

# Programmes d'analyse d'images et de suivis d'EROS

R. Ansari      ansari@lal.in2p3.fr  
É. Aubourg    aubourg@hep.saclay.cea.fr  
É. Lesquoy    lesquoy@hep.saclay.cea.fr  
C. Magneville   cmv@hep.saclay.cea.fr

# Table des matières

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Introduction</b>   | <b>3</b>  |
| <b>2</b> | <b>Les différents format de fichiers de PEIDA</b>                       | <b>5</b>  |
| 2.1      | fichiers Datacard . . . . .   | 5         |
| 2.2      | fichiers FITS . . . . .   | 5         |
| 2.3      | fichiers PPF . . . . .  | 5         |
| <b>3</b> | <b>La reconstruction photométrique</b>                                  | <b>8</b>  |
| 3.1      | Les principales étapes de la reconstruction photométrique . . . . .     | 8         |
| 3.2      | Les fichiers manipulés par stars++ . . . . .                            | 10        |
| 3.3      | Arguments de la ligne de commande de stars++ . . . . .                  | 11        |
| 3.4      | Les paramètres de stars++ (DataCard) . . . . .                          | 12        |
| 3.5      | L'association et la fusion de listes d'étoiles . . . . .                | 22        |
| 3.6      | Exemple de commandes pour la fabrication de fichiers de suivi . . . . . | 23        |
| 3.7      | Reconstruction d'images synthétiques . . . . .                          | 24        |
| <b>4</b> | <b>Rééchantillonnage et images compositées</b>                          | <b>25</b> |
| <b>5</b> | <b>Réduction Flat-field et Offset</b>                                   | <b>26</b> |
| 5.1      | redimg . . . . .  | 26        |
| 5.2      | reducall . . . . .  | 26        |
| 5.3      | acqfiltmed . . . . .  | 27        |
| <b>6</b> | <b>Quelques programmes utilitaires</b>                                  | <b>27</b> |
| 6.1      | dirphotos . . . . .   | 27        |
| 6.2      | header . . . . .  | 28        |
| 6.3      | rdheader . . . . .  | 28        |
| 6.4      | fitpacz . . . . .   | 28        |
| 6.5      | erosdate . . . . .  | 29        |
| 6.6      | findingt . . . . .  | 29        |
| 6.7      | imgdump . . . . .   | 30        |
| 6.8      | isole . . . . .   | 30        |
| 6.9      | mosaic . . . . .  | 30        |
| 6.10     | Utilitaires pour suivis . . . . .                                       | 30        |
| <b>7</b> | <b>Le contenu des fichiers de Suivi</b>                                 | <b>31</b> |
| 7.1      | GlobInfo . . . . .  | 31        |
| 7.2      | StarInfo . . . . .  | 32        |
| 7.3      | TimeInfo . . . . .  | 33        |
| 7.4      | Mesure . . . . .  | 35        |

|   |           |
|---|-----------|
| <b>8 Programmes de manipulation des fichiers de suivis</b>          | <b>36</b> |
| 8.1 Utilitaires de lecture des suivis . . . . .                     | 36        |
| 8.2 Dump de la structure TimeInfo, timf2asc . . . . .               | 37        |
| 8.3 Association de fichiers de suivi, suiviass . . . . .            | 38        |
| 8.4 Ré-alignement photométrique, recalib++ et ntreCALIB++ . . . . . | 38        |
| 8.5 Création d'un fichier de suivi réduit, svtrsel. . . . .         | 39        |
| <b>9 nbsread (C et C++)</b>   | <b>41</b> |
| <b>10 Variables d'environnement</b>                                 | <b>50</b> |
| 10.1 Variables pour le debugging. . . . .                           | 51        |
| 10.2 Variables pour le pilotage de stars++. . . . .                 | 51        |
| <b>11 Exemple commenté de sortie stars++</b>                        | <b>53</b> |

# 1 Introduction

Ce document contient une description des programmes d'analyse d'images et de fichiers de suivi développés pour le traitement des données d'EROS. Ces programmes sont basés sur les bibliothèques de classes et de fonctions **PEIDA++**<sup>1</sup> Photométrie et Etude d'Images Destiné à l'Astrophysique. La description de ces bibliothèques de classes est en dehors du cadre de cette note.

Les différents types de fichier manipulés par ces programmes sont présentés dans le premier paragraphe. Les principales étapes de la reconstruction photométrique, effectuées par le programme `stars++` sont ensuite décrites, suivi de la présentation d'autres programmes de traitement des images (`rebin`, `redimg`, ...) et quelques programmes utilitaires (`header`, `listphotos`, ...).

La présentation du contenu des fichiers de suivi d'EROS, et de l'architecture des programmes d'analyse de ces suivis (**nbsread**) fait l'objet de la dernière partie de cette note.

Les fichiers entête, (.h), bibliothèques et programme exécutable de PEIDA sont installés dans un répertoire défini par la variable d'environnement **EROSBASEREP**. Les exécutables et les bibliothèques se trouvent dans un sous-répertoire dont le nom est celui du système d'exploitation (commande `uname`), complété par celui du compilateur utilisé.

```
$EROSBASEREP/HP-UX-g++/ : Hewlett-Packard, g++
$EROSBASEREP/AIX-g++/ : IBM AIX, g++
$EROSBASEREP/OSF1-cxx/ : Compaq (ex. Digital Equipment) OSF1, cxx
$EROSBASEREP/Linux-g++/ : Linux, compilateur g++ (GNU)
$EROSBASEREP/SunOS-KCC/ : Sun, compilateur KAI (KCC)
```

Exemple de chemins d'accès aux différentes composantes de PEIDA, pour un système DEC/OSF :

```
$EROSBASEREP/Include/ : Les fichiers entête (.h)
$EROSBASEREP/OSF1-cxx/Libs/ : Bibliothèques (.a)
$EROSBASEREP/OSF1-cxx/ShLibs/ : Bibliothèques partagées
$EROSBASEREP/OSF1-cxx/Exec/ : Les exécutables.
```

Sur la majorité des plateformes (DEC-OSF, Linux, SunOS, SGI IRIX64 en V\_Nov99) des bibliothèques partagées (*shared library*) de PEIDA peuvent être utilisées. Dans ce cas, il faut que le chemin d'accès aux bibliothèques partagées soit défini au moment de l'exécution. La variable d'environnement **LD\_LIBRARY\_PATH** (DEC-OSF, Linux et SunOS) ou **LD\_LIBRARY\_PATH LD\_LIBRARYN32\_PATH LD\_LIBRARY64\_PATH** (SGI - IRIX64) doit alors être définie. Utilisez par exemple sous `csh`, la commande `setenv` :

```
csh> setenv LD_LIBRARY_PATH \
      .:$EROSBASEREP/OSF1-cxx/ShLibs/:$LD_LIBRARY_PATH
```

Les programmes de reconstruction photométrique utilisent les bibliothèques suivantes :  
**libcutils.a** : Utilitaires et fonctions mathématiques (C)

---

<sup>1</sup>PEIDA :

**libOutils++.a** : Classes et fonctions mathématiques et utilitaires (C++)

**libImages++.a** : Classes et fonctions de manipulation des images et des fichiers au format FITS (C++)

**libStarReco++.a** : Reconstruction photométrique. (C++)

**libfsv.a** : Gestion des fichiers de suivi (C)

Les trois bibliothèques de classes et fonctions C++ ont été regroupées dans la bibliothèque partagée libpeida.so et les fonctions C dans la bibliothèque liberosc.so.

Les bibliothèques de PEIDA possèdent un numéro de version, et un numéro de révision. Les versions majeures sont identifiées par un nom symbolique, correspondant en général à un numéro de révision unitaire. ceux-ci sont imprimés par tous les exécutables utilisant ces bibliothèques :

```
PEIDA Version 3.8 Revision 1 (V_Nov99) - Nov 10 1999 01 :43 :25 cxx
```

Numéro de Version 3.8, Révision 1, le nom symbolique V\_Nov99. La date et l'heure de compilation, ainsi que le compilateur utilisé sont aussi indiqués.

## 2 Les différents format de fichiers de PEIDA

Les fichiers utilisés par PEIDA peuvent être regroupés en quatre grandes catégories :

- **Datacard** : les fichiers datacard permettent de spécifier les paramètres de traitement.
- **FITS**<sup>2</sup> : c'est le format usuel des fichiers image utilisé en astrophysique.
- **PPF**<sup>3</sup> : est le format de fichiers commun aux objets des classes de PEIDA.
- **Suivi** : les fichiers de suivi, contenant les courbes de lumière. La structure des fichiers de suivi est décrite dans une note séparée (Note Peida No 1).

### 2.1 fichiers Datacard

Ce sont des fichiers texte ordinaire (ascii) qui sont manipulés par les objets de la classe **DataCards** de PEIDA. Les instructions DataCard sont constituées d'un mot-clé commençant par '@' en début d'une ligne, suivie éventuellement par une ou plusieurs valeurs. Toutes les autres lignes sont ignorées par le programme, facilitant l'inclusion de commentaires.

### 2.2 fichiers FITS

Le format **FITS** est très répandu en astronomie et en astrophysique. Il permet la sauvegarde de données de type entier ou réel structurées en tableau à plusieurs dimensions.

Ces fichiers, sont utilisables dans le cadre de PEIDA à travers les classes **FitsImage**<T> (**FitsImageU2**, **FitsImageI2**, **FitsImageI4**, **FitsImageR4**). Seuls les fichiers FITS représentant des tableaux à deux dimensions (images) peuvent être lus et écrits par les objets de ces classes.

### 2.3 fichiers PPF

De nombreuses classes de PEIDA héritent de la classe **PPersist**. Les objets de ces classes ont des méthodes de lecture et d'écriture sur des fichiers au format PPF. Ce format, ainsi que l'identification de type d'objet sont reconnus par PEIDA, permettant ainsi de créer les instances d'objets représentés par ces fichiers. Ceux-ci sont en format binaire ne peuvent être pas être facilement modifiés manuellement, sauf pour les objets des classes **ZFidu** et **DVList**. **Matrix**, **Vector**, **Poly**, **Poly2**, **RzImage** et **Image**<T>, **StarList**, **Transfo** et **ZFidu** constituent une liste non exhaustive de classes créant des objets persistants (**PPersist**). La date de création est aussi codée dans l'entête des fichiers **PPersist** (à partir de la version 3.7).

---

<sup>2</sup>Flexible Image Transport System

<sup>3</sup>Peida Persistent-object File

- **StarList** sont des listes d'étoiles. Les fichiers de référence de stars++ sont des fichiers PPF représentant la liste des étoiles détectées sur l'image de référence.
- **RzImage, Image<T>** Les RzImage et les images typées, (ImageU1, ImageU2, ImageI2, ImageI4, ImageR4) peuvent aussi être écrites et lues dans des fichiers PPF. Les FitsImage<T> héritent des images typées.
- **ZFidu** Les instances de la classe ZFidu peuvent être utilisées pour manipuler des zones fiducielles sur les images. Celles-ci sont définies à partir d'une région acceptable rectangulaire (GR) ou circulaire (GC), à laquelle on soustrait éventuellement certaines zones (cercles ou rectangles. Les ZFidu peuvent aussi contenir un tableau de mauvais pixels (ImageU1). Les fichiers PPF représentant des ZFidu contiennent la liste des rectangles et des cercles sous forme ASCII, et peuvent être modifiés à l'aide d'un éditeur de texte. Les rectangles et les cercles sont respectivement spécifiés par :

```
R XMin XMax YMin YMax
C XC YC R
```

Un exemple de fichier PPF ZFidu :

```
PEIDA-PPersistFile V3                                LITTLE-ENDIAN                                0^C-
-----
-----
GR 100 500 200 700
--Bad-Zone-List--
R 2 130 150 300 350
R 255 230 250 300 380
R 255 100 105 203 207
C 1 400 450 35
C 255 300 600 23
C 255 260 340 28
ZZZZZ-End-of-Bad-Zone-List
```

Afin de faciliter la gestion des fichiers, différents répertoires peuvent être spécifiés. ceux-ci peuvent être définis à travers des variables d'environnement (setenv). Pour certains programmes, ces chemins d'accès peuvent aussi être définis par les instructions du fichier DataCard pour l'exécutible stars++. Les instructions DataCard sont prioritaires par rapport aux variables d'environnement.

Quatre zones (répertoires), peuvent être définis :

- **Images** : Ce chemin d'accès peut être spécifié à l'aide de la variable d'environnement PEIDA\_IMAGES ou par l'instruction @IMAGEDIR dans le fichier DataCard. Ce chemin est utilisé pour tous les fichiers image (FITS).
- **Références** : Le chemin d'accès pour les fichiers de référence (StarList) peut être défini à l'aide de la variable d'environnement PEIDA\_REF ou par l'instruction @REFDIR dans le fichier DataCard.
- **Suivis** : Le chemin d'accès pour les fichiers de suivi créés ou mis à jour par le programme peut être défini à l'aide de la variable d'environnement PEIDA\_SUIVI ou

- par l'instruction @SUIVIDIR dans le fichier DataCard.
- **Travail** : Un répertoire de travail peut être défini à l'aide de la variable d'environnement PEIDA\_WORK ou par l'instruction @WORKDIR dans le fichier DataCard.

### 3 La reconstruction photométrique

Pour la constitution des courbes de lumière, une série d'image d'un même champ doit être traitée. La détection d'étoiles est effectuée sur l'image de référence, et la position des étoiles sont ensuite imposées sur chacune des images après recalage géométrique, à partir des positions de référence. Les flux de référence permettent en outre un alignement photométrique relatif des images. Des listes **StarList** d'étoiles de type **RzStar** sont adaptées à ce type de traitement effectué par le programme `stars++`. Un fichier de référence de `stars++` est donc un fichier PPF, contenant une **StarList(RzStar)**. La figure 1 montre l'interaction du programme `stars++` avec les différents fichiers.

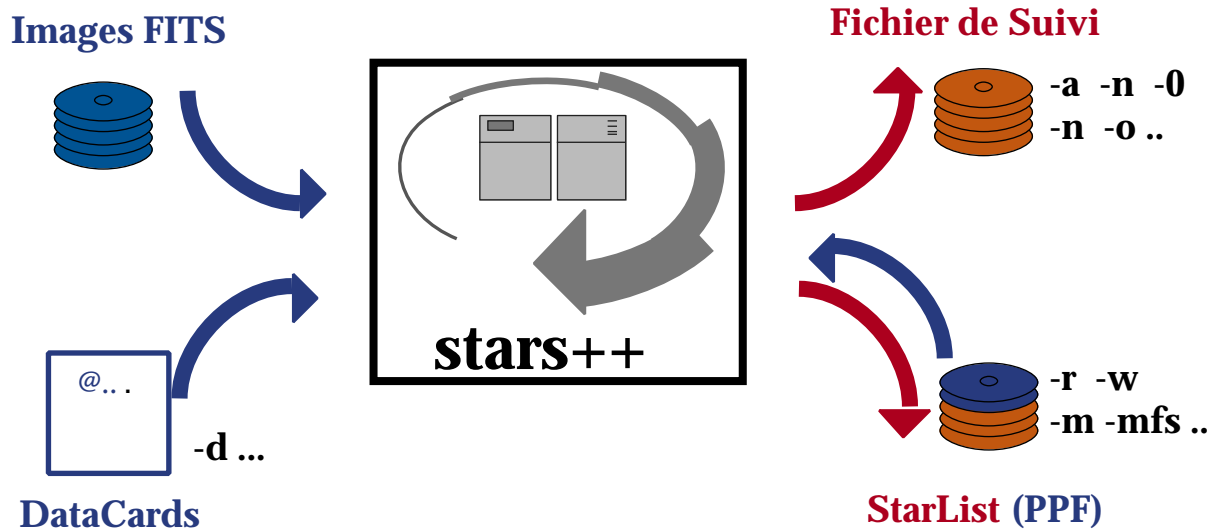


FIG. 1 – `stars++` et ses différents fichiers

#### 3.1 Les principales étapes de la reconstruction photométrique

##### Détecteur d'étoiles (MakeStarList())

La recherche d'étoiles est la première étape de la reconstruction des images. Les étoiles de la liste des brillantes (Bright) sont les étoiles de flux élevé, mais non saturée sur l'image de référence ont (Voir @DETETOILE, @STARLIST). Trois détecteurs d'étoiles peuvent être utilisés :

- **SimpleFind** est un détecteur simple, mais rapide. Il est basé sur la recherche de maximum locaux, au-dessus d'un certain seuil par rapport au fond de ciel.
- **DAOFind** implémente l'algorithme de détection de DAOPHOT.
- **CorrFind** effectue la recherche d'étoiles sur une image obtenue par filtrage de l'image originale. Ce filtrage consiste à calculer une image des corrélations avec la forme de PSF.

Une fois que le catalogue de référence a été établi, la recherche d'étoiles sur l'image courante n'est effectuée que si le calage par proximité échoue. L'utilisation de **SimpleFind** est alors appropriée.

– **Ajustement de position (DoFitPosSig())**

La position des étoiles est déterminée par deux ajustements à une dimension (Fit-1D) après projection suivant les deux axes X,Y, autour de la position approximative fournie par le détecteur d'étoiles, ou la position prédite à partir de la position de référence et de la transformation géométrique de calage. Les paramètres de largeur de PSF suivant les deux axes sont aussi calculés pour chaque étoile. Les positions calculées pour l'image sur laquelle la détection d'étoiles a été faite sont conservées comme positions de référence dans la liste d'étoiles. Les positions calculées par CorrFind étant plus précises, elles sont conservées comme positions de référence. Pour DAOFind et SimpleFind, la position fournie par le détecteur d'étoiles est remplacée par celle obtenue par l'ajustement 1D, pour les étoiles où l'ajustement n'échoue pas. (voir @FIT1D @TYPSPF @WINSIZE)

– **Recalage géométrique (DoCalFitPos())**

Plusieurs méthodes complémentaires sont disponibles pour la détermination de la transformation de calage. La logique de contrôle, ainsi que l'appel à DoFitPosSig() se trouve dans la fonction DoCalFitPos() et ses paramètres sont contrôlés par @ALIGEOM @TRFINI @NOCALPEIDA . Dans la méthode la plus simple, une estimation du décalage géométrique pour l'image courante est obtenue en recherchant les étoiles brillantes autour de leurs positions de référence. Cette méthode ne convient que pour des petits décalages (jusqu'à  $\sim 30$  pixels maximum, pour des images EROS-2). Si cette méthode échoue, une recherche d'étoile est menée sur l'image courante, suivi du recalage géométrique de cette liste sur celle de référence. Là aussi, différentes méthodes ont été mises en oeuvre (*classes Transfo, AligQuick*) :

Histogrammation des différences de positions ( $\delta_x \delta_y$ ) entre paires d'étoiles de deux listes.

Association de triangles semblables.

Une transformation géométrique précise est ensuite calculée en comparant les positions de l'ensemble des étoiles sur l'image courante à celles de la référence (voir @TRANSFO).

– **Calcul PSF 2D (DoPSF2DFit())** : Les paramètres de forme de la PSF sont déterminés sur des étoiles brillantes par un ajustement complet de chacune de ces étoiles. Ces paramètres sont ensuite fixés pour la détermination des flux de l'ensemble des étoiles. (Voir @PSF2D, @TYPSPF dans la liste des datacards)

– **Calcul des flux (DoGlobStFit())** : Pour le calcul des flux, les paramètres de forme de la PSF sont fixés et les positions des étoiles sont calculées à partir des positions de référence et la transformation géométrique de calage pour l'image courante. Le flux de chaque étoile est alors déterminé par un ajustement sur les valeurs des pixels autour de l'étoile, en laissant libre le fond de ciel, l'amplitude de l'étoile,

et s'il y a lieu, celles de ses proches voisines. `DoGlobStFit()` fournit donc pour chaque étoile, la valeur du fond de ciel et du flux, ainsi que les erreurs correspondantes. (Voir `@WINSIZE @TYPPSF`).

- **Calcul des flux (classe `StLMPhotometrie`)** : Une deuxième méthode de calcul de flux des étoiles, avec les paramètres de forme de la PSF et les positions fixés, est disponible. Cette méthode, implementée à travers la classe `StLMPhotometrie` est inspirée de la photométrie plaques EROS-1. L'image est découpée en pavés ayant un recouvrement (zone de garde). Un ajustement global des flux de toutes les étoiles d'un pavé est effectuées. Le flux est conservé pour les étoiles se trouvant hors de la zone de garde, donc loin du bord de pavé. (Voir `@LMPHOT @LMPHOTZONE`).
- **Alignement photométrique relatif (`DoCalibr()`)** : Cette étape calcule une fonction de correction à appliquer aux flux mesurés afin de tenir compte des différences dans les conditions d'observation. Les valeurs de flux sont ramenés en moyenne aux valeurs des flux référence. (Voir `@PARCAL`)
- **Calcul des résolutions (`DoResol()`)** : Le calcul de la dispersion des mesures et de l'écart systématique des flux reconstruits est effectué dans cette partie. Un ajustement des erreurs externes (dispersion des mesures) en fonction de  $\text{Log}_{10}(\text{Flux})$  est aussi réalisé, ainsi que le calcul d'une correction aux erreurs estimées par les fonctions d'ajustement.

### 3.2 Les fichiers manipulés par `stars++`

Les données en entrée de `stars++` sont constituées principalement de fichiers images à traiter. Les noms des fichiers image doivent être spécifiés en arguments de la ligne de commande. Actuellement, seul les fichiers image au format FITS peuvent être traités.

Un fichier référence au format PPF, contenant une copie de la `StarList` complète, peut être créé par `stars++`, à partir d'une image de référence et peut servir ensuite comme catalogue d'entrée pour le traitement d'une séquence d'images. Ces fichiers portent en général le nom de l'image correspondante, avec une extension `.ref`. Le fichier `gn40014tbr6i2426.ref` est le nom de la référence créée à partir de l'image FITS `gn40014tbr6i2426.fits`. `stars++` permet également la fabrication d'un fichier de référence "moyenné". Dans ce cas, le flux de référence pour chaque étoile est remplacé par la moyenne des flux ajustés sur les images traitées. Le nom de ce fichier est contrôlé par le datacard `@REFMOYNAME`, option de commande `-m`.

Normalement, les résultats de reconstruction sont stockés dans un fichier de suivi. leur contenu est décrit dans la dernière partie de cette note. Les résultats globaux des différentes étapes de traitement se trouve aussi dans le "fichier log" (`stdout`, à l'écran).

Afin de faciliter la manipulation des noms de fichiers, plusieurs autres paramètres peuvent être définis par des variables d'environnement ou des instructions `DataCard`.

L'extension des fichiers images peut être définies par la variable d'environnement

PEIDA\_IMAGEXT ou l'instruction @IMAGEXT. La valeur par défaut de ce paramètre est .fits. De manière similaire, l'extension de nom pour les fichiers référence est contrôlé par PEIDA\_REFEXT ou @REFEXT, avec .ref comme valeur par défaut. Le nom des fichiers de suivi est spécifié par PEIDA\_SUIVINAME ou @SUIVINAME, avec ccd.suivi comme valeur par défaut.

Le nom du fichier DataCard utilisé est déterminé par l'option -d NomDatacard de la ligne de commande ou par la variable d'environnement PEIDA\_DATACARD.

### 3.3 Arguments de la ligne de commande de stars++

Les noms des fichiers image et référence (sans extension si on utilise les variables IMAGEXT et REFEXT) à traiter doivent être spécifiés par les arguments de commande. En outre, des flags optionnels permettent de contrôler le fonctionnement de stars++. Certains de ces flags ont une action identique à des instructions Datacard ou bien des variables d'environnement. Les options et les flags de la ligne de commande sont prioritaires par rapport aux instructions DataCard, elles-même prioritaires par rapport aux variables d'environnement. On peut aussi obtenir l'impression de toute ou partie des informations de la StarList à travers l'option -psl de la ligne de commande, ou de manière équivalente par @PRTSTLIST dans le fichier DataCard.

Une mini aide en ligne est disponible par la commande stars++ -h

```
Commande> stars++ -h
PEIDA Version 3.4 Revision 36 (V_Mar97) -- Sep 22 1997 16:34:11 cxx
```

```
***** Programme Stars *****
```

```
Usage: stars++ [-Flags] [-Options ...] Reference [Image1 Image2 ...]
```

```
-> Flags d'activation/ desactivation -rwmsa0nb :
```

```
-w : Ecriture de starlist apres creation (Fichier .refxx)
```

```
-r : lecture du fichier de reference .refxx
```

```
-m : Flux moyennes pour la reference -> moy.refxx
```

```
-s : Couleur secondaire (0:Catalogue 1:Reference)
```

```
-n : Pas d'écriture de fichier de suivi
```

```
-a : Extension de fichier de suivi existant
```

```
-0 : Pas d'écriture de reference ds le fichier de suivi (0=zero)
```

```
-b : Ajustement en X,Y des brillantes seulement
```

```
-version : Impression du numero de version
```

```
-> Options de commande avec arguments:
```

```
-d DatacardFile : Nom de fichier datacard
```

```
-o SuiviName : Nom de fichier de suivi en sortie
```

`-mfs RefMoyName` : Calcul reference moyennee avec specification denom  
`-num CCDList` : Liste des numeros de CCD a traiter (Ex 4-8,12,15)  
`-psl ListImages,FgPrtList,FgPrtSti` : Impression de liste d'etoiles  
`-i StarList` : impose les positions de la starlist pour la reference

Les arguments entre [] sont optionnels. Le nom `Reference` désigne le nom d'un fichier image qui sera traitée alors comme image de référence, si le flag `-r` (pour `-reference` ou `-read`) n'est pas présent. Dans le cas contraire, `Reference` désigne le nom d'un fichier catalogue de référence (`StarList`, `.ref`) dont le contenu servira de base à la reconstruction des autres images. Si le flag `-w` (pour `-write`) est présent, l'objet `StarList` créé après traitement de l'image de référence est écrit dans un fichier `.ref` (format PPF). Le flag `-m` active la création et écriture d'un catalogue de référence, avec des valeurs de flux (`FluxRef`) calculées par une moyenne des valeurs de flux des étoiles sur chacune des images de la liste, exceptée la référence. Le flag `t -mfs` active la même opération et permet de spécifier le nom du fichier de référence en sortie. Le flag `-s` (`-secondaire`) peut être utilisé pour projeter une référence sur une nouvelle image pour la création d'une référence secondaire. Afin d'accélérer le traitement, on peut se contenter de retrouver la position des étoiles du sous ensemble des brillantes pour le calcul du calage géométrique (flag `-b`).

Les flags `-n` `-a` `-0` concernent les manipulations de fichiers de suivi. Le flag `-n` désactive la création ou la mise à jour des suivis. Le flag `-a` (pour `-append`) active le mode de mise à jour des fichiers de suivi existant, avec le résultat de reconstruction des Images 1,2,... Si le flag `-0` (zéro) est présent, la mesure correspondante à la référence n'est pas écrite dans le suivi.

Les flags `-d` et `-o` permettent de spécifier un nom de fichier `DataCard` et le nom de fichier de suivi en sortie, et `-num` la liste des numéros de CCD à traiter.

Le flag `-i StarList` d'imposer la position des étoiles uniquement, et non les flux, à partir d'une `StarList`. L'argument qui suit le flag `-i` est le nom du fichier `StarList` à partir de laquelle les positions seront imposées pour la reconstruction photométrique des image. Cette option permet en particulier l'utilisation d'un détecteur d'étoile autre que ceux disponibles dans PEIDA.

`-psl` active l'impression du contenu de la `StarList` dans le `LogFile` (Voir la description de l'instruction `DataCard @PRTSTLIST`).

`Image1 Image2 ...` constitue la liste des noms d'images à traiter.

### 3.4 Les paramètres de stars++ (DataCard)

Les paramètres de reconstruction d'image peuvent être modifiés à l'aide des instructions contenus dans un fichier lu à l'initialisation. Malgré la présence des valeurs par défaut, il est conseillé d'utiliser un fichier `Datacard`, avec des valeurs de paramètres adaptées au traitement souhaitée.

Les instructions `DataCard` sont constituées d'un mot-clé commençant par '@' en début d'une ligne, suivie éventuellement par une ou plusieurs valeurs. Toutes les autres

lignes sont ignorées par le programme, facilitant l'inclusion de commentaires. le fichier exemple.datacard contenant une liste commentée et à jour de tous les datacards est inclus dans le module StarReco++, avec les sources de la librairie.

- ▶ La définition des répertoires :
  - IMAGEDIR** RepertoireImages  
@IMAGEDIR /exp/eros/Images/
  - SUIVIDIR** RepertoireSuivis  
@SUIVIDIR /exp/eros/Test/
  - REFDIR** RepertoireReference  
@REFDIR /exp/eros/Test/
  - WORKDIR** RepertoireDeTravail  
@WORKDIR /exp/eros/Test/
  
- ▶ Noms des fichiers :
  - SUIVINAME** Nom de Fichier de Suivi (avec extension)  
@SUIVINAME eros2.sv
  - REFMOYNAME** Nom de fichier de référence (StarList) moyennée (sans extension) @REFMOYNAME erosmoy
  - IMAGEXT** Extension du nom des fichiers Images (format FITS)  
@IMAGEXT .fits
  - REFEXT** Extension du nom des fichier StarList (référence)  
@REFEXT .ref
  
- ▶ Liste des CCD (pour les noms d'images au format EROS-I :
  - CCDLIST** Numero des CCD (Ex : CCDLIST 1-3,7 ...)  
@CCDLIST 13
  - NUMPATT** n (Nb de chiffres, complétés a zéro, pour les numéros de CCD)  
@NUMPATT 2
  
- ▶ Impression des paramètres utilisés pour l'execution :
  - LIST** Niveau (Bits 0..5 Min=0,Max=63)  
@LIST 63
  
- ▶ Gestion de zones d'images
  - IMAGZONE** OffX OffY SizX SizY  
@IMAGZONE 140 200 512 512  
Pavé 512x512 à partir du pixel (140,200). Le premier pixel de l'image est le (0,0)  
@IMAGZONE 300 450 -1 -1  
Toute la sous-image à partir du pixel (300,450)
  
- ▶ Gestion de zones de StarList
  - STLRANGE** xmin xmax ymin ymax

Pour ne garder dans la StarList que les étoiles ayant une position comprise entre  $x_{min}$ - $x_{max}$  et  $y_{min}$ - $y_{max}$ . Ceci permet une gestion précise des problèmes de recouvrement quand, par exemple, on coupe un CCD 2048x2048 en plusieurs zones. ( $x_{min} \geq x_{max} \Rightarrow x_{min} = -1.e19$   $x_{max} = 1.e19$  idem pour  $y_{min}, y_{max}$ )

```
@STLRANGE 0 1024 0 1024
```

- Définition de taille des pixels (en mm, microns,...) sur l'image de référence et sur l'image courante (X,Y)

**PIXELSIZE** PxSz\_X\_Ref PxSz\_Y\_Ref PxSz\_X\_Cur PxSz\_Y\_Cur

Exemple : EROS-I\_comp 9.2 9.2 23. 23.

EROS-II\_nocomp 15. 15. 15. 15.

```
@PIXELSIZE 15. 15. 15. 15.
```

- Gestion des zones fiducielles :

**ZFIDUFILE** NomFichier

Pour imposer une zone fiducielle a partir d'un fichier ZFidu, de nom NomFichier se trouvant dans le répertoire WorkDir

```
@ZFIDUFILE myfidu.ppf
```

**BADRECT** mask Xmin Xmax Ymin Ymax

**BADCIRC** mask Xcentre Ycentre Rayon

Pour masquer des rectangles et des cercles. On peut spécifier un ensemble de plusieurs rectangles et cercles.

mask = 0 tous les masques

= 1 masque de type *pixel*

= 2 masque de type *ciel/etoile*

```
@BADRECT 0 200 350 700 750
```

```
@BADRECT 2 400 450 100 250
```

```
@BADCIRC 0 500 500 75
```

**AUTOZF** npmin lowbad highbad deb

Pour constituer une zone fiducielle avec détection automatique des mauvaises zones.

- npmin : nombre minimum de mauvais pixels mauvais adjacents

à partir duquel on tue la zone.

npmin  $\geq 0$  : les zones fiducielles sont des rectangles

npmin  $< 0$  : les zones fiducielles sont des cartes de pixels

- lowbad,highbad : définition des mauvais pixels (highbad  $\leq$  lowbad, on prend les valeurs de la carte DYNCCD

- deb : flag de debug,  $\geq 10 \rightarrow$  une image fits des masques est créée.

**ZFIDUCLR**

Pour ne pas utiliser la zone fiducielle sauvegardée avec la StarList, dans le fichier de référence.

```
@ZFIDUCLR
```

## NOWRITEFIDU

Pour utiliser le masque (ZFidu) mais ne pas l'écrire dans le fichier de référence

@NOWRITEFIDU

## NORDQUALITE

Pour ne pas tenir compte des infos Qualité dans l'entête FITS.

@NORDQUALITE

### ► Paramètres de la liste d'étoiles (StarList) :

**STARLIST** NbMaxStar NbBright VoisWinSize

- NbMaxStar : Ordre de grandeur du nombre d'étoiles sur l'image. Cette valeur est destinée à optimiser la gestion mémoire. (Ex : 1000 ou 3000 pour une image avec plusieurs milliers d'étoiles, 100 ou 200 pour une image avec quelques centaines d'étoiles.
- NbBright : Nombre maximum d'étoiles sélectionnées pour la liste des brillantes.
- VoisWinSize : Taille de la fenêtre de recherche des voisines. (Ex : VoisWinSize=11, Pour chaque étoile, on conserve la liste de ses voisines se trouvant dans un pavé 11\*11 centré sur l'étoile).

@STARLIST 3000 250 11

### ► Détecteur d'étoiles :

**DETETOILE** TypDet ThrNbSig DetStarPar[10]

- TypDet : Choix du détecteur d'étoiles (0,1,2,3)
  - =0 Ancienne détecteur PEIDA (ImageU2 uniquement)
  - =1 SimpleFind
  - =2 DAOFind
  - =3 CorrFind
- ThrNbSig : Seuil du détecteur d'étoiles (en nombre de sigmas du fond de ciel)
- DetStarPar[] : Autres paramètres du détecteur d'étoiles.
  - DAOFind : fwhm Sharp-low Sharp-hi Round-lo Round-hi  
Nav Nsum deb debhi Ideb Jdeb
  - CorrFind : rCorr(=sigma) seuil seuil2 rInt2 rExt2  
seuilNoise rNoise xlCorr surEch

si xlCorr = 0 lCorr = 2\*rCorr\*surEch+1

> 0 lCorr = xlCorr\*surEch

< 0 lCorr = -xlCorr\*rCorr\*surEch

@DETETOILE 1 5. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.

@DETETOILE 2 2. 3.8 -0.1 1.1 -1.1 1.1 1. 1. 0. 0. 0.

@DETETOILE 3 2. 1.6 0.4 0 1.6 2.6 -0.25 1.5 0 1 0

### ► Forme de la PSF pour les ajustements 1D et 2D

**TYPSPF** TypPSF1D TypPSF2D TermPSF2[0-4]

- TypPSF-1D : 1 = Gaussienne, 2 = Gaussienne Intégrée

- TypPSF-2D :
  - 1,2= Gaussienne, Gaussienne Integree
  - 3,4= Dev.Lim O(3) Gauss.,GaussInteg
  - 5,6= Dev.Lim O(2) Gauss.,GaussInteg. (B4 variable)
  - 7,8= Moffat et Moffat Integree

- TermPSF2 : [0]=(expo moffat/B4) [1]=terme B6 [2]=terme B2  
 @TYPPSF 1 1 0. 0. 0. 0. 0.

- Taille de pavés pour les ajustements d'étoiles :

**WINSIZE** Fit1D-WSz Fit2D-WSz

- Fit1D-WSz : Taille de base de la fenêtre pour ajustement  $X, Y / \sigma_{x,y}$  par projection (1D)
- Fit2D-WSz : Taille de base de la fenêtre pour l'ajustement des flux des étoiles (2D)

@WINSIZE 9 7

- Paramètres de l'ajustement numérique en 1D

**FIT1D** NSample NbProj NbPasSig SigMin SigMax NbPasPos PosMin PosMax  
 FgTabul

- NSample : Nombre de points pour l'ajustement. Celui-ci est donc effectué sur une projection en X ou Y, d'une longueur égale à NSample pixels.
- NbProj : Nombre de lignes/colonnes projetées
- NbPasSig : Le nombre de pas pour la détermination de sigma, (paramètre de largeur de la PSF).
- SigMin,Max : Bornes mini et maxi pour sigma, avec un pas de (SigMax-SigMin)/NbPasSig
- NbPasPos : Nombre de pas pour détermination de la position.
- PosMin,Max : Bornes mini et maxi pour la position. La position varie de 0 à NSample sur les pixels d'une projection.
- FgTabul : Si non nul, tabulation des ajustements activée

@FIT1D 5 2 300 0.2 3.2 300 1. 4. 1

@FIT1D 11 3 300 0.2 6.2 350 2. 9. 1

- Photométrie avec StlLMPhotometrie

**LMPHOT** LMPHOTPaveSz LMPHOTGarde LMPHOTNSig1 LMPHOTNSig2 LMPHOT-  
 FracFond

Cette carte permet de choisir l'ajustement des flux d'étoiles avec StlLMPhotometrie.

- LMPHOTPaveSz : taille des pavés d'ajustement global
- LMPHOTGarde : taille de zone de garde
- LMPHOTNSig1 : définition de la zone centrale d'ajustement de PSF en nombre de sigmas

- LMPHOTNSig2 : définition de la zone externe (zone d'influence de l'étoile, en nombre de sigmas de PSF.
- LMPHOTFracFond : Fraction (en nombre) des pixels du pavé utilisé pour l'ajustement du fond de ciel

@LMPHOT 50 5 1. 4. 0.25

**LMPHOTZONE LMPHOTXMin LMPHOTXMax LMPHOTYMin LMPHOTYMax**

Cette carte permet de limiter la zone d'ajustement au rectangle défini par LMPHOTXMin LMPHOTXMax LMPHOTYMin LMPHOTYMax. Par défaut, toute la zone couverte par les étoiles de la StarList est découpée en pavés pour l'ajustement.

@LMPHOTZONE 200. 500. 100. 400.

► Ajustement de PSF - 2D

**PSF2D NbMaxStars PaveSz**

- NbMaxStars : Nb Maxi d'étoiles pour la détermination de la PSF 2D
- PaveSz : Taille de la fenêtre pour FitPSF-2D
  - < 0 TaillePave = Abs(PaveSz) × (SigmaX+SigmaY) + 1
  - > 0 TaillePave = 2\*PaveSz + 1

@PSF2D 35 -4.5

► Calage géométrique des images

**ALIGEOM AlGMet AlGNbTrg AlGNbH2d AlGNbProp AlGDCut AlGDAss**

- AlGMet : Sélection de la méthode de calage. Bit 0 (=1), pour appliquer la méthode d'histogramation en permettant une translation seulement, Bit 1 (=2) méthode d'histogramation en autorisant en plus une petite rotation et homotétie, Bit 2 (=4) pour appliquer la méthode d'association des triangles semblables. La valeur 0 (AlGMet=0) permet d'appliquer les 3 méthodes dans l'ordre ci-dessus.
- AlGNbTrg : Nombre d'étoiles utilisées par la méthode des triangles. Compte tenu de la combinatoire, ce nombre ne peut en principe dépasser une vingtaine.
- AlGNbH2d : Nombre d'étoiles utilisées par la méthode d'histogramation. Ce nombre peut-être quelques centaines.
- AlGDCut : Nombre d'étoiles utilisées pour affiner la transformation géométrique (propagation) trouvée dans la première étape.
- AlGDCut : Translation maximal pour la méthode d'histogramation.
- AlGDAss : Distance maximale d'association d'étoiles dans la phase de propagation. C'est aussi la taille du bin des histos 2D des méthodes 1 et 2.

@ALIGEOM 0 16 100 500 200. 2.

**TRFINI Dx Dy Teta Hom Sym**

Initialisation de la transformation geometrique 1->2

- Dx : decalage en X (unite de de l'image 2)
- Dy : decalage en Y (unite de de l'image 2)

- Teta : angle de rotation (en radians)
- Hom : facteur d'homotétie
- Sym = 0 Pas de symétrie, 1 (x <> -x), 2 (y <> -y), 3 (x,y <> -x,-y)

$$(x_2, y_2) = T(x_1, y_1)$$

$$x_2 = Dx + h \times (sx \cos(\theta)) \times x_1 + sy * \sin(\theta) \times y_1$$

$$y_2 = Dy + h \times (-sx \sin(\theta)) \times x_1 + sy * \cos(\theta) \times y_1$$

$$(Sym = 0 \rightarrow (sx, sy) = (1, 1), Sym = 1 \rightarrow (sx, sy) = (-1, 1)$$

$$Sym = 2 \rightarrow (sx, sy) = (1, -1), Sym = 3 \rightarrow (sx, sy) = (-1, -1))$$

Priorités pour initialisation de la transformation :

1. carte TRFINI
2. carte PIXELSIZE pour déterminer l'homotétie Courante/Référence
3. valeurs de la taille des pixels dans la listes d'étoiles de référence et dans l'entête FITS de l'image courante, pour déterminer l'homotétie de la transformation courante → référence
4. Initialisation a l'identité

@TRFINI 0. 0. 0. 1. 0

**TRANSFO** flag fmin fmax

flag = 0 Toutes les étoiles

flag = 1 Les étoiles de la liste des brillantes seulement.

flag = 2 Toutes les étoiles avec fmin<FluxRef<fmax

@TRANSFO 2 1000. 50000.

**NOCALPEIDA**

Pour désactiver le calage par proximité de PEIDA.

@NOCALPEIDA

- Paramètres d'association des listes d'étoiles :

**ASSOCST** AssDMax

Distance maximum d'association (pour assocbr++)

@ASSOCST 5.

- Calcul du bruit des pixels

**DYNCCD** TypNoise MinADU MaxADU Gain RONoise RefFond RefSFond

- TypNoise : 0 ou 1 ou 2, méthode de calcul du bruit (0, 1 ou 2, la valeur "théorique correcte" étant 1)

0 : Bruit = 1 pour tous les pixels, 0 pour les pixels hors dynamique (>MaxADU ou <MinADU)

1 : Bruit =  $\sqrt{((RONoise/Gain)^2 + ValPix/Gain)}$  + marquage des pixels hors dynamique

2 : Bruit =  $\sqrt{(RefSFond^2 + (ValPix - RefFond)/Gain)}$  + marquage des pixels hors dynamique

3 : Bruit =  $\sqrt{Abs(ValPix)}$  + marquage des pixels hors dynamique

- Min,MaxADU : Valeurs mini,maxi acceptables pour les pixels
- Gain : Gain de la chaine de lecture (en Electrons/ADU)
- ROnoise : Bruit de lecture (en Electrons)
- RefFond,RefSFond : Valeur moyenne et sigma du fond (TypNoise=2)

Normalement, les paramètres des CCD sont lus dans l'entête fits et les valeurs de la carte data ne sont utilisées que pour des fichiers FITS incomplets. On peut néanmoins forcer les valeurs de DYNCCD en spécifiant la carte FORCEDYNCCD.

```
@DYNCCD 2 2 32000 6. 20. 140. 15.25 @FORCEDYNCCD
```

► Alignement photométrique relatif

```
PARCAL TypCal FgCal MinNbSt FracLow1 FracLow2 FracHigh1 FracHigh2 PA1  
PA2
```

Plusieurs méthodes d'alignement sont actuellement implantées, sélectionnée par le paramètre TypCal. Sauf pour TypCal=9, FracLow1 FracLow2 FracHigh1 FracHigh2 permettent de définir deux zones en flux de référence des étoiles utilisées pour l'ajustement des constantes de calibration. Les zones en flux sont définies par une fraction du nombre total d'étoiles ordonnées en flux référence. Par exemple, pour FracLow1, 2= 0.2, 0.65, la zone à bas flux s'étendra d'un flux minimum correspondant au seuil des 20% des étoiles les moins brillantes jusqu'à un flux maximum, correspondant au seuil des 65% des étoiles les moins brillantes. Les deux zones peuvent se chevaucher. La signification de FgCal, MinNbSt, Par1, Par2 dépend de la méthode de calibration choisie. Pour la calibration de type 9, les paramètres ont tous des significations particulières (voir ci-dessous).

- TypCal=0 :  $F_{cal} = a_1 \times F_{brut}$ . Les étoiles entre [ FracLow2 - FracHigh2 ] sont utilisées si FracLow2 < FracHigh2. Sinon, étoiles entre [ FracLow1 - FracHigh2 ] .

```
@PARCAL 0 0 100 0.20 0.99 0.65 0.99 0 0
```

- TypCal=1 : Alignement photométrique relatif linéaire. Deux ensembles de constantes, deg=1, deg=2 sont déterminés.

$$\begin{aligned} F_{cal} &= a_0 + a_1 \times F_{brut} \\ F_{cal} &= b_0 + b_1 \times F_{brut} + b_2 \times F_{brut}^2 \end{aligned}$$

Si FgCal=11, l'alignement de second degré est utilisé pour l'alignement des flux à la relecture du fichier de suivi. **Déconseillé** On utilise les bonnes étoiles entre [ FracLow1, max(FracLow2, FracHigh2) ] . Si MinNbSt > 0, sa valeur est imposée comme le nombre minimum d'étoiles acceptée pour la calibration dans la passe finale.

```
@PARCAL 1 0 100 0.20 0.99 0.65 0.99 0 0
```

- TypCal=2,3 : Dans ces cas, on détermine des fonctions de calibration ayant deux formes analytiques différentes sur les deux zones, à haut flux ( $FracHigh1, FracHigh2$ ), et à bas flux ( $FracLow1, FracLow2$ ). Le point de raccordement  $F_0$  est calculé automatiquement.

-TypCal=2

$$\begin{aligned} F_{cal} &= a_0 + a_1 \times F_{brut} & F_{brut} > F_0 \\ F_{cal} &= a_0 + a_1 \times F_{brut} + b_2 \times (F_{brut} - F_0)^2 & F_{brut} < F_0 \end{aligned}$$

-TypCal=3

$$\begin{aligned} F_{cal} &= F_{brut} \times (a_0 + a_1 \times \log(F_{brut})) & F_{brut} > F_0 \\ F_{cal} &= F_{brut} \times (a_0 + a_1 \times \log(F_0) + b_1 \delta_{lfb} + b_2 \delta_{lfb}^2) & F_{brut} < F_0 \quad \delta_{lfb} = \log(F_{brut}/F_0) \end{aligned}$$

@PARCAL 2 0 100 0.20 0.70 0.65 0.99 0 0

- TypCal=4 : Dans ce cas aussi, la fonction d'alignement photométrique relatif est constitué de deux formes analytiques avec un point de raccordement (pivot).

$$\begin{aligned} F_{cal} &= F_{brut} \times (a + b_1 \times \log(F_{brut})) & F_{brut} > F_0 \\ F_{cal} &= F_{brut} \times (a + b_1 \times \log(F_0) + b_2 \delta_{lfb}) & F_{brut} > F_0 \quad \delta_{lfb} = \log(F_{brut}/F_0) \\ F_{cal} &= F_{brut} \times (a + b_1 \times \log(F_0) + b_2 \delta_{F1}) & F_{brut} < F_1 \quad \delta_{F1} = \log(F_1/F_0) \end{aligned}$$

On calcule d'abord un premier coefficient de correction sur la zone a haut flux. Un point pivot (F0) est calculé ensuite sur cette même zone en flux. On ajuste ensuite les termes  $a$  et  $b_1$  sur la zone a haut flux, pour des flux supérieurs à F0. le terme  $b_2$  est calculé ensuite sur la zone a bas flux, pour des flux inférieurs a F0. F1 correspond à la borne inférieure de la zone à bas flux.

- TypCal=8 :  $F_{cal} = a_0 + a_1 \times F_{brut}$  Ajustement des constantes de calibrations sur nuage de points (idem TypCal=1), avec des paramètres de nettoyage des mesures (PA1, PA2) Les étoiles entre [ $FracLow1, \max(FracLow2, FracHigh2)$ ] sont utilisés, FgCal détermine le nombre de passages (2 ou 3) de calcul, avec 0 comme valeur par défaut. Les étoiles ayant un écart  $\Delta F = Abs(F_{cal} - F_{Ref}) > PA1/2 \times Err$  sont éliminés de l'ajustement dans passage 2,3/1

@PARCAL 8 3 -50 0.50 0.50 0.997 0.997 7. 12.

@PARCAL 8 0 -50 0.50 0.50 0.997 0.997 -1. -1. Valeurs par défaut.

- TypCal=9 : Ajustement en 3 intervalles par méthode de raccord/pivot :  
haut flux (HF) : Ajustement de droite avec nettoyage sur le nuage de points  
moyen flux (MF) : Ajustement de parabole (par intervalles de flux)  
bas flux (BF) : Ajustement de droite (par intervalles de flux)

- FgCal définit le nombre d'étoiles les plus brillantes à éliminer pour l'ajustement fit HF (négatif  $\rightarrow$  0).

- MinNbSt : Le nombre d'intervalles de type "haut flux" (def.  $< 0 \rightarrow 7$ )
- FracLow1 : Le nombre d'étoiles dans les intervalles à haut flux (cf FgCal) (def.  $< 0 \rightarrow 75$ )
- FracLow2 : nombre d'\* dans les intervalles à bas flux (cf FgCal) (def.  $< 0 \rightarrow 200$ )
- FracHigh1 : Valeur du flux brut en ADU en dessous de laquelle on ne considère plus les intervalles pour l'ajustement. Si  $< 0$  calcul automatique a  $\text{abs}(\text{FracHigh1})\%$  du maximum de l'histogramme de  $\log_{10}(F_{\text{brut}})$ .  
**Attention** : les bornes de l'histo sont fixées à 15 bins de 0. a 10.
- FracHigh2 : Nombre d'étoiles brillantes à prendre pour le fit HF (def.  $< 0 \rightarrow 0$ )
- PA1 : Pourcentage d'étoiles à nettoyer dans les calculs de rapports  $F_{\text{brut}}/F_{\text{Ref}}$  pour chaque intervalle (def.  $< 0 \rightarrow 0.2$ )
- PA2 : Pourcentage d'étoiles à nettoyer dans le fit (HF) du nuage  $F_{\text{brut}}$  vs.  $F_{\text{Ref}}$  (def.  $< 0 \rightarrow 0.2$ )

@PARCAL 9 50 7 75 200 -0.5 100 0.2 0.2

@PARCAL 9 -1 -1 -1 -1 -1 -1 -1 -1 Valeurs par défaut.

Il est possible de ne pas executer l'étape d'alignement photométrique et forcer les constantes d'alignement, pour un alignement linéaire.

#### **FORCECAL A0 A1**

$F_{\text{cal}} = a_0 + a_1 \times F_{\text{brut}}$ , TypCal=1.

@FORCECAL 0. 1.

Il est possible de ne pas executer l'étape de calcul de la résolution et des paramètres de correction des erreurs du fit. Ceci est pratique quand on veut faire un fichier de suivi avec des images prises dans des filtres différents.

#### **NORESOL**

@NORESOL

- Calcul des flux de références moyennées avec flux calibrés

#### **CALMOYFLX Cal1Min Cal1Max**

Si cette carte est présente, la moyenne est calculée avec les flux calibrés : ( $F_{\text{cal}} = a_1 \times F_{\text{brut}}$ ,  $a_1 = \text{Cal1}$ ), sinon avec les flux brutes. Seules TypCal = 0,1,8 sont valides si cette carte est présente et Cal1Min, Cal1Max définissent le domaine acceptable pour  $a_1 = \text{Cal1}$ .

@CALMOYFLX 0.7 1.4

- Generation d'images synthétiques

**IMGSYNTH FgSynth ListImages** Avec FgSynth = ab en notation décimal :

- **b = 0** → Image synthétique avec fond local (nom avec extension .syn)
- **b = 1** → Image du fond seulement (voir SyntheseImage( , , b, ...))
- **b = 2** → Image synthétique avec fond constant.
- **a > 0** → Création de l'image soustraite également (nom avec extension .sub)
- **ListImages** : 1 bit / image pour laquelle on créer l'image synthétique. (Ex 13 = Bit 0 + 2 + 3 Images 0 (Ref) 2 et 3)

- @IMGSYNTH 10 1
- ▶ Niveau d'Impression/Debug général du programme. Des niveaux d'impression/debug pour des modules particuliers peuvent être activés à l'aide de variables d'environnement. **PRTLEVEL** PrintLevel DebugLevel (0..5 0..5)  
@PRTLEVEL 2 1
  - ▶ Impression de la liste d'étoiles  
**PRTSTLIST ListImages FgPrtStl FgPrtStInfo**
    - ListImages : 1 bit / image pour laquelle on veut imprimer la liste (Ex 13 = Bit 0 + 2 + 3 Images 0 (Ref) 2 et 3)
    - FgPrtStl : 0 pour imprimer les étoiles de la liste des brillantes seulement, n>0 pour imprimer 1 étoile / n de la liste, n<0 pour imprimer 1 étoile / (-n) trié par ordre des flux décroissants.
    - FgPrtStInfo : Niveau d'impression de pour chaque étoile. Utiliser FgPrtStInfo=7 pour imprimer toutes les informations d'une liste de RzStar.
- @PRTSTLIST 1 100 7
- ▶ Choix de l'observatoire. 1 définit LaSilla pour le calcul du temps sidéral et légal (défaut), 0 sinon.  
**SETOBS ObsId**  
@SETOBS 1

### 3.5 L'association et la fusion de listes d'étoiles

Dans le schéma de traitement habituel, les images prises dans chaque couleur sont traitées de manière indépendante. Une association des listes d'étoiles obtenues dans chaque couleur doit alors être effectuée avant la création des fichiers de suivi. Cette étape s'effectue à l'aide du programme **assocbr++**. Il est possible de fusionner les listes d'étoiles obtenues dans les deux couleurs afin de compléter ces listes et d'améliorer la précision de position des étoiles, en particulier des étoiles de flux faible dans une seule couleur. Cette fusion peut être réalisée par le programme **mergebr**.

Le programme **assocbr++** utilise les mêmes variables d'environnement et data-cards que **stars++**. Un fichier DataCard peut aussi être spécifié. Outre les instructions DataCard relatives aux paramètres généraux (Répertoires, Niveau d'impression, ...), les paramètres de calage géométrique de listes d'étoiles peuvent être spécifiés (@ALIGN @ASSOCST @TRFINI). **assocbr++** permet aussi la création d'un NTuple diagramme C-M à l'aide de l'option **-nt**.

L'aide en ligne de **assocbr++** peut être obtenu par l'option **-h** :

```

Commande> assocbr++ -h
PEIDA Version 3.6 Revision 3 (V_Fev99) -- Feb 23 1999 16:59:18 cxx

***** Programme AssocBR *****

assocbr++ -h

```

```

Usage: assocbr [-Flags] [-Options ...] NomRef1 NomRef2 [NomOut1]
-> Flags d'activation/ desactivation -wn :
    -w : Ecriture de starlist apres association (Fichier .refxx)
    -n : Ne pas faire d'association
    -version : Impression du numero de version

-> Options de commande avec arguments:
    -d DatacardFile : Nom de fichier datacard
    -nt NTupleFile : Creation NTuple DiagCM ds fichier NTupleFile (ppf)
    -num CCDList : Liste des numeros de CCD a traiter (Ex 4-8,12,15)
    -psl ListImages,FgPrtList,FgPrtSti : Impression de liste d'etoiles

```

**assocbr++ associe les étoiles d'une première liste NomRef1 à celles d'une seconde liste NomRef2 et réécrit éventuellement la première liste avec les numéros de référence croisée des étoiles associées dans la seconde liste, ainsi que les distances d'association correspondantes. Si aucun nom n'est spécifié pour le fichier de sortie, NomOut1, le fichier NomRef1 sera surécrit.**

L'aide en ligne de mergebr peut-être obtenu par la commande mergebr sans arguments :

```

Commande> mergebr
PEIDA Version 3.4 Revision 36 (V_Mar97) -- Sep 22 1997 16:34:11 cxx
dMax=2 minfluxr=1500 minfluxb=2500
nsth2d=100 nsttrg=16 nguides=500
dass=2 dcut=-200 deg=1
usage: mergebr [-d dMax] [-i] [-f minfxr,minfxb]
             [-t nsth2d,nsttrg,nguides,dass,dcut,deg]
             inr inb outr outb
defaults: -d=2, -f=1500,2500 -t=100,16,500,2,-200,1

```

**inr inb outr outb** spécifient les noms des fichiers StarList dans les deux couleurs (Rouge, Bleue *r*, *b*) en entrée et en sortie. L'option **-f** permet de spécifier une limite inférieure pour les flux rouges (*minfxr*) et bleues (*minfxb*). **-d dMax** indique la distance maximum d'association. L'option **-t** permet la définition des paramètres de calage géométrique (voir la carte @ALIGEOM ci-dessus), avec *deg* étant le degré du polynome de la transformation géométrique.

### 3.6 Exemple de commandes pour la fabrication de fichiers de suivi

Voila un exemple des commandes nécessaire pour la constitution de fichiers de suivi à partir de deux séries d'images Bleue et Rouge. On suppose que la référence doit être fabriqué à partir d'une image compositée.

- Constitution des StarList Bleue et Rouge, à partir des images composées compB.fits et compR.fits. (*compB/R.fits* → *compB/R.ref*). L'option `-n` inhibe l'écriture du fichier de suivi et `-w` active la sauvegarde sur fichier de la StarList construite.
 

```
stars++ -wn -d marly_comp.d compB
stars++ -wn -d marly_comp.d compR
```
- On fusionne les deux StarList, et on recalcule les flux à partir de l'image composée (ou d'une bonne image non composée), en surécrivant les fichiers de référence fusionnés :
 

```
mergebr -d 2.5 compB.ref compR.ref comp_mrgB.ref comp_mrgR.ref

stars++ -wn -d marly_comp.d -i comp_mrgB comp_mrgB
stars++ -wn -d marly_comp.d -i comp_mrgR comp_mrgR
```
- On fabrique une référence avec des flux moyennés :
 

```
stars++ -rn -d marly_moy.d -mfs ref_moyB comp_mrgB imgB1 imgB2
imgB3 imgB4 imgB5
stars++ -rn -d marly_moy.d -mfs ref_moyR comp_mrgR imgR1 imgR2
imgR3 imgR4 imgR5
```
- On associe les deux listes d'étoiles, [Bleu → Rouge] et [Rouge → Bleu]. On surécrit les fichiers `ref_moyB.ref` et `ref_moyR.ref` :
 

```
assocbr++ -w -d marly_ass.d ref_moyB ref_moyR
assocbr++ -w -d marly_ass.d ref_moyR ref_moyB
```
- On traite les premières images et on crée les fichiers de suivi dans les deux couleurs, en écrivant la référence comme première mesure (Pas d'option `-0`) :
 

```
stars++ -r -d marly.d -o SBleu imgB1 imgB2 imgB3 imgB4 imgB5
stars++ -r -d marly.d -o SRouge imgR1 imgR2 imgR3 imgR4 imgR5
```
- On traite les images suivantes en rajoutant les mesures dans les fichiers de suivi (option `-a`) :
 

```
stars++ -ra -d marly.d -o SBleu imgB6 imgB7 imgB8 imgB9 imgB10
stars++ -ra -d marly.d -o SRouge imgR6 imgR7 imgR8 imgR9 imgR10
stars++ -ra -d marly.d -o SBleu imgB11 imgB12 imgB13 imgB14 imgB15
stars++ -ra -d marly.d -o SRouge imgR11 imgR12 imgR13 imgR14
imgR15
```

### 3.7 Reconstruction d'images synthétiques

`synth++` permet de synthétiser une image à partir d'une StarList. `synth++` construit cette liste d'étoiles en effectuant les différentes étapes de la reconstruction photométrique, (Détection, Ajustement de positions, Détermination de PSF, et ajustement de flux), suivies de `SyntheseImage(StarList *stl, ...)`. Les arguments et options de la commande `synth++` sont décrits ci-dessous :

```
Commande> synth++ -h
PEIDA Version 3.4 Revision 36 (V_Mar97) -- Sep 22 1997 16:34:11 cxx
```

Usage: synth [-pfn] [-cuts fmin,fmax,scale] [-d DataCard] NomIn No-  
mOut [NomSub]

-p : Fond de ciel constant = starList->Fond  
-f : Image du fond de ciel seulement  
-rp : Utilisation PosX/YRef  
-rpC : Projection sur courante de PosX/YRef  
-n : Image synthetisee sans bruit  
-cuts fmin,fmax,scale : fmin<=Flux<=fmax, FluxGen=Flux\*scale  
-d DataCard : Nom de fichier DataCard  
-psl FgPrtList,FgPrtSti: Impression de liste d'etoiles  
NomIn : Nom de l'image en entree  
NomOut : Nom de l'image synthetisee en sortie  
NomSub : (Optionnel) Image\_Sub = In-Out

L'option `-p` impose un fond de ciel constant, alors que `-f` permet de construire l'image des fond de ciel ajustés pour chaque étoile. `rp` impose l'utilisation des positions référence, et `-rpC` projète les positions référence sur l'image courante, à partir de la transformation géométrique. Si l'option `-n` est spécifiée, le programme ne rajoute pas du bruit aux valeurs de pixels obtenues. `-d` permet de spécifier le nom d'un fichier DataCard, avec les mêmes instructions que pour `stars++`. `-cuts` permet de définir une coupure sur le flux des étoiles, ainsi qu'un facteur multiplicatif pour renormaliser les flux. Si un troisième argument `NomSub` est spécifié, l'image des différences (`ImgIn - ImgSynth`) est calculée et écrite sous le nom `NomSub`.

## 4 Rééchantillonnage et images composites

Afin d'augmenter le rapport signal/bruit, on peut souhaiter sommer plusieurs images. Il est nécessaire de commencer par les réaligner, et donc les rééchantillonner (les pixels ne tombent pas forcément en face...).

```
Commande> rebin
PEIDA Version 3.4 Revision 36 (V_Mar97) -- Sep 22 1997 16:34:11 cxx
Usage rebin [OPTIONS] dest sources...
```

```
OPTIONS :
  -s scale (def=0.7)
  -m marge
  -q quart [klmn]
  -c      (aligfond)
  -z zfidufile
  -S      spline
```

La commande `rebin` accepte en argument une image destination (somme), et une série d'images source. Les images sources sont réalignées sur la première, puis rééchantillonnées, et sommées.

L'option `-s` permet de choisir un rééchantillonnage par splines plutôt que la méthode bilinéaire utilisée par défaut. Le temps de calcul est beaucoup plus long, mais le résultat est meilleur, surtout avec peu d'images (augmenter le nombre d'images gomme la distorsion introduite par la méthode bilinéaire) et si on choisit de ne pas suréchantillonner (le suréchantillonnage compense la perte intrinsèque d'information de la méthode bilinéaire). Même avec beaucoup d'images et du suréchantillonnage, on note un élargissement de PSF avec la méthode bilinéaire.

L'option `-s` permet de spécifier une échelle de suréchantillonnage. 0.7 est un bon compromis (images deux fois plus grandes).

L'option `-m` donne, en pixels, la marge à ajouter sur les côtés de l'image d'origine pour définir l'image finale (permet de limiter la zone perdue à cause des translations entre images).

L'option `-c`, vivement recommandée, permet de réajuster le fond de ciel d'une image à une autre. Il est quand même conseillé de choisir des images avec des fonds de ciel voisins.

L'option `-z zfidufile` permet de fournir une zone fiducielle (typiquement les zfidu standard donnant les défauts des CCD).

L'option `-q [klmn]` permet un découpage automatique par quarts de CCD : on ne relira qu'un quart (avec une petite marge) des images sources.

## 5 Réduction Flat-field et Offset

### 5.1 redimg

Pour effectuer la réduction d'une image FITS brute.

```
usage: redimg [-o offset] [-f flat] imabrut [imared]
      args par défaut dans FLATS/DATA_RED. Nom complet pour args explicites
```

### 5.2 reducall

Réduction de données fits automatiquement (cmv La Silla 14/07/96) On réduit automatiquement les images du type Eros2 : réduction de toutes les images brutes de \$DATA\_BRUT, flat-offset et offset sont choisis automatiquement dans FLATS, le nom de l'image réduite peut être créé automatiquement, l'image est écrite dans \$DATA\_RED.

```
usage: reducall <-C n> <-f c>
          [-d d] [-o z] [-p z] [-c r/b ou 0/1] [-s z]
OBLIGATOIRE -c : camera (0/1 ou r/b)
OBLIGATOIRE -f : filtre (b, r, ...)
            -o : objet (2 zeza)
            -p : champ (3 zeza)
```

```

-C : CCD      (0-7)
-s : sous-image
      si pas specifie, seul t est traite
-d : Date (jj/mm/aa ou amjj)
-S : numero de sequence

```

### 5.3 acqfiltmed

Pour créer un flat field ou un offset filtré à partir d'une liste de flats field ou d'offsets. Ce programme sert juste d'interface a la fonction FiltMed décrite dans Images++.

```

usage Main_FiltMed:
  -prt print_level (%d)
  -filt nom de l'image filtree (%s)
  -off  nom de l'offset (%s)
  -img liste des fichiers a filtrer (%s %s ...)
  -moy % des points centraux a moyenner (%f)
  -ncut % de points minimum pour filtrage (%f)
  -dyn lowbad,highbad en ADU (%f,%f)

```

## 6 Quelques programmes utilitaires

### 6.1 dirphotos

Programme permettant de rechercher des images FITS ayant un nom générique de type Eros 2.

```

usage: dirphotos [-O] [-F] [-o z] [-p z] [-c n/c] [-C n]
                [-s z] [-f c] [-B] [-R] [-t c] [-d d] [-T] [dir]
-O : Obscurite (-o ob)
-F : Flat      (-o ko)
-o : objet     (2 zeza)
-p : champ     (3 zeza)
-c : camera    (0/1 ou r/b)
-C : CCD      (0-7)
-s : sous-image
-f : filtre    (b, r, ...)
-t : traitement (b, r, ...)
-d : Date (jj/mm/aa ou amjj, '?' admis, ou +/-
d)
-T : trie

```

## 6.2 header

Lecture d'entête FITS avec possibilité de modification des clefs. Ne marche que pour des images de type U2 pour la partie modification des clefs.

```
usage: header fichier {[-q{ids} cle]} [-o fichier_out {-i/d/s cle valeur}].
```

## 6.3 rdheader

Lecture d'entete FITS. Le (les) noms de clefs fits peuvent être précisées.

```
usage: rdheader <nom_fichier_fits> [nom_clef_fits ...]
```

## 6.4 fitpacz

Fit de fonction de Paczinski monochrome ou polychrome, avec ou sans blending. Possibilité de tester les diverses configurations grace a une simulation interne.

```
usage: fitpacz -f nom_data
  -h help detaille
  -u u0,u0min,u0max,stu0 (def -1,1,-1,0.1)
  -T t0,t0min,t0max,stt0 (def -1,1,-1,0.25)
  -t tau,taumin,taumax,stau (def -1,1,-1,0.25)
  -1 f1,f1min,f1max,stf1 (def -1,1,-1,0.05)
  -2 f2,f2min,f2max,stf2 (def -1,1,-1,0.05)
  -a bl1,bl1min,bl1max,stbl1 (def 1.,0.,1.5,0.05)
  -b bl2,bl2min,bl2max,stbl2 (def 1.,0.,1.5,0.05)
  -F bit (def 7)

  -M max_step,nstopmx (def 100,3)
  -C stop_chi2,eps (def 0.001,1.e-8)
  -S dt1,dt2,stdt,err simulation (def -5,-5,-0.1,1.)
  -s u0_mc,t0_mc,tau_mc,f1_mc,f2_mc,bl1_mc,bl2_mc gene MC
      (def -1.,-1.,-1.,-1.,-1.,1.,1.)
  -n npt_max (def 5000)
  -i niu0,nit0,nitau,nif,nit Initialisation auto (def 10,10,10,10,3)
  -l print,print_ini (def 0,0)

-----> Pilotage du fit: (-u,-T,-t,-1,-2,-a,-b,-F,-M,-C,-l
p: initialisation du parametre (u0,t0,tau,f1,f2,bl1,bl2)
pmin,pmax: (parametre fixe si pmax<=pmin)
step: (parametre fixe si = 0)
  stu0 = step en u0
  stt0 = step en fraction de tau (def=0.25*tau)
  sttau = step en fraction de tau (def=0.25*tau)
  stf1 /f2 = step en fraction de f1 /f2 (def=0.05*f1 /f2)
```

```

    stb11 /b12 = step en b11 /b12 (def=0.05)
bit : 1=fit_c1, 2=fit_c2, 4=fit_c1+c2, 8=fit_c1+c2_blend
-----> Pilotage de l'initialisation automatique: (-i)
niu0,nit0,nitau,nif: initialisation automatique des parametres
    du fit. Dans ce cas la variation se fait entre pmin et pmax
    definis avec -u,-T,-t,-1,-2. Si pmin>pmax on varie en pmax et pmin.
    - Si niu0 <=0 parametre fixe a la valeur u0 de -u (idem autres par)
nit = nombre d'iterations
-----> Pilotage de la simulation: (-S,-s)
dt1,dt2,stdt simulation de t0-dt1 a t0+dt2 par pas de stdt
    - si dt1,dt2,stdt < 0. alors simulation de t0-|dt1|*tau
      a t0+|dt2|*tau par pas de stdt*tau
    - err : l'erreur sur les flux est err*sqrt(flux)
-----> Structure du fichier de data: (-f)
i_%%d c_%%f temps_%%f flux_%%f erreur_flux_%%d flag_%%d
    ou i = numero du points, c = couleur (0,1,...)
    flag si <=0 point tue

```

## 6.5 erodate

Affiche la date au format EROS (amjj ou jours depuis 01/01/1990).

```
usage: erodate [-t] [-o deltaj] [jj/mm/aaaa]
```

Par défaut, la date est affichée au format amjj, par exemple 8c16 pour le 16 mars 1998. Le changement de jour se fait à midi, conformément à la norme EROS de date d'observation. L'option -t permet d'afficher le nombre de jours écoulés depuis le 01/01/1990. L'option -o permet de spécifier un décalage en nombre de jours : erodate -o -1 affiche la date de la veille.

## 6.6 findimgt

Trouve la transformation géométrique entre deux images.

```
usage: findimgt [-m methode] [-r rayon] [-s seuil] [-t taille ssimage]
              [-x orgx] [-y orgy] image1 image2
methode : 1 - circrfix
          2 - peida FindStars
          3 - daofind
          4 - corrfind bary
```

Le programme affiche la transformation et la translation entre les images image1 et image2. On peut spécifier la méthode de reconstruction d'étoiles, ainsi que ses paramètres (rayon et seuil). La méthode est appliquée sur une sous-image de taille et position paramétrables.

## 6.7 imgdump

Affiche en ASCII le contenu d'une portion d'image

```
usage: imgdump x1 x2 y1 y2 image
```

Affiche la portion entre (x1, y1) et (x2, y2) en ascii (x y pixel).

## 6.8 isole

Extrait une sous-image

```
usage: isole x1 x2 y1 y2 imagein imageout
```

imageout contiendra la sous-image entre (x1, y1) et (x2, y2) de image.

## 6.9 mosaic

```
usage: mosaic [-b binning, def=8] [-f] img result
  img est une des images. Les autres sont cherchees
  au meme endroit
```

Les images d'origine doivent avoir un nom standard EROS 2. Il suffit de spécifier un des CCD, et result contiendra une image de toute la mosaïque, rebinnée par défaut d'un facteur 8. L'option -t permet de réajuster le fond de ciel entre les sous-images.

## 6.10 Utilitaires pour suivis

Les programmes suivants permettent de manipuler globalement les fichiers de suivi :

- **transmi** : Conversion des fichiers de suivi en ou depuis le format MI (Machine Indépendant). L'utilisation de ce programme n'est en principe plus nécessaire.

```
cmd> transmi -h
```

```
Transmi_Erreur: Arguments pas specifie !
```

```
Usage : transmi TOMI/FRMI FileIn FileOut [FgAsc]
```

```
- TOMI/FRMI : Suivi ->/<- Format de transfert
```

```
- FgAsc : 0 (X), 1(X->IBM-VM), 2(IBM-VM->X)
```

- **checkfsv** : Permet de vérifier l'intégrité du fichier de suivi. Le code de retour est non nul si le fichier est illisible ou corrompu. Ce programme permet aussi de tronquer le fichier de suivi (supprimer des mesures en fin de fichier).

```
cmd> checkfsv -h
```

```
Usage: CheckFSV [-restore] NomSuivi
```

```
ou CheckFSV [-swapdefaut] NomSuivi
```

```
ou CheckFSV [-tronq nmes] NomSuivi
```

```
Rc=0 -> Fichier OK, RC > 0 = Nb de mesure OK
```

```
-restore : Tronquer le fichier au nombre de mesure OK
```

```
-swapdef : Changer la valeur de Flg Swap a SWAPDEFAULT
```

- tronq nmes : Tronquer le fichier au nombre de mesure nmes
- **concatfsv** : Permet de concatener plusieurs fichiers de suivi en un seul. Tous les fichiers doivent avoir le même format. Seul le GlobInfo et le StarInfo du premier fichier sont copiés dans le fichier de suivi de sortie. Il est possible de supprimer certaines mesures à l'aide de l'option `-killmes`. `concatfsv` peut être utilisé pour réécrire un fichier dans le format natif du byte-swap de la machine.

```
cmd> concatfsv -h
```

```
Usage: ConcatFSV [-killmes M1,M2,...] FSVOut FSVIn1 FSVIn2 [FSVIn3 ...]
Specification des mesures a supprimer: Mi = NumFile:NumMes
1<=NumFile<=NFile (MaxNFile=100) 1<=NumMes<=NbMesFile
```

## 7 Le contenu des fichiers de Suivi

Les fichiers de suivi contiennent des courbes de lumière obtenues à partir de la mesure de luminosité des étoiles sur une série d'images.

$$\left\{ \begin{array}{ll} (t_1, t_2, \dots, t_m, \dots) & m = 1, \dots, M \\ (Star)_s & (\phi_1, \phi_2, \dots, \phi_m, \dots)_s \quad s = 1, \dots, N \end{array} \right\}$$

$t_m$  Informations ou mesures se rapportant globalement à une image (*TIMEINFO*)

$(Star)_s$  Informations indépendantes d'une image particulière, liées à une étoile.

(*STARINFO*)

$(\phi_m)_s$  Mesures obtenues sur une image, pour chacune des étoiles. (*MESURE*)

La structure des fichiers de suivi et les fonctions permettant de les manipuler (`libfsv.a`) sont décrites dans une note séparée. Nous décrivons ici le contenu des fichiers de suivi EROS-2, correspondant aux structures (C, C++) du fichier `fsvst.h` et aux fichiers de suivi de **TypeSuivi=10**.

### 7.1 GlobInfo

La section GlobInfo regroupe les informations communes à toutes les étoiles et toutes les mesures du fichier. La structure ci-dessous représente les informations GlobInfo pour les fichiers de suivi EROS-2. La fonction `PrtGlobInfo (GLOBINFO *glinf, int lp)` permet d'imprimer le contenu de GlobInfo, avec un niveau d'impression `lp = 0..2`.

```
struct GlobInfo {
    int_4 NumCCD;
    int_4 Couleur;
    int_4 NumChamp;
    int_4 IRes[4];
    float AlphaHr;
```

```

float DeltaDg;
float GainCCD;
float FRes[2];

double PolxyBR[3][2];
double PolxyRB[3][2];
double LargX[2],LargY[2],MidX[2],MidY[2];
double XMin[2],XMax[2],YMin[2],YMax[2];
int_4 DegPolTG;
};
typedef struct GlobInfo GLOBINFO;

```

**NumCCD** : Le numéro de CCD

**Couleur** : Filtre, Caméra

**NumChamp** : Codage de zone et numéro de champ EROS-2

**IRes[4]** : Réservés pour une utilisation ultérieure

**AlphaHr, DeltaDg** : L'ascension droite (heures décimales), déclinaison (degrés décimales) du centre du champ.

**GainCCD** : Gain du CCD, en e/ADU

**FRes[2]** : Réservés pour une utilisation ultérieure

**PolxyBR...DegPolTG** : Transformations géométriques de calage des listes d'étoiles dans les deux couleurs ( $Coul1 \rightarrow Coul2, Coul2 \rightarrow Coul1$ ). La fonction `GlobInfoToTransf( GLOBINFO *gli, TRANSFO *t1, TRANSFO *t2)` permet de construire les structures `TRANSFO  $T_{1 \rightarrow 2}$`  et  `$T_{2 \rightarrow 1}$`  à partir d'une structure `GLOBINFO`.

## 7.2 StarInfo

La section `StarInfo` des fichiers de suivi EROS-2 correspond à la structure suivante. Son contenu peut être imprimé par `PrtStarInfo( STARINFO *sti, int n, int lp)`, avec un niveau d'impression `lp= 0,1`.

```

struct StarInfo {
    int_4 NumEt, XRef;
    float FlxMean,FlxSig;
    float FluxRef;
    float XPos,YPos;
    float DisMin,DisM2,DisM2R;
    uint_2 FgRef;
    uint_2 NbVois,Voisin[8];
};
typedef struct StarInfo STARINFO;

```

**NumEt** : Numéro d'étoile ( $1 \dots N_*$ )

**XRef** : Numéro de référence croisée de l'étoile associée dans l'autre couleur ( $1 \dots N_{\star}^s$  si associée)

**FlxMean, FlxSig** : Flux moyen et dispersion de la distribution des flux le long de la courbe de lumière. Ces variables ne sont pas calculées par `stars++` et devront être calculées et remplies pour des fichiers de suivi complets, par un programme de filtrage.

**DisMin, DisM2, DisM2R** : Distance d'association et deuxième étoile la plus proche. `DisM2R` n'est pas remplie.

**FgRef** : Les flags référence de l'étoile, voir `star.h`.

**NbVois, Voisin[8]** : Nombre et numéro des huit premières voisines.

### 7.3 TimeInfo

En plus de la structure `TimeInfo`, une structure réduite `TimeInfoU` (U : User) permet une manipulation plus aisée des informations `TimeInfo` les plus usuelles. `DecodeTim(TIMEINFO *time, TIMEINFOU *timeu)` permet d'extraire la structure `TimeInfoU` à partir d'un `TimeInfo`. Les fonctions `PrtTimeInfo(TIMEINFO *tim, int n, int lp)` et `PrtTimeInfoU(TIMEINFOU *tim, int n, int lp)` permettent d'imprimer le contenu de ces structures, avec un niveau d'impression `lp=0 .. 3`.

```
struct TimeInfo {
    int_4    NumPhoto;
    int_4    TStart, Expose;

    int_4    NbOkPS, NbOkGF;
    float    Fond, SigFond;

    float    SigX, SigY, Rho;
    float    SgX, SgY;
    float    PxSiz[2], CrVal[2];
    float    DelX, DelY;

    double   PolxyRC[3][2];
    double   PolxyCR[3][2];
    double   LargX[2], LargY[2], MidX[2], MidY[2];
    double   XMin[2], XMax[2], YMin[2], YMax[2];
    int_4    DegPolTG;

    float    Absorption;
    float    AirMass;

    int_4    FgCalib;
    float    Calib[8];
}
```

```

float  Resol[3][6];
float  PFitErr[2][5];

uint_2 PSF1D, PSF2D;
uint_2 Version, Revision;
int_4  DateHeure;
uint_2 TSid, ObsId;
int_4  DTU;
float  FRes;
};
typedef struct TimeInfo  TIMEINFO;

struct TimeInfoU {
  int_4  NumPhoto;
  int_4  TStart, Expose;
  float  Fond, SigFond;
  float  SigX, SigY, Rho;
  float  DelX, DelY;
  float  Absorption, AirMass;
  int_4  FgCalib;
  float  Calib[8];
  float  PFitErr[2][5];
};
typedef struct TimeInfoU  TIMEINFOU;

```

**NumPhoto** : Numéro de photo EROS-2

**TStar, Expose** : Le moment (date) et la durée d'exposition en secondes. TStart est exprimé en nombre de secondes écoulées depuis le 1<sup>er</sup> Janvier 1990. On peut exprimer ainsi des dates comprises entre 1930 - 2050.

**NbOkPS, NbOkGF** : Nombre d'étoiles pour lesquelles l'ajustement 1D (NbOkPS), et l'ajustement de flux (NbOkGF) ont réussi.

**Fond, SigFond** : Valeur du fond de ciel et de sa dispersion sur l'image.

**SigX, SigY, Rho** : Les paramètres de la PSF déterminé sur l'image.

**SgX, SgY** : Paramètres de largeur de PSF déterminé par l'ajustement 1D.

**PxSiz[2], CrVal[2]** : Taille des pixels de l'image (en mm) et l'offset de position du pixel (0,0).

**DelX, DelY** : Décalage moyenne de l'image par rapport à la référence.

**Absorption, AirMass** : L'absorption moyenne  $\langle F_{cal}/F_{brut} \rangle$  et l'épaisseur d'air.

**PolxyRC, PolxyCR, ... DegPolTG** : Transformations géométriques de calage  $T_{Ref \rightarrow Cur}, T_{Cur \rightarrow Ref}$ .

La fonction `TimeInfoToTransf(TIMEINFO *tminf, TRANSFO *t1, TRANSFO *t2)` permet l'extraction de deux structures TRANSFO à partir de t TIMEINFO.

**FgCalib, Calib[8]** : Paramètres de calibration.

**Resol[3][6]** : Résolutions photométrique pour 6 intervalles de flux. Resol[1] est la valeur moyenne du FluxRef dans l'intervalle, Resol[2], [3] sont la valeur moyenne et l'écart type de la distribution de  $(F_{cat} - F_{Ref})$ .

**PFitErr[2][5]** : Paramètres de l'ajustement des erreurs externes, permettant de corriger les erreurs provenant de l'ajustement des flux de PEIDA.

**PSF1D, PSF2D** : Type de PSF 1D, 2D utilisé.

**Version, Revision** : Numéro de version et de révision de PEIDA utilisé pour traiter cette image. Version = 1000\*Peida\_Version : 3.4 → 3400, Revision = Peida\_Revision.

**DateHeure** : codage de la date et de l'heure (légale) de prise du cliché.

(Utilise la fonction LibsUtil : DateTstoInt(const char \*date, const char \*ts) et les mots-clé de l'entête FITS : DATE-OBS TM-START )

**TSid** : codage de l'heure sidérale de prise du cliché (précision 2 sec).

(Utilise la fonction LibsUtil : DateTstoInt(const char \*date, const char \*ts) et des mots-clé de l'entête FITS : DATE-OBS TS-START )

**ObsId** : code de l'observatoire (ESO-LaSilla = 1).

**DTU** : codage de la date et de l'heure TU de prise du cliché.

(Utilise la fonction LibsUtil : DateTimetoInt(const char \*date, const char \*ts) et du mot-clé de l'entête FITS DATE-EXP )

**FRes** : Réservés pour un usage ultérieur.

## 7.4 Mesure

Afin de diminuer l'espace occupé par les informations Mesure, celles-ci sont écrites sous forme comprimée dans le fichier de suivi. La structure MESURE est l'image de la section Mesure des fichiers de suivi d'EROS-2. La structure MesureU (U : User) correspond aux informations décodées et peut être obtenue à partir de Mesure à l'aide de la fonction DecodeMes( MESURE \*mes, MESUREU \*mesu). Les fonctions Prt-Mesure( MESURE \*mes, int n, int lp) et PrtMesureU( MESUREU \*mesu, int n, int lp) permettent d'imprimer le contenu de ces structures.

```
struct Mesure {
    float Flux, Xi2;
    int_2 ErrFlux, Fond;
    int_2 Flx0, Fnd0, Xi20;
    int_2 S9Pix, PixMax;
    uint_2 X, Y;
    uint_2 SigX, SigY;
};
typedef struct Mesure MESURE;

struct MesureU {
    float Flux, FluxB;
```

```

float  Xi2,ErrFlux,ErrFluxB;
float  Fond;
float  Flx0,Fnd0,Xi20;
float  S9Pix,PixMax;
float  X,Y;
float  SigX,SigY;
};
typedef struct MesureU  MESUREU;

```

**Flux, FluxB** : Flux calibré et flux brute. La structure Mesure ne contient que la valeur du flux brute. Le flux calibré est calculé par la fonction `Calibre_F_E(MESUREU *mesu,TIMEINFOU *timu)`.

**Xi2** : Le  $\chi^2$  de l'ajustement de flux, fond. Négatif si l'ajustement a échoué.

**ErrFlux, ErrFluxB** : Erreur sur le flux calibré et le flux brut.

**Fond** : La valeur du fond de ciel, obtenu par l'ajustement du flux pour l'étoile.

**Flx0, Fnd0** : La valeur du flux, fond et  $\chi^2$  de l'ajustement 1D.

**Xi20** :  $\chi^2$  de l'ajustement par projection 1D. Négatif si échec.

**S9Pix, PixMax** : Somme des pixels sur un pavé 3\*3 centré sur l'étoile (fond soustrait) et la valeur du pixel maximum (fond non soustrait).

**X, Y** : Position de l'étoile ajustée sur l'image courante (Fit 1D).

**SigX, SigY** : Paramètres de largeur de l'étoile calculés par le Fit 1D.

## 8 Programmes de manipulation des fichiers de suivis

### 8.1 Utilitaires de lecture des suivis

`scansv` permet de lire les fichiers de suivi, et d'en imprimer tout ou partie du contenu. Il permet aussi le calcul des résolutions photométriques. Ce programme utilise la variable d'environnement `STARS_SUIVI`. `ntfsv` est une version de `scansv` compilé avec `HBOOK`, permettant la création de NTuples, suivant les options de la ligne de commande. Les arguments et options de la ligne de commande sont identiques entre `scansv` et `ntfsv`. L'aide en l'aide peut être obtenu par `scansv/ntfsv -h`:

```

Usage: Ntfsv [-Flag] NomSuivi
  -inq : Marque une pause apres lecture TimeInfo
  -act prfg,clfg,ntfg : Flags de controle
  -prt prt1 : Niveau d'impression (0->3)
  -his nbin,MinLog,MaxLog : Binning histo resolution
  -mes m1-m2 : Traitement des mesures m1 a m2
  -numet n1-n2,incn : Traitement des etoiles n1 a n2 (C1)
  -xpos min-max : Coupure min <= XPosEtoile <= max (C2)
  -ypos min-max : Coupure min <= YPosEtoile <= max (C3)
  On fait C1 & C2 & C3, Valeur par default = Tout

```

```

prfg (PrintFlag) : 0 , 1= Glob+TimeInfo
                2= +StarInfo, 3= +MesuresU (=default), 4= +Mesures
clfg (Flag Courbe de lumiere) : 0(=default),1
ntfg (Bits Flag Ntuple): B0(=default) TimeInfo, B1 Resol, B2 Mesures

```

L'option `-act` permet d'activer l'impression des différentes composantes de suivi (`prfg` non nul, la création de ntuples de courbes de lumière pour les étoiles sélectionnées. Chacun des bits de `ntfg` active la création d'un type de ntuple. Les noms des variables des ntuples sont similaires à celles des structures présentées dans le paragraphe précédant.

- Bit 0 (=1) Ntuple TimeInfo (Num=300) avec une entrée par image (TimeInfo). Les variables `FlxMi`, `DFlxi`, `SigFlxi`,  $i=1..6$  correspondent aux trois termes de résolution pour chacun des intervalles de flux (`TimeInfo.Resol[0,1,2][i]`).
- Bit 1 (=2) NTuples des résolutions par intervalles de flux, pour chacune des mesures (images) traitées (Num=500+m, m=numéro de mesure). Le nombre d'intervalles et les bornes en  $\text{Log}_{10}(F_{ref})$  sont déterminés par l'option `his`. Pour chaque mesure, il y a une entrée dans le ntuple correspondant pour chaque intervalle de flux. `NumBin`, `XBin`, `NSt`, `FlxMoy`, `DFlx`, `SigFlx`, `DFlxErr`, `SFlxErr`, `NStRef`, `FRefMoy` sont les noms des variables de ces ntuples. `NumBin`, `XBin` correspondent au numéro d'intervalle et l'abscisse ( $\text{Log}_{10}(F_{ref})$ ) du point milieu de l'intervalle. `NStRef`, `NSt` sont le nombre total d'étoiles dans cet intervalle (Ref) et le nombre de celles ayant une mesure en flux correcte. `FlxMoy`, `FRefMoy` sont les valeurs moyennes des flux et des fluxRef des étoiles dans chaque intervalle. `DFlx`, `SigFlx` sont la valeur moyenne et la dispersion de la distribution  $F_{Ref} - F_{cal}$  et `DFlxErr`, `SFlxErr` celles de la distribution  $F_{Ref} - F_{cal}/ErrFit$  pour chaque intervalle.
- Bit 2 (=4) Ntuple (Num=400) pouvant contenir jusqu'à un maximum de dix mesures pour les étoiles sélectionnées. `Xi2m` et `Xi m` correspondent respectivement à `MesureU.Xi2` et `MesureU.Xi20`, pour la mesure `m`.

L'option `-prt` permet de modifier le niveau d'impression des différentes structures (`StarInfo`, `TimeInfo`, ...). `-mes` permet de traiter uniquement les mesures numéro `m1` à `m2`. Les options `-numet`, `-xpos`, `-ypos` permettent de sélectionner un sous ensemble d'étoiles à étudier. `incn` représente le pas de la boucle sur les numéros d'étoiles qui sont traitées.

## 8.2 Dump de la structure TimeInfo, timf2asc

Transcription ASCII des variables les plus importantes de la structure `TIMEINFO` d'un fichier de suivi. Possibilité de ne sélectionner que certaines images à partir de coupures sur ces paramètres.

```

timf2asc [options] NomSuivi
        [-t min,max coupure en temps (jours %f)]

```

```

[-e min,max  coupure en temps d exposition (%d)]
[-f min,max  coupure en fond de ciel (%f)]
[-s min,max  coupure en sigma fond de ciel (%f)]
[-x min,max  coupure en sigma X (%f)]
[-y min,max  coupure en sigma Y (%f)]

[-r min,max  coupure en Rho (%f)]
[-X min,max  coupure en decalage en X (%f)]
[-Y min,max  coupure en decalage en Y (%f)]
[-a min,max  coupure en absorption (%f)]
[-m min,max  coupure en masse d air (%f)]
[-n sortie ASCII pour ntuple (sans chaine de char)]
[-l numero de mesure de 1 a nmes (def 0 a nmes-1)]

```

### 8.3 Association de fichiers de suivi, suiviass

Pour associer les etoiles de deux fichiers de suivi.

```

usage: suiviass <-n> <-l> -d datacard suivil suivi2
-d datacard: fichier datacard type stars++
-n : pour test sans re-ecrire le fichier de suivi
-l : pour ne faire l'alignement geometrique que de 1 -> 2
      seul suivil est update
suivil : nom complet du premier  fichier de suivi
suivi2 : nom complet du deuxieme fichier de suivi

```

### 8.4 Ré-alignement photométrique, recalib++ et ntrelib++

Pour re-aligner photométriquement les flux bruts des images d'un fichier de suivi par rapport au flux de référence décrit dans STARINFO. Seule est changée la valeur des coefficients de calibration et de résolution de la structure TIMEINFO. Il est aussi possible de changer les valeurs des flux de référence, soit en refaisant une moyenne des flux bruts d'un choix d'images du fichier de suivi, soit en relisant un fichier contenant une starliste. ntrelib++ exécute le même travail que recalib++ en fournissant en plus des n-uples de vérification. Les calibrateurs de Peida 0, 1 (10 et 11), 8 et 9 sont disponibles.

```

recalib++
<-n NomSuivi (%s)>
[-w flag d'écriture]
[-mes numes1,numes2 (%d,%d) de 1 a N] (
[-prt print level code: PrtLevel*10+lp (%d)]
[-dbg numes_a_debugguer (%d)] (
[-xran Xmin,Xmax (%f,%f)]

```

```

[-yran Ymin,Ymax (%f,%f)]
[-typ typcal (%d) choix -1(no cal),0,1ou10,11,8ou80,9ou91 [def=9]
[-refw NomStl (%s) pour ecrire la starliste utilisee
[-refr NomStl (%s) pour lire la starliste a utiliser
      a la place de StarInfo pour calibrer]
[-refm a,b,c-d,e,f (..%d..) pour refaire un flux de reference
      moyenne avec les mesure a,b... de 1 a N]
[-refma pour corriger les images par l'absorption si refm]

```

Calib 91

```

[-nbril NKBRIL,NGBril,Percent_Clean_Br (%d,%d,%f)]
[-dpolm DPolM (%d)]
[-fmin FBcutMin (%f)]
[-pclean Percent_Clean (%f)]

```

Calib 0,1,10,11,8,80

```

[-parc MinNbSt,FracLow,FracHigh (%d,%f,%f)]

```

Calib 0,1,10,11

```

[-xi2 MaxXi20,MaxXi2 (%f,%f)]

```

Calib 8,80

```

[-nsig npass,nsig1,nsig2 (%d,%f,%f)]

```

Calib 91 ..ET.. DoResol

```

[-int Nst_Bin_DoResol,Nst_Bin_DoResol0,NBin_DoResol0 (%d,%d,%d)]

```

## 8.5 Création d'un fichier de suivi réduit, svtrsel.

Pour écrire un fichier de suivi réduit à partir de plusieurs fichiers de suivi et en sélectionnant sur les étoiles. Le cas de séries de fichiers de suivi bicouleurs est traité ainsi que les problèmes de références croisées. Un exemple de fichier de datacard se trouve dans svtrsel.data.

svtrsel:

```

-d nom_fichier_datacard (def=NULL)
-l nom_fichier_suivi_1 (def=NULL)
[-2 nom_fichier_suivi_2] (def=NULL)
[-l niveau_de_debug] (def=0)

```

svtrsel.data:

```

-----
C ceci est une carte commentaire

```

```

C liste des fichiers des suivi en entree

```

```

C... couleur nom_du_fichier_de_suivi

```

```
1 /home/usr223/mnt/cmv/Run/ReSv/totoR1.suivi
2 /home/usr223/mnt/cmv/Run/ReSv/totoB1.suivi
```

```
1 /home/usr223/mnt/cmv/Run/ReSv/totoR2.suivi
1 /home/usr223/mnt/cmv/Run/ReSv/totoR3.suivi
```

```
2 /home/usr223/mnt/cmv/Run/ReSv/totoB2.suivi
```

C Liste des etoiles a copier

C numero etoile de 1 a n, numero de couleur de 1 a 2

C e"couleur" num\_lere\_etoile num\_derniere\_etoile

C ou e"couleur" num\_etoile

C... e1 et1 et2 : on copie pour la couleur 1 de et1 a et2

C... e1 et1 : on copie pour la couleur 1 l'etoile et1

C... e2 et1 et2 : on copie pour la couleur 2 de et1 a et2

C... e2 et1 : on copie pour la couleur 2 l'etoile et1

e1 1

e2 1 100

e1 1000 1050

e2 4990 5060

e1 5000 5050

e1 15000

-----

## 9 nbsread (C et C++)

**Lib :** libnbsread++.a et libnbsread.a  
**Include :** nbsread.h nbgene.h

### - nbsread

C'est un programme qui permet de lire des fichiers de suivi pour 2 couleurs associees. Il fournit des points d'entrees permettant a l'utilisateur de piloter une analyse sans devoir s'occuper de la machinerie relative a la lecture de fichiers de suivis.

nbsread nbsread.dataccd.Exemple

Les noms des fichiers de suivi a lire doivent etre mis dans un fichier dont un descriptif est donne dans le fichier *AnaSuiv/nbsread.dataccd.Exemple*. La lecture de plusieurs fichiers de suivi pour plusieurs CCD dans 2 couleurs est permise. Ce fichier permet aussi de selectionner des etoiles par leur numero ou un intervalle de numeros, le programme renvoie *FgSelSt* non nulle si l'etoile a ete selectionnee par les datacards.

### - Structure des donnees

Le programme lit les etoiles les unes apres les autres. Pour chaque etoile il remplit diverses structure ou tableau de structures permettant l'analyse de sa courbe de lumiere (la description des structures se trouve dans *fsvst.h*) :

```
EXTERN GLOBINFO globu : decodage de GLOBINFO
EXTERN STARINFO staru[NCOULMX] : decodage de STARINFO
EXTERN int_4 nmes[NCOULMX] : nombre de mesures pour le CCD courant
                           et les couleurs 0 et 1
EXTERN TIMEINFOU *timeu[NCOULMX] : tableau de structures TIMEINFOU
EXTERN MESUREU *mesu[NCOULMX] : tableau de structures MESUREU
EXTERN double *date[NCOULMX] : tableau des dates en jours
EXTERN float *ampli[NCOULMX] : tableau des amplifications montecarlo
EXTERN int_4 nmesure[NCOULMX] : nombre de mesure valides
                           pour le CCD courant
EXTERN int_4 *indexu[NCOULMX] : tableau des images valides
                           rangees chronologiquement
EXTERN STARCALU starcal : structure de magnitude absolue
EXTERN STARCUTU starcut : structure de coupure sur les magnitudes
                           et les couleurs (permet de ne pas lire
                           certaines etoiles pour gagner du temps)
```

Les tableaux de structures ont *nmes[0]* / *nmes[1]* elements correspondant au nombre total d'images pour les couleurs 0 et 1 dans les fichiers de suivi d'un CCD.

## - Les points d'entree utilisateurs et structure de nbsread

Ce sont les routine commençant par la lettre *U...()* ayant un nom en majuscule.

```
Initialisation du job {

void UINIT(void); <***** USER
Initialisation montecarlo

Boucle sur les CCD a lire {

  ---> Lecture de GLOBINFO et de TIMEINFOU pour un CCD
void UDATCLEAN(int coul); <***** USER
Classement des images valides par dates croissantes
void UINITCCD(void); <***** USER
Initialisation montecarlo pour le CCD courant

  Boucle sur les etoiles a lire {

    ---> lecture de MESUREU
    Calcul des magnitudes absolues
    Coupure sur les magnitudes absolues et la couleur des etoiles
    Generation montecarlo pour chaque etoile
      (eventuellement plusieurs generation sur la meme etoile)
    void UEVT(void); <***** USER

  } Fin de la boucle sur les etoiles

  void UENDCCD(void); <***** USER

} Fin de la boucle sur les CCD

void UEND(void); <***** USER
Fin du montecarlo, impression des statistiques des generations

} Fin du job
```

### - void UINIT(void)

Tout ce qui ne doit etre fait qu'une fois par job definitions de parametres generaux, ouverture hbook etc...

### - UDATCLEAN(int icoul)

Permet de tuer des images Si `date[icoul][imes]` est mise a *GRAND2* l'image correspondante ne sera pas prise en compte dans la sequence en temps. Si `date[icoul][imes]` est mise a *GRAND2\_KILL*, l'image correspondante ne sera ni decodee ni prise en compte (structure *MESUREU* mis a zero). Ceci permet d'eliminer des images ou il y a eu des erreurs de codage qui conduisent a un core dump.

#### - Classement et elimination des images

Les images sont ensuite rangees chronologiquement en ne tenant pas compte des images tuees. Pour les images ayant la meme date, seules celles de plus grand indice sont prises en compte. Le tableau `indexu[icoul][i]` est fournit : il contient les indices des images valides rangees chronologiquement et sans doublons. L'indice de `indexu` varie de 0 a `nmesure[icoul]` valeur qui represente le nombre d'images utilisables pour l'analyse. Par exemple, l'indice de la 5ieme image valide de la couleur bleue (reference secondaire) est :

```
j = indexu[1][5];
```

et l'acces au flux brut est donne par :

```
fb = mesu[1][j].FluxB;
```

Toutes les images sont encore accessibles directement, par exemple le flux brut de la 7ieme image (la 7ieme des fichiers de suivi sans aucun choix ni classement chronologique) est, pour la reference primaire (couleur rouge) :

```
fb = mesu[0][7].FluxB;
```

#### - UINITCCD(void)

C'est un point d'entree a la fin de l'initialisation du CCD courant. A ce niveau toutes les informations de *TIMEINFOU*, les notions d'images valides et le classement chronologique sont connus. Les variables *TFIRST* et *TLAST* representent respectivement la valeur de `date[icoul][i]` la plus petit et la plus grande pour les images valides. A ce niveau peuvent etre faites des initialisation montecarlo propres a un CCD.

#### - Calcul des magnitudes absolues

Par default, il n'y a pas de calcul de magnitudes absolues. L'utilisateur peut fournir une routine de prototype :

```
void mycalib(float, float, STARCALU *);
```

puis la connecter (dans *UNIT* ou *UINITCCD*) au prototype :

```
void (*Calibration_Absolue)(float, float, STARCALU *, void*);
```

en écrivant : `Calibration_Absolue = mycalib;`. Cette routine sera appelee automatiquement pour chaque etoile avec les arguments :

```
Calibration_Absolue(staru[0].FluxRef, staru[1].FluxRef, &starcal, uparms);
```

et remplira la structure *starcal*. Dans sa propre routine, l'utilisateur peut remplir des magnitudes *B,V,R,I* ainsi que le type de l'étoile (*SP=1* ou *GR=2* ou *indefini=0*) et son rayon en unités de rayon solaire. Le remplissage de toutes ces valeurs n'est pas obligatoire, il dépend de ce qui doit être fait après, soit par l'utilisateur dans son analyse, soit par la simulation montecarlo (l'effet de taille finie par exemple a besoin du rayon de l'étoile). Pour passer des paramètres supplémentaires à la fonction, il faut initialiser le pointeur par l'assignation :

```
Calibration_Absolue_UParms = &user_data_block
```

et l'adresse sera passée comme 4ème argument de *Calibration\_Absolue*. *user\_data\_block* doit être déclarée comme variable *globale* dans le fichier ou comme variable *static* dans la routine où est faite l'assignation.

#### - **Coupage sur les magnitudes absolues et la couleur des étoiles**

Pour des raisons de rapidité, l'utilisateur peut couper sur la magnitude absolue ou la couleur de l'étoile en remplissant la structure *starcut*. Le temps de lecture de l'étoile est économisé, et le point d'entrée *UEVT()* n'est pas appelé.

#### - **Generation montecarlo pour chaque étoile**

L'effet de microlentille est simulé sur chaque étoile (ou plusieurs fois par étoile). Une description détaillée des possibilités de simulation est donnée plus loin.

#### - **UEVT(void)**

C'est l'endroit où l'utilisateur met le code d'analyse et de traitement des courbes de lumières. À ce niveau les étoiles sélectionnées dans le fichier datacard par une carte *@SELET* sont identifiées par le flag *FgSelSt>0*. Les variables de niveau d'impression, debug, etc... sont accessibles :

```
iet[NCOULMX] : numero de l'étoile (ref ou ref sec)
FgSelSt : flag de selection
idebug[4] : niveau de debug [iet1,iet2,ccd,niveau de debug]
ffprt : niveau d'impression
etc...
```

#### - **void UENDCCD(void)**

Point d'entrée à la fin du traitement d'un CCD. Il permet de fermer ou d'imprimer toute structure relative à un CCD.

#### - **void UEND(void)**

Point d'entrée à la fin du job. C'est l'endroit pour désallouer les tableaux et pour imprimer les conclusions du job

### - Arguments de l'exécutable

```
[-h] : help tres tres detaille  
[-et et1,et2,etinc] : numeros des etoiles a traiter  
[-prt niv] : niveau de print  
[-dbg et1,et2,ccd,niv] : debug de certaine etoiles/ccd  
[-kill mes1 mes2 mes3 ...] : pour tuer les images mes1...
```

-kill permet de tuer les images mes1,mes2,mes3,... de *chaque* fichier de suivi pour un CCD et une couleur. C'est particulierement utile si, par exemple, la premiere image de chaque fichier de suivi est l'images de reference : on s'en debarasse en indiquant :

```
-kill 1
```

Pour les arguments de nbsread, conformement aux conventions prises pour les routines de lecture de fichiers de suivi, les numeros d'etoiles et d'images vont de *1 a nmes[]* (et non pas de *0 a nmes[]-1*).

### - Comment creer ses propres routines utilisateur

Il faut ecrire un fichier dans lequel sont definies les point d'entree utilisateurs : *UINIT()*, *UDATCLEAN(int)*, *UINITCCD()*, *UEVT()*, *UENDCCD()* et *UEND()*. Le nom du fichier doit obligatoirement s'appeler

```
nbsreadu_myroutine.cc (meme si on compile en C)  
nbsreaduF_myroutine.f pour la partie Fortran (si il y a des hbook)
```

myroutine est un nom choisi librement par l'utilisateur. La compilation et le link s'effectue en executant :

```
make -f Makeus++ nbsread_myroutine (si on travaille en C++)  
make -f Makeus nbsread_myroutine (si on travaille en C )
```

Un exemple detaille a ete sauve dans la base CVS :

```
nbsreadu_squel.cc , nbsreaduF_squel.f
```

Pour l'executer et regarder ses impressions, copiez *nbsreadu\_squel.cc* et *nbsreaduF\_squel.f* dans un de vos repertoires, et recuperez dans la base CVS *AnaSuiv/Makeus++* (ou *AnaSuiv/Makeus* si vous voulez compiler et linker en C). Compilez et linkez :

```
make -f Makeus++ nbsread_squel  
(ou make -f Makeus nbsread_squel)
```

Creez un fichier *fichier.dataccd* contenant des fichiers de suivi. Executez *nbsreadu\_squel* en tapant la ligne de commande suivante :

```
nbsreadu_squel fichier.dataccd -prt 1 \  
-mc 2 -u0 0.01,0.5 -t0 1,-1,-1 -tau 2,10,-1 -tf 2,0,2 -bl 1
```

(cette procedure est egalement decrite aussi dans *nbsreadu\_squel.cc*)

### - Initialisation du montecarlo et generation

L'initialisatuon peut etre effectuee, soit par arguments de l'executable :

```
[-mc ngenstar]
[-u0 U0MinI,U0MaxI,U0SimI]
[-t0 T0MinI,T0MaxI,T0SimI]
[-tau TauMinI,TauMaxI,TauSimI]
[-tf type(1ou2),UMinI,UMaxI,USimI,Usu0_Cut]
[-bl Blending (1ou2)]
[-seed seed1,seed2,seed3]
```

soit en remplissant directement la structure *MONTECARLO mc* dans *UNIT()* et eventuellement *UNITCCD()* pour l'initialisation du *t0*. *nbsread -h* donne un help on-line tres detaille sur la facon de piloter la generation. La description des variables de la structure *mc* se trouve dans le fichier *nbgene.h*. La description des possibilites du montecarlo est donnee ci-apres en fonction des variables de la structure *mc* :

```
----- CHOIX DU TYPE DE SIMULATION -----
mc.Taille_Finie = 0 : generation taille ponctuelle
                   1 :          taille finie tirage plat en U = Rs(proj)/Rs
                   2 :          taille finie tirage plat en UC (U = UC*Rs)
mc.Blending = 1 : simulation du "blending" sur l'etoile principale (la + brillante)
                2 : simulation du "blending" sur l'etoile cachee (la -
brillante)

----- GENERATION PLATE -----
si max<=min
  - TauSim=TauSimI  Tau du phenomene
  - T0Sim=T0SimI    instant du maximum si T0SimI>0
  - U0Sim=U0SimI    parametre d'impact (en Re) si U0SimI>0
  - USim=USimI      projection rayon etoile dans plan naine (en Re) si USimI>0
si max>min
  - TauSim est engendre avec une loi plate entre TauMinI ET TauMaxI
  - T0Sim est engendre avec une loi plate entre T0MinI ET T0MaxI
    si T0MaxI<T0MinI et T0SimI<0 on prend plat sur toute la periode
  - U0Sim est engendre avec une loi plate entre U0MinI ET U0MaxI
  - USim est engendre avec une loi plate entre UMinI ET UMaxI si mc.Taille_Finie=0
    entre UMinI*Rs ET UMaxI*Rs si mc.Taille_Finie=1

----- GENERATION SELON UNE LOI DETERMINEE PAR L'UTILISATEUR -----
-
l'utilisateur peut fournir la loi de tirage aleatoire pour u0,tau et u:
```

```

- mc.TireU0(float), mc.TireT0(float), mc.TireTau(float) et
  mc.TireU(float) sont 3 pointeurs sur des fonctions
  "float (*TireU0)(float alea)"
  Il sont initialises a 0 dans nbsread.c
- mise en oeuvre (ex: pour le tirage de tau plat en log):
  1-/ dans nbsreadu.c, l'utilisateur ecrit les routines de tirage
    ( alea est uniformement distribue entre [0.,1.[ )
    float my_tiretau(float alea)
    {
      float tau;
      tau = log(mc.TauMinI)
          + alea * ( log(mc.TauMaxI) - log(mc.TauMinI) );
      return( (float) exp( (double) tau ) );
    }
  2-/ dans UINIT (par exemple)
    mc.TireTau = my_tiretau;

```

Evidemment dans l'hypothese d'un tirage avec effet de taille finie le rayon de l'etoile (starcal.Rstar) doit etre rempli dans la calibration absolue donnee par l'utilisateur.

Pour une generation avec effet de blending, l'utilisateur doit une routine de simulation de l'effet de blending et la connecter au prototype mc.Get\_Par\_Blending dans UINIT() ou UINITCCD().

```
mc.Get_Par_Blending = my_blending_gene
```

Pour une simulation avec effet de blending sur l'etoile la plus brillante du couple, il faut remplir

```
mc.coeff_Arec1[ic] pour les 2 couleurs (ic=0,1)
```

Arec est le coefficient de blending :

```
Ampli(blend) = (Ampli-1.)*Arec +1
```

Pour une simulation avec effet de blending sur l'etoile la moins brillante du couple, il faut remplir

```
mc.coeff_Arec2[ic] pour les 2 couleurs (ic=0,1)
```

Pour une simulation avec effet de blending sur l'etoile la plus brillante du couple et avec effet de taille finie, il faut remplir

```
mc.coeff_Arec1[ic] pour les 2 couleurs (ic=0,1)
```

```
mc.starcal1.Rstar : rayon de l'etoile la plus brillante
```

Pour une simulation avec effet de blending sur l'etoile la moins brillante du couple et avec effet de taille finie, il faut remplir

```
mc.coeff_Arec2[ic] pour les 2 couleurs (ic=0,1)
```

```
mc.starcal2.Rstar : rayon de l'etoile la moins brillante
```

### - Initialisation des aleatoires

Le generateur utilise est drand48 standard unix. Il est prevu 3 variables permettant d'initialiser le generateur :

```
mc.iseed1, mc.iseed2, mc.iseed3
```

A la fin du job l'etat des graines est automatiquement imprime et permet de redemarrer un autre job pour la suite de la sequence d'aleatoires.

### - Exemple de fichier datacard

Cf AnaSuiv/nbsread.dataccd.Exemple

```
4                nombre de CCD a traiter
@CCD 03          numero du 1er CCD a traiter
@COUL 1          numero de la couleur de reference
/home/suivi/file1_coull1_ccd03.suivi
/home/suivi/file2_coull1_ccd03.suivi
@COUL 2          numero de la couleur secondaire (facultatif)
/home/suivi/file1_coul2_ccd03.suivi
/home/suivi/file2_coul2_ccd03.suivi
/home/suivi/file3_coul2_ccd03.suivi
@FINCCD         carte de fin de CCD
@CCD 01          numero du 2sd CCD a traiter
@COUL 1          numero de la couleur de reference
/home/suivi/file1_coull1_ccd01.suivi
/home/suivi/file2_coull1_ccd01.suivi
@COUL 2          numero de la couleur secondaire (facultatif)
/home/suivi/file1_coul2_ccd01.suivi
@FINCCD         carte de fin de CCD
@CCD 05          numero du 3ieme CCD a traiter
@COUL 1          numero de la couleur de reference
/home/suivi/file1_coull1_ccd05.suivi
/home/suivi/file2_coull1_ccd05.suivi
/home/suivi/file3_coull1_ccd05.suivi
/home/suivi/file4_coull1_ccd05.suivi
@COUL 2          numero de la couleur secondaire (facultatif)
/home/suivi/file1_coul2_ccd05.suivi
/home/suivi/file2_coul2_ccd05.suivi
/home/suivi/file3_coul2_ccd05.suivi
@FINCCD         carte de fin de CCD
@CCD 07          numero du 4ieme CCD a traiter
@COUL 1          numero de la couleur de reference
/home/suivi/file1_coull1_ccd07.suivi
/home/suivi/file2_coull1_ccd07.suivi
```

```

/home/suivi/file3_coull_ccd07.suivi
@FINCCD          carte de fin de CCD
@END             carte de fin de fichier datacard

```

```

.... liste d etoiles a selectionner: FgSelSt != 0 dans nbsread.h
@SELET 1001
@SELET 1234
@SELET 2032 2050
@SELET 5044
@SELET 10001 10100
@SELET 6022

```

A l'exception de la 1ere carte et des noms de fichiers de suivi toute carte valide commence par un caractere @. Il est fournit un moyen de selectionner des etoiles dans le fichier datacard et de les identifier dans *UEVT()*. grace au flag *FgSelSt*.

```

@SELET 1001          selection de l'etoile 1001 (attention et=[1,...])
@SELET 1001 2001    selection des etoiles de 1001 a 2001 inclusif

```

### - Commentaire sur les structures des fichiers de suivi et de nbsread

Dans le fichier de suivi sont codees les structures :

```

GLOBINFO STARINFO MESURE TIMEINFO

```

La description des diverses structures et variables qu'elles contiennent se trouve dans *fsvst.h*. Pour gagner de la place, certaines d'entre elles (*TIMEINFO, MESURE*) ont des variables codees de facon non triviale dans les mots de 16 bits. A la relecture du fichier de suivi, *nbsread* decode automatiquement les variables pour les ranger en clair dans des structures utilisateurs (dont le nom se termine par un *U*). Notamment la structure *MESUREU* decode la structure *MESURE* et renvoie le flux et les erreurs calibrees et tenant compte des informations codees dans la structure *TIMEINFO*.

```

TIMEINFO  -> TIMEINFOU (qui ne contient qu'une partie des infos)
MESURE    -> MESUREU

```

Les routines de decodage sont dans *fsvst.h* mais ne doivent en pratique pas etre utilisees directement par l'utilisateur de *nbsread* :

```

TIMEINFOU * DecodeTim( TIMEINFO *time, TIMEINFOU *timeu);
MESUREU * DecodeMes( MESURE *mesc, MESUREU *mesu);

```

### - Routines pratiques utilisables dans nbsread

```

int read_timeinfo(int ic,int imes,TIMEINFO *tim)

```

Comme *TIMEINFOU* ne contient qu'une partie des informations de la structure *TIMEINFO* du fichier de suivi, il peut etre pratique de pouvoir recuperer *TIMEINFO*. *read\_timeinfo* permet de recuperer *TIMEINFO* pour la couleur *ic* (0,1) et la mesure *imes* (0,*nmes[ic]-1*).

```

void PrtGlobInfo (GLOBINFO *glinf, int lp);
void PrtStarInfo (STARINFO *sti, int n, int lp);
void PrtMesure (MESURE *mes, int n, int lp);
void PrtMesureU (MESUREU *mesu, int n, int lp);
void PrtTimeInfo (TIMEINFO *tim, int n, int lp);
void PrtTimeInfoU (TIMEINFOU *tim, int n, int lp);

```

Ces routines permettent d'imprimer les diverses structures d'un fichiers de suivi. Elles sont dans *fsvt.h*

### - Routines de travail avec les transformations

Elles permettent, par exemple, de calculer les coordonnees sur l'image courante a partir des coordonnees de references de STARINFO. Les rptines se trouvent dans *fsvt.h* et *transfost.h*.

```

void GlobInfoToTransf(GLOBINFO *gli, TRANSFO *t1, TRANSFO *t2);
void TimeInfoToTransf(&timloc,&t1,&t2);
void CordTransf(double xS, double yS, double *xD, double *yD,
                TRANSFO *transf)

```

Voici un exemple d'utilisation a mettre dans *UEVT()* pour l'etoile courante.

```

TIMEINFO timloc;
TRANSFO t1,t2;
double Xcur[2],Ycur[2],x,y;
for(ic=0;ic<NCOULMX;ic++) {
  for(k=0;k<nmes[ic];k++) {
    read_timeinfo(ic,k,&timloc);
    TimeInfoToTransf(&timloc,&t1,&t2);
    -- des coordonnees de reference vers les courantes
    x = staru[ic].XPos; y = staru[ic].YPos;
    CordTransf(x,y,&Xcur[ic],&Ycur[ic],&t1);
    PrintTransfo(&t1);
    PrintTransfo(&t2);
    -- des coordonnees courantes vers celles de reference
    CordTransf(Xcur[ic],Ycur[ic],&x,&y,&t2);
  }
}

```

## 10 Variables d'environnement

Les variables d'environnement sont definies sous l'interpreteur de commandes c-shell par la commande *setenv* et la commande *export* sous *bash* :

```
csh> setenv NOM_DE_LA_VARIABLE valeur_de_la_variable
ex: csh> setenv PDEBUG_ALIGQUICK 5
bash> export NOM_DE_LA_VARIABLE=valeur_de_la_variable
ex: bash> export PDEBUG_ALIGQUICK=5
```

## 10.1 Variables pour le debugging.

- **PEIDA\_DATACARDS** : nom par défaut du fichier contenant les datacards (classe DataCards).
- **PEIDA\_WORK** : répertoire du fichier qui contient les datacards (classe DataCards).
- **PEIDANICE** : niveau de priorité (ordre UNIX nice, classe PeidaInitiator).
- **PEIDAWORKPATH** : chaîne de caractères retournée par la fonction PeidaWorkPath() (fichier utils.cc).
- **ACQ\_TZ** : nom du fuseau horaire (classe TimeZone). Si la variable n'est pas définie, nous sommes en France.
- **PDEBUG\_FOND** : niveau de debug des méthodes EstimeFdSg et FondSigmaCiel (classe Image<T>).
- **PDEBUG\_FONDLOCAL** : niveau de debug de la méthode FondLocal (classe GridBgnd).
- **PDEBUG\_KILLBAD** : niveau de debug de la fonction KillBadZn.
- **PDEBUG\_GENERALFIT** : niveau de debug de la classe GeneralFit.
- **PDEBUG\_ALIGQUICK** : niveau de debug de la classe AliGQuicK.
- **PDEBUG\_CIRCRFIX** : niveau de debug de la classe CircRFix<T>.
- **PDEBUG\_CORRFIND** : niveau de debug de la classe CorrFind.
- **CORRFIND\_RELOADIMG** : nom de l'image de corrélation pour une relecture directe (classe CorrFind).
- **PDEBUG\_CORREL** : niveau de debug de la fonction ImgCorrCoef\_MakeNoyau.
- **PDEBUG\_QUALITE** : niveau de debug du programme de qualité des images.
- **PDEBUG\_REBIN** : niveau de debug de la classe RebinImage<T>.
- **PDEBUG\_MAKESTLIST** : niveau de debug de la fonction MakeStarList.
- **PDEBUG\_STLIST** : niveau de debug de la classe StarList.
- **PDEBUG\_TRANSFO** : niveau de debug de la classe Transfo.

## 10.2 Variables pour le pilotage de stars++.

- **PEIDANOCORE** : pour que stars++ ne produise pas de core dans le cas d'un plantage.
- **PEIDA\_IMAGES** : nom du répertoire où se trouvent les images à traiter.
- **PEIDA\_REF** : nom du répertoire où se trouvent les références.
- **PEIDA\_SUIVI** : nom du répertoire où se trouvent les fichiers de suivi.
- **PEIDA\_WORK** : nom du répertoire de travail.

- **PEIDA\_IMAGEEXT** : extension des noms des fichiers images (ex : .fits).
- **PEIDA\_REFEXT** : extension des noms de fichiers de référence (ex : .ref).
- **PEIDA\_SUIVINAME** : nom à donner au fichier de suivi.

## 11 Exemple commenté de sortie stars++

### Execution (avec PrtLevel=2 de

```
stars++ -w -d eros2+.datacard gs20011tbr7j2530 gs20011tbr7j2631 gs20011tbr7
```

---

### Impression du numéro de version, des valeurs des paramètres :

```
PEIDA Version 3.5 Revision 1 (V_Fev98) -- Feb 16 1998 10:04:30 cxx
```

```
***** Programme Stars *****
```

```
stars++ -w -d eros2+.datacard gs20011tbr7j2530 gs20011tbr7j2631 gs20011tbr7
InitStars_Info: Datacard File = eros2+.datacard
InitCalLarg_Info: FitWidth= 5 Wf2d,Garde,NSample= 4 3 11 NProj=1
InitFormelD_Info: PSF= Gaussienne (1) NSample=11 NMxSig,Pos= 200 200
... SigMin,Max= 0.5 4.5 PosMin,Max= 2 9 (Tabul=1)
=====> StPrinParam : (ObsId=1)
ImageZone: Offset X,Y= 0 0 SizeX,Y= 1124 1124 PxSzX,Y= -1 -1 (BitSgn=-
1)
.Det.Etoile TypDet= 1 ThrNbSig= 2 DetStarPar=
. 2.1 -0.1 1.1 -1.1 1.1 1 1 0 0 0 0 0 0 0 0 0 0 0 0
..star_list NbMaxSt= 3000 NbBright= 250 VoisWSz= 21
...P.S.F: TypPSF1D= 1 TypPSF2D= 1 TermPSF2= 0 0 0 0
....Fit1D_WSz= 13 Fit2D_WSz= 11 NbStMaxPSF2D= 35 (-4.500000)
.....CutXi2: 99 99 Transfo: 0 30000 250000 - Assoc-MaxDass= 5
.....AliGeom: (0) NStTrg/H2d/Prop= 16 100 500 DCut=200 DAss=2 NoCalP=0
.....DynCCD Force= 0 TypNoise=2 Min,MaxADU= 10 55000
.....DynCCD Gain= 2 RONoise= 10 Fond,SFnd= 140 15.25
.....Calib. Type 1 Flag= 0 MinNbStar= -50 CutXi2,Xi20= 99 9999
.....Calib. FracLow1,2= 0.5 0.5 High1,2= 0.65 0.99
.....PrintLevel= 2 DebugLevel= 0
.....PrintStarList ListImages= 0 FgPrt,FgPSti= 0 7
==> ZFiduFile = (Dir= /loceros/users/eros2/PTest/) Clr=0 NoWrt=0 Qual=0
--ZFidu::Print()-- (NBadRect= 0 NBadCirc= 0 BadPixMap= No)
FiduRect: XMin/Max= -4e+19 4e+19 YMin/Max= -4e+19 4e+19
--Fin de ZFidu::Print()--
ZFiduAuto: NpMin 0, 1<dyn<-1, deb=0
==> Limitation Positions Stl[1]: x=[-1e+19,1e+19] y=[-1e+19,1e+19]
> Control: FgRdCat= 0 FgWrtCat= 1 FgMoyFR= 0 FgSuivi= 7
> FgSec= 0 FgBrOnly= 0 NFile= 3
> CCDList: 1 ccd # -1
```

```

>> Noms: Suivi= ccd.suivi ImgExt= .fits RefExt= .ref NumFmt= %d
>> Noms: Suivi= ccd.suivi RefMoy= moyflx
>>> ImagDir= /data/Images/Red_1/ SuivDir= /loceros/users/eros2/PTest/
>>> RefDir= /loceros/users/eros2/PTest/ WorkDir= /loceros/users/eros2/PTest/
--- UserFlag[0..4]= -1 -1 -1 -1 -1
--- UserCut[0..4]= 0 0 0 0 0
---UserCutMin[0..4]= -99999 -99999 -99999 -99999 -99999
---UserCutMax[0..4]= 99999 99999 99999 99999 99999

```

---

**Traitement de l'image de référence : détection d'étoiles et constitution de la liste (Star-List), recherche des voisins et constitution de la liste des brillantes.**

```

==> Traitement de l'image gs20011tbr7j2530 (Reference) (NumChamp=-
1)
Pixels: total= 1263376 bad= 6760 fracbad= 0.535074%
RzImage(uint_2)[1124*1124] Org: 0 0 PxSz= 0.015 0.015 (Id=0 Nom=)
NbSat,Nul=4 6756 Min,Max= 10 54730 Moy,Sig= 854.134 2056.35 Fond,Sig= 641.6
Date,Heure= 25/10/97 21:21:55 (Tobs = 246662515 TPose = 120)
Fin_ReadFits+CalculFond CPUTime: Total= 5.26646 (Partial= 5.26646) Sec.
ElapsedTime(hh:mm:ss): Total= 00:00:05 (Partial= 00:00:05)
SimpleStFind_Info : 6642 stars detectes (NbFlg=34988) (FlgPmx=33079)
StarList::Sort: NOK= 6642 (NbTot=6642) FluxMedian=1699.37
StarList::FindBright: NBrOk= 250 (NBright=250) NbTot=6642 MeanFlux=72433.3
StarList::FindVoisins_Warning: 0 etoiles vires (Pixel occupe)
StarList::FindVoisins_Warning: 0 etoiles trop voisines (fusionnees)
StarList::FindVoisins: NbIsol(21,19)= 532 826 (SizeVoisList= 17342)
MakeStarList_Info: NbTot=6642 SelFit=6642 NBrOk= 250 - FlxMed=0
.. Fnd,Sg= 641.625 22.2599 Seuil=686.145 NbIsol(21,19)= 532 826
Fin_MakeStarList CPUTime: Total= 7.21638 (Partial= 1.94992) Sec.
ElapsedTime(hh:mm:ss): Total= 00:00:07 (Partial= 00:00:02)
gs20011tbr7j2530.fits: coul=2 ccd=1

```

---

**Ajustement de la position des étoiles et de leurs largeurs (Fit-1D)**

```

DoFitPosSig_Info(Bright): 243 FitOk OKok= 243 / 250 (MeanFlux= 198444)
.. DelX,Y= 0.16 -0.04 SigX,Y= 2.012 1.480 m(s)Dx,y= 0.16(0.33) -
0.04(0.33)
StarList::FindVoisins_Warning: 128 etoiles vires (Pixel occupe)
StarList::FindVoisins_Warning: 69 etoiles trop voisines (fusionnees)
StarList::FindVoisins: NbIsol(21,19)= 554 858 (SizeVoisList= 16521)
DoFitPosSig_Info: 5339 FitOk OKok= 4840 sur 6642 (80.4% , 72.9%)
...DelX,Y= 0.06 -0.03 SigX,Y= 1.883 1.363 m(s)Dx,y= 0.06(0.52) -
0.03(0.38)

```

```
Fin_DoCalFitPos  CPUTime: Total= 10.9996  (Partial= 3.78318) Sec.
ElapsedTime(hh:mm:ss): Total= 00:00:11  (Partial= 00:00:04)
CalStlRefLimits_Info: Min/Max X= 10.695 1119.74  Y= 3.4746 1119.48
CalTransfCur: fg=0  TrCurRef_res=9.66725e-16 , TrRefCur_res=9.66725e-
16
Fin_CalTransfCur  CPUTime: Total= 11.0329  (Partial= 0.033332) Sec.
ElapsedTime(hh:mm:ss): Total= 00:00:11  (Partial= 00:00:00)
```

---

**Détermination des paramètres de PSF** Ceux-ci sont calculés à partir de l'ajustement non-linéaire à deux dimensions des étoiles brillantes.

```
DoPSF2DFit_Info: Nst= 17 (/35/38) SigX= 2.089 SigY= 1.616 Rho= 0.064
Fin_DoPSF2DFit  CPUTime: Total= 12.9661  (Partial= 1.93326) Sec.
ElapsedTime(hh:mm:ss): Total= 00:00:13  (Partial= 00:00:02)
```

---

**Ajustement du flux des étoiles :** Les positions sont imposées à partir des valeurs de référence et de la transformation de calage (identité sur l'image de référence). Les listes des voisines et des brillantes isolées sont recalculées.

```
DoGlobStFit_Info: PSF SigX,Y= 2.08922 1.61583  Rho= 0.0642699
StarList::Sort: NOK= 6286 (NbTot=6642) FluxMedian=4687.71
StarList::FindBright: NBrOk= 250 (NBright=250) NbTot=6642 MeanFlux=196387
StarList::FindBright: NBrOk= 250 (NBright=250) NbTot=6642 MeanFlux=196387
DoGlobStFit_Info: 6413 FitOk OKOk= 6207 sur 6443 (99.5% , 96.3%)
DoGlobStFit_Info: MeanFlux= 196387  MeanFond= 779.982 (Fond=641.625)
Fin_DoGlobStFit  CPUTime: Total= 16.8327  (Partial= 3.86651) Sec.
ElapsedTime(hh:mm:ss): Total= 00:00:17  (Partial= 00:00:04)
```

---

**Constitution d'une liste réduite pour les calages ultérieurs**

```
StarList::Sort: NOK= 1000 (NbTot=1000) FluxMedian=-9e+19
StarList::FindBright: NBrOk= 500 (NBright=500) NbTot=1000 MeanFlux=196618
StarList::Extract: Ok (NbStars= 1000)
```

---

**Ecriture du fichier StarList de référence :** La liste est éventuellement nettoyée si demandé dans les datacard

```
==> Ecriture StarList CCD 0 (/loceros/users/eros2/PTest/gs20011tbr7j2530.r
Limitation des positions de la StarListe:
--ZFidu::Print()-- (NBadRect= 0  NBadCirc= 0  BadPixMap= No)
FiduRect: XMin/Max= -1e+19 1e+19  YMin/Max= -1e+19 1e+19
--Fin de ZFidu::Print()--
StarList::Sort: NOK= 6286 (NbTot=6286) FluxMedian=4687.71
```

```
StarList::FindBright: NBrOk= 250 (NBright=250) NbTot=6286 MeanFlux=196387
StarList::FindVoisins_Warning: 0 etoiles vires (Pixel occupe)
StarList::FindVoisins_Warning: 0 etoiles trop voisines (fusionnees)
StarList::FindVoisins: NbIsol(21,19)= 575 887 (SizeVoisList= 14901)
WriteStarList_Info: NbMaxSt=6900 NbSt=6286 NbBr=250 NbPSF= 250 axr= 6
Fin_WriteStarList CPUTime: Total= 17.466 (Partial= 0.616642) Sec.
ElapsedTime(hh:mm:ss): Total= 00:00:17 (Partial= 00:00:00)
```

---

**Création du fichier de suivi : Ecriture GlobInfo/ StarInfo, éventuellement les mesures (TimeInfo/ Mesure) pour l'image de référence**

```
==> Creation Suivi CCD 0 (/loceros/users/eros2/PTest/ccd.suivi)
Fin_FSVWrtStars CPUTime: Total= 17.666 (Partial= 0.199992) Sec.
ElapsedTime(hh:mm:ss): Total= 00:00:17 (Partial= 00:00:00)
FSVWrtMes_Info: Mes 1 (T= 2854.89) NumPhoto= 1171025030 (fga=0)
Fin_FSVWrtMesure CPUTime: Total= 17.8993 (Partial= 0.233324) Sec.
ElapsedTime(hh:mm:ss): Total= 00:00:18 (Partial= 00:00:01)
```

---

**Traitement de la l'image suivante : impression des informations de calcul de temps (TSideral, TLegal), vérification de la dynamique des pixels de l'image lue**

```
==> Traitement de l'image gs20011tbr7j2631 (NumChamp=-1)
Pixels: total= 1263376 bad= 6903 fracbad= 0.546393%
==== TLegInfo : Changement d'heure a 12:00:00
--- TLegInfo : Selection par Jour de semaine >= dd et mois (Flag = C) -
--
Butees: 1er Dimanche apres le 8 Mars - 1er Dimanche apres le 8 Octobre
-> 01 Janvier 97 - 09/03/1997 : -3 heures
-> 09/03/1997 - 12/10/1997 : -4 heures
-> 12/10/1997 - 31 Decembre 97 : -3 heures
**Site : La Silla (Chili) , T Sol.Moy.-TU -4.71533 (heures)
RzImage(uint_2)[1124*1124] Org: 0 0 PxSz= 0.015 0.015 (Id=0 Nom=)
NbSat,Nul=0 6903 Min,Max= 10 54469 Moy,Sig= 760.902 1980.64 Fond,Sig= 544.5
Date,Heure= 26/10/97 21:30:05 (Tobs = 246749405 TPose = 120)
Fin_ReadFits+CalculFond CPUTime: Total= 23.0824 (Partial= 5.18313) Sec.
ElapsedTime(hh:mm:ss): Total= 00:00:23 (Partial= 00:00:05)
gs20011tbr7j2631.fits: coul=2 ccd=1
```

---

**Ajustement de la position des étoiles et de leurs largeurs (Fit-1D) : On calcule la transformation géométrique de calage. Si la première méthode (par recherche d'étoiles très brillantes (VBr) proches de leur positions de référence) échoue, on fait une détection d'étoiles sur l'image courante et on applique les méthodes de calage de liste**

d'étoiles (Technique d'histogramation des différences de positions, ou la technique de recherche de triangles semblables).

```
DoFitPosSig_Info(VBr): OKok=25/25 DelX,Y= -2.18 ( 0.08) 6.00 ( 0.29) SzPv:
DoFitPosSig_Info(Bright): 239 FitOk OKok= 239 / 247 (MeanFlux= 175825)
.. DelX,Y= -2.15 5.98 SigX,Y= 1.983 1.393 m(s)Dx,y= 0.03(0.08) -
0.02(0.12)
DoFitPosSig_Info: 4764 FitOk OKOk= 4348 sur 6286 (75.8% , 69.2%)
...DelX,Y= -2.14 5.99 SigX,Y= 1.913 1.346 m(s)Dx,y= 0.00(0.34) 0.01(
Fin_DoCalFitPos CPUTime: Total= 25.9156 (Partial= 2.83322) Sec.
ElapsedTime(hh:mm:ss): Total= 00:00:26 (Partial= 00:00:03)
CalTransfCur: fg=0 TrCurRef_res=0.0978229 , TrRefCur_res=0.0978114
Fin_CalTransfCur CPUTime: Total= 25.9323 (Partial= 0.016666) Sec.
ElapsedTime(hh:mm:ss): Total= 00:00:26 (Partial= 00:00:00)
```

---

### Calcul des paramètres de PSF-2D :

```
DoPSF2DFit_Info: Nst= 17 (/35/38) SigX= 2.027 SigY= 1.524 Rho= 0.078
Fin_DoPSF2DFit CPUTime: Total= 27.4822 (Partial= 1.54994) Sec.
ElapsedTime(hh:mm:ss): Total= 00:00:28 (Partial= 00:00:02)
```

---

**Ajustement du flux des étoiles :** les positions sont imposées à partir des valeurs de référence et de la transformation géométrique de calage. Les paramètres de la PSF sont fixées à leurs valeurs calculées dans l'étape précédente

```
DoGlobStFit_Info: PSF SigX,Y= 2.02688 1.52409 Rho= 0.0778833
DoGlobStFit_Info: 6158 FitOk OKOk= 5943 sur 6201 (99.3% , 95.8%)
DoGlobStFit_Info: MeanFlux= 180673 MeanFond= 685.506 (Fond=544.515)
Fin_DoGlobStFit CPUTime: Total= 30.9988 (Partial= 3.51653) Sec.
ElapsedTime(hh:mm:ss): Total= 00:00:31 (Partial= 00:00:03)
```

---

### Alignement photométrique relatif :

```
DoCalibR0_Info: Rap_flux[2690*] moy= 0.91067 sig= 0.0776901, max_histo= 0.9
DoCalibR1_Info: MeanFlxCur/Ref= 180673/198273 Rap= 0.92 n1,2=3143 6223
DoCalibR1_Info: Pass 1 A= 1.09011 (NbStar=2616)
DoCalibR1_Info: A0.1= 0.38 1.085 A0.2= -0.11 0.914 5.90135e-
05 (NbSt=2594)
Fin_DoCalibR CPUTime: Total= 31.0154 (Partial= 0.016666) Sec.
ElapsedTime(hh:mm:ss): Total= 00:00:31 (Partial= 00:00:00)
```

---

**Calcul des résolutions photométriques :** Pour différents bin de flux moyen  $fr$  ( $2.5 * \text{Log}_{10}(fr)$ ),  $m$  indiquent le décalage relatif moyen du flux calibré par rapport au flux de référence

( $\langle (F_c - F_r) / F_r \rangle$ ,  $ex$  représente la dispersion moyenne  $\text{Sigma}((F_c - F_r) / F_r)$ ,  $rap = m/ex$  est le décalage moyen exprimé en nombre d'écart standard, et  $ef$  correspond à la dispersion moyenne calculée à partir de l'erreur fournie par l'ajustement du flux des étoiles (étape `DoGlobStFit()`).  $ef$  et  $ex$  seraient égaux dans le cas parfait. Une paramétrisation `Err_ext`, `Err_fit` de  $ex$ ,  $ef$  est aussi calculée.

```
DoResol[ 0]: fr= 375233 (13.94) -- m= 0.003 ex= 0.004 rap= 0.683 ef=
DoResol[ 4]: fr= 53249 (11.82) -- m= -0.001 ex= 0.006 rap= -
0.220 ef= 0.005
DoResol[ 8]: fr= 23027 (10.91) -- m= -0.002 ex= 0.010 rap= -
0.154 ef= 0.009
DoResol[12]: fr= 10072 (10.01) -- m= -0.007 ex= 0.020 rap= -
0.329 ef= 0.019
DoResol[16]: fr= 5695 ( 9.39) -- m= -0.018 ex= 0.035 rap= -
0.509 ef= 0.032
DoResol[20]: fr= 3345 ( 8.81) -- m= -0.006 ex= 0.047 rap= -
0.129 ef= 0.052
DoResol[24]: fr= 1882 ( 8.19) -- m= -0.027 ex= 0.080 rap= -
0.332 ef= 0.093
DoResol: Err_ext 5.71747 0.996 0.00461027 0 -999 (x2r= 2.64)
... Err_fit 4.01055 0.794 0 19.6617 2.9776 (x2r= 0.26)
Fin_DoResol CPUTime: Total= 31.1154 (Partial= 0.099996) Sec.
ElapsedTime(hh:mm:ss): Total= 00:00:31 (Partial= 00:00:00)
```

---

**Mise à jour du fichier de suivi :** écriture de la nouvelle mesure correspondante à l'image courante (`TimeInfo`, `Mesure`).

```
FSVWrtMes_Info: Mes 2 (T= 2855.9) NumPhoto= 1171026031 (fga=0)
Fin_FSVWrtMesure CPUTime: Total= 31.2821 (Partial= 0.16666) Sec.
ElapsedTime(hh:mm:ss): Total= 00:00:32 (Partial= 00:00:01)
```

---

### Traitement de l'image suivante

```
==> Traitement de l'image gs20011tbr7j2934 (NumChamp=-1)
Pixels: total= 1263376 bad= 4955 fracbad= 0.392203%
RzImage(uint_2)[1124*1124] Org: 0 0 PxSz= 0.015 0.015 (Id=0 Nom=)
NbSat,Nul=0 4955 Min,Max= 10 54236 Moy,Sig= 890.897 1938.03 Fond,Sig= 688.6
Date,Heure= 29/10/97 21:30:55 (Tobs = 247008655 TPose = 120)
Fin_ReadFits+CalculFond CPUTime: Total= 36.4652 (Partial= 5.18313) Sec.
ElapsedTime(hh:mm:ss): Total= 00:00:37 (Partial= 00:00:05)
gs20011tbr7j2934.fits: coul=2 ccd=1
DoFitPosSig_Info(VBr): OKok=24/25 DelX,Y= -1.17 ( 0.08) 19.75 ( 0.30) SzPv:
DoFitPosSig_Info(Bright): 236 FitOk OKok= 236 / 247 (MeanFlux= 95401.2)
```

```

.. DelX,Y=  -1.08  17.86 SigX,Y=  2.197  1.630 m(s)Dx,y=  0.09(0.28) -
1.89(0.15)
DoFitPosSig_Info: 4621 FitOk OKOk= 4166 sur 6286 (73.5% , 66.3%)
...DelX,Y=  -1.13  17.89 SigX,Y=  1.976  1.543 m(s)Dx,y=  -0.05(0.38)  0.03(
Fin_DoCalFitPos  CPUTime: Total= 39.2651 (Partial= 2.79989) Sec.
ElapsedTime(hh:mm:ss): Total= 00:00:40 (Partial= 00:00:03)
CalTransfCur: fg=0 TrCurRef_res=0.120285 , TrRefCur_res=0.120254
Fin_CalTransfCur  CPUTime: Total= 39.2818 (Partial= 0.016666) Sec.
ElapsedTime(hh:mm:ss): Total= 00:00:40 (Partial= 00:00:00)
DoPSF2DFit_Info: Nst= 18 (/35/38) SigX= 2.151 SigY= 1.749 Rho= 0.069
Fin_DoPSF2DFit  CPUTime: Total= 41.1484 (Partial= 1.86659) Sec.
ElapsedTime(hh:mm:ss): Total= 00:00:42 (Partial= 00:00:02)
DoGlobStFit_Info: PSF SigX,Y= 2.15055 1.74908 Rho= 0.0690197
DoGlobStFit_Info: 6140 FitOk OKOk= 5952 sur 6252 (98.2% , 95.2%)
DoGlobStFit_Info: MeanFlux= 183572 MeanFond= 805.709 (Fond=688.624)
Fin_DoGlobStFit  CPUTime: Total= 44.7149 (Partial= 3.56652) Sec.
ElapsedTime(hh:mm:ss): Total= 00:00:46 (Partial= 00:00:04)
DoCalibR0_Info: Rap_flux[2675*] moy= 0.926971 sig= 0.0569476, max_histo= 0.
DoCalibR1_Info: MeanFlxCur/Ref= 183572/198273 Rap= 0.93 n1,2=3143 6223
DoCalibR1_Info: Pass 1 A= 1.07561 (NbStar=2626)
DoCalibR1_Info: A0.1= 0.32 1.071 A0.2= 0.19 0.915 0.000137056 (NbSt=2
Fin_DoCalibR  CPUTime: Total= 44.7315 (Partial= 0.016666) Sec.
ElapsedTime(hh:mm:ss): Total= 00:00:46 (Partial= 00:00:00)
DoResol[ 0]: fr= 375233 (13.94) -- m= -0.001 ex= 0.005 rap= -
0.296 ef= 0.001
DoResol[ 4]: fr= 52863 (11.81) -- m= -0.003 ex= 0.008 rap= -
0.427 ef= 0.005
DoResol[ 8]: fr= 22936 (10.90) -- m= -0.005 ex= 0.012 rap= -
0.423 ef= 0.011
DoResol[12]: fr= 10023 (10.00) -- m= -0.010 ex= 0.024 rap= -
0.406 ef= 0.023
DoResol[16]: fr= 5649 ( 9.38) -- m= -0.012 ex= 0.041 rap= -
0.289 ef= 0.040
DoResol[20]: fr= 3312 ( 8.80) -- m= 0.004 ex= 0.055 rap= 0.074 ef=
DoResol[24]: fr= 1851 ( 8.17) -- m= -0.033 ex= 0.098 rap= -
0.342 ef= 0.120
DoResol: Err_ext 5.4784 0.96 0.00550896 0 -999 (x2r= 2.07)
... Err_fit 4.59505 0.834 0 19.465 2.9376 (x2r= 0.21)
Fin_DoResol  CPUTime: Total= 44.8315 (Partial= 0.099996) Sec.
ElapsedTime(hh:mm:ss): Total= 00:00:46 (Partial= 00:00:00)
FSVWrtMes_Info: Mes 3 (T= 2858.9) NumPhoto= 1171029034 (fga=0)
Fin_FSVWrtMeasure  CPUTime: Total= 44.9982 (Partial= 0.16666) Sec.

```

ElapsedTime(hh:mm:ss): Total= 00:00:46 (Partial= 00:00:00)

---

## Résumé global :

Fond, SigFond désignent les valeurs calculées pour le fond de ciel et sa dispersion pour chaque image, BadPixels le nombre de pixels hors dynamique. NbDet, NbSel indiquent le nombre d'étoiles détectées, et le nombre d'étoiles conservées après nettoyage. Les paramètres de la PSF calculés pour chaque image par l'ajustement non linéaire 2D et par le fit 1D sont indiqués, ainsi que le nombre d'étoiles pour lesquelles l'ajustement de flux a été réussi (FitOk). DelX, DelY indiquent le décalage géométrique moyenne selon les deux axes, entre l'image et la référence (unités de pixels de l'image courante). Les paramètres calculés pour l'alignement photométriques sont rappelés, ainsi que la valeur de l'absorption (Absor), correspondant à la valeur moyenne de FluxRef/Flux pour de étoiles de flux moyen.

```
----- Resume global -----
---- Nb de champ traite 1 (/1)
=====> Pour le Champ # -1 (NbDet=6642  NbSel=6642)  NImages= 3 (/3)
.... Image 0 (gs20011tbr7j2530): Fond= 641.6 SigFond= 22.26 Bad-
Pixels= 0.54%
Fit2D: SigX,Y= 2.09 1.62 Rho= 0.0643 FitOk= 6413 Absor= 1.000
Fit1D: SigX,Y= 2.01 1.48 DelX,Y= 0.00 0.00 (MoyFlx,Fnd= 196387 78
Calib: 10 A0..5= 0 1 0 0 0 0
.... Image 1 (gs20011tbr7j2631): Fond= 544.5 SigFond= 20.34 Bad-
Pixels= 0.55%
Fit2D: SigX,Y= 2.03 1.52 Rho= 0.0779 FitOk= 6158 Absor= 1.085
Fit1D: SigX,Y= 1.98 1.39 DelX,Y= -2.14 5.99 (MoyFlx,Fnd= 180673 68
Calib: 10 A0..5= 0.380427 1.08462 -0.107372 0.913776 5.90135e-05 0
.... Image 2 (gs20011tbr7j2934): Fond= 688.6 SigFond= 22.62 Bad-
Pixels= 0.39%
Fit2D: SigX,Y= 2.15 1.75 Rho= 0.0690 FitOk= 6140 Absor= 1.071
Fit1D: SigX,Y= 2.20 1.63 DelX,Y= -1.13 17.89 (MoyFlx,Fnd= 183572 80
Calib: 10 A0..5= 0.317527 1.07113 0.193507 0.915372 0.000137056 0
```