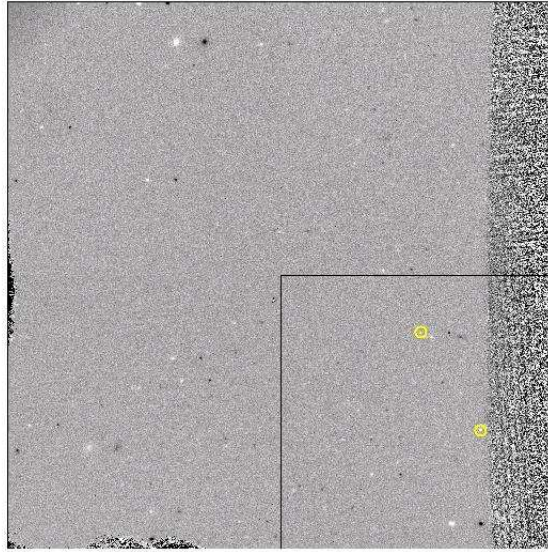


図 6: 生データを直後に取られた生データで割った画像。2つの星が黄色の○で囲まれている。

```
mc> imcheck hdfn1_1.lst
```

6.2.5 有効カウント、COADD

次にサイエンスデータとして有効なカウントについてです。MOIRCS の場合、検出器が飽和するいわゆるサチュレーションレベルは、光の入射強度や検出器上の位置で変わってしまい、あまり有効な指標にはなりません。ただ目安としては、COADD=1 の通常のデータの場合、およそ 22000 ADU (ADU はピクセルのカウントの事) (channel-1 の場合、channel-2 だと 25000ADU 程度) 以下にデータの値があれば安全でしょう。COADD=3 の場合は $22000 \times 3 = 66000$ ADU が目安になります。

COADD とは、全く同じ場所である程度の積分時間をとりたいにもかかわらず、近赤外観測のように明るい背景光によって 1 回の積分時間が短く制限されてしまう場合に、各ピクセルから読み出したカウント値を FITS ファイルに書き出さずにメモリに蓄えておき、また次の積分で読み出されたカウント値をメモリ上で足しあげておいて、複数回積分した後にその合計のカウント値を FITS ファイルに書き出すという手法のことで、各露出毎に FITS を生成する必要が無いので、観測時間の無駄 (オーバーヘッド) を縮める効果があります。

COADD の回数も各 FITS ファイルのヘッダに情報として与えられていて、

```
mc> hselect @hdfn1_1.lst FRAMEID,COADD,EXP1TIME,EXPTIME yes
```

MCSA00014017	2	50.000	100.000
MCSA00014019	2	50.000	100.000
MCSA00014021	2	50.000	100.000
MCSA00014023	2	50.000	100.000
MCSA00014025	2	50.000	100.000
MCSA00014027	2	50.000	100.000

MCSA00014029	2	50.000	100.000
MCSA00014031	2	50.000	100.000
MCSA00014033	2	50.000	100.000

のように見ることができます。この場合 COADD=2 で、前節で見た積分時間 100 秒というのは実は 1 回 50 秒の積分を 2 回 COADD した結果だったことがわかります (EXP1TIME というキーワードが 1 回の積分時間を示していて、EXPTIME は合計の積分時間 COADD × EXP1TIME の値になっています)。

6.2.6 タスクの実行

前節までの画像チェックで大きな問題がない事が分かったら、いよいよ mcsall の実行に移ります。

mcsall では 2 つの検出器のデータを独立に処理します。このタスクは 6.2 節で紹介した一次処理プロセスを一括で処理するために作られた、一種のパイプラインですので、非常に多くのパラメータを持っています。

ちなみに、2 つの検出器のデータをモザイクするためには、mcsall で歪み補正 (6.2 節における工程 6) まで処理したデータを使ってタスク dmosimg (次節で説明) を実行することになります。

mcsall を実行する前にまずパラメータのチェックを行います。

```
mc> lpar mcsall
  inlist = "hdfn1_1.lst"   The list of raw images (listprep output)
  resimg = "hdfn1_no1_ch1.fits" The name of the final resulting image
  (jump = 1)               Jump to the i-th procss
  (bye = 8)                Stop after the i-th procss
  (disp = yes)            Display the process?
  (config = "dir_mcsred$DATABASE/ana_dec05.cfg") the name of mcsred config
  (dosf = yes)            Step1: Use a self-flat instead of external flat
  (sflat = "")           Step1: the name of output sky flatframe.
  (extflat = "")         Step1: if dosf-, supply the name of flatframe.
  (dodk = no)            Step1: subtract the dark frame ?
  (dark = "")           Step1: the name of the user-supplied Dark Frame
  (rail = no)           Step1: do the rail remain on chip 1?
  (sthresh = 1.25)      Step1: threshold for object mask.
  (msksize = 4.)        Step1: object expansion size (pix).
  (mskcr = yes)         Step1: cosmic-ray cleaning during making mask?
  (nself = 3)           Step3: 2 x nself frames are used for making sky
  (skipqms = no)        Step4: skip qmsepsky process (cautious!)?
  (moresky = no)        Step4: more constant sky subtraction after qmse
  (crrej = yes)         Step5: cosmic-ray cleaning during mcsgeocorr?
  (minfw = 5)           Step7: Minimum FWHM for matching catalog
  (maxfw = 15)          Step7: Maximun FWHM for matching catalog
```

(satur = 22000)	Step7: Saturation counts
(thres = 5)	Step7: SExtractor: detect_thresh
(conn = 16)	Step7: SExtractor: connected pixel
(combine = "average")	Step8: type of final combine?
(reject = "sigclip")	Step8: type of rejection
(weight = "sigma")	Step8: weight -sigma or exptime?
(lsigma = 3)	Step8: lower sigma clipping factor
(hsigma = 3)	Step8: upper sigma clipping factor
(nmat = 40)	Step8: nmatch parameter in xyxymatch
(mode = "q")	

各パラメータの説明：

inlist — listprep で作成された入力リストファイル。

resimg — 最終合成画像名 (出力)。必ず ".fits" をつけてください。

jump, bye — もし mcsall が 6.2 節で述べた工程のどこか途中で終了してしまった場合、jump パラメータで指定する番号の工程から再スタートさせることができます。逆に mcsall を途中で止めたい場合は bye パラメータで指定する番号の工程で終了させることができます。

disp — 自動で処理の途中経過の画像表示を行うかどうかを指定します。常に途中経過を見ていて、異常が起きていない事を確認する事が推奨されますが、リモート解析などで画像表示をさせるのが困難な場合はこれを no にします。mcsall の実行前に ds9 などの画像表示ツールが立ち上がっていることを確認してください。

config — 処理しようとしている観測データに対応するコンフィギュレーションファイルを指定します。MOIRCS の昇温-冷却が行われる度に歪み補正に関連するデータベースは変更されます。観測データの歪み補正及びその後の位置合わせを正しく行うためにはそのデータに合ったコンフィギュレーションファイルを指定する必要があります。以下にコンフィギュレーションファイル名と対応する観測時期を示します。

ana_jan05.cfg	2005 年 1 月 (機能試験観測)
ana_dec05.cfg	2005 年 12 月 (機能試験観測)
ana_jan06.cfg	2006 年 1 月-3 月
ana_mar06.cfg	2006 年 4 月-7 月
ana_aug06.cfg	2006 年 8 月-10 月
ana_nov06.cfg	2006 年 11 月 - 2007 年 1 月
ana_feb07.cfg	2007 年 2 月-3 月
ana_may07.cfg	2007 年 4 月-6 月
ana_aug07.cfg	2007 年 7 月-10 月初
ana_oct07.cfg	2007 年 10 月末-2008 年 2 月

ana_feb08.cfg

2008年2月-7月

ana_aug08.cfg

2008年8月-2009年7月

例えば今回の講習で使うデータは2005年12月9日の観測データですので、ana_dec05.cfgを入力することになります。

なお、最新の情報はMCSREDパッケージ内のREADMEかmcsgeocor.clのヘッダに情報があります。

dosf, sflat, extflat — dosfはセルフフラットを使用するかどうかを指定するパラメータです。セルフフラットを使う場合にはこれをyesにし、sflatパラメータに作成されるセルフフラットフレームのファイル名を入力します。セルフフラットを作成せず、すでに手元にある別のフラット(ドームフラット等)を代わりに使用する場合は、dosf=noにし、extflatに使用するフラットフレーム名を入力します。

ドームフラットの方はセルフフラットに比べて天体やフリンジの影響(例えば7.2節、7.3節参照)を受けない利点がありますが、全体的な感度の傾きがドームフラットでは天体を観測した時と多少異なるという欠点もあります。また、ドームフラットを使う場合は同じ観測ランで取得されたドームフラットを用います。これは、MOIRCSの昇温-冷却が行われる度に2-3%レベルで各ピクセルの感度の変動が見られるためです。

今回の講習ではセルフフラットを使った処理を行うので、dosfは”yes”にし、sflatにフラット名を入れます。既に存在するファイル名を入れない様に注意してください。

dodk, dark — ダーク引きを行うかどうか、また行う場合に用いるダークフレームを指定するパラメータです。**MOIRCSデータでは通常ダーク引きは行っていません。**これはダーク自体のカウントレベルが比較的小さいことに加えて、実際に天体やドームフラットの観測の場合には、ラテント画像又は残光と呼んでいる検出器への光の入射強度に依存した(今のところ正確に測定できていない)成分がデータに加わっていて、これがダークに比べてかなり大きいことが理由です。このラテント成分がのっていることによってフラットフレームの作成に系統的な誤差(おおよそ2-3%から最大8%程度と推定されています)が生じます。

rail — MOS観測の後の撮像などで、しばしばchannel-1画像の左端100pixel程がブランクになっているケースがあります。これはMOSマスクの安全対策レールが取り残されてしまったもので、これが見られた場合、rail=yesにして実行してください。

sthresh, msksize — 工程1で天体マスクを作る際、スカイの何シグマ以上を天体とみなすか(sthresh)、その天体とみなされたシグナルの周囲を何ピクセル増やすか(msksize)に関するパラメータです。通常デフォルトでOKです。

msker — 工程1で天体マスクを作る際、宇宙線などの除去をするか(IRAF cosmicraysタスクを利用します)を決めます。狭帯域フィルタなどの長時間データではこれをyesにします。

nself — スカイ引きの工程で用いるメジアンスカイを何枚のフレームから作成するか

を指定するパラメータです。ある 1 枚の天体フレームに対するメジアンスカイはその天体フレームの直前の `nself` 枚及び直後の `nself` 枚、合わせて $2 \times nself$ 枚から作られることとなります。通常 `nself=3-4` が使われることが多いです。個々のフレームの積分時間や観測時のスカイコンディションに応じて程よい値を選びます (A.2 節参照)。

`skipqms` — スカイの引け残りをフィット面で引く工程をするかしないかを決めます。通常 No。

`moresky` — スカイの引け残りをフィット面で引く工程の後、その画像のメジアンを正確に求めて引き去るという処理をするかどうかを決めます。通常フィット引きで十分な精度になるので、スキップしても問題ありません。

`crrej` — 光学歪み補正をする前に宇宙線などの処理をするかどうかを決めます。光学歪み補正は画素間の情報を補間するので、宇宙線やホットピクセルなどの影響がそのピクセルの周囲に広がってしまいます。歪み補正をする前に処理をします。通常は Yes。

`minfw, maxfw` — 画像の位置合わせの工程で使う天体の大きさ (FWHM の最小値と最大値) の範囲を指定します。`minfw` には処理しようとしているデータの最小 PSF より少し小さいくらいの値を入れておきます。これによって各画像の星よりもずっと小さいバッドピクセルや宇宙線が位置合わせの工程に悪影響を及ぼすことを防ぐことができます。`maxfw` には観測データの最大 PSF の 2 倍の値、 $2-3 \times \text{FWHM}$ 程度を入力しておくといいでしよう。あまりにサイズの大きな天体では位置を決める精度が落ちてしまう危険性があります。

`satur, thresh, conn` — 位置合わせの工程で使うための天体を検出する際に用いる SExtractor のパラメータを指定します。それぞれ SExtractor での `SATUR_LEVEL`, `DETECT_THRESH`, `DETECT_MINAREA`, の各パラメータに対応しており、天体にフラグを立てるサチュレーションレベル (サチってしまっている天体では位置が正確に求まらない危険性があるので、これを除くために使います) や、どれくらい有意にスカイより高いカウント (`thresh`) のピクセルがどの程度つながっていたら天体と見なすか (`conn`) を指定します。`conn` パラメータは FWHM 値の 2-3 倍が目安です。また、`thresh` パラメータは通常 SExtractor の `DETECT_THRESH` に使われる値よりも大きめに設定します (A.6 節参照)。ただし、あまり検出天体が少ないと、カタログマッチング (検出天体カタログを比較し同じ天体を同定すること : 天体の位置が合う様に画像をシフトさせる時使います) で失敗するため、2-5 の範囲で試してみてください。サチュレーションレベルの目安は 6.2.5 節で述べた安全なデータカウント範囲の 22000-25000 ADU を入れておく事になります。通常はデフォルトからいじる必要はありません。なお、COADD されたデータについても、自動的に補正を入れるので COADD 分の補正について心配する必要はありません。

`combine, reject, weight, lsigma, hsigma` — 整約された個々の画像を 1 枚の最終画像にするために使う合成アルゴリズムパラメータです。画像が 5-6 枚以下と少ない場合は `combine=median` にする必要がありますが、それ以外は `combine=average` に適切な rejection (`pclip`, `sigclip`, `avsigclip` 等) をかけた方が SN が良い画像になります。`weight` は画像の rms を評価してウェイトとするか、ヘッダの露出時間でウェイトにするか、という選択に

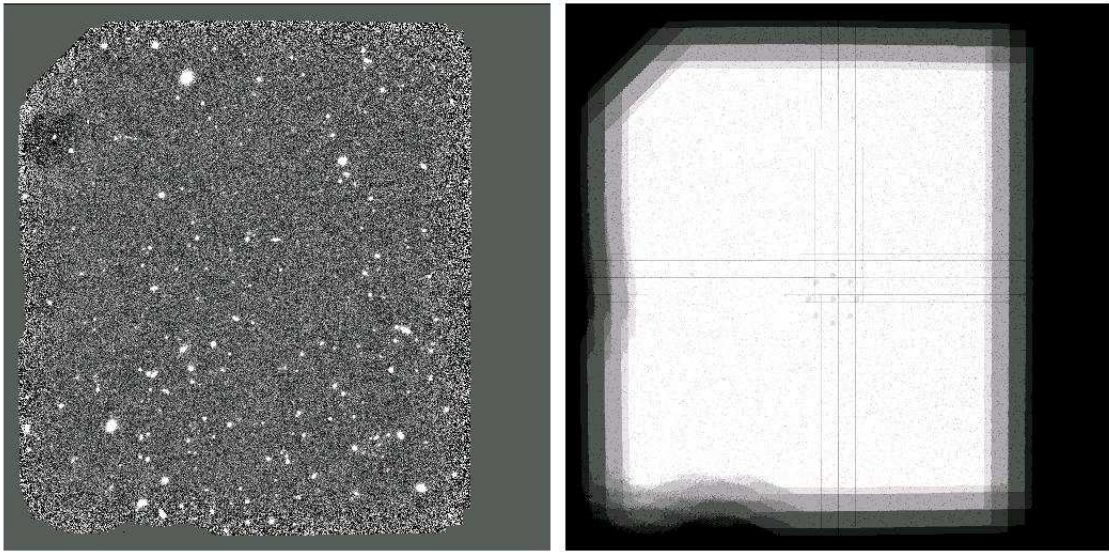


図 7: mcsall で足し合わせた画像 (左) と exposure map (右)

なります。赤外はスカイが大きく変動しそれに応じて画像の rms も変動するので、ここでは rms ウェイトを使っています。

nmat – 個々の画像を合成する際、天体を検出し、カタログマッチ (IRAF xyxymatch) を行う事で個々の画像の相対的なシフト量を評価します。この時に使われる天体数の最大値です。通常はいじる必要はありませんが、計算時間が膨大になるので、50 以上にはいけません。

準備ができれば、いよいよランします。例えばシーイングが 4.5-6.0 ピクセル程度の観測データの場合、

```
mc> mcsall hdfn1_1.lst hdfn1_no1_ch1.fits dosf+ sflat=Selflat_1_ch1.fits minfw=4.
maxfw=15 thres=4 conn=10 config=dir_mcsred$DATABASE/ana_dec05.cfg
```

のように実行します。デフォルトでよいパラメータは省略可能です。

タスクの所用時間はコンピューターの性能に依存します。一例を挙げると、Intel Xeon 3.4 GHz, 4GByte memory の linux PC を用いた場合で、16 枚のフレームのデータの処理に約 25 分かかります。

mcsall が終わると図 7 のような足し合わせ画像が出力されます。足し合わせた画像とともにログファイルや exposure map なども出力されます (6.3.3 節参照)。また、mcsall の各段階でどのようなファイルが出力されるかについては Appendix B を参照してください。

上の例では channel-1 (hdfn1_1.lst) のデータの足し合わせでしたが、足し合わせができれば channel-2 の方のデータに対しても同様に mcsall を実行します。

6.3 画像のモザイク

channel-1 と channel-2 のデータ画像のモザイクを行うためにタスク `dmosimg` が用意されています。`dmosimg` は歪み補正済みの2つの検出器のデータを入力すると、入力した2枚の画像をモザイクした画像を出力します。

作業手順としては、

- 1) `mcsall` で歪み補正 (6.2 節で述べた工程 6) まで処理したデータ (ファイル名は `gcSBsbfMCSA000?????.fits`) のリストを `dmosimg` に入力。
- 2) `dmosimg` はそれぞれの組 (channel-1 と channel-2) をモザイクした画像を出力。
- 3) `gsxcat` を用いて各モザイク画像において位置合わせのための天体検出 (6.2 節の工程 7)
- 4) `gmkgrimages` を用いて各モザイク画像の位置合わせ及び足し合わせ (6.2 節の工程 8)

と結局、6.2 節で述べた `mcsall` の工程の 6) と 7) の間に `dmosimg` の処理を加えた形になります。

画像のモザイクを行う際にはモザイクする2つの画像のゼロ点 (1ADU/sec あたりの明るさ) を合わせるために、前もって2つの検出器の感度の比を求めておく必要があります。`dmosimg` ではこの2つの検出器の感度比を表すパラメータ `sc` が次のように定義されています。

$$\text{channel-1 の感度} = \text{sc} \times \text{channel-2 の感度}$$

6.3.1 モザイクタスク `dmosimg`

`dmosimg` は `mcsall` でフラットフィールド、スカイ引き及び歪み補正の処理をされたデータのリストを入力すると、各フレームの channel-1 と channel-2 をモザイクした画像を出力します。

最初に次の例のようにして入力ファイルリストを用意します。入力リストは `mcsall` で出力される画像のうち `gcSBsbfMCSA000?????.fits` を順番に並べたものです (`mcsall` で出力されるファイルについては Appendix B 参照)。

```
mc> files gc*fits > gcall.lst
mc> type gcall.lst
gcSBsbfMCSA00014017.fits
gcSBsbfMCSA00014018.fits
gcSBsbfMCSA00014019.fits
gcSBsbfMCSA00014020.fits
....
```

dmosimg の入力リストには同時に観測された channel-1 と channel-2 のデータがセットで存在していることが前提となっています。また、リストは奇数番号のファイル (channel-1 データ) で始まり、偶数番号のファイル (channel-2 データ) で終わっている必要があります。

次にパラメータのチェックです。

```
mc> lpar dmosimg
inlist = ""                input distortion-corrected file list
froot =                    the root name for mosaicked individual images
(sc = 0.8)                Sensitivity scale factor of chip 2.
(config = "dir_mcsred$DATABASE/ana_feb07.cfg") the name of mcsred config
(sky = yes)               subtract sky?
(corx = 0.)               additional dx for ch2 to default mosaic rule
(cory = 0.)               additional dy for ch2 to default mosaic rule
(displ = yes)              Display the result?
(mode = "q")
```

inlist — 上で作成した入力ファイルリストを指定します。

froot — 出力されるモザイク画像のルート名を指定します。例えば、froot=tmos と指定すると、出力されるファイル名は次のようになります。

tmos_000?.fits (?=1,2,3,...)

mos_gcall.lst ... 出力画像のリスト。(mos_入力リスト名)

sc — 前節で述べた channel 間の感度比を表すパラメータです。sc は基本的には天体データとともに取得した標準星データから求めたそれぞれの channel でのゼロ点の差から求めます (標準星データの処理については 6.4 節参照)。より簡単には、観測された画像の平均スカイの値を各チャンネルについて求め、その比を計算することで推定できます (精度はスカイのラテント成分の量で制限されます)。

$$sc = 10^{[ZP(ch1) - ZP(ch2)]/2.5}$$

ここで ZP(ch1) と ZP(ch2) はそれぞれ channel-1 と channel-2 のゼロ点です。ディザ幅をある程度以上の大きさで観測した場合には、mcsall でそれぞれの channel で独立に足し合わせた後に channel-1 の足し合わせ画像の右端と channel-2 の足し合わせ画像の左端にある程度の幅の共通部分ができるので、この境界近くの領域に比較的明るい天体が両方の画像に写っている場合にはそれらの天体のカウントを測って感度比を決めることができることもあります。

現在使われている各広帯域フィルターでの sc の目安を以下に紹介しておきます (MCSRED のタスクで作成したスカイフラットやドームフラットを使用した場合の値です)。

20060601 - Ks+CSL: sc=0.813 (under the use of the mcsred flatfield data)



図 8: dmosimg でモザイクした画像

```
# 20061210 - J+CSL:   sc=0.8913 (under the use of the mcsred flatfield data)
# 20061210 - H+CSL:   sc=0.9120 (under the use of the mcsred flatfield data)
```

config — mcsall の時と同様に観測データに合ったコンフィギュレーションファイルを指定します。

sky — スカイ引きを行うかどうかを指定します。” sky=yes” にした場合、モザイクを行う前に各 channel の真ん中付近の領域 ([700:1300,700:1300]) でメジアン値を求め、その値をスカイとして画像全体から引き去ります。

corx, cory — 取得された時の高度が低い場合など、重力たわみによってデフォルトのモザイクルールから実際のモザイクルールがずれてくる場合があります。このずれ量が分かっている場合、corx, cory パラメータにそれを入れて補正します。Channel-1 と 2 の境界付近にある天体がそれぞれのチャンネル間をディザで行き来するケースがあれば、これを使って補正量を評価する事ができます。MCSRED にはこの値を評価するためのタスク、findms と moscorcalc が実装されていますが、一般にはこの補正は微小で無視できるので、解説は割愛します。

準備ができれば実行します。例えば、

```
mc> dmosimg gcall.lst tmos sc=0.813 config=dir_mcsred$DATABASE/ana_dec05.cfg
```

図 8 のようなモザイク画像が対応する枚数分出来上がります。デフォルト設定における出力される各モザイク画像のフォーマットは 3569 pix × 2048 pix で、ピクセルスケールは 0.117 arcsec/pix になっています。dmosimg は画像データのカウント値を channel-1 の方に

合わせるので、出来上がったモザイク画像のゼロ点は channel-1 に対して求めた値を使います。

Elevation が 45 度以下のデータでは装置のたわみによって dmosimg におけるデフォルト設定のモザイク規則では合成の精度があまり良くないので注意が必要です。corx, cory パラメータで解決できる場合は良いですが、うまく合成されない場合は mcsall で足し合わせた各 channel のデータに WCS を張って channel 間の相対位置を決定するなどの作業が必要になります。

モザイク合成された各画像が完成したら、次にこれらの合成を行います。そのためには各画像の天体を検出し、その天体と同じ座標に来るように位置合わせをした上で最終合成を行います。天体検出を行うタスクが gsextpat、位置合わせと最終合成を行うタスクが gmkgtrimages です (mcsall は内部でこれらを使用しています)。

6.3.2 天体検出 gsextpat

gsextpat は MOIRCS 画像で天体検出を行うためのタスクです。パラメータは次のようになっています。

```
mc> lpar gsextpat
  infile = "@mos_gcall.lst" Input image name (allow @list forma
  detmin = 12                -DETECT_MINAREA parameter in sextractor (int).
  thresh = 5.                -DETECT_THRESH parameter in sextractor (real).
  minfw = 5.                 Minimum size of objects for output catalog.
  maxfw = 15.                Maximum size of objects for output catalog.
  (disp = yes)               Display and check result by display?
  (satur = 15000.)          -SATUR_LEVEL parameter in sextractor.
  (recent = no)              Execute re-centering for SExtractor output?
  (nlimit = yes)             Limit the number of SExtractor catalog objects
  (auto = yes)               Automatically define the detection region
  (xfr = 0.145)              detection avoiding region factor in x
  (yfr = 0.05)               detection avoiding region factor in y
  (cx0 = 296)                Beginning x coordinages for cut-out region
  (cy0 = 102)                Beginning y coordinages for cut-out region
  (cwx = 1456)                x size of the cut-out region
  (cwy = 1844)                y size of the cut-out region
  (mode = "q")
```

inlist — 天体検出にかける画像ファイル名、又は画像ファイルのリストを入力します (画像のリストを入力する場合はリストファイルの名前の頭に '@' をつけ加えます)。

detmin, thresh — SExtractor の DETECT_MINAREA, DETECT_THRESH の各パラメータを指定します (6.2.4 節の thres, conn の項参照)。

minfw, maxfw — SExtractor によって検出された天体のうち位置合わせに用いる天体のサイズ (FWHM) の範囲を指定します。これらのパラメータで指定される範囲外のサイズの天体はカタログから除かれます (6.2.4 節の minfw, maxfw の項参照)。

satur — SExtractor の SATUR_LEVEL パラメータを指定します (6.2.4 節参照)。デフォルトで大丈夫でしょう。

disp — 天体検出結果を ds9 等でチェックする時 yes にします。通常 yes。

recent — 検出された天体の中心座標を center タスクを用いて再評価するかどうかを指定します。通常不要です。

nlimit — nlimit オプションを使うと明るい方から 69 個までの天体のみのカタログが出力されます。

auto, xfr, yfr — 位置合わせに用いる天体の座標範囲を計算して決定するかどうかを決めます。yes の場合、画像中心から

(入力画像の x 方向サイズ) × xfr、(入力画像の y 方向サイズ) × yfr
の部分について、位置合わせ用天体を検出します。

cx0, cy0, cwx, cwy — 上の検出位置決めパラメータ auto が no の時、この値で決まる領域について、天体検出をします。cx0, cy0 が始点の X 座標、Y 座標を表し、cwx, cwy が用いる領域の X 方向、Y 方向の大きさを示します ([cx0:cx0+cwx-1, cy0:cy0+cwy-1] の領域の天体が位置合わせに使われます)。

cx0, cy0, cwx, cwy パラメータのデフォルト設定は mcsall のように各 channel 独立に処理する場合を想定した値になっているので、モザイク画像の位置合わせを行う場合には、例えば cx0=100, cy0=100, cwx=3369, cwy=1848 のような値を入力する事になりますが、auto=yes で自動決定するのが楽でしょう。

```
mc> gsxtract @mos_gcall.lst 12 5 4 10 auto=yes xfr=0.1 yfr=0.05
```

detmin, thresh などの天体検出の条件を決めるパラメータや用いる天体のサイズ範囲を指定する minfw, maxfw のパラメータは、最終的な (位置合わせに使われる) 天体数が 100 以下になるように設定しておくことをお勧めします (nlimit オプションを "yes" にしておくことで、カタログされる天体数を明るい順に 69 個までに制限することができるので便利です)。天体数が少なすぎたり、又は多すぎたりすると次の位置合わせの工程で失敗することがあります。

6.3.3 位置合わせと足し合わせ gmkgtrimages

gmkgtrimages は各フレームの相対位置を測定して位置合わせをして、最後に足し合わせを行うタスクです。

```

mc> lpar gmkgtrimages
  inlst = "mos_gcall.lst" The list of input images with associated catalo
  output = "hdfn1mos_no1.fits" Name of result image
  (checkbox = no)           Automatically define the final image size?
  (xbox = 300)           Dither box size in x in pixels
  (ybox = 300)           Dither box size in y in pixels
  (xc1 = 150)            Relative x coord of 1st image in dither box
  (yc1 = 150)            Relative y coord of 1st image in dither box
  (allcomb = yes)        Execute the final combine process?
  (rail = no)            Do the rail remain on chip 1?
  (combine = "average")  Type of the final combine operation
  (reject = "avsig")     Type of rejection
  (zero = "none")        Image sky zero level offset
  (weight = "sigma")     Weight: sigma or exptime?
  (fitgeo = "shift")     geomap type: SHOULD BE rotate/shift
  (lsigma = 3.)          Lower sigma clipping factor
  (hsigma = 3.)          Upper sigma clipping factor
  (nmat = 40.)           xyxymatch: nmatch parameter
  (tol = 10)             xyxymatch: lolerance parameter
  (nrej = 5)             xyxymatch: nreject parameter
  (sepa = 10)            xyxymatch: separation parameter
  (rat = 6)              xyxymatch: ratio parameter
  (fstop = no)           process will stop after xyxymatch for check
  (fsresume = no)        Resume process using the result by fstop=yes?
  (skip = no)            Skip all processes except the final combine?
  (disp = yes)           display the final result?
  (gtrlst = "")          List of images for final combine
  (mode = "q")

```

inlst — 入力ファイルリストを指定します。前節の gsexcat と同じファイルを指定します (ただし '@' は必要ありません)。

output — 出力される足し合わせ画像のファイル名を指定します。

checkbox — 出力される足し合わせ画像のサイズフォーマットをタスクに自動決定させるかどうかを指定します。

xbox, ybox — 入力される画像のサイズに対する出力される足し合わせ画像のサイズフォーマットの増分を示します。xbox, ybox に観測時に使ったディザ幅を指定しておくと、どの方向に振ったフレームに対しても端が切れることなくすっぽり収まる足し合わせ画像のサイズを確保できます (ただしディザ幅ピッタリにすると厳密に正しく中心をとらないとすぐにいずれかのフレームの端が切れてしまうので、通常はディザ幅よりさらに余裕をもった値を入力しておきます)。

xc1, yc1 — 入力リストの最初の画像と出力される足し合わせ画像の相対シフト量を指定します。入力リストの最初の画像の (X_PIXEL, Y_PIXEL) の点が足し合わせ画像の (X_PIXEL+xc1, Y_PIXEL+yc1) の座標にくるように座標系がとられます。

allcomb — 足し合わせを行うかどうかを指定します。"allcomb= no" を入力すると、gmkgtrimages は各フレームの相対シフト量を測って位置合わせを行った段階で終了します。通常は yes。

combine, reject, zero, weight, lsigma, hsigma — 足し合わせをする際の imcombine のパラメータを指定します。combine は average, median, sum など足し合わせの方法を示します。reject は sigma clipping などの統計的にはずれたデータを除去するアルゴリズムを指定します。sigma clipping 系の除去アルゴリズムを用いる場合には lsigma, hsigma で sigma clipping factor (何 σ 以上平均値からはずれていたら除去するか) を高低それぞれに対して指定します。zero は定数を加減算することでスカイの高さ (メジアン値) を合わせるかどうかを指定します。weight は重み付き平均で足し合わせる時の重みの付け方を指定します。各フレームの積分時間で重みをつける "exptime" と、各フレームの pixel-to-pixel 統計の分散の逆数で重みをつける "sigma" のオプションがあります。

fitgeo — 位置合わせを行う際にどのような座標変換 (geomap) を行うかを指定します。XY 方向の平行移動で合わせる "shift" と、XY シフトに回転も含めた "rotate" のいずれかを選択します。なお、高度の低い画像を含むデータの場合は、"general" を指定する事でよりマッチ精度が上がる事もあります。

nmat,tol,nrej,sepa,rat — カタログマッチングタスク (IRAF xyxymatch) のパラメータです。通常はデフォルトで大丈夫ですが、マッチに失敗する場合は rat を 6-9 の範囲で振ったり、tol や sepa を適当にいじる事で改善する場合があります。詳細は xyxymatch のヘルプを参照ください。

fstop, sfresume — fstop=yes で実行すると、カタログマッチをして位置補正データベースを作成した段階で停止します。カタログマッチがうまくいかず、パラメータをサーチする必要がある場合、fstop=yes にしながら、上の xyxymatch 関連のパラメータを修正してトライします。うまく行った場合、fstop=no sfresume=yes と指定する事で、そのマッチング結果を用いて実際に位置合わせ済み画像の作成と、最終合成画像の作成に入ります。

skip — "skip=yes" にすると、gmkgtrimages は相対シフト量の測定と位置合わせの処理を飛ばして、最後の足し合わせのみを行います。一度 gmkgtrimages を実行した後で、足し合わせの方法や除去アルゴリズムだけを変更して、再度足し合わせ作業を行いたい場合などに使います。この場合、gtrlst にすでに位置合わせの処理が済んでいる画像のリストを入力する必要があります。また、最終合成画像リストの中で、合成に使用したくない画像があった場合、gtrlst に指定するファイルリストを編集した上で skip=yes で再処理する事でそうする事が可能になります。

一度 gmkgtrimages を実行後に、今度は重み付け平均の代わりにメジアンを使い、minmax の除去アルゴリズムで足し合わせたい場合には、以下の様にします。

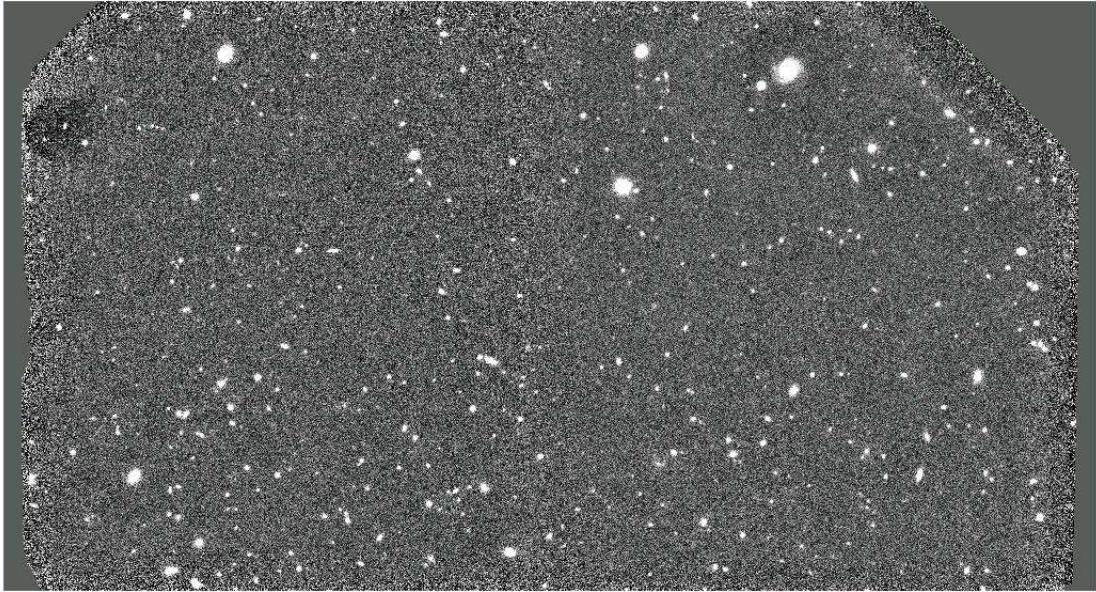


図 9: モザイク画像を gmkgtrimages で足し合わせた画像

```
mc> gmkgtrimages mos_gcall.lst MOSRESULT2.fits comb=median reject=minmax  
gtrlst=GTRmos_gcall.lst skip+
```

gtrlst — 足し合わせの直前の位置合わせを済ませた画像のリストの名前を指定します。通常は “GTR” (inlst で指定したファイル名) という名前のファイルが最初の合成時に作成されますので、それを指定します。

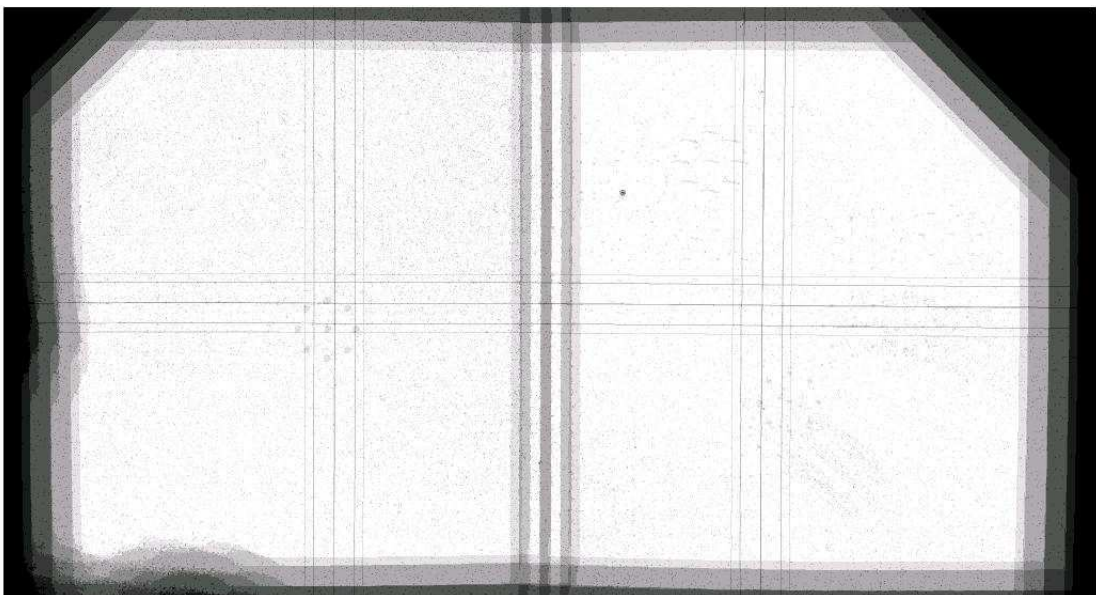


図 10: 足し合わせたモザイク画像に対する exposure map

gmkgtrimages では出力画像（足し合わせた結果及び足し合わせる直前の位置合わせした各フレーム）は1秒積分あたりのカウントにスケールされています（A.7節参照）。

足し合わせが完了すると図9のような画像が出力されます。足し合わせた画像データと位置合わせされた各フレーム（及び各フレームの bad region に対するマスク画像）の他に、以下のファイルが出力されます。

- gmp 入力リスト.log — ログファイル（上の例の場合 gmpmos_gcall.lst）
- exp_出力画像ファイル名.pl — exposure map（1ADU が1秒積分に相当している画像、exp_MOSRESULT.fits.pl）
- sgm_出力画像ファイル名 — sigma image（足し合わせする時の各ピクセルでのフレーム間の標準偏差の画像、sgm_MOSRESULT.fits）
- 出力画像ファイル名.pl — 足し合わせ画像に対する bad region のマスクデータ（MOSRESULT.pl）。マスク領域が全データ数の半数以上を占める（つまり実行積分時間が半分以下）の領域がマスク対象になります。

6.3.4 最終合成データについて

フラット・スカイ引き処理された画像を最終合成する際、合成画像の間で写っている星の明るさやシーイングサイズがずれていないか調べ、合成画像全てでそれらが一致する用にスケールしたり PSF を修正したりしてから、最終的な合成をする、という事がよく行われます。

MCSRED ではこれらを扱っていません。一つには、MOIRCS は視野が広いと言えど、上の事を正確に評価できる程に十分な数の明るい天体が視野に入らない事が多い、という事情があります。高精度な測光を目指すなどの特殊な用途には、それ相応のきめ細かい扱いが必要になります。MCSRED においては、合成に使った全データの平均的な結果が最終画像になっている、と言う点に注意すべきでしょう。

例えば、最終合成画像のフラックス校正の際に注意が必要です。一般的にはメインターゲットの観測の前か後に標準星を観測します。空が安定していれば、標準星で校正した結果をそのまま最終合成画像に適応しても大きな間違いは起こらないと言えます。しかし、もし合成したデータの一部が透明度の悪い状況でのデータだった場合、最終合成画像はそれらの影響を受けたものになりますから、単純なゼロ点の適用は系統誤差を生む要因になります。全データのうち、極力標準星観測に時間的に近い、あるいは安定していたと分かっている時間帯のデータだけを使って校正し直せば、比較的安全な結果になると考えられます。

データを正しく扱うにはどうしたら良いかを、常に考えながら MCSRED を利用してください。

6.4 標準星データ

得られたデータのフラックス較正のために、通常の撮像観測では予め明るさが正確に分かっている星(標準星)をターゲットとセットで観測します。ある明るさの星からの光が、単位時間当たり何カウント来るのかが分かれば、逆にあるカウントの星が何等なのかが分かるわけです。これを測光ゼロ点(1ADU/secあたりの明るさ)と言います。特殊な高精度測光を目的とする場合を除けば、標準星は大抵天体データとなるべく近いコンディション(天体を観測した直前又は直後、天体となるべく近い airmass 等)で観測するのが普通です。

一般に標準星は明るいいため、わざとデフォーカスして星像を拡大して検出器が飽和しない様にして取得したり、後述の「部分読み出し」が使って(2009年以降では稀)短時間露出して取得します。後者の様な取得法ではパイプライン的な扱いはできず、解析補助の目的で MCSRED のタスクを個別に使う事で作業を簡単化する、という形になります(6.4.3節参照)。一方前者の場合は mcsall をそのまま利用できます。

6.4.1 部分読みについて

MOIRCS にはシャッターがないので、可能な最低露出時間は検出器の読み出し時間で決まります。現在使われている「ダミー読み入り全体読み出し」では、21秒積分が最低露出時間となっています。しかし、読み出す領域を小さく制限する事によって、最短露出時間を21秒より小さくする事も可能です。これを「部分読み」と呼びます。典型的には、視野中央の 1024×1024 や 512×512 のみを読み出します。部分読みしたかどうかの情報は FITS ファイルのヘッダの PRD-MIN[12], PRD-RNG[12] というキーワードに記録されています。なお、2008年11月に部分読みモードでの問題が明らかになり、それ以降の観測では基本的に部分読みを避ける様に運用がされています。

6.4.2 標準星データの処理

この解析資料で扱う標準星データは全体読み出しのデータですので、mcsall を使って処理できます(部分読みデータの処理については6.4.3節参照)。

標準星データ解析でまず注意すべき事として、**一般に標準星はデフォーカスされているので、安易に複数フレームを合成してはいけません。**デフォーカス像は位置や条件で形状が変わるため、それを合成するとカウントをロスしたり、意図せず付加してしまったりします。デフォーカスされたデータは必ず1枚1枚測光し、平均を取るなどしてください。

上記の事から、標準星を mcsall で解析する場合、スカイ引き(あるいは歪み補正)まで進んだ段階で処理を止める必要があります。mcsall を途中で止めたい場合には、“bye=6”と指定しておく、最終合成のために天体検出をする工程に入る直前でタスクが終了します(6.2.6節参照)。

もう一つ注意すべき事として、**標準星データの処理で重要な点は天体データを処理した時と全く同じフラットを用いてください。**短時間露出で枚数の少ないデータでは精度のあるフラットフレームが作成できません。

ここでは天体データと同様に2005年12月9日の観測の標準星データ処理の例を紹介합니다。まず、6.2.1節で作ったヘッダリストを見てみます。下の方まで見ていくと、

```
mc> less header.txt
```

```
MCSA00014102   HDFN1   KS      100.000
MCSA00014103   HDFN1   KS      100.000
MCSA00014104   HDFN1   KS      100.000
MCSA00014105   HDFN1   KS      100.000
MCSA00014106   HDFN1   KS      100.000
MCSA00014107   FS33    KS      30.000
MCSA00014108   FS33    KS      30.000
MCSA00014109   FS33    KS      30.000
MCSA00014110   FS33    KS      30.000
MCSA00014111   FS33    KS      30.000
MCSA00014112   FS33    KS      30.000
MCSA00014113   FS33    KS      30.000
MCSA00014114   FS33    KS      30.000
MCSA00014115   TWILIGHT   KS      13.000
MCSA00014116   TWILIGHT   KS      13.000
```

と、HDFN の観測の直後に FS33 という天体が観測されていることが分かります。この FS33 は UKIRT Faint Standards (Leggett e al. 2006, MNRAS, 373, 781 を参照) と呼ばれる近赤外撮像観測でよく使われる標準星リストにある標準星の一つです。このデータを処理してゼロ点を求めます。

MOIRCS は 2 つの検出器があるので、それぞれの検出器に標準星を入れて観測する必要があります。通常まずどちらかの channel に標準星を入れて、画像上のいくつかの点で観測してから、もう片方の channel をまた数回観測するというを行います。ここでは MCSA00014107 から MCSA00014114 まで 4 枚のフレームに渡って標準星 FS33 を観測していて、最初の 2 枚では channel-2 に標準星を入れ (つまり MCSA00014108 と MCSA00014110 に標準星が写っている)、後の 2 枚では channel-1 (MCSA00014111 と MCSA00014113) に標準星を入れて観測しています。

これを mcsall で処理するためにまずリストを作ります。

```
mc> listprep /mb24h/kajiswms/GOODSN/MOIRCS_UH/FITS/ FS33 14107 14114
```

標準星が入っていない方のチップには関係のない (となりの) 空が写っていますが、スカイ引きには有用なため、標準星が写っているかどうかにかかわらず一緒に処理を行います。mcsall に入力する前に一応この標準星データが全体読み出しデータかどうかを確認しておきます (部分読み出しのデータの処理には mcsall は使えません)。

```
mc> hselect @FS33_1.lst FRAMEID,PRD-RNG1,PRD-RNG2 yes
```

```
MCSA00014107   2048   2048
MCSA00014109   2048   2048
MCSA00014111   2048   2048
MCSA00014113   2048   2048
```

PRD-RNG1, PRD-RNG2はそれぞれX、Y軸方向の部分読み出しの範囲を示していて、2048 pixel ということは画像全体を意味しますので、部分読み出しを使っていないことがわかります。

全体読み出しであることを確認したら、mcsall を実行します。

```
mc> mcsall FS33_1.lst FS33_ch1.fits dosf- extflat=Selfflat_1_ch1.fits bye=6
```

フラットは天体データで使ったものと同じものを使います (dosf="no" にして、extflat パラメータに天体データで用いたフラットフレームを入力します)。また、bye=6 と指定して、最終合成直前のフレームが完成した段階でタスクを止めています。なお、本質的なのはスカイ引きまでの工程ですので、工程5の歪み補正は必ずしも必要ではありません。しかし、工程6の象限間溝の補間は、標準星が象限境界のそばに落ちてしまったような場合、ある程度有効でしょう。

同様に、channel-2についても同様に mcsall を実行して、データの一次処理は終了です。

```
mc> mcsall FS33_2.lst FS33_ch2.fits dosf- flat=Selfflat_1_ch2.fits bye=6
```

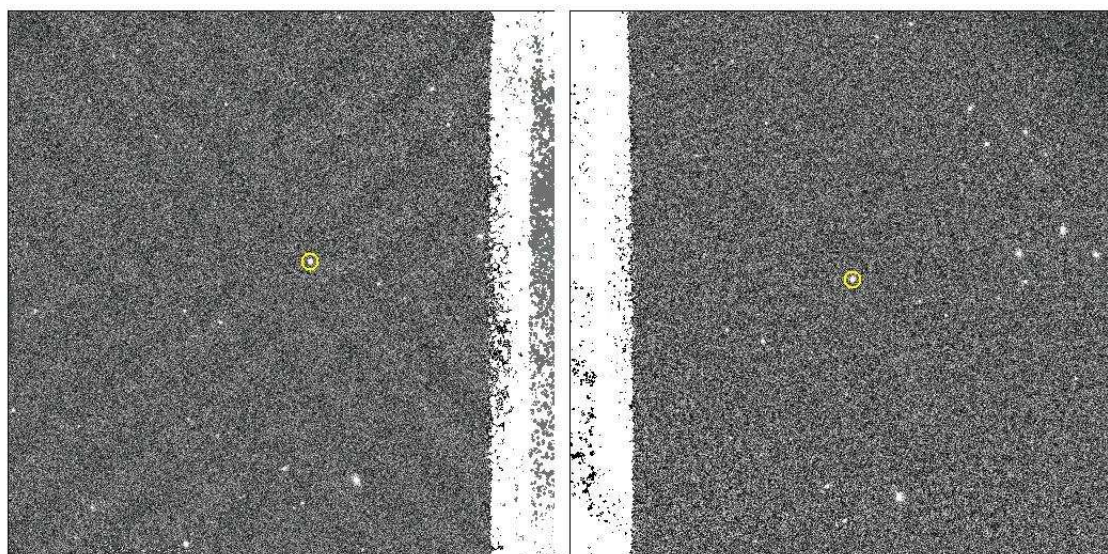


図 11: スカイ引きまで処理した標準星データ (左: channel-1, gcSBSbfMCSA00014111.fits, 右: channel-2, gcSBSbfMCSA00014108.fits)。黄色い○で囲まれている天体が標準星 FS33.

処理がスカイ引きまで進むと図 11 のような画像ができています。

ここでは非常に (説明が) 簡単な方法でゼロ点を求めてみます。まず上に述べた UKIRT Faint Standards の文献を見ると、FS33 の K バンドでの等級は $K = 14.296 \pm 0.011$ であることがわかります。あとは MOIRCS のデータ上でこの星がどれだけの単位時間あたりのカウント値になっているかを測ればゼロ点を求めることができます。

とりあえず channel-1 の方の画像を表示させます。例えば

```
mc> display gcSBsbfiMCSA00014111.fits 1 zr- zs- z1=-100 z2=150
```

ここでは天体のカウント値を測るのにデータのチェックでも使った `imexamine` タスクを使ってみます。IRAF ver.2.12以降の `imexamine` では、天体からの光を誤差を抑えつつ最大限拾うために必要な測光領域(開口)のサイズをタスクが自動的に決めてくれるので便利です。早速やってみます。以下の様に実行します。

```
mc> imexamine
```

すると、`ds9` などの画像表示ツール上に円環状の点滅するカーソルが出ます。FS33 (図11参照) の上にカーソルを持っていき、`'a'` と入力します

```
# COL LINE COORDINATES
# R MAG FLUX SKY PEAK E PA BETA ENCLOSED MOFFAT DIRECT
1132.68 1109.94 1132.68 1109.94
15.71 19.01 335871. 1.91 8265. 0.02 -50 6.88 5.26 5.36 5.24
```

ここで“FLUX”というところの値が天体のカウント値で、“R”というののは測光に使われた開口の半径です。上の `header.txt` で見たように、このデータが 30 秒積分であることを考慮すると、1 秒あたりのカウントは、 $335871/30 = 11195.7$ ADU/sec となります。このカウント値が $K = 14.296$ の明るさに相当しているので、ゼロ点は

$$ZP(\text{ch1}) = 14.296 + 2.5 \times \log_{10}(11195.7) = 24.42$$

と計算されます。channel-2 の方でも同様に `imexamine` でカウント値を測ってみると、例えば `gcSBsbfiMCSA00014108.fits` では、

```
# COL LINE COORDINATES
# R MAG FLUX SKY PEAK E PA BETA ENCLOSED MOFFAT DIRECT
1058.47 1042.94 1058.47 1042.94
16.07 18.79 410440. 2.842 9299. 0.07 67 3.49 5.38 5.48 5.36
```

となっていて、

$$ZP(\text{ch2}) = 14.296 + 2.5 \times \log_{10}(410440/30) = 24.64$$

となります。これらのゼロ点を使って天体データ上で検出された天体の明るさを大雑把に求めることができるようになります (より正確な測光の技法については、専門の教科書などで勉強してください)。

ちなみに、例えばこの channel-1 と channel-2 のゼロ点を使って channel 間の感度比を求めてみると、

$$sc = 10^{[24.42-24.64]/2.5} = 0.817$$

となって、6.3.1節で述べた値とほぼ一致していることがわかります。

なお、デフォーカスの程度が大きく自動での開口設定がうまく働かない場合など、カーソルの位置から自分で決めた半径で開口を設定し、その中で測光をしたくなる場合があります。その場合、自動で光分布の中心を決めるのを止め、自分で星のカウントが0になる半径を評価し、その値を `imexamine` に教えて測光する事になります。以下にその場合の例を示します。ここでは同じデータを使っています。

まず、自動で光分布中心を決める工程を止めます。

```
mc> rimexam.center=no
```

次に、星のシグナルの裾野がどこまで伸びているかを評価します。画像上で十分星の裾野が無くなる半径を見積もり、その値を `rplot` で指定します。

```
mc> rimexam.rplot=40
```

こうしておいてから `imexam` を実行し、星の上で”r”カーソルを押します。すると図12のように、指定した天体の輝度プロファイルを表す図が半径40pixelまで表示されます。この輝

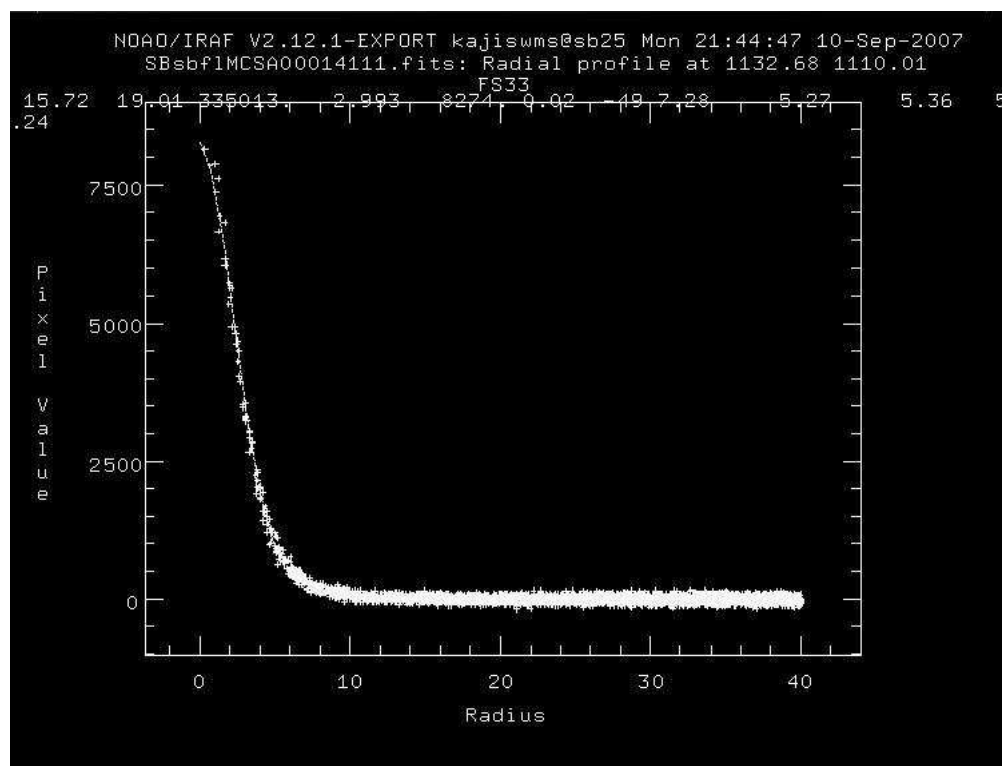


図 12: `rimexamine` で表示領域を明示的に指定した場合の光度プロファイルの表示例。 `imexamine` で天体上にカーソルを持っていき `r` を入力する。

度プロファイルを見てどの程度外側までいくと空と見分けがつかなくなるかの見当をつけて、どれくらいの半径で天体のカウント値を測るかを決めます。ここでは例えば（一旦'q'

で imexamine を終了して)

```
mc> rimexam.iterations=1 (これで自動開口決めルーチンが無効化されます)
mc> rimexam.radius=15
mc> rimexam.buffer=5
mc> rimexam.width=10
```

とでもしておきます。radius が天体のカウント値を測る半径を示し、buffer が天体を測る半径からスカイを見積もる内側の半径まで距離を、width がスカイを見積もる円環の幅を表します。この例の場合、半径 15 ピクセルの円で天体のカウント値を測り、半径 20 から 30 ピクセルの円環でスカイを見積もってこれを天体から引き去るということになります。半径を決めたら再度 imexamine を実行してカーソルを FS33 まで持っていき、今度は 'a' を入力することで測光を行います。

測光が終わったら、以下の様にして忘れずにパラメータをデフォルトに戻しましょう。

```
mc> unlearn rimexamine
```

より本格的には、IRAF の apphot パッケージを勉強してみてください。例えば、http://optik2.mtk.nao.ac.jp/subaru_red/autumn07/resume/spcam lec.pdf に吉田二美さんによる詳しい解説があります。

6.4.3 部分読み出しデータの処理

最後に、部分読み出しによる標準星データの処理を解説します。6.4.1 節で述べたように、明るい標準星の観測などでは短時間露出を行うために検出器の部分読み出しを使うことがあります (図 13)。これらのデータに対しては mcsall で一括処理することはできません。この部分読み出しデータの処理のために MCSRED では prmask、cutpr といったタスクが用意されています。prmask が部分読み出しデータにおける天体マスクを作成するタスク、cutpr が部分読み出しデータのうち読み出し部分だけを切り出すタスクです。

部分読み出しデータの処理のおおまかな流れは次のようになります。

- 1) prmask で天体マスクを作成。
- 2) フラットフレームで割り算 (標準星データの場合は天体データの処理で使ったものと同じフラットフレームを使う)
- 3) sbselfsky でメジアンスカイ引きを行う
- 4) cutpr で読み出し部分だけを切り出す
- 5) 必要に応じて (通常の qmsepskysb の代りに) tsubanomaly を使ってスカイの引け残りを処理する

ここで tsubanomaly は qmsepskysb と似たような処理を (読み出し部分を切り出したデータのように) サイズが異なるデータに対しても行うタスクです。

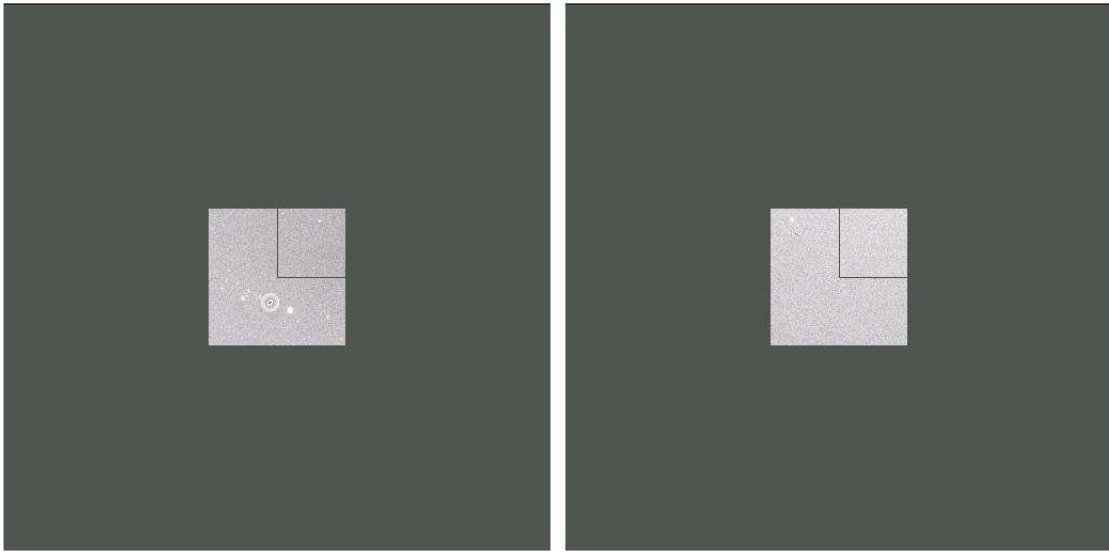


図 13: 部分読み出しデータの例 (左: channel-1, 右: channel-2)。この例では真ん中の 512pix × 512 pix の部分だけが読み出されている。

歪み補正を行う場合には `cutpr` で読み出し部分を切り出す前に `mcsgeocorr` 等を使用します。ただし部分読みはほとんど標準星観測時のみに使われているので、通常、部分読みデータに歪み補正を施すことはあまりないと思われます。

以下は例です。まずリストを作ります。/RAW というデータにある画像を使っています。

```
mc> listprep /RAW fs103 39115 39126
```

この結果を `prmask` に渡します。パラメータはほぼデフォルトで問題ないはずです。

```
mc> prmask fs103k_1.lst
mc> prmask fs103k_2.lst
```

次にフラットフィールドリングをします。ここは IRAF タスク `imarith` を用います。カレントディレクトリを見ると、`prmask` で出力された生データリスト、`bisfs103k_?.lst` (?は 1 か 2) があります。これを使ってフラット化後の画像リストをまず作成します。ここでは `awk` を用いてリスト作成をしています。

```
mc> !awk 'print "sf"$0' bisfs103k_1.lst > flbsfs103k_1.lst
mc> !awk 'print "sf"$0' bisfs103k_2.lst > flbsfs103k_2.lst
```

これらのファイルを用いてフラット化を行います。サイエンスデータで作成したフラット (ここでは `SFlat_ch?.fits` とします) を用いて割り算します。

```
mc> imarith @bisfs103k_1.lst / SFlat_ch1.fits @flbsfs103k_1.lst
```


すると、カレントディレクトリに割り算した結果のファイル (flMCSA****.fits) ができます。
次はメジアンスカイ引きです。作成したフラット済みデータのリストファイルを、次の sbselfsky タスクに渡します。

```
mc> sbselfsky flbsfs103k_1.lst 0  
mc> sbselfsky flbsfs103k_2.lst 0
```

ここで nself パラメータを 0 に指定しています。これは、リスト内のデータのうち、自分自身を除く全てのデータでメジアンスカイを作成する場合に使用します。

これが終わった段階で測光に入ってもあまり問題はないと思われませんが、例えば画像に風車の様なパターンが見られる場合、これを引く処理をすると精度が上がります。以下はその例ですが、実際に星が写っている画像だけ個別に処理する事になります。

まず、cutpr タスクで、部分読みデータ部分だけを切り出します。以下は例です。

```
mc> cutpr sbsfMCSA00039121.fits ctsbsfMCSA00039121.fits
```

結果画像を tsubanomaly タスクに通せば、よりフラットな画像にすることができます。

```
mc> tsubanomaly ctsbsfMCSA00039121.fits order=5
```

オーダーはあまり高くする必要はないでしょう。3-5 で十分です。この出力として、SBctsbsfMCSA00039121.fits というファイルができます。画像表示して、フラットになっているのを確認してください。

6.5 mcsall の後始末

mcsall では各工程において色々なファイルが出力されます (Appendix B 参照)。MCSRED にはこれらの出力ファイルのうち途中経過のものを掃除するタスク cleanall があります。例えば、

```
mc> ceanall hdfn1_1.lst level=0
```

のように mcsall での入力リストを指定して使います。level パラメータはどれだけのファイルを消してしまうかの度合いを表すパラメータです。

- level=0: tar でまとめるのみ。生画像以外すべて tar ファイルに残る。
- level=1: スカイ引き済み画像、マスクファイル、天体カタログ、フレーム間相対位置情報ファイルを残し、残りの (比較的再生容易な) ファイルを消す。
- level=2: 最終結果とリスト以外は全て消す。

6.6 MCSRED 使用上の注意

MCSRED を使用する際にはソースの CL スクリプトを読まれることを強く推奨します。スクリプト中にあるパラメータをチューニングすることによって、よりよい結果が得られる可能性が十分あります。ソースファイルは最初に tar ファイルを展開したディレクトリの下に MCSRED/ というディレクトリにあります。

このテキストは講習会資料であり、最新情報になっているとは限りません。より新しい情報は、MCSRED ディレクトリにある、README_****.txt (****はバージョン) というファイルにまとめられています。こちらにはより完全な MCSRED の各タスクの説明があります (ユーザー向けであり、初学者向けではありません)。最新情報は常にこの README ドキュメントになりますので、必ずこちらを参照するようにしてください。また、さらに新しい情報は、revise.log というファイルに常に更新されます。***tmp.tar という名前で配布される中間最新バージョンでは、こちらが最新情報のソースになります (日本語が主)。

MCSRED は常にユーザーの意見やニーズを受けて変化しています。もし要望がありましたら、検討しますのでご相談ください。

最後に、MCSRED は田中壺が自分のニーズに応じて作成している個人プログラムです。断り無く MCSRED を使用するのはいくらも構いませんが、それで得られた結果に対して、著者はいかなる責任も負えませんので、十分な理解をした上でご使用ください。

MCSRED の使用に際しては、以下の Web サイトを必ずご参照ください。

<http://www.naoj.org/staff/ichi/MCSRED/mcsred.html>

ご意見、ご質問は以下までお願いいたします。

田中 壺
国立天文台ハワイ観測所
E-mail: ichi "at-mark" naoj.org

7 mcsall では行われない工程

この講習資料では mcsall を中心に取り扱いますが、MOIRCS 撮像データ処理において mcsall では取り扱われない処理が必要な状況もあり得ます。ここでは、そのような例をいくつか紹介します。

7.1 NODDING 観測のデータ処理

Nodding は広がった天体の観測に用いられる手法で、MOIRCS では GETOBJECT コマンドにおいて “SKYNOD” オプションを指定することにより実行されます。この観測モードにおいては通常のディザを用いた観測とは異なり、

天体 1 -> スカイ 1 -> 天体 2 -> スカイ 2 -> 天体 3 ...

のように、1 フレーム天体を観測する度に視野を振ってスカイフレームを取得して、また天体を撮るといった作業を繰り返します。このようなデータに対しては、MCSRED では noddata パッケージをロードする事により、比較的たやすく一次処理が可能です。現在以下のようなタスクが用意されています。²

- subskyimage — スカイフレームに対する天体マスク作成、スカイ引き、フラットフィールドを行う。
- gcrsbimg — 歪み補正を行う (1 つの channel 用)。
- tiltskycor — 引け残りスカイの補正を行う (「風車パターン」の除去のみ)。
- mosgcsbimg — gcrsbimg で歪み補正を行ったデータをモザイクする。
- mossbimg — 両 channel のデータについて、光学歪み補正およびモザイクを一括して行う (=gcrsbimg+mosgcsbimg)。
- skycorgmk — 最初の gmkgtrimage を実行した後、スカイを引いて再合成しなおす。

ここではおおよその作業の流れのみを述べておきます。

1) hselect などデータをチェック (subskyimage は全ての入力フレームが同じ積分時間を前提としていることに注意)。

2) listprep で channel-1 と channel-2 のリストを作成。

3) subskyimage でスカイ引き、フラットフィールド、スカイフレームの天体にマスクの各作業を行う。

4) 上の工程で引け残ったスカイ成分のうち、いわゆる「風車パターン (A.3 参照)」のスカイ成分を引く。

5a) 相対感度比が分かっている場合、mossbimg を実行。

5b) 相対感度比が分からない場合、あるいはモザイク不要の場合、gcrsbimg で歪み補

² ヘッダキーワードの未整備のため、2007 年以前のデータでは動かない時があります。

正し、必要に応じて `mosgsbimg` でモザイク。

- 6) `gsxcat` で位置合わせ用の天体検出 (ディザ観測と同様)。
- 7) `gmkgtrimages` で位置合わせ及び足し合わせ (ディザ観測と同様)。
- 8) `skycorgmk` で背景領域を指定しすべての画像の背景レベルを合わせ、再合成。

なお、上の工程を一括して行う、`nodmosall` というタスクもあります。詳しくは最初に MCSRED の tar ファイルを展開したディレクトリの中の MCSRED/ディレクトリにある `MOIRCS_SKYNOD_REDUCE.txt` を参照してください。

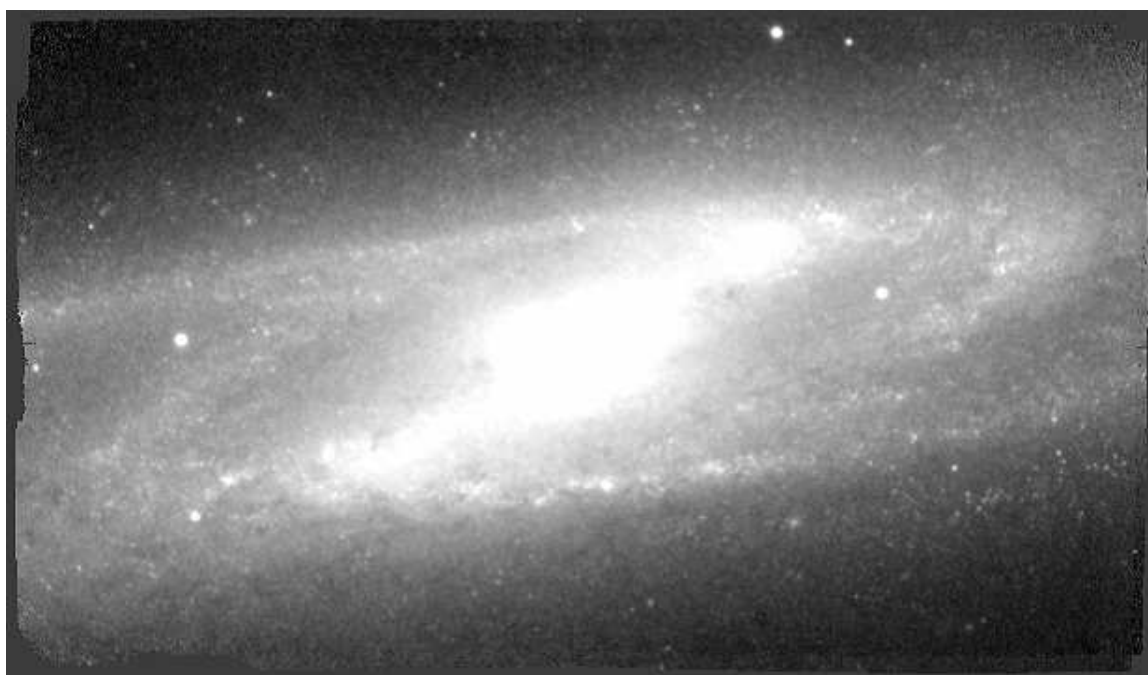


図 14: `noddata` パッケージによる処理例

7.2 一度足し合わせた画像から天体マスクを作る

`mcsall` では最初の `mcs_mksflat` タスクにおいて、各生画像に対して天体を検出してマスクを作成していますが、当然ながら複数枚のフレームを位置合わせして足し合わせた画像においてはより高い S/N で、より暗いところまで天体を検出することができます。一般に天体のより暗いところまでマスクをかけてメジアンスカイフレームを作ってスカイ引きを行った方が出来上がり画像のスカイのでこぼこをより小さくすることができます (図 15)。特に視野中に明るい天体がある場合や非常に長時間に渡る深いデータの場合はこの効果が大きくなります。

この作業を行うには、例えば `mcsall` で一度足し合わせの処理まで行って、その画像上で天体検出を行ってマスクを作り、それを使って再度フラットフィールドやメジアンスカイ引きの工程からやり直すというような方法があります。

現状 MCSRED には `invmask` という一回足し合わせた画像から新たにマスクを作り、それを使ってフラットフレームを作り直すというタスクが用意されています (IRAF v2.12.1

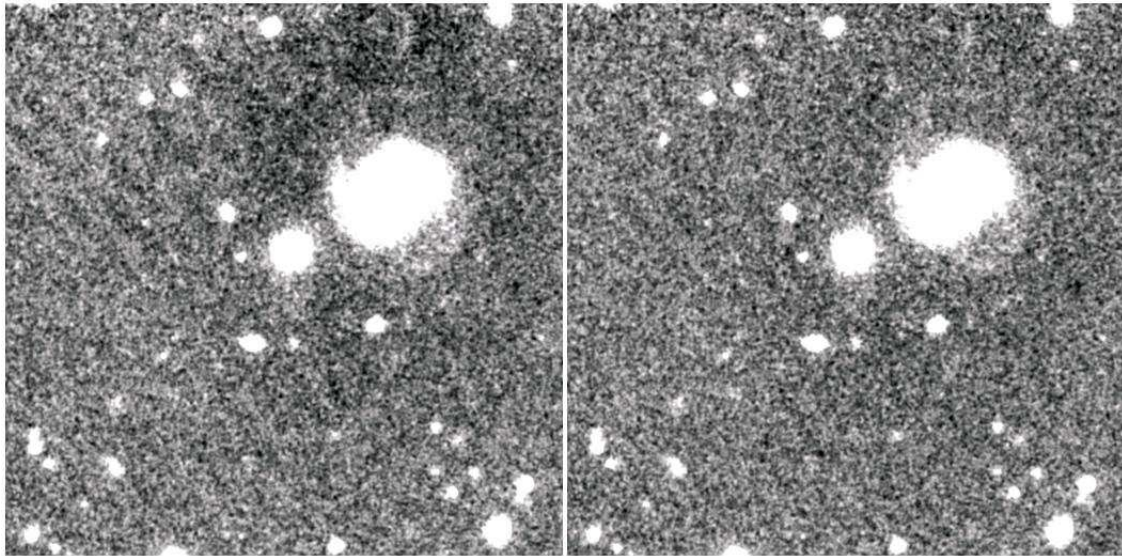


図 15: mcsall で足し合わせた画像 (左) と一度足し合わせた画像から天体マスクを作って再処理を行った画像 (右)。(今回のデータではそれ程大きな違いはないかも...)

においては、invmask 中で使われている geoxytran の direction パラメータが存在しません。ソースファイル invmask.cl 中の geoxytran の direction パラメータの記述部分を削除する必要があります)。

invmask を使った処理のおおまかな流れは

- 1) 1回 mcsall を行ったディレクトリで invmask を実行する。
- 2) invmask は足し合わせ画像を使って各フレーム用の新しい天体マスクを作る (objmsk000?????.pl、?????はフレーム ID ナンバー)
- 3) mcsall と invmask を実行したディレクトリにある生データ (MCSA000?????.fits) と新しい天体マスク (objmsk000?????.pl) を別のディレクトリに移す
- 4) 移したディレクトリに移動して mcsall を実行

これで新しいマスクを使って、フラットフィールドやスカイ引き、最終的な足し合わせまでの処理が行われます。

7.3 フリンジパターンの処理

天体データをドームフラットで割るとよくわかりますが (図 16)、MOIRCS データにはフリンジパターンが乗っています (特に K バンドや channel-2 で強い傾向があります)。これらはスカイコンディションがメジアンスカイが作られるタイムスケールで安定している場合にはメジアンスカイ引きの工程で引き去られますが、比較的短いタイムスケールで激しく変動している場合などにはスカイ引きを行った後でも引け残ってしまいます。また、セルフスカイフラットにも (多くの場合せいぜい 1-2% 以下のレベルでの影響ですが) このパターンが乗ることになります。

現在 MCSRED にはフリンジ処理のためのタスクは実装されていませんが、Weihao Wang

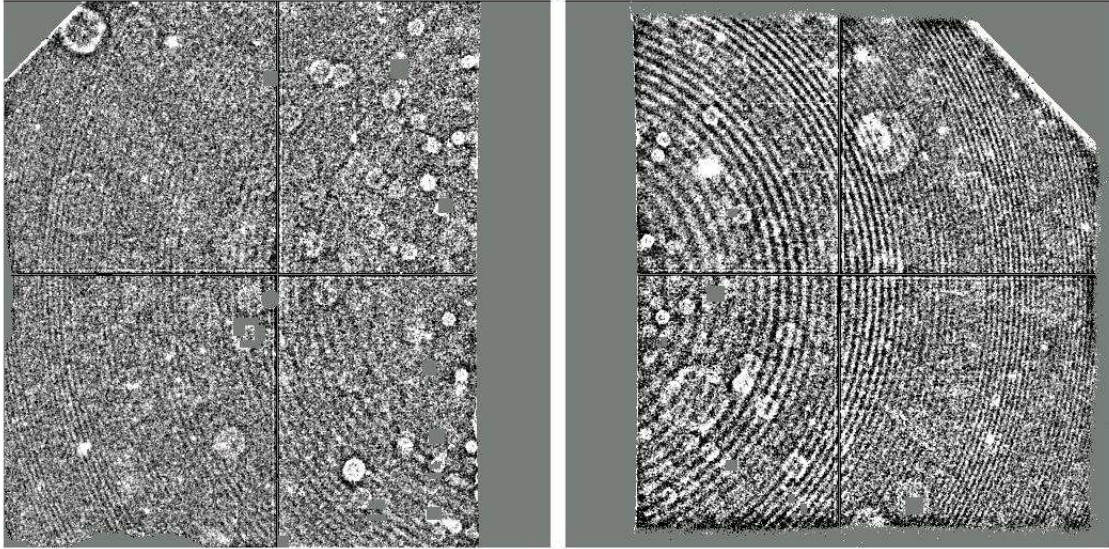


図 16: 生データ (左 : MCSA00014017.fits, 右 : MCSA00014018.fits) をドームフラットで割り大雑把にスカイを引いて、 $\sigma = 2\text{pix}$ のガウシアンでなました画像。画像全体に渡る大きな縞模様がフリンジパターン (小さな丸はゴミや bad pixel がガウシアンでなまったもの)。

氏による IDL ソフトウェアが既に一般公開されています。すばる望遠鏡の MOIRCS ページにリンクがありますので、利用をお勧めします。ここでは図 17 にフリンジ処理の方法の 1 例を紹介します (この工程はこのテキスト著者の鍛冶澤 賢氏により既にタスク化されていますので、必要な場合、コンタクトをされると良いでしょう)。

なお、現在 MOIRCS ではフリンジパターンが表れない新しいフィルターを導入しつつあります。詳細については MOIRCS のウェブサイトをご参照ください。

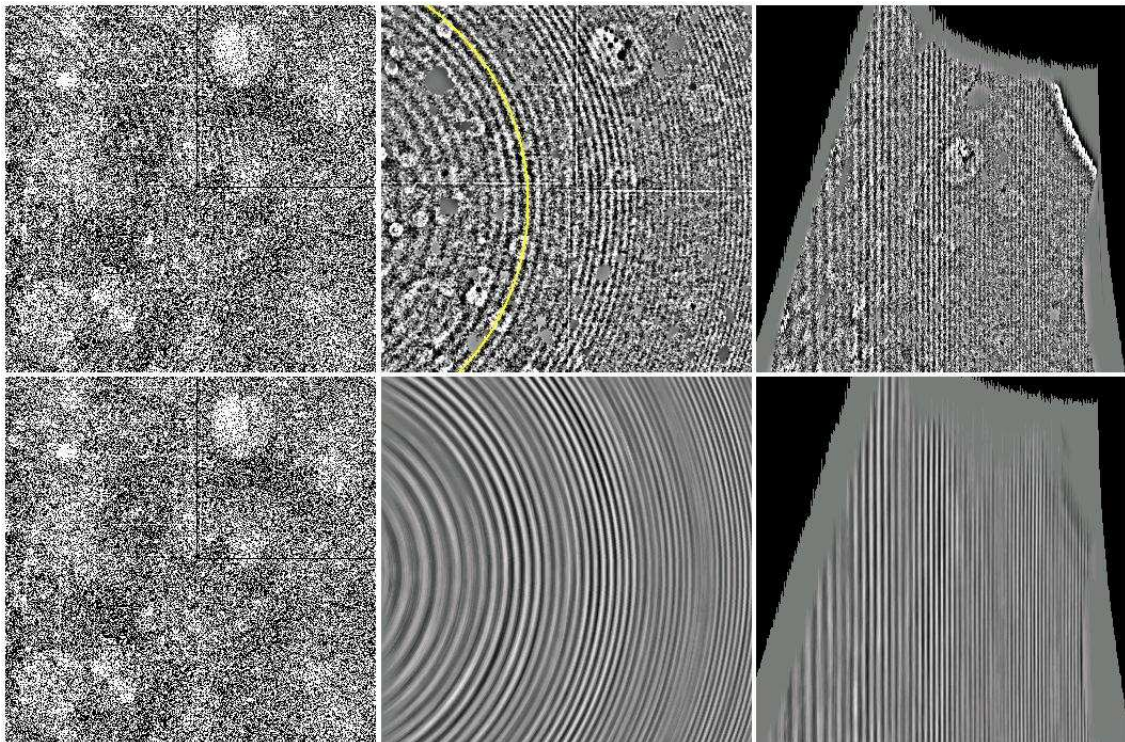


図 17: フリンジ処理の一例。左上: 生データをドームフラットで割ってスカイを引いた画像。真ん中上: 天体にマスクをしてガウシアンでなましてフリンジパターンを目立たせて、パターンの中心を見積もる。右上: 横軸に半径、縦軸に position angle をとった座標系に座標変換することで、フリンジパターンは縦方向の縞模様になる。右下: 縦方向にメジアンフィルターをかける。真ん中下: もとの座標系に逆変換。左下: もとの画像から差っ引く。

A mcsall で使われる各タスクについて

ここでは一括実行タスク mcsall の中に含まれている各タスクについて、簡単に説明します。より詳しい解説は、MCSRED ディレクトリ内に含まれる README ファイルにありますので、そちらを参照してください。

A.1 mcs_mksflat

mcs_mksflat はセルフスキフラットを作成するためのタスクです。また、その課程で作られる各フレームに対する天体マスクは後のメジアンスキ引きなどの工程でも使用されるため、mcsall では仮にセルフフラットを使用しない (dosf=no) 場合においてもまず始めに mcs_mksflat が実行されます。図 18 はセルフスキフラットの一例を示しています。ダークは 2005 年 12 月以降のデータでは比較的小さな値であり、通常 of データ処理においてダーク引きは行われていません。

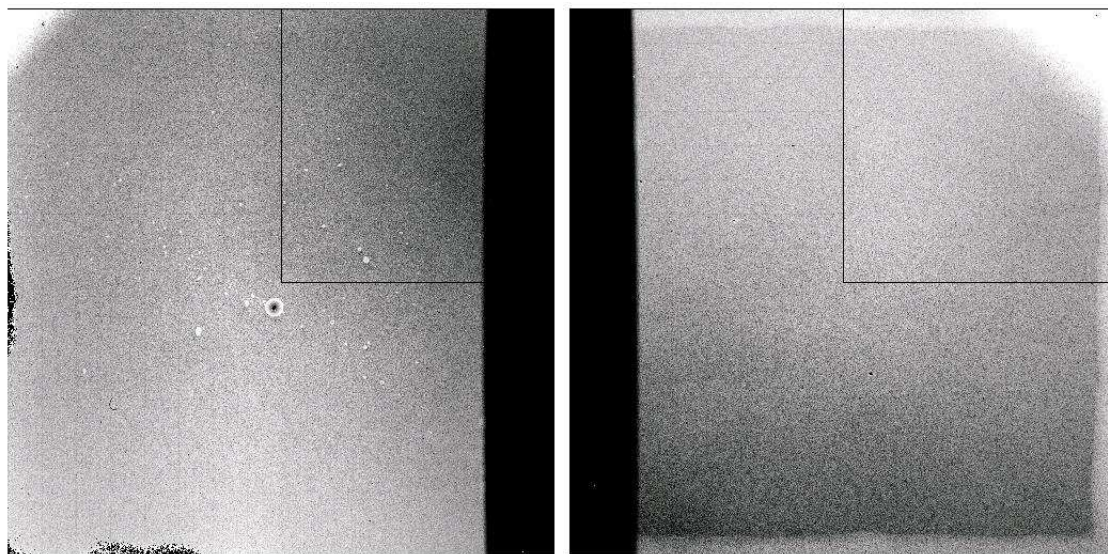


図 18: K バンドのセルフフラット (左 : channel-1, 右 : channel-2)

主な工程は

- 1) (ダーク引き)
- 2) 天体マスク作成
 - 2-1) 各フレームを直後のフレームで割り算。
 - 2-2) 象限ごとに低次関数で fit したものを差し引いて、スカイを平らにする。
 - 2-3) bad pixel にマスクをしてメジアンフィルタをかけてから pixel-to-pixel 統計 (ピクセル間での生カウント数の散らばり) をとる。
 - 2-4) 有意に値の高いピクセルとその回りを天体としてマスクを作る。
- 3) セルフフラット作成。
 - 3-1) 各フレームのメジアン値が最初のフレームに合うよう to かけ算で高さ合わせ。
 - 3-2) 入力された全フレームを天体マスクをかけながらメジアンで足し合わせる

- 3-3) 足し合わせた画像のメジアン値を測って、その値で割ることで規格化する
- 4) (作成したフラットを用いて入力ファイルのフラットフィールドを行うことも可能)

なお、ドームフラットデータの作成のためには、MCSRED の `mkdome` というタスクを利用できます。

A.2 sbselfsky

`sbselfsky` はメジアンスカイ引きを行うタスクです。フラットフィールド処理された画像のリストを入力すると、各フレームに対してその直前及び直後の指定された枚数 ($2 \times \text{nsf}$) のフレームを使ってメジアンスカイを作り、それを用いてスカイ引きを行います。この工程を通す事で、画像に乗るフリンジや検出器起因の模様などを減らすことができます。

メジアンスカイ作成に用いる枚数は多いほど `pixel-to-pixel` 統計におけるノイズの軽減が期待できますが、その分より長い時間に渡るスカイの変動の影響を受けてしまうというトレードオフの関係になっているので、データの積分時間やスカイコンディションに応じてほどよい妥協点を探るとよいでしょう。また、スカイが非常に安定した条件のデータでは、この工程を省略した方が結果が良いという事もあり得ます。

なお、入力リストが3枚の場合は `nsf=1` としますが、そのフレーム以外の2枚を比較して、カウントの小さいほう (天体でない) を選びスカイ画像とします。 `nsf=0` と入力すると、リストから自分自身を除く全てのフレームを使ってメジアンスカイを作成します。特殊例として、 `nsf=-1` とすると、自分自身も含む全画像をメジアンスカイ作成に使用します。

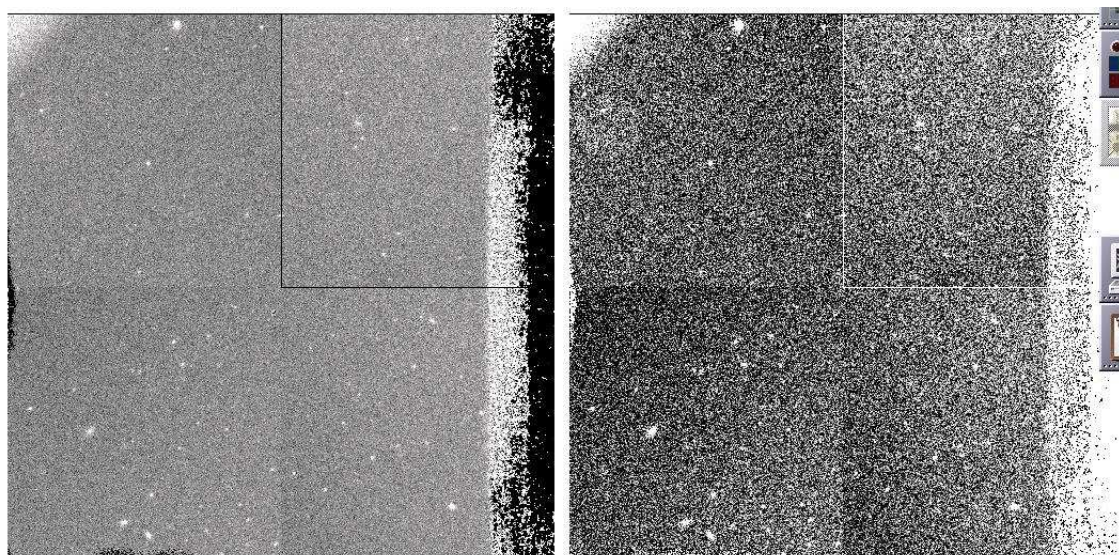


図 19: フラットフィールドした画像 (左, `fIMCSA00014021.fits`) とさらにメジアンスカイ引きを行った画像 (右, `sbfIMCSA14021.fits`)

主な工程は

- 1) 各フレームに対して直前、直後のフレームのリストを作成
リスト全体の枚数がメジアンスカイ作成に指定された枚数 ($2 \times \text{nsf}$) と比較して少ない場合、自らを除く全てのフレームでメジアンスカイを作る

2) メジアンスカイ作成

2-1) メジアンスカイの材料フレームの中で天体フレーム自身と積分時間が異なるものは積分時間の比でスケールする

2-2) 天体にマスクをかけながら足し合わせる

3) 作成したスカイフレームを天体フレームから差し引く

A.3 qmsepskysb

qmsepskysb は画像の各象限ごとに低次関数でスカイのフィッティングを行い、それをもとの画像から差し引くタスクです。mcsall においてはメジアンスカイ引きを行った後の画像のスカイの引き残りを処理するために使われます。

MOIRCS は4つの象限を独立同時に11.5秒かけて読み出しています。この比較的長い読み出し時間の間に雲などが来てスカイの明るさが急激に変化すると、変化の様子がそのまま画像上に記録され、風車のようなパターンとしてデータに乗ってしまいます(図19左: 風車になるのは象限ごとに同時に読み出すためです)。又、検出器への光の入射強度に依存して強さや傾きが変動する残像成分があるので、スカイの激しく変動している時などは、結果的に風車模様にさらに別のパターンが追加された様な引け残りパターンが、データによく見られます。qmsepskysb ではこれをきれいに差し引くために4つの象限それぞれ独立にスカイのフィッティング及びその差し引きを行っています。

なお、象限の境界付近に大きな銀河や明るい星がいたりすると、これに影響されフィッティングが失敗し、不自然な段差や、激しい波模様が画像に逆に付加されることがあるので、この工程の結果を良く確認しておくことが重要です(こうなった場合は、tiltskycor などの別タスクで回避する手段もあります)。

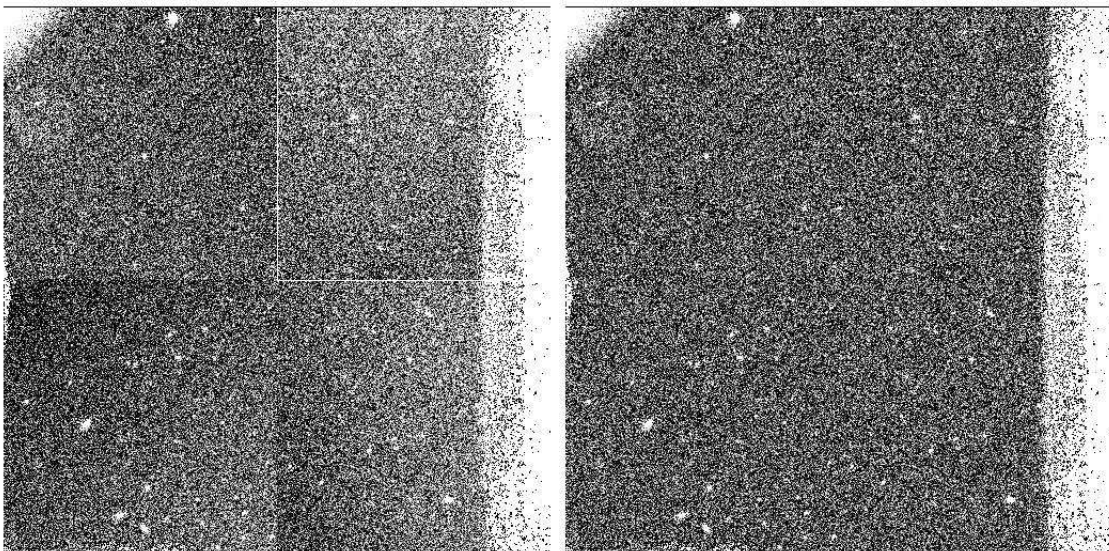


図 20: メジアンスカイ引きした画像 (左, sbfMCSA00014021.fits) と qmsepskysb でスカイの引け残りを処理して quadcor で象限間の溝を埋めた画像 (右, SBsbfMCSA14021.fits)

主な工程は

- 1) フィッティングに悪影響を及ぼす部分をマスク
 - 1-1) bad pixel、天体にマスクをして、channel-1(channel-2) の右端(左端) の光が入っていない部分を隣り合う領域を反転させたものを使って埋めておく。
 - 1-2) 3×3 のメジアンフィルターをかけて極端な値のピクセルを除く。また、channel-1(channel-2) の左上(右上) 角の部分はけられ(K-band においては熱源) があって急な勾配となっているのでこれにマスクをかけておく。
- 2) 各象限を切り出して `imsurfit` を使ってフィッティング
- 3) フィッティングされた低次関数をもとの画像から引く
- 4) 各象限をもとの一枚の画像に張り付ける

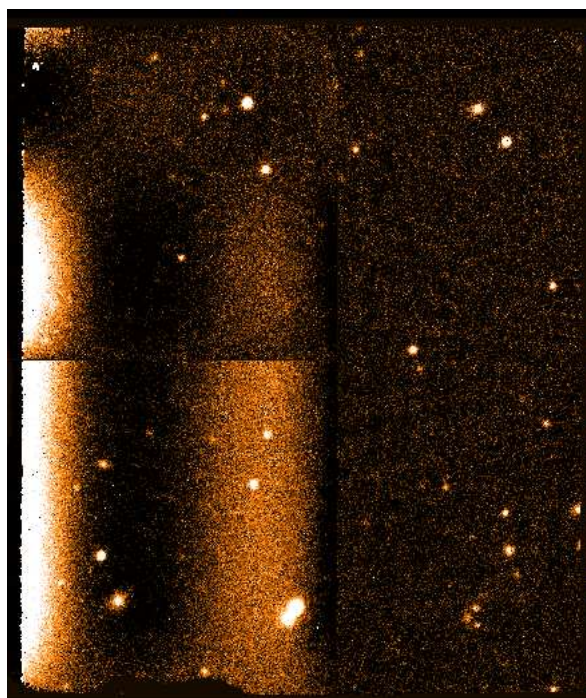


図 21: フィットスカイ引きに失敗した例。ここでは分光モード用のルールが視野の端に入ってしまったため、それに引きずられて第2、第3象限のフィット面が波打ってしまった。

A.4 quadcor

`quadcor` は画像の各象限の間の境界上の感度の低い溝の部分まわりのピクセルの値を使って補間するタスクです。このタスクは元来、ピクセルストレージのエラーによって生じていた各象限の間のずれを補正するために作られたタスクであり、2005年6月以前のデータに対してはこの補正を行います。このエラーは2005年6月に修正されていて、これ以後のデータに関しては象限間にずれはなく、単にその境界上に感度がない溝がある状態となっています。2005年6月以降のデータに関しては `fixpix` を用いてこの溝を埋める処理を行います。

なお、この補間された部分は実際はマスクされ解析には使われていないので、現在では標準星データの様な1枚1枚を解析する場合以外、あまり意味のある工程ではありません。

A.5 mcsgeocorr

mcsgeocorr は歪み補正を行うタスクです。

入力されたコンフィギュレーションファイルが指定する歪み補正データベースを用いて geotran を行うことで、画像を歪みのない状態に変換します。又、channel-1 と channel-2 は生画像では若干ピクセルスケールが異なりますが、それを同じ 0.117 (arcsec/pixel) に修正します。

MOIRCS の昇温-冷却が行われる度に、内部光学系の状態がわずかですが変化し、それに応じて歪み補正パラメータは変化します。それに応じて、解析に使うデータベースは変更されなければなりません。ですので、MOIRCS の昇温-冷却が行われる度に、歪み較正用領域の観測が行われ、そのデータから歪み補正用のデータベースが作られています。**観測データの歪み補正を正しく行うためにはそのデータに合ったコンフィギュレーションファイルを指定する必要があります。** config ファイルの選択については、6.2.6 節の mcsall のパラメータで説明したとおりです。

オプションとして歪み補正の画像変換の前に cosmicray タスクを使って、宇宙線除去の処理を加えることもできます。歪み補正の際は画素の再サンプリングが行われるため、元来 1-2 ピクセルに落ちていた宇宙線がその直近の画素全てに影響を与えてしまいます。宇宙線が再サンプリングで広がる前に、ある程度取り除く事で個々のフレームがきれいな結果になります。

A.6 gsextract

gsextract は SExtractor を用いて MOIRCS 画像の位置合わせのための天体検出を行うタスクです。タスクのパラメータによって SExtractor の天体検出に関わる主なパラメータを指定しますが、これらはデータのシーイングサイズや宇宙線の影響の度合などに応じて調節します。特に注意する点として、thresh パラメータを通常 SExtractor で使われる DETECT_THRESH の典型的な値よりも大きめに設定することがあります。これは歪み補正の画像変換によって隣り合うピクセル同士が相関を持ち、それによって画像の pixel-to-pixel の標準偏差が過小評価されてしまう効果を補正するためです。

パラメータによって指定された範囲のサイズを持つ天体のみがカタログされることに加えて、近くの天体と blend していたり、サチっているピクセルを含む天体もカタログから除かれます。また、nlimit オプションを使うと明るい方から 69 個までの天体のみのカタログが出力されます。

主な工程は

- 1) パラメータで指定された範囲の画像を切り出し、bad pixel を fixpix を使って補間する
- 2) SExtractor を実行
- 3) 出力された天体カタログから指定された範囲外のサイズ (FWHM) を持つ天体を除く
- 4) (center タスクを使って検出された各天体の中心の再評価を行うことも可能)

A.7 gmkgtrimages

gmkgtrimages は画像の位置合わせ及び足し合わせを行うタスクです。gsexcat で出力された各画像の天体カタログを使ってカタログマッチングを行い、相対位置を計算しています。

xyxymatch タスクによるカタログマッチングは mcsall の中でも最も失敗しやすい工程の一つですので、いきなり終わりまで流そうとせず、fstop=yes としてカタログマッチング工程で止めて動作を確認する事が有効です。もし失敗した場合は xyxymatch の nmat, tol, rat, nrej といった各パラメータなどを変更してみて再トライします。うまく最後まで行くパラメータが見つかったら、次に fstop=no fsresume=yes として流しなおすと、マッチング工程から先 (画像の位置あわせと最終合成) を行ってくれます。

カタログマッチングがどうしても失敗する場合は、各カタログで検出されている天体の数がひどく違うという事がないか、チェックしてみてください。シーイングの急変、天候条件の急変によって、急に天体数が変わることはよくあります。こういうデータは入力ファイルリストから外してしまった方が多い場合が多いですが、状況によっては必要な場合もあるでしょう。gsexcat の工程まで遡って、天体検出する領域をぎりぎりまで広くとってみる、検出閾値を下げてみるなどの、個別の対処が必要になります。

xyxymatch に失敗したフレームがある場合は、gmkgtrimages はそのフレームを足し合わせからは除いて最終合成へ進みます。失敗フレームの情報は、failed_cat_”入力リスト” という名前のファイルに保存されます。

chkbox は各フレームの大雑把な位置関係を FITS ヘッダから計算して自動的に足し合わせ画像のサイズフォーマットを決定するオプションですが、現在では各フレーム間の位置関係において回転が含まれないこと (平行移動のみ) を前提としています。互いに回転しているフレーム同士の位置合わせ及び足し合わせを行う場合には chkbox オプションは使わず、xbox, ybox, xcl, ycl のパラメータで足し合わせ画像のサイズと座標を自分で設定する必要があります。

カタログマッチの後には、各画像を入力リストの 1 枚目の画像と同じ位置に天体が来るように、各フレームの位置合わせが行われます。ここでは xy 方向の平行移動か、回転を許すか、という選択肢が推奨されています。しかし、複数の日にまたがるデータや、低高度で大気歪みを受けているデータなどを合成する場合は、fitgeo=general を選んでみるのも有効かもしれません。

gmkgtrimages においては入力データは全て EXPTIME=1sec にスケールされ、足し合わせ画像も 1 秒積分にスケールされたデータになります。もとの積分時間の情報は ORG_EXP というヘッダキーワードに記録されます。

位置合わせされた各フレーム (GTgcSBsbflMCSA000?????.fits) とともに各フレームに対応したマスクデータ (GTgcSBsbflMCSA000?????.pl) も出力されますが、各フレームにおいてなんらかの理由で追加でマスクをしたい領域などがある場合には、このマスクデータを編集して再度足し合わせを行うとよいでしょう。skip=yes として gtrlst パラメータに

作成された位置合わせ済みフレームのリスト (GTR_***.lst という名前のリストがあるはず) を指定すれば、再合成工程のみをやってくれます。

主な工程は

1) 位置合わせ

- 1-1) 各フレームの天体カタログを使って xyxymatch を実行し、入力リストの最初のフレームとのカタログマッチングを行う
- 1-2) 最初のフレームと対応づけられたカタログを使って、各フレームと最初のフレームとの相対位置 (変換式) を geomap で計算
- 1-3) 各フレームを geotran を使って最初のフレームに位置合わせ
- 1-4) bad pixel/region のマスク画像にも同様の変換を行って、位置合わせした各フレームに対するマスク画像を作成

2) 足し合わせ

- 2-1) 位置合わせした各フレームを積分時間で割り、1秒当たりのカウント値にスケール
- 2-2) (足し合わせの重みに使う場合には) 各フレームの [700:1300,700:1300] の領域での pixel-to-pixel の標準偏差を計算
- 2-3) imcombine でマスク、重みを使いながら足し合わせ

B mcsallの各工程で出力されるファイル

0) 入力リスト作成 (listprep)

channel ごとのリスト (“froot”_ [12].lst)

1) 天体マスク、セルフフラット作成 (mcs_mksflat)

生 (or ダークを引いた) データリスト (bis”inlist”)

天体マスク (omskMCSA000?????.pl)

セルフフラットフレーム (flat で指定)

2) フラットフィールド (imarith)

フラットフィールドされたフレーム (fMCSA000?????.fits)

そのリスト (flbs”inlist”)

3) メジアンスカイ引き (sbsselfsky)

メジアンスカイを引いたフレーム (sbflMCSA000?????.fits)

そのリスト (sbfbs”inlist”)

4) スカイ引け残りを低次関数 fitting で処理 (qmsepskysb)

低次関数を引き去ったフレーム (SBsbflMCSA000?????.fits)

そのリスト (SBsbflbs”inlist”)

5) 象限間の溝の補正 (quadcor)

溝を処理したフレーム (SBsbflMCSA000?????.fits ファイル名は変更されずに処理)

6) 歪み補正 (mcsgeocorr)

歪み補正済みフレーム (gcSBsbflMCSA000?????.fits)

そのリスト (dSBsbflbs”inlist”)

7) 天体検出 (gsextcat)

検出した天体のカタログ (gcSBsbflMCSA000?????.cat)

そのリスト (CATdSBsbflbs”inlist”)

8) 位置合わせ及び足し合わせ (gmkgtrimages)

入力リストの最初のフレームとマッチングされたカタログ (gcSBsbflMCSA000?????.gmp)

位置合わせ画像変換用データベース (gcSBsbflMCSA000?????.gtr geomap の出力ファイル)

位置合わせされた天体フレーム (GTgcSBsbflMCSA000?????.fits)

そのリスト (GTR_”inlist”)

位置合わせされたマスク画像 (GTgcSBsbflMCSA000?????.pl)

足し合わせ画像 (output で指定)

exposure map (exp_”output”.pl)

sigma 画像 (sgm_”outout”)

位置合わせ工程のログ (gmpdSBsbflbs”inlist”)

位置合わせに失敗して足し合わせに含まれなかったフレームのリスト (failed_cat_”inlist”)