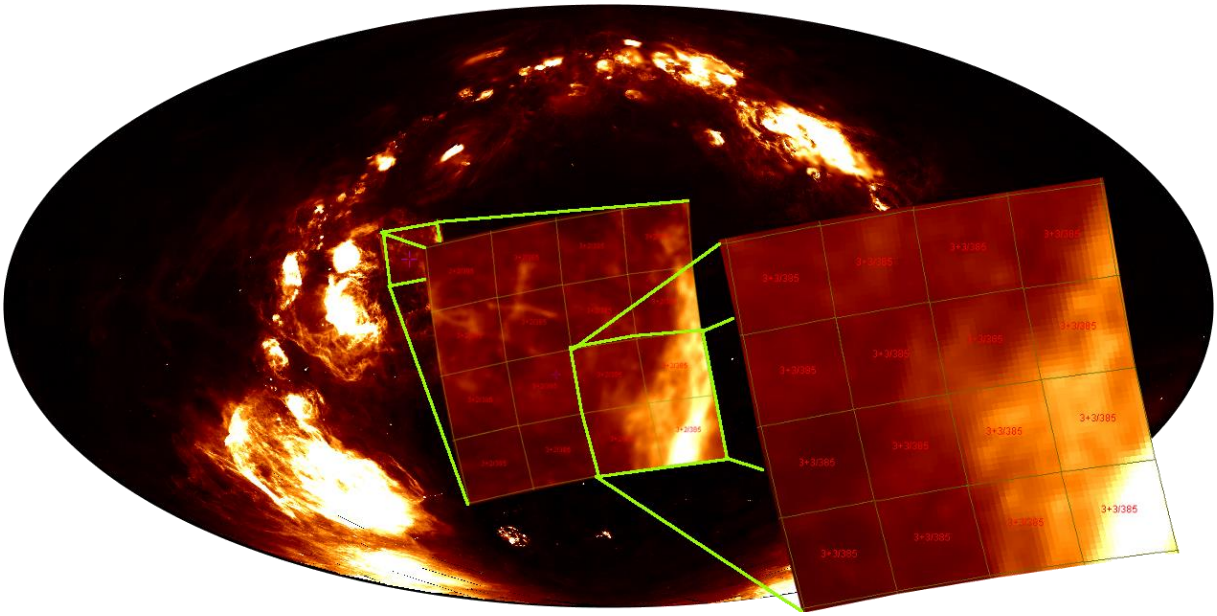
Hipsgen

Manuel de l’utilisateur

Dédié aux versions 12.135 et suivantes



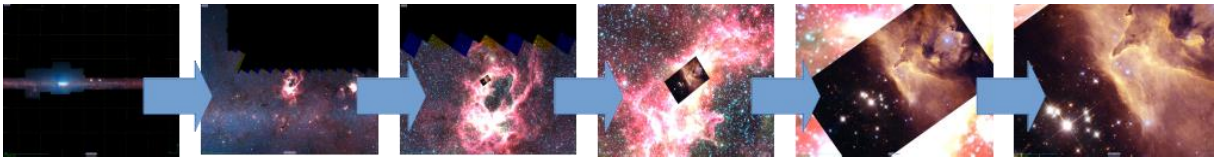
Hipsgen – Manuel de l’utilisateur
Pierre Fernique

Janvier 2023 – avec additions 26/1/2024, 23/4/2024 et 27/5/2024

Centre de Données astronomiques de Strasbourg
Observatoire astronomique de Strasbourg

Introduction

Hipsgen est un logiciel destiné à générer un Hierarchical Progressive Survey, ou HiPS, à partir d'un ensemble d'images, encore appelé « relevé du ciel ». Un HiPS est à la fois un format et un protocole de visualisation des données astronomiques, basé sur un « tuilage » hiérarchique qui permet de zoomer et se déplacer dans des images astronomiques représentées comme un ensemble global cohérent. Il est décrit dans un standard international IVOA « HiPS 1.0 ».



HiPS est destiné en priorité à un usage scientifique. Les méthodes employées minimisent à la fois les déformations spatiales mais également conservent si besoin la dynamique (valeurs des pixels) des images originales.

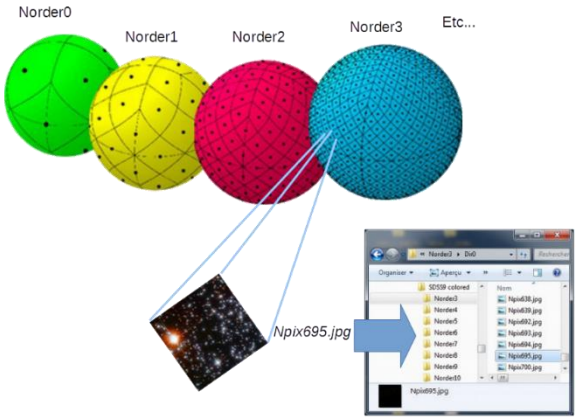
Une fois le HiPS créé, il sera distribuable au moyen d'un classique serveur web (Apache, nginx ou équivalent) et visualisable par n'importe quel outil astronomique compatibles HiPS tels que Aladin (Desktop ou Lite), WWT, Stellarium ou encore DIGISTAR, localement ou à travers Internet.

Hipsgen est en logiciel développé par le CDS. Il partage le même code qu'Aladin Desktop. Il peut être utilisé soit via l'interface graphique d'Aladin via le menu « *Outils -> Génération d'un HiPS* », soit en ligne de commande. Ce document détaille l'utilisation en ligne de commande qui offre davantage de contrôles et un usage plus adapté à la production de grands HiPS. Si vous souhaitiez utiliser l'interface graphique, veuillez-vous référer au manuel d'utilisation d'Aladin Desktop[1].

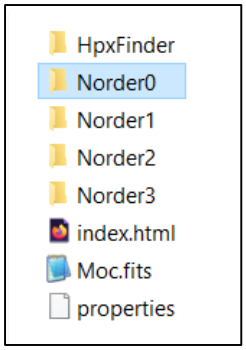
Vous pouvez télécharger la dernière version d'Hipsgen à l'adresse suivante : https://aladin.cds.unistra.fr/java/Hipsgen.jar , ou simplement utiliser le code d'Aladin Desktop comme expliqué ci-dessus.

[1] https://aladin.cds.unistra.fr/java/AladinManuel.pdf

Structure d'un HiPS



En faisant court, un HiPS, ou Hierarchical Progressive Survey, est une mosaïque d'images astronomiques stockée sous forme de tuiles hiérarchiques utilisant le découpage du ciel HEALPix. La création d'un HiPS consiste à générer la mosaïque, le découpage, puis la génération de l'arborescence des tuiles qui constituent le HiPS final. Les tuiles obtenues sont des fichiers images de même taille regroupés dans des répertoires suivant une hiérarchie décrite dans le standard IVOA.



En plus des tuiles, un HiPS fournit quelques fichiers supplémentaires, optionnels ou non : index.html – une page HTML permettant de visualiser le HiPS au moyen du client Web Aladin Lite, Moc.fits – un fichier qui décrit la couverture spatiale du HiPS, et properties – un fichier texte fournissant les propriétés du HiPS (nom, identificateur, description, caractéristiques techniques, etc). Enfin, vous retrouverez souvent un répertoire HpxFinder. Il est ajouté par Hipsgen pour stocker l'index spatial nécessaire à la construction du HiPS ainsi que pour offrir certaines fonctions avancées.

Pour plus de détails, veuillez-vous référer au standard HiPS IVOA[2]

Principe de fonctionnement d'Hipsgen

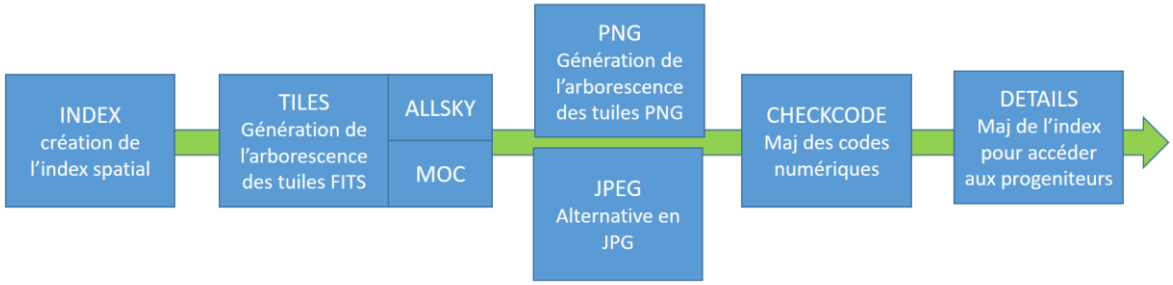
Hipsgen propose toute une panoplie d'actions liées à la génération et à la manipulation d'un HiPS, qui va de l'indexation préalable à la génération des différents formats de tuiles HiPS. Ces actions pourront être exécutées séparément ou automatiquement les unes après les autres. Hipsgen permet également d'effectuer des opérations annexes telles que la duplication, la concaténation de 2 HiPS, ou encore la vérification de la cohérence d'un HiPS déjà créé.

La génération d'un HiPS suit habituellement les 5 étapes suivantes :

1. **INDEX :** Génération d'un index spatial. Cette première étape permet à Hipsgen de mémoriser les tuiles HiPS concernées par chaque image originale (encore appelée « progenitor »). Ces informations de localisation vont être stockées dans le répertoire dédié : « *HpxFinder* ».
2. **TILES** : Génération des tuiles HiPS. En se basant sur les informations de localisation obtenues lors de la phase précédente, Hipsgen va calculer une à une les tuiles HiPS dans un format conservant la dynamique des pixels des images originales. Il s'agit des tuiles « FITS » habituellement de taille 512x512 pixels. Cette étape génère d'une part toutes les tuiles ayant la résolution optimale (Norder***x*** où *x* est en entier correspondant à la profondeur/résolution maximale du relevé), mais également l'ensemble des tuiles de la hiérarchie HiPS jusqu'à la résolution la plus faible (Norder***0***).
   A la fin de cette étape, Hipsgen va également mettre à jour les fichiers annexes : « *properties* », « *Moc.fits* », et si besoin « *Allsky.fits* » (cf standard IVOA HiPS).

[2] https://www.ivoa.net/documents/HiPS/

3. **PNG** : Génération d'un deuxième jeu de tuiles HiPS dans le format d'image compressée PNG. Cette étape utilise les tuiles FITS calculées à l'étape précédente.
   A la fin de cette étape, Hipsgen va mettre à jour les fichiers annexes « *properties* » et si besoin « *Allsky.png* ».
   Cette étape peut être remplacée par JPEG qui opère le même traitement mais dans un format d'image compressée JPEG.
4. **CHECKCODE** : Calcule les clés numériques. Cette étape va mettre à jour le fichier « *properties* » en y insérant les valeurs de clés numériques qui permettra de vérifier rapidement que le HiPS n'a pas été endommagé, par exemple suite à un transfert.
5. **DETAILS** : Met en forme les informations d'indexation spatiale créées à la toute première étape (INDEX) afin d'offrir un moyen pratique d'accéder aux images originales (progéniteurs) via des liens directs depuis le HiPS.



Mise en œuvre

Syntaxe : Hipsgen, en ligne de commande, répond à la syntaxe suivante :

```
java –jar Hipsgen.jar –options in=repSrc out=repDst autresParams… ACTIONs
```

Le répertoire source indiqué par le paramètre « in » contient les images d'origine. Ce répertoire peut lui-même être subdivisé en sous-répertoires, Hipsgen prendra en compte toutes les images contenues dans ce répertoire et ses sous-répertoires. A noter que si vous n'avez qu'une unique image d'origine, vous pouvez directement indiquer son nom de fichier plutôt que le répertoire qui la contient.

Le répertoire destination indiqué par le paramètre « out » contiendra le HiPS qui sera généré.

Les actions : Elles sont indiquées sous la forme d'un ou plusieurs mots clés décrivant la séquence des étapes à opérer. Par convention, elles s'écrivent en majuscules même si Hipsgen ne fait pas de distinction. En l'absence d'actions mentionnées, les 5 actions indiquées précédemment (INDEX, TILES, PNG, CHECKCODE, DETAILS) seront exécutées en séquence.

Les paramètres : Ils précisent certains éléments requis pour effectuer les actions. Ils suivent la syntaxe « *nomDuParam=valeurs…* ». Tout comme pour les actions, les majuscules et minuscules du nom du paramètre n'ont pas d'importance. D'autre part, certains paramètres supportent des alias (synonymes) pour compatibilité avec les versions antérieures d'Hipsgen et/ou pour prendre en compte le vocabulaire spécifique IVOA obscore équivalent (cf. section « Metadonnée »).

Lorsque la liste des paramètres s'allonge, la solution de les mettre tous à la queue leu leu sur la ligne de commande peut s'avérer fastidieuse, et ce, d'autant plus si vos paramètres contiennent des espaces vous obligeant à les mettre entre guillemets. Vous pouvez regrouper une partie des paramètres, ou la totalité de ceux-ci dans un fichier qui sera pris en compte par Hipsgen au moyen de l'option « -param=*filename*». Ce fichier doit contenir un

```
1 in=/data/SCUBA
2 out=/data/ScubaHips
3 id=CDS/P/Scuba/Test
4 blank=0
```

paramètre par ligne, suivant la syntaxe « *clé=valeur* »[3]. A noter que les actions ainsi que les options (ex : « -color ») ne sont pas à mettre dans ce fichier de paramètres. Si c'est le cas, elles seront ignorées.

```
java -jar Hipsgen.jar -param=/dir/param.txt …
```

Les options : Elles contrôlent le fonctionnement général d'Hipsgen, (ex : -color : mise en couleur des messages, -n : non-exécution effective des actions, etc…). Elles sont précédés d'un tiret.

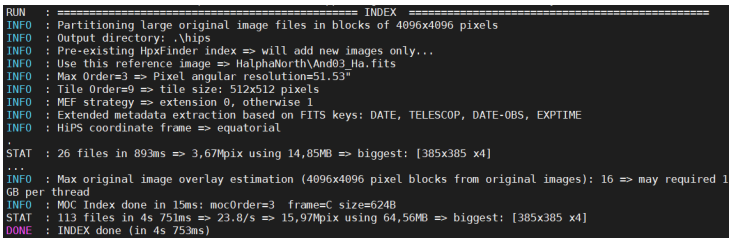
Aide en ligne : Hipsgen offre une aide en ligne pour toutes les actions, paramètres et options. La liste exhaustive de ces éléments s'obtient par l'option « -h » (reproduite à la fin de ce document). La page de manuel spécifique à l'un d'eux s'obtient par l'option « -man *xxx* » où *xxx* est une action ou un paramètre. A défaut, ce sera l'ensemble des pages de manuel qui sera restitué.

```
Ex : java -jar Hipsgen.jar -man TILES
```

Lancement d'Hipsgen via Aladin.jar : A noter qu'il est également possible de lancer Hipsgen avec le code d'Aladin Desktop en utilisant la commande suivante :

```
java -jar Aladin.jar -hipsgen …
```

Terminal : Hipsgen fournit de nombreuses indications au fur et à mesure des actions qu'il opère. Pour vous aider à vous y repérer, et lorsque le terminal où il s'exécute le permet Hipsgen met en œuvre un code couleur pour repérer facilement les informations, les statistiques, les alertes, voire les erreurs. Vous pouvez désactiver ce mécanisme par l'option « -nocolor » ou au contraire le forcer « -color » si Hipsgen n'a pas détecté correctement un terminal « colorisable ».



Visualisation, distribution et publication

Lorsque votre HiPS aura été généré, vous pourrez le visualiser directement à partir de la page de garde « index.html » à l'aide d'un simple navigateur Web, soit localement, soit à distance à travers internet. Dans ce deuxième cas, il vous faudra au préalable « installer » le HiPS que vous avez généré sur votre serveur Web (apache, nginx ou équivalent). Comme un HiPS généré par Hipsgen[4] n'est constitué que de fichiers, une simple copie dans un répertoire accessible par votre serveur web fera parfaitement l'affaire sans modification de la configuration du serveur Web.

Pour publier votre HiPS notamment si vous souhaitez qu'il apparaisse dans les menus des outils tels Aladin ou DIGISTAR, veuillez-vous reporter au standard HiPS déjà précédemment cité[5].

[3] Les lignes vides et les lignes de commentaires commençant par le caractère # sont autorisées.
[4] A noter que les HiPS peuvent être implémentés sous d'autres formes (bases de données, fichiers dédiés) tant qu'ils peuvent être consultés selon le standard IVOA HiPS.
[5] https://www.ivoa.net/documents/HiPS/

Génération d'un HiPS à partir d'images FITS

Vous disposez de l'ensemble des images FITS d'un relevé du ciel et vous souhaitez en générer un HiPS. C'est l'utilisation la plus courante d'Hipsgen. Cet ensemble d'images peut se réduire à une unique image, typiquement une carte cartésienne du ciel complet, ou au contraire être composé de milliers voire de millions d'images pour un gros relevé tel que PanSTARR.

Prérequis : Vos images doivent toutes se situer dans le répertoire (ou sous-répertoires) « in » mentionné dans la commande Hipsgen. Cela peut être fait physiquement ou via un ou plusieurs « liens symboliques », voire un montage réseau, mais en gardant à l'esprit qu'un accès disque rapide à ces images est fortement recommandé. Toutes ces images doivent disposer d'une calibration astrométrique. Il s'agit de mots clés contenus dans chaque entête d'image FITS permettant de connaitre la position de chaque pixel sur la voute céleste, et réciproquement. Pour vous assurer que vos images disposent bien de ces informations, et dans une syntaxe reconnue par Hipsgen, il vous suffit de charger l'une d'elle dans Aladin Desktop et de vérifier qu'il est possible d'afficher la grille de coordonnées. Hipsgen supporte la plupart des calibrations astrométriques courantes.

La commande Hipsgen : La génération d'un HiPS complet à partir de votre jeu d'images FITS va s'effectuer en indiquant simplement le répertoire de vos images « in=*xxx* » et le répertoire où sera généré votre HiPS « out=*xxx* » ainsi qu'un identifiant pour votre HiPS. C'est le minimum requis pour générer un HiPS.

```
java –jar Hipsgen.jar in=/data/img out=/data/hips id=HERE/P/myhips[6]
```

Identification : L'identification d'un HiPS est indispensable. Elle se fait au moyen du paramètre « id=… ». L'identificateur suit la convention HiPS[7] à savoir « ivo://*AUTORITE*/**P**/*xxx* ». Le préfixe *ivo://* peut être omis, *AUTORITE* est un mot ou une abréviation - généralement en majuscules - de votre institut de rattachement, ***P*** indique qu'il s'agit d'un HiPS « **P**ixels »[8] , et *xxx* est le label spécifique à votre HiPS. Ce label peut lui-même incorporer des '/' pour décrire éventuellement des sous-catégories d'une même mission.

Pour exemples, le HiPS du DSS couleur généré par le CDS a comme identificateur « *CDS/P/DSSColor* », le HiPS de la mission XMM, instrument EPIC généré par l'ESA répond à l'identificateur « *ESAVO/P/XMM/EPIC* ».

```
Ex: in=… out=… id=HERE/P/myhips …
```

Notez que vous pourrez également modifier l'identificateur de votre HiPS en éditant a posteriori le fichier « properties » (cf. la section « Metadonnées d'un HiPS »).

Compression : Si vos images sont compressées, soit de manière externe par GZIP, BZIP2 ou ZIP, soit de manière interne par une compression RICE, GZIP1, GZIP2 ou HCOMPRESS vous n'avez pas besoin de les décompresser au préalable, Hipsgen le fera au fur et à mesure, gagnant du temps de traitement et vous évitant de disposer de l'espace disque nécessaire à la décompression totale du relevé.

Multi-extensions : Vos fichiers FITS peuvent contenir une ou plusieurs extensions, (HDU dans la terminologie FITS). Par défaut Hipsgen prend en compte la première extension image. Le paramètre hdu=n1,n2-n3,…|all vous permet de modifier ce comportement par défaut. Il est ainsi

[6] Dans la suite de ce manuel, pour des raisons de concision, ce paramètre pourra être omis des exemples.
[7] Cette convention dérive du standard IVOA Identifiers (https://www.ivoa.net/documents/IVOAIdentifiers/)
[8] Il existe d'autres types de HiPS, par exemple des HiPS **C**atalogues.

possible d'indiquer expressément l'indice ou les indices des extensions images requises, voire prendre en compte la totalité des extensions images.

Paramètres additionnels courants

Le format FITS des images astronomiques offre un certain nombre de variantes qu'Hipsgen devra éventuellement prendre en compte. Il peut s'agir d'ajuster le nombre de bits utilisés pour coder chaque pixel (BITPIX), de la prise en compte d'une valeur nulle spécifique (BLANK) ou encore d'un éventuel changement linéaire à appliquer aux valeurs des pixels (BZERO, BSCALE).

Blank : Le paramètre blank=*nnnn* permet d'indiquer à Hipsgen une valeur spécifique utilisée dans les images d'origine pour les pixels considérés comme nuls, c'est-à-dire non définis. Cette valeur aurait dû être mentionnée directement dans chaque fichier FITS au moyen du mot clé FITS « *BLANK* », mais il n'est pas rare qu'elle soit oubliée. La valeur numérique *nnnn* correspond à celle stockée dans le fichier FITS sans application d'un changement linéaire (bzero/bscale cf. ci-dessous). Dans le cas d'une image FITS codée en nombres réels (BITPIX=-32 ou BITPIX=-64), la valeur NaN sera toujours considérée comme nulle même si un *blank* spécifique a été mentionné[9]. Si *nnnn* n'est pas une valeur numérique, il sera considéré comme une alternative au mot clé FITS « BLANK »[10].

Ex : … `in="…" out="…" blank=0`

Bitpix : Le paramètre « bitpix=nn » permet de modifier le codage des pixels. Il indique le codage utilisé pour les tuiles FITS quel que soit le codage des images d'origines. Il suit le standard FITS à savoir : 8, 16, 32 ou 64 pour un codage sur un entier suivant le nombre de bits indiqués, et -32 ou -64 pour un codage en réel en 32 ou 64 bits. Par défaut Hipsgen conserve le codage d'origine, ou tout du moins de la première image traitée (image étalon).

Dans le cas d'une conversion, la plage des valeurs peut être réduite ce qui va éventuellement introduire des arrondis. Pour contrôler cet effet, le paramètre « dataRange=min max » permet d'indiquer à Hipsgen l'intervalle des valeurs d'origine à prendre en compte. Dès lors Hipsgen va ajuster au mieux la conversion des valeurs en utilisant l'opération de changement linéaire propre au standard FITS, à savoir introduire un changement linéaire au moyen des facteurs BZERO et BSCALE suivant la formule : *pixval = bzero + bscale*pixcoded*. A noter que les images d'origine peuvent avoir elles-mêmes déjà utilisé un changement linéaire au moyen de leur propres BZERO et BSCALE, les bornes min et max du paramètre « dataRange » sont alors à considérer en tant que valeurs des pixels après avoir appliqué le changement linéaire (et non pas la valeur codée dans le fichier FITS d'origine). Si cet intervalle n'est pas indiqué explicitement, Hipsgen analysera le contenu de la première image à traiter (image étalon – cf paramètre « img=xxx » décrit ci-après) afin d'en déterminer les valeurs.

Ex : `in="…" out="…" bitpix=16 dataRange="-11000 32000"`

A noter que les images d'origines n'ont pas besoin de partager toutes les mêmes codages. Ceci est vrai pour le BITPIX, mais également pour les 3 autres paramètres BLANK, BZERO et BSCALE.

Tuiles compressées 8 bits: Hipsgen va générer des tuiles FITS et des tuiles compressées PNG. Pour les tuiles PNG, les pixels sont codés sur 8 bits réduisant d'autant la dynamique des valeurs possibles. Ainsi de manière similaire à un changement de BITPIX dans le cas de tuiles FITS, le paramètre « pixelCut=min max [fct] » vous permettra d'indiquer explicitement la plage des valeurs d'origines à prendre en

[9] L'utilisation d'un mot clé FITS BLANK pour un codage réel -32 ou -64 est surprenant, mais pas interdite.
[10] L'utilisation d'un mot clé FITS alternatif au mot conventionnel BLANK est assez rare. On peut trouver par exemple « BADVAL »

compte et la « fonction de transfert » à y appliquer pour obtenir les 255 valeurs possibles des tuiles compressées : log, sqrt, linear (par défaut), asinh, pow2. Si cet intervalle n'est pas mentionné, Hipgen analysera la première image (image étalon) à traiter et en déduira automatiquement la plage qui lui semble appropriée. A noter que les valeurs considérées comme nulles (cf blank ci-dessus) apparaîtront transparentes dans les tuiles compressées PNG, et en noir pour l'alternative JPEG (cf. action JPEG).

Ex : `in="…" out="…" pixelCut="-8000 16000 sqrt"`

La détermination d'un intervalle de valeurs pour la génération des images 8 bits peut s'avérer difficile notamment lorsque le relevé du ciel comporte des images originales avec une très grande amplitude de valeurs. Il s'agit de trouver le meilleur compromis pour éviter de générer un HiPS dont certaines tuiles compressées seraient trop foncées et d'autres trop claires. Une solution alternative existe pour les relevés d'images pointées. Ces relevés ne formant pas une mosaïque continue sur le ciel il peut être possible de déterminer un intervalle spécifique à chaque zone observée. Avec le paramètre « pixelCut=byRegion… » Hipsgen déterminera des zones observées indépendantes, évaluera pour chacun d'elle un intervalle spécifique en analysant la distribution des pixels des tuiles FITS concernées. Puis il sauvegardera ces seuils dans l'entête des tuiles FITS sous les mots clés « *CUTMIN* » et « *CUTMAX* ». Ces valeurs seront utilisées lors de la génération des tuiles compressées. Le HiPS obtenu sera ainsi bien plus homogène visuellement.

Lorsque Hipsgen effectue une évaluation automatique pour déterminer un intervalle de valeurs, soit sur l'image de référence, soit sur les régions indépendantes, le paramètre pixelCut peut être utilisé pour fournir deux pourcentages indiquant la proportion des valeurs des pixels à restituer. Ainsi l'exemple ci-dessous demande à Hipsgen de déterminer les seuils minimum et maximum pour restituer entre les 3% et les 98.5% de la distribution des valeurs des pixels, et cela pour chaque région indépendante du relevé.

Ex : `in="…" out="…" pixelCut="3% 98.5% byRegion"`

Fond du ciel : Suivant l'origine des images, le fond du ciel peut avoir été déjà soustrait ou pas. Il peut donc être nécessaire d'effectuer un ajustement « du fond du ciel » pour pouvoir en faire une mosaïque sans effet « patchwork ». Parfois, la valeur du fond du ciel n'a pas été soustraite, mais simplement indiquée dans l'entête de chaque image au moyen d'un mot clé dédié (ex : BACKGRD). Le paramètre « skyVal=key » vous permet d'indiquer à Hipsgen quel mot clé doit être pris en compte pour soustraire le fond de ciel. Ce paramètre peut également prendre les valeurs : « auto » - indique à Hipsgen de faire une évaluation du fond du ciel sur chaque image[11], « info% » - représente le pourcentage centré de l'histogramme de détection automatique du fond de ciel qu'Hipsgen doit conserver, ou encore « min% max% » - représente l'intervalle des pourcentages de cet histogramme.

Ex1 : `in="…" out="…" skyVal=BACKGRD`
Ex2 : `in="…" out="…" skyVal=99%`
Ex3 : `in="…" out="…" skyVal="0.3% 99.7%"`

Temps d'exposition : Les images originales peuvent également avoir un temps d'exposition spécifique. Toujours dans le but d'obtenir une mosaïque homogène, et dans le cas où cette information a été mentionnée dans l'entête FITS de chaque image originale, il peut être nécessaire[12] de demander à

[11] L'évaluation du fond du ciel par Hipsgen utilise 5 zones de l'image réparties comme sur la face d'un dé, et prend en compte les moyennes des mesures sur les 3 zones dont les valeurs ne sont pas les extrema.
[12] Cette méthode n'est pas restreinte au temps d'exposition, elle peut être utilisée pour n'importe quelle valeur de l'entête FITS en tant que facteur de pondération.

Hipsgen de moyenner les valeurs des pixels en fonction du temps d'exposition. Le paramètre « expTime=*fitskey* » permet d'indiquer le mot clé FITS concerné.

Ex3 : `in="…" out="…" expTime=EXPTIME`

Image étalon : Lorsqu'Hipsgen traite un jeu d'images, il va se baser sur les caractéristiques de l'une d'entre elles (la première lors de la lecture séquentielle du répertoire « in ») et en déduire entre autres les paramètres bitpix, order, dataRange, pixelCut à appliquer. En utilisant le paramètre « img=*filename* », il est possible d'indiquer explicitement quelle image « représentative » doit être utilisée.

Ex : `in=/data/img out="…" img=/data/img/Image32.fits`

Rééchantillonnage : HiPS est basé sur un découpage du ciel en HEALPix, soient 12 losanges de surfaces sphériques identiques à l'ordre le plus faible, chacun subdivisé en 4 à chaque ordre supplémentaire. La résolution maximale est obtenue à la 29ème subdivision et correspond à une taille angulaire d'environ 400 µas par losange. Les tuiles HiPS regroupent un carré de NxN losanges Healpix, par défaut 512x512 (N est nécessairement une puissance de 2). Le paramètre « tileWidth=*nnn* » permet de modifier ce défaut, soit pour faire des tuiles plus petits (ex : 64x64) ce qui diminue les bords « nuls » dans le cas d'observations pointées, soit pour faire des tuiles plus grandes (ex : 1024x1024) pour réduire le nombre total de tuiles du HiPS. Sachant que la tuile est l'élément de base que charge un client HiPS pour réaliser la visualisation il est important de conserver une taille « raisonnable » en terme de Ko à transférer/manipuler.

Hipsgen détermine le nombre de subdivisions requises en fonction de la taille des tuiles (512x512 par défaut) et la résolution angulaire des images d'origine. Il se base sur la première image traitée (image étalon). Il choisira automatiquement l'ordre nécessaire pour disposer d'une résolution angulaire des pixels HiPS égale ou juste supérieure à la résolution initiale. L'ordre 0 (k-tile dans la table ci-contre[13]) correspond aux 12 tuiles de départ. L'ordre 9 correspond à 3 714 728 tuiles, et si les tuiles ont une taille de 512x512 pixels, elles offrent une résolution angulaire par pixel de 805,2 mas. Chaque ordre supplémentaire est deux fois plus précis en résolution angulaire, mais ajoute 4 fois le volume de l'ordre précédent. En utilisant le paramètre « order=*nn* » il est possible de sous-échantillonner, ou d'augmenter l'échantillonnage qu'Hipsgen avait prévu par défaut.

k	$N_{side} = 2^k$	N_{pix}	θ_{pix}	$k_{tile,512}$	$N_{tile,512}$	$\theta_{tile,512}$
0	1	12	58°.6			
1	2	48	29°.3			
2	4	192	14°.7			
3	8	768	7°.33			
4	16	3072	3°.66			
5	32	12,288	1°.83			
6	64	49,152	55′.0			
7	128	196,608	27′.5			
8	256	786,432	13′.7			
9	512	3,145,728	6′.87	0	12	58°.6
10	1024	12,582,912	3′.44	1	48	29°.3
11	2048	50,331,648	1′.72	2	192	14°.7
12	4096	201,326,592	51″.5	3	768	7°.33
13	8192	805,306,368	25″.8	4	3072	3°.66
14	2^{14}	3.22×10^{9}	12″.9	5	12288	1°.83
15	2^{15}	1.29×10^{10}	6″.44	6	49152	55′.0
16	2^{16}	5.15×10^{10}	3″.22	7	196608	27′.5
17	2^{17}	2.06×10^{11}	1″.61	8	786432	13′.7
18	2^{18}	8.25×10^{11}	0″.81	9	3,145,728	6′.87
19	2^{19}	3.30×10^{12}	0″.40	10	12,582,912	3′.44
20	2^{20}	1.32×10^{13}	0″.20	11	50,331,648	1′.72
21	2^{21}	5.28×10^{13}	0″.10	12	201,326,592	51″.5
22	2^{22}	2.11×10^{14}	50.3 mas	13	805,306,368	25″.8
23	2^{23}	8.44×10^{14}	25.1 mas	14	3.22×10^{9}	12″.9
24	2^{24}	3.38×10^{15}	12.6 mas	15	1.29×10^{10}	6″.44
25	2^{25}	1.35×10^{16}	6.29 mas	16	5.15×10^{10}	3″.22
26	2^{26}	5.40×10^{16}	3.15 mas	17	2.06×10^{11}	1″.61
27	2^{27}	2.16×10^{17}	1.57 mas	18	8.25×10^{11}	0″.81
28	2^{28}	8.65×10^{17}	0.786 mas	19	3.30×10^{12}	0″.40
29	2^{29}	3.46×10^{18}	0.393 mas	20	1.32×10^{13}	0″.20

Ex : `in="…" out="…" tileWidth=256 order=13`

[13] Table extraite de l'article Fernique et al. (2015A&A...578A.114F). Les colonnes représentent : *k* – l'ordre HEALPix, *Nside* – la résolution dans la terminologie HEALPix, *Npix* – le nombre de pixels HEALPix, *ϑ-pix* – la résolution angulaire, *k-tile* – l'ordre des tuiles HiPS, *N-tile* – le nombre de tuiles HiPS, *ϑ-tile* – la résolution angulaire d'un pixel de la tuile.

Hipsgen utilise un algorithme de ré-échantillonnage bilinéaire pour « convertir » les pixels des images d'origines dans la grille HEALpix. Hipsgen n'offre pas d'alternative à cet algorithme, sauf si les pixels d'origine sont déjà repérés dans une grille HEALPix (cf. la section « Générer un HiPS depuis une carte HEALPix).

Recouvrement : Hipsgen est utilisé aussi bien sur des jeux d'images pointant sur des objets astronomiques, ou sur des relevés du ciel sous forme d'une mosaïque d'images dont les bords se recouvrent partiellement. Dans les deux cas, plusieurs images d'origines peuvent contribuer par recouvrement à la valeur des pixels calculés dans les tuiles HiPS. Le paramètre « mode=*param* » permet d'indiquer la méthode qu'Hisgen doit utiliser pour combiner plusieurs valeurs de pixels d'origine. Les paramètres supportés sont : overlayNone – sans recouvrement, une seule valeur de pixel sera prise en compte, overlayMean – toutes les valeurs seront moyennées, overlayAdd – toutes les valeurs seront additionnées.

Ex : `in="…" out="…" mode=overlayMean`

A noter que dans le cas de la moyenne (mode par défaut), le signal sur bruit sera d'autant augmenté que le nombre d'images contribuant à la valeur finale est important. La conséquence est qu'il est normal de pouvoir repérer des objets astronomiques dans le HiPS qui étaient pourtant non détectables dans chaque image individuelle.

D'autre part, Hipsgen (version 12.134 et supérieures) dispose de l'action COUNT qui permet de générer un HiPS alternatif (répertoire dédié HipsCounter) recensant, pour chaque pixel HiPS, le nombre d'images ayant contribuées à son calcul (cf. section « Génération d'un HiPS de poids »).

Images suspectes : Il peut arriver que dans le jeu d'images originales, certaines d'entre elles aient une calibration astrométrique imprécise, voire incorrecte. Le résultat sur le HiPS dépendra de la nature de l'erreur et consiste généralement à un léger décalage des objets sur le ciel, facilement repérable en cas de recouvrement de plusieurs images d'origine sur la même région. Mais une erreur particulièrement ennuyeuse est une taille angulaire de pixel erronée. Si l'erreur est importante ces pixels vont être en mesure d'altérer une grande zone du HiPS. Pour éviter cela, Hipsgen rejette automatiquement, pour un lot de plusieurs images, toutes les images dont les pixels ont une projection angulaire sur le ciel suspecte car fortement disproportionnée en longitude par rapport à la latitude[14]. Par défaut un ratio supérieur à 3 sera considéré comme peu probable. Le paramètre « maxRatio=x » permet de modifier cette valeur limite, et la valeur 0 supprime totalement le test. La liste des images écartées est fournie à la fin du traitement.

Ex : `in="…" out="…" maxRatio=1.5`

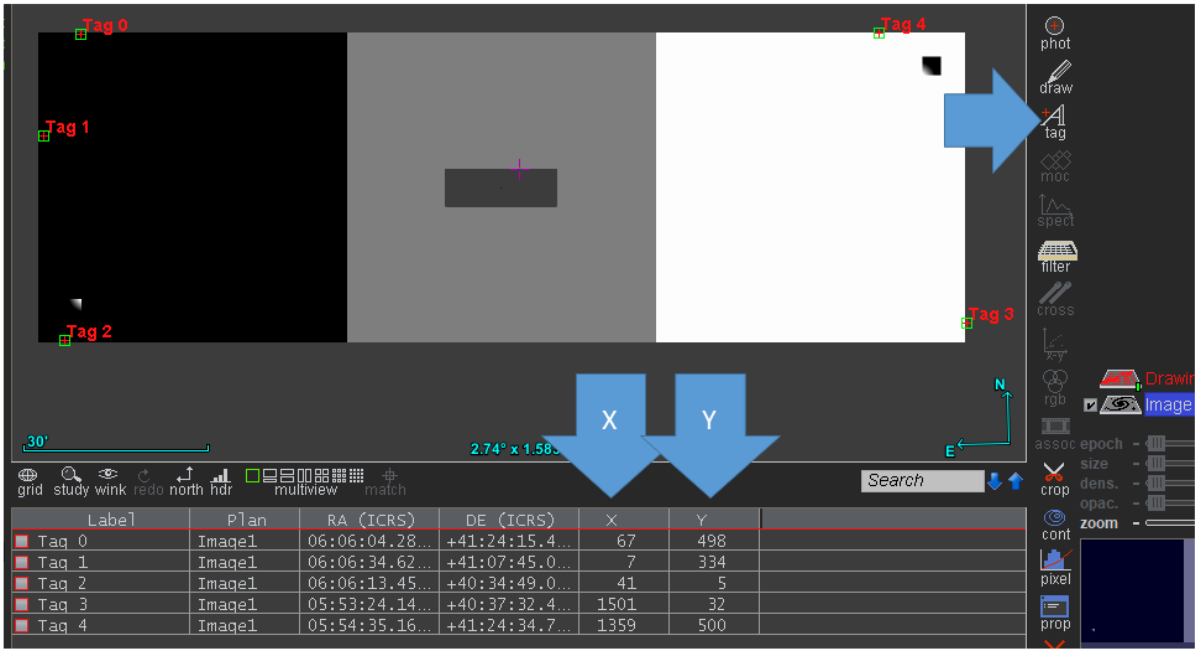
Champ de vue : Certaines images originales peuvent avoir un champ de vue plus petit que la totalité des pixels de l'image : bords altérés, présence d'un cartouche, champ du télescope plus restreint que l'image, … Afin d'écarter les pixels originaux « non observés » de telles images, Hipsgen peut utiliser un polygone ou un cercle où seuls les pixels internes seront pris en compte. Il peut s'agir du même polygone/cercle pour toutes les images, ou seulement pour une partie des images, voire pour chaque image individuellement. Ce polygone/cercle sera décrit dans un fichier ayant le même nom que l'image, respectivement que le répertoire, auquel il est associé mais avec l'extension « *.fov* ». Dans le cas d'un fichier « fov » associé à un répertoire, toutes les images qui s'y trouvent seront concernées

[14] A noter que ce filtrage ne s'applique pas aux images couvrant une grande partie du ciel en projection cartésienne ou moldweide.

sauf pour celles qui disposent de leur propre fichier « fov ». Pour un polygone le format attendu est une liste de couples X Y séparés par une virgule et/ou un espace, un couple par ligne, décrit dans le sens antihoraire sur l'image. Pour un cercle, Hipsgen attend une unique ligne avec un triplet représentant le centre X Y et son rayon R. Les coordonnées sont à prendre selon la convention FITS à savoir que le centre du pixel en bas à gauche est aux coordonnées 1,1. Il est indispensable d'utiliser le paramètre « fov=true » pour que ces fichiers additionnels soient bien pris en compte par Hipsgen. Dans le cas d'une unique forme commune à toutes les images, il est également possible d'indiquer directement les coordonnées de celle-ci en tant que valeur du paramètre (ex : fov=2,2,1000,2,1000,500,4,390).

Ex : `in="…" out="…" fov=true`

Astuce : *L'obtention des couples X,Y peut être facilitée via Aladin Desktop en utilisant l'outil « tag ».*



Lorsque les bords à supprimer sont constants, le paramètre « border=N E S O » vous permet de supprimer un certain nombre de pixels respectivement en haut, à gauche, en bas et à droite de toute les images. Une unique valeur est également possible si tous les bords sont identiques.

Ex : `in="…" out="…" border="10 50 20 50"`

Lorsque la forme du champ de vue est un rectangle ou une ellipse, aligné dans l'image, mais dont le centre, voire la dimension, n'est pas identique d'une image à l'autre, le paramètre « shape=rectangle|ellipse» va demander à Hipsgen de déterminer par lui-même les caractéristiques du masque pour chaque image[15].

Ex : `in="…" out="…" shape=ellipse`

Hiérarchie : Lors de la construction de la hiérarchie des tuiles, chaque tuile d'ordre ***x*** est générée à partir de ses 4 tuiles filles de l'ordre ***x***+1. Le paramètre « mode=*param* » permet de spécifier la méthode d'agrégation des pixels : treeMean – la moyenne des 4 pixels fils, treeMedian – la médiane

[15] Cette méthode n'est à utiliser que dans le cas où une valeur BLANK ne peut être définie (typiquement lorsque la valeur du bord - généralement zéro - est également utilisée dans la zone des pixels à prendre en compte. Cette situation est de fait assez courante (ex : GALEX, etc).

des 4 pixels fils, treeFirst – un des 4 pixels, treeMiddle – un des 2 pixels de valeurs intermédiaires. Si ce n'est pas précisé, Hipsgen utilise la moyenne pour les HiPS issues d'images FITS, et la médiane pour les HiPS issus d'images couleurs[16] (cf. « Comment générer un HiPS à partir d'images couleurs compressées »). Le paramètre « mode » peut indiquer plusieurs modes de contrôle en indiquant plusieurs valeurs, voire de ne préciser que le suffixe d'une valeur pour l'appliquer simultanément. Les 2 exemples ci-dessous sont équivalent. Ils appliquent le calcul de la moyenne, aussi bien pour les recouvrements (cf. paragraphe précédent) que pour la hiérarchie.

```
Ex1 : in="…" out="…" mode="overlayMean treeMean"
Ex2 : in="…" out="…" mode=mean
```

Partitionnement : Dans le cas d'un grand relevé, et d'autant plus s'il a des recouvrements très denses, Hipsgen peut avoir besoin de beaucoup de ressources afin d'être en mesure de calculer les valeurs des pixels des tuiles. Pour éviter les débordements mémoires, Hipsgen va subdiviser le traitement des très grandes images originales en blocs de 4096x4096 pixels. Le paramètre « partitioning=*nnn*|false » permet d'ajuster la taille de ces blocs, voire même de supprimer cette option si vous disposez de suffisamment de mémoire RAM. Ce paramètre doit s'appliquer dès la première étape de la génération d'un HiPS à savoir l'index spatial (INDEX).

```
Ex : in="…" out="…" partioning=false
```

Système de coordonnées : Par défaut Hipsgen génère le HiPS en se basant sur le système de coordonnées équatoriales ICRS. Ainsi les tuiles HiPS seront disposées en fonction de l'équateur et des pôles équatoriaux. Le paramètre « frame=equatorial|galactic|ecliptic » permet de modifier ce comportement par défaut. Il est cependant préférable de conserver tant que faire ce peut le système équatorial afin de pouvoir bénéficier de l'usage de certains outils HiPS combinant plusieurs HiPS, avec la contrainte d'un unique référentiel.

```
Ex : in="…" out="…" frame=galactic
```

Génération d'un HiPS à partir d'images couleurs compressées

Vous disposez d'un ensemble d'images couleurs png ou jpg et vous souhaitez en générer un HiPS.

Prérequis : Vos images doivent disposer d'une calibration astrométrique. Hipsgen va chercher ces informations soit dans les fichiers images directement, sous la forme de mots clés AVM[17], ou d'une entête WCS stockée dans chaque fichier image, ou séparément dans un fichier de même nom mais dont l'extension aura été remplacée par « .hhh ». Pour s'assurer de cela, il vous suffit de charger l'une de ces images dans Aladin Desktop et vérifier que la grille de coordonnées peut être activée. Si ce n'est pas le cas, il vous faudra « calibrer » manuellement (par exemple via le menu d'Aladin *Image -> Calibration astrométrique*), ou automatiquement (par exemple via *Astrometry.net*) chacune de ces images.

Tuiles en couleur : Contrairement à un HiPS basé sur des images FITS en niveaux de gris, Hipsgen va générer directement des tuiles couleurs correspondantes aux images couleurs originales. Par défaut

[16] Le choix de la médiane pour les images couleurs permet de visualiser plus facilement les grandes structures, nébulosités, filaments, etc. La moyenne est la méthode naturelle pour conserver la photométrie entre échelle dans le cas des brillances de surface.

[17] Astronomy Visualization Metadata – http://virtualastronomy.org : mots clés utiliser pour décrire des images astronomiques à usage médiatique.

les tuiles produites reprendront le même format de compression que les images originales, ou plus exactement de la première image originale traitée (image étalon). Le paramètre « color=png|jpg » vous permettra de modifier ce comportement par défaut. Le format JPEG est plus rapide à générer, et plus économe en volume. Le format PNG dispose d'un canal de transparence qui permet au client HiPS de visualiser simultanément un HiPS PNG partiel au-dessus d'un autre HiPS.

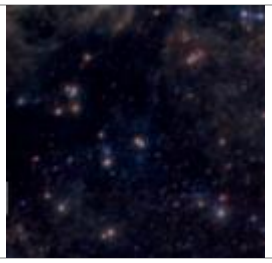
```
in="/data/img " out="/data/hips" color=jpeg …
```

Recouvrement : Les recouvrements des images originales peuvent être traités avec diverses méthodes. Le paramètre « mode=*xxx* » permet d'indiquer à Hipsgen le traitement à opérer en cas de plusieurs contributeurs à un pixel HiPS : overlayMean – moyenne (canal couleur par canal couleur), overlayAdd – addition (canal couleur par canal couleur), overlayNone – un seul contributeur. Cependant ces méthodes sont bien plus adaptées au traitement des images originales FITS pour lesquelles la valeur des pixels représentent une grandeur physique. Une quatrième option est possible « mode=overlayFading ». Elle offre une alternative intéressante pour les images couleurs. Hipsgen va calculer la moyenne des contributeurs mais en utilisant une pondération qui sera inversement proportionnelle à la distance au bord de l'image correspondante. Une manière compliquée pour décrire un « fondu-enchainé » entre les images permettant d'atténuer les transitions entre images.

Hiérarchie : Lors de la construction de la hiérarchie des tuiles, chaque tuile d'ordre ***x*** est générée à partir de ses 4 tuiles filles de l'ordre ***x***+1. Le paramètre « mode=*param* » permet de spécifier la méthode d'agrégation des pixels : treeMean – la moyenne des 4 pixels fils, treeMedian – la médiane des 4 pixels fils, treeFirst – un des 4 pixels, treeMiddle – un des 2 pixels de valeurs intermédiaires[18]. Si ce n'est pas précisé, Hipsgen utilise la médiane pour les HiPS issus d'images couleurs. Les calculs sont effectués sur chaque canal couleur (rouge, vert et bleu) indépendamment.


treeMean


treeMedian


treeMiddle


treeFirst

Contrôle et performances

La génération d'un HiPS sur un grand jeu d'images est un traitement lourd aussi bien du point de vue de l'espace disque, que des ressources mémoires ou des capacités de calcul dont vous disposez. Il s'agit de lire potentiellement des milliards de pixels, d'effectuer divers calculs sur chacun d'eux, et de les réécrire. Plusieurs paramètres peuvent vous permettre d'adapter le fonctionnement d'Hipsgen en fonction des ressources dont vous disposez.

« Not run » : En tout premier lieu, l'option « -n » vous permet de vérifier à l'avance ce que va faire Hipsgen en fonction des paramètres que vous lui avez passés. Techniquement, cette option inhibe l'écriture sur le disque. Vous ne risquez aucun dommage même si vous vous êtes trompés dans vos paramètres. En revanche vous verrez s'afficher les informations techniques du traitement (cf. section

[18] La méthode « treeMiddle » met particulièrement en évidence les grandes structures. La méthode « treeFirst » accentue les sources ponctuelles. L'effet est cumulatif à chaque niveau de zoom.

« Les traces de Hipsgen » à la fin de ce document). C'est bien pratique si vous avez un doute, notamment si l'action que vous souhaitez entreprendre modifie un HiPS déjà généré et que vous pourriez endommager par erreur.

Pilote : La génération d'un HiPS passe souvent par quelques essais préalables pour adapter les différents paramètres et vérifier de visu (par Aladin Desktop ou tout autre client compatible HiPS) le résultat obtenu. Or le temps de génération d'un HiPS peut être long rendant fastidieuses ces étapes de mise au point. L'option « pilot=*nnn* » permet de calculer un HiPS « pour voir » uniquement sur *nnn* images originales. Vous pouvez ainsi ajuster vos paramètres jusqu'à déterminer les meilleures options, puis relancer le calcul, cette fois-ci sur la totalité des images, en supprimant simplement le paramètre « pilot=*nnn* ». A noter que si vous générez le HiPS final dans le même répertoire destination que votre pilote et que celui-ci ne comportait qu'un petit nombre d'images, il est alors judicieux d'ajouter l'option « -clean » ce qui va accélérer le traitement en évitant à Hipsgen d'avoir à vérifier constamment s'il a déjà traité ou non chaque image source (cf. la section « Reprise d'un calcul HiPS » ci-après).

```
Pilot1: in=/data/img out=/data/hips pilot=100
Pilot2: -clean in=/data/img out=/data/hips pilot=100 skyVal=auto …
Exécution définitive: -clean in=/data/img out=/data/hips …
```

Les étapes "pilotes" permettent également de noter les performances obtenues. Elles dépendent bien sûr de la puissance de votre machine de calcul, mais aussi des options que vous aurez choisies pour la création du HiPS. Lors de la génération des tuiles (action « TILES »), des statistiques de vitesse de traitement sont affichées toutes les 30s vous permettant de comparer le cout temporel par rapport au résultat attendu. Sur une petite machine (4 cœurs/4Go), vous pouvez escompter de 500 à 1000 tuiles par minutes, et jusqu'à 5000 à 7000 tuiles par minutes pour un gros calculateur (50 cœurs/128Go). Ainsi, un relevé du ciel complet tel que 2MASS à une résolution HiPS de 800mas qui demande plus de 3 millions de tuiles va prendre une quinzaine d'heures de traitement sur un gros calculateur.

```
.............
STAT  : 610/17977 tiles + 439 nodes in 12s 796ms (3.4%) by 8/8 threads
STAT  : RAM cache: 67 items/300 using 132,73MB/5,28GB freeRAM=7,73GB (opened=67 reused=672 released=0)
.............................
STAT  : 1906/17977 tiles + 1367 nodes in 43s 232ms (10.6% 2554 tiles/mn(=10,64Mpix/s) EndsIn:6m 4s) by 8/8 th
STAT  : RAM cache: 175 items/300 using 346,7MB/5,28GB freeRAM=7,51GB (opened=175 reused=2159 released=0)
.............................
STAT  : 3054+1/17977 tiles + 2161 nodes in 1m 13s (17.0% 2258 tiles/mn(=9,41Mpix/s) EndsIn:6m 106ms) by 8/8 t
STAT  : RAM cache: 282 items/300 using 558,67MB/5,28GB freeRAM=7,25GB (opened=282 reused=3574 released=0)
```

Astuce : *La version 12 et suivantes d'Aladin Desktop peut afficher un HiPS même en cours de génération (à partir de l'étape TILES) sans même stopper Hipsgen. C'est pratique pour les impatients, et aussi pour vérifier le résultat avant la fin du traitement. L'arborescence HiPS étant en cours de génération, il sera éventuellement nécessaire de zoomer jusqu'aux tuiles d'ordre maximal pour voir le résultat partiel. Notez que pour pouvoir prendre en compte l'avancée des calculs, il est indispensable de recharger le HiPS dans un nouveau plan d'Aladin Desktop.*

Multitâche : Hipsgen accélère considérablement son traitement s'il dispose de plusieurs cœurs de calculs. Par défaut, il va prendre tous les cœurs disponibles sur la machine. Si cette machine doit effectuer d'autres tâches simultanément, l'option « maxthread=*n* » va vous permettre de limiter le nombre de threads utilisés par Hipsgen et par conséquent les cœurs mis à contribution. A noter qu'il n'est pas utile de réduire le nombre de threads pour économiser l'impact mémoire (pratique conseillée dans les versions antérieures de Hipsgen), Hipsgen le fera automatiquement si cela s'avère nécessaire.

Mémoire RAM : Une des ressources fortement sollicitées par Hipsgen est la mémoire RAM. Tout comme les cœurs de calculs, plus vous lui en donnerez mieux il se portera. Hipsgen étant un

programme « java », le contrôle de la mémoire RAM se fait par le paramètre « -Xmx » spécifique à votre « interpréteur java ». Une bonne pratique est de compter 1Go de RAM pour chaque cœur de calculs. L'exemple ci-dessous alloue 10 giga octets.

```
Ex: java –Xmx10g –jar Hipsgen.jar in=/data/img out=/data/hips …
```

Si vous disposez d'une machine particulièrement costaude vous pouvez envisager de prendre la quasi-totalité de votre mémoire RAM (attention à ne pas en prendre plus car sinon vous allez « swapper » et donc au contraire ralentir – drastiquement - le traitement) tout en inhibant le découpage en blocs qu'Hipsgen opère sur les grosses images (>4096x4096). Puisque vous avez de la mémoire, autant en profiter. L'exemple ci-dessous alloue 255 giga octets.

```
Ex: java –Xmx255g –jar Hipsgen.jar partioning=false in …
```

Au contraire, si votre machine n'est qu'un petit portable avec très peu de RAM (typiquement <1Go), vous aurez intérêt à imposer une taille de blocs plus petite que le défaut. Hipsgen sera en mesure de générer votre HiPS même sur une petite machine, simplement il faudra être davantage patient.

```
Ex: java –Xmx700m –jar Hipsgen.jar partioning=512 in …
```

Cache disque : Dans le cas où vos images originales sont compressées, Hipsgen devra les décompresser au fur et à mesure pour les traiter. Pour cela il a besoin d'espace disque. Comme Hipsgen va potentiellement utiliser plusieurs fois la même image au cours de son traitement, il gagnera du temps si l'opération de décompression n'est pas répétée inutilement. Ainsi plus le cache disque sera volumineux, plus vite le traitement sera opéré. Pour les images FITS compressées, le cache disque par défaut est désigné par le système d'exploitation de votre machine. Le paramètre « cacheSize=*xxx* » vous permet d'augmenter la taille par défaut (500Go ou moins s'il n'y a pas assez de place sur la partition concernée). Notez que *xxx* est exprimé en Mo. Si la partition du cache par défaut est trop petite, le paramètre « cache=*path* » vous permettra d'indiquer un répertoire alternatif. Finalement, le paramètre « cacheRemoveOnExit=false » vous permettra de conserver le contenu du cache, et donc les images décompressées qui s'y trouvent, afin d'accélérer une reprise éventuelle de traitement. A charge pour vous de supprimer ce cache manuellement une fois le HiPS final obtenu.

Pour les images couleurs JPEG ou PNG, lorsqu'elles sont particulièrement volumineuses (>4096x4096), un cache disque est également mis à contribution mais directement géré par les librairies graphiques java. C'est donc via une option de l'interpréteur java « -Djava.io.tmpdir=*path* » que vous pourrez désigner un répertoire cache alternatif.

```
Ex: java -Djava.io.tmpdir=/path/to/tmpdir –jar Hipsgen.jar …
```

Réduction de la taille des tuiles : Avant tout, notez que vous pouvez déjà disposer de tuiles compressées PNG (ou JPEG) par l'action « PNG » (resp. JPEG). Libre à vous de supprimer les tuiles FITS par la suite via l'action « CLEANFITS » afin d'obtenir au final un HiPS uniquement destiné à la visualisation, mais de moindre volume. Cette approche n'est cependant pas satisfaisante si justement vous souhaitiez conserver et exploiter les tuiles FITS pour disposer de toute la dynamique des valeurs des pixels. Hipsgen vous offre deux possibilités pour réduire la taille des tuiles FITS : la compression GZIP et/ou la suppression des bords. Ces deux méthodes sont à utilisées qu'en cas de strict besoin. Elles sont encore susceptibles d'évoluer, et les tuiles FITS produites ne sont pour l'heure que reconnues par Aladin Desktop.

Compression GZIP : Hipsgen permet d'appliquer une compression GZIP sur les tuiles FITS, soit à la volée avec l'option « -gzip », ou a posteriori via l'action « GZIP ». Celle-ci va gzipper toutes les tuiles

FITS ainsi que le fichier « Allsky.fits » s'il a été généré[19]. C'est une opération très efficace en terme de volume, mais longue, du même ordre de temps que la génération des tuiles FITS elle-même. Réciproquement, via l'action GUNZIP vous pourrez décompresser les tuiles FITS qui auraient été gzippées au préalable.

```
Ex1: -gzip in=/data/img out=/data/hips INDEX TILES …
Ex2: out=/data/hips GZIP
Ex3: out=/data/hips GUNZIP
```

Suppression des bords : La deuxième solution n'est pas à proprement parler une méthode de compression, mais une opération "d'élagage" des bords des tuiles FITS. Elle n'est intéressante que pour les HiPS morcellés[20]. Elle a l'avantage d'être très rapide, voire quasi instantanée si elle est utilisée lors de la génération des tuiles FITS. Elle a également l'avantage de ne pas restreindre l'accès direct aux valeurs des pixels[21]. Les tuiles FITS restent des images FITS classiques, mais de tailles éventuellement restreintes et avec les mots clés dédiés XTRIM et YTRIM mémorisant les bords « à l'origine » ayant été supprimés, ZNAXIS1 et ZNAXIS les dimensions d'origine. La réduction des bords peut être appliquée immédiatement lors de la génération des tuiles FITS si l'option « -trim » est spécifiée. Elle peut être également appliquée, respectivement supprimée a posteriori via les actions « TRIM » et « UNTRIM ».

```
Ex1: -trim in=/data/img out=/data/hips INDEX TILES …
Ex2: out=/data/hips TRIM
Ex3: out=/data/hips UNTRIM
```

Reprise du calcul de HiPS

La génération d'un HiPS sur un très gros relevé peut prendre plusieurs heures, voire jours. Il n'est donc pas étonnant que le traitement puisse être interrompu, volontairement ou non.

Reprise après interruption involontaire

Vous patientez depuis plusieurs heures pendant la génération de votre HiPS. Malheureusement le calcul a été interrompu par inadvertance. Vous pouvez bien entendu relancer le calcul, sans même effacer les tuiles déjà générées. Cependant vous pouvez préciser l'option « mode=keeptile » qui évitera de régénérer les tuiles déjà produites ce qui accélèrera d'autant le traitement.

```
in=/data/img out=/data/hips mode=keeptile …
```

Reprise pour changer le mode de calcul

En revanche, si vous souhaitez reprendre tout le calcul depuis le début (par exemple si vous souhaitez changer certains paramètres tel que le *bitpix* du HiPS), il vous faudra supprimer le premier traitement, soit partiellement, soit totalement. Pour cela vous disposer d'une série d'actions de nettoyage : CLEAN– supprime tout le HiPS excepté le fichier properties, CLEANINDEX – supprime l'index spatial (HpxFinder), CLEANTILES – supprime toutes les tuiles, CLEANFITS – supprime toutes les tuiles FITS,

[19] Attention, les tuiles FITS qu'elles soient gzippées ou non, garderont leur extension « .fits ». A charge au client de détecter une compression éventuelle grâce au magic code en début de fichier, plutôt qu'en se basant sur l'extension de fichier.

[20] Cette approche est particulièrement efficace sur les HiPS d'objets pointés pour lesquels les tuiles HiPS sont très « creuses », autrement dit, elles ont beaucoup de pixels nuls, et principalement sur les bords, et pour une proportion de plus en plus grande en remontant la hiérarchie HiPS.

[21] Cette propriété est indispensable pour certains outils HiPS travaillant « coté serveur », par exemple pour extraire rapidement d'un groupe de HiPS les valeurs d'une SED sur une position donnée.

CLEANPNG – supprime toutes les tuiles PNG, CLEANJPEG – supprime toutes les tuiles JPEG, CLEANWEIGHT – supprime toutes les tuiles de poids (cf. section Reprise d'un HiPS). Vous pouvez également utiliser l'option « -clean »[22] qui supprimera automatiquement uniquement les éléments qui seront régénérés.

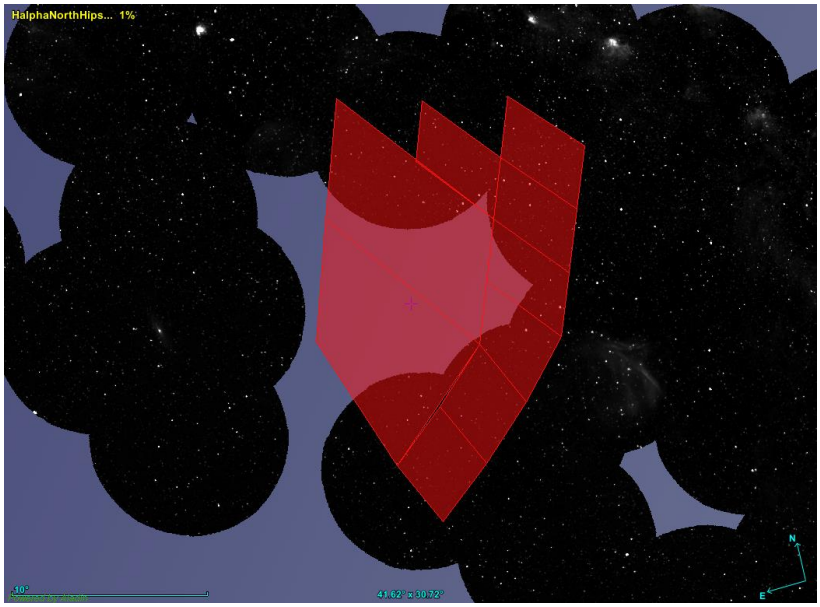
```
-clean in=/data/img out=/data/hips …
```

Reprise pour compléter des zones

Il peut arriver qu'un HiPS généré révèle des défauts que vous souhaitez corriger sans pour autant relancer la totalité du calcul. Ainsi, lorsque certaines images originales ont été oubliées, puis ajoutées, il est souvent pratique de relancer la génération du HiPS, mais uniquement sur les zones concernées. Le paramètre « region=… » est là pour ça. Il consiste à indiquer les numéros de losanges HEALPix que vous souhaitez recalculer sous la forme suivante : *orderN/npix1 npix2 … orderM/npix…*[23]. Vous pouvez alternativement indiquer le nom d'un fichier MOC décrivant la région à reprendre (format MOC binaire FITS).

Le paramètre « mode=… » contrôle la manière dont les nouvelles tuiles vont être insérer dans le HiPS existant : mergeKeep – conserve les pixels des tuiles déjà calculées et n'ajoute que les nouveaux pixels, mergeOverwrite – écrase les pixels des tuiles déjà calculées avec les nouvelles valeurs (c'est l'option par défaut), mergeMean – effectue la moyenne entre les anciens pixels et les nouveaux, mergeAdd – additionne les anciens pixels et les nouveaux.

```
in="…" out="…" mode=mergeKeep region="3/213 215 4/849 851 857 859 881 883-884"
```



Astuce : *L'obtention de la région peut être facilité via Aladin Desktop en entourant la zone concernée par l'outil « Draw », puis en générant un MOC (Multi-Order Coverage) à partir de la zone détourée (menu Coverage -> Gererate a spatial MOC based on the selected drawing object ».*

Pondération des pixels : Une opération de reprise n'aura pas le même résultat si vous disposez toujours de l'ensemble des images originales (plus éventuellement quelques nouvelles images), ou si au contraire vous n'avez plus que les nouvelles images. Dans le premier cas, le résultat final sera identique à une reprise complète du calcul. En revanche, si vous ne disposez que des nouvelles images, la

[22] Cette option remplace l'ancienne option « -f » (toujours acceptée)
[23] Cela correspond à la syntaxe d'un MOC ASCII

pondération associée à chaque pixel aura été perdue. Par exemple supposons qu'un pixel HiPS ait été obtenu par la moyenne de N images originales, et que le deuxième traitement ajoute une nouvelle image sur la zone, dans le premier cas, le pixel final suivant la méthode mergeMean prendra la valeur *(N x ancienneValeur) + nouvelleValeur) / (N+1)* alors que dans le deuxième cas ce sera *(ancienneValeur + nouvelleValeur) / 2*.

Pour pouvoir compléter avec de nouvelles images et utiliser une pondération liée au nombre de progéniteurs, il est nécessaire de positionner le paramètre « incremental=true »[24] lors de la génération du premier HiPS. Ce paramètre indique à Hipsgen que pour chaque tuile FITS générée, il doit également conserver une tuile des « poids » associés à chaque pixel. Ces tuiles de poids seront utilisées ultérieurement dans le cas d'une reprise. Notez que cette méthode coute cher en espace disque car elle va globalement doubler la taille du HiPS finale. Si vous estimez que vous n'avez plus besoin de ces tuiles de poids, vous pouvez les supprimer via l'action « CLEANWEIGHT ».

```
in="/data/img " out="/data/hips" incremental=true …
```

A noter qu'Hipsgen ne prend pas en compte les éventuelles HDU FITS supplémentaires dédiés aux poids ou au masquage associés à chaque pixel original. Techniquement le calcul ne pose pas de problème, malheureusement il existe de trop grandes disparités sur la manière d'intégrer ces poids et masques dans le format FITS. Si vous voulez les utiliser, il vous faudra générer préalablement un jeu d'images prenant en compte ces poids et masques.

Mise à jour d'un HiPS

La mise à jour d'un HiPS intervient généralement pour ajouter des nouvelles images sans avoir à relancer complètement le calcul[25]. En général vous ne disposez plus des images originales précédentes, et vous souhaitez simplement en ajouter des nouvelles. La méthode la plus sûre consiste à créer un nouveau HiPS avec les nouvelles images, puis à le concaténer au HiPS original.

Concaténation de 2 HiPS

Hipsgen permet de concaténer 2 HiPS au moyen de l'action « CONCAT ». Ces 2 HiPS doivent nécessairement être compatibles, c'est-à-dire avoir le même ordre HiPS maximal et le même codage des pixels (bitpix).

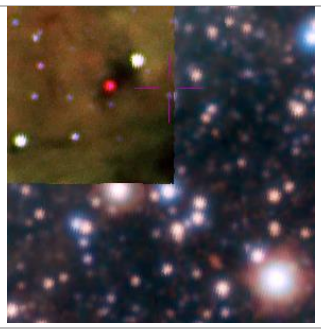
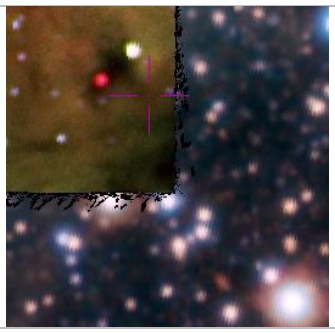
La concaténation s'opère par intégration du premier HiPS dans le deuxième. A la fin de l'opération le deuxième HiPS aura été modifié et contiendra les données communes. Pour des raisons de performances, il est préférable de choisir le plus petit des HiPS à intégrer dans le plus gros. Par défaut les tuiles des deux HiPS vont être fusionnées en utilisant une moyenne. Dans le cas où le paramètre « incremental=true » a été utilisée lors de la création des deux HiPS (cf. section précédente), cette moyenne sera pondérée par le nombre d'images d'origines concernées (progenitors). Le paramètre « mode=*xxx* » permet de modifier ce comportement par défaut : mergeOverwrite : les pixels du HiPS à inclure vont remplacer les pixels du second HiPS même s'ils existaient au préalable, mergeKeep : seuls les nouveaux pixels seront inclus, ceux déjà calculés seront conservés tels que, mergeAdd : les pixels seront additionnés (sans pondération).

[24] Option « -live » dans les versions antérieures d'Hipsgen (toujours reconnue).
[25] Il n'est en revanche pas possible de « supprimer » les pixels qui concernent des images dont on voudrait finalement se débarrasser.

```
Ex: in=/data/hips1 out=/data/hips2 mode=mergeOverwrite CONCAT
```

La concaténation va également intégrer les informations des images originales du HiPS que l'on ajoute dans l'index spatial (HpxFinder) du HiPS receveur.

Concaténation de HiPS couleurs : La concaténation de HiPS couleurs s'opère de la même façon. Les résultats sont bien meilleurs en format PNG. Les HiPS couleurs JPG ne disposant pas de canal de transparence, la fusion se fera uniquement pour les pixels « non noirs ». Or la compression JPEG entraine des légères altérations des valeurs qui vont engendrer des pixels « presque » noirs. Ils ne pourront être fusionnés correctement.

	
PNG HiPS	JPEG HiPS

Mise à jour par Génération & Concaténation

Pour vous faciliter la vie, la création d'un HiPS suivi de sa concaténation peut s'effectuer en une seule action « APPEND ». Celle-ci a l'avantage de s'assurer de la compatibilité des deux HiPS car elle reprend les mêmes paramètres de traitement du HiPS final pour la génération du HiPS à fusionner. Il est bien sûr possible de modifier ce défaut, notamment par le paramètre « mode ». A noter que la présence des images originales du HiPS receveur ne sont pas nécessaires à cette opération, seules les images du HiPS à créer, puis à fusionner.

```
Ex: in=/data/img out=/data/hips APPEND
```

Génération d'un HiPS par combinaisons arithmétiques

Hipsgen offre 4 modes pour combiner des HiPS par additions, soustractions, multiplications et divisions : « mergeAdd, mergeSub, mergeMul, mergeDiv ». Ces options associées au possibilité de concaténation des HiPS vont permettre de générer un HiPS par étapes successifs. Un exemple illustratif peut être l'obtention d'un HiPS à partir d'une part des « images » donnant les décomptes des photons, et d'autre part les temps d'exposition.

```
1) Génération du HiPS de la somme des comptages:
   in=/data/countImg out=/data/hips mode=add

2) Génération du HiPS des temps d'exposition :
   in=/data/timeImg out=/data/hipsTime mode=add

3) Génération du HiPS final de flux:
   in=/data/hipsTime out=/data/hips mode=div CONCAT
```

Génération d'un HiPS à l'aide d'un cluster de calcul

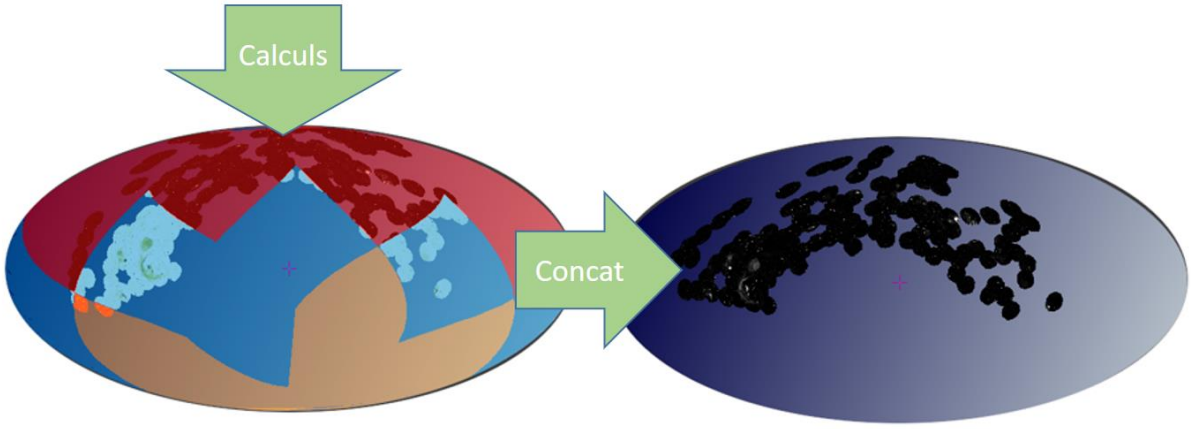
Si les données originales sont très volumineuses, il peut être judicieux de recourir à plusieurs machines de calcul, chacune chargée de générer un HiPS partiel, ceux-ci étant fusionnés à la fin du traitement afin d'obtenir un unique HiPS final.

La stratégie pour répartir le travail va dépendre des capacités disques de chaque machine de calcul. Un disque pour tout le monde, ou des espaces disques indépendants.

Disque commun : La situation la plus pratique - mais peut-être pas la plus rapide - est celle où toutes les machines de calcul accèdent à un même support disque commun. Dans ce cas, le découpage du calcul va être déterminé par des régions disjointes au moyen du paramètre « region » (cf ci-dessus). Une fois les calculs terminés, la fusion des HiPS partiels s'opèrera au moyen d'actions « CONCAT » pour chaque HiPS partiel sur le HiPS final. Si les régions ont bien été déterminées de manière disjointe le mode de fusion n'a pas d'impact sur le résultat final. Si ça n'a pas été le cas, il est recommandé d'opter pour « mode=overwrite » afin de fusionner correctement les tuiles qui auraient été calculées partiellement plusieurs fois.

```
Machine1: in=/data/img out=/data/hips  region="0/0-3"
Machine2: in=/data/img out=/data/hips2 region="0/4-8"
Machine3: in=/data/img out=/data/hips3 region="0/9-11"

Concat1: in=/data/hips2 out=/data/hips CONCAT
Concat2: in=/data/hips3 out=/data/hips CONCAT
```



Disques indépendants : Dans le cas où les machines du cluster ne partagent pas un même disque, ou encore pour obtenir de meilleures performances sur les accès disques, le découpage du calcul nécessitera soit la duplication complète des données originales sur chaque disque de chaque machine, soit le partitionnement de ces images originales. Dans le premier cas, la méthode est finalement similaire à celle utilisée dans le cas d'un disque commun. Dans le second cas, le partitionnement des images originales doit absolument garantir que toutes les images originales concernées par une des régions de calcul se retrouvent bien sur le disque de la machine chargée du calcul.

```
Machine1: in=/data/imglot1 out=/data/hips    region="0/0-3"
Machine2: in=/data2/imglot2 out=/data2/hips2 region="0/4-8"
Machine3: in=/data3/imglot3 out=/data3/hips3 region="0/9-11"

Copie manuelle1 : scp machine2:/data2/hips2 machine1:/data
Copie manuelle2 : scp machine3:/data3/hips3 machine1:/data
```

```
Puis sur la machine1:
Concat1: in=/data/hips2 out=/data/hips CONCAT
Concat2: in=/data/hips3 out=/data/hips CONCAT
```

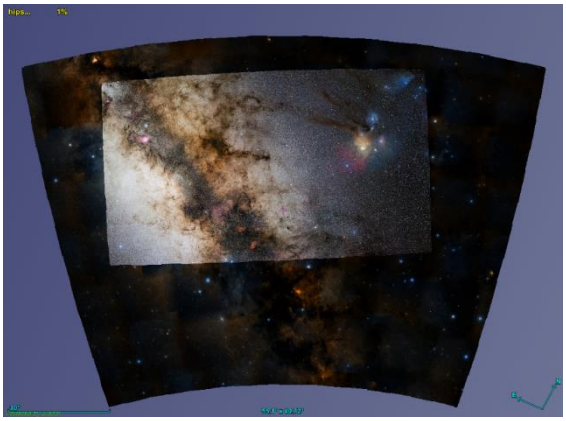
Astuce : *Il est possible d'obtenir les listes des images concernées pour une zone de calcul en détournant l'usage de l'action « INDEX » de la manière suivante. Sur une machine disposant de la totalité des images originales, lancez la commande suivante : in="/data/img" out="/data/hips" order=0 INDEX, puis récupérer les noms des fichiers concernés pour chaque région à partir du contenu des tuiles d'index spatial mémorisées dans le répertoire /data/hips/HpxFinder/Norder0/Dir0…*

L'intérêt d'effectuer un calcul de HiPS sur plusieurs machines plutôt que sur une seule doit être évalué au cas par cas. Le temps nécessaire à la sélection puis de mise en place des images originales ajouté aux déplacements des résultats intermédiaires n'est pas toujours gagnant par rapport au temps de calcul sur une unique machine. D'autre part, comme un relevé du ciel est généralement disponible en plusieurs longueurs d'ondes, il est souvent préférable de paralléliser la génération des HiPS en utilisant une machine par « bande de longueur d'onde » car les données sources sont par construction indépendantes, et il n'y a pas de fusions à effectuer à la fin des traitements.

Génération d'un HiPS composé d'images « outreachs » pointées

Vous disposez d'un ensemble de jolies images couleurs (png ou jpg) pointées sur quelques objets astronomiques et vous souhaitez en générer un HiPS (par exemple les images du JWST)

Prérequis : Vos images doivent disposer d'une calibration astrométrique. Reportez-vous à la section « Génération d'un HiPS à partir d'images couleurs ».



Préparation des données : Une des particularités des images de diffusion des connaissance (« outreachs ») c'est de souvent s'intéresser plusieurs fois aux mêmes objets astronomiques. Or par défaut de Hipsgen effectue une moyenne de toutes les images disponibles. En cas de décalages astrométriques, et/ou d'images très hétéroclites (longueur d'ondes, résolutions), le résultat sera surprenant, voire décevant. Pour ces images, il est parfois préférable de les ajouter, une à une, sur un premier HiPS généré, en « écrasant » les pixels concernés au moyen du paramètre « mode=overwrite »[26]. Le format recommandé pour une telle manipulation est PNG (cf. section « concaténation de 2 HiPS). L'ordre de traitement de ces images supplémentaires est également important. Il est judicieux de commencer par les images ayant la plus grande surface sur le ciel pour finir par les plus petites. Ainsi, vous devrez stocker vos images dans au moins deux répertoires distincts (ex : *allTogether* et *individual*) de telle manière à ce que vous puissiez aisément les manipuler, d'abord globalement pour la génération globale d'un HiPS, puis individuellement pour les ajouts a posteriori.

[26] A noter que le paramètre « incremental=true » est inutile car sans effet dans ce contexte (pas de pondération). Il ne ferait que ralentir le calcul et augmenter le volume du HiPS.

Hipsgen en action : L'action APPEND permet d'ajouter à un HiPS déjà généré de nouvelles images. Dans le cas présenté ici, ces images seront ajoutées une à une séquentiellement[27].

1) Génération d'un HiPS couleur PNG à partir de tout un jeu d'images:
```
in=/data/allTogether out=/data/hips color=png
```

2) Ajouts d'images individuelles en superposition du premier HiPS :
```
in=/data/individual/image1.jpg out=/data/hips mode=overwrite APPEND
in=/data/individual/image2.jpg out=/data/hips mode=overwrite APPEND
…
```

Génération d'un HiPS couleur à partir de 3 HiPS en niveaux de gris

L'action « RGB » d'Hipsgen permet de générer un HiPS couleur à partir de 3 HiPS en niveau de gris calculés au préalable. La commande est la suivante :

```
Ex: inRed=hips1 inGreen=hips2 inBlue=hips3 out=hipsRGB RGB
```

Astuce : *Si les HiPS d'origine sont issus d'un HiPS cube (cf. section « Génération d'un HiPS cube ») les paths des hips pourront être suffixés par « [nn] » où nn est le numéro de la tranche concernée dans le cube (en commençant à 0 pour la première).*

Méthode basique : C'est la méthode par défaut. Chaque composante couleur (rouge, verte et bleue) des pixels à calculer reprendra la valeur du pixel de chaque HiPS en niveaux de gris, ramené à une échelle de 256 valeurs. Par défaut Hipsgen prend en compte l'intervalle des valeurs et la fonction de transfert utilisés pour générer les tuiles compressées des HiPS sources (cf. le paramètre pixelCut). Il est possible de modifier ce choix au moyen des paramètres « cmRed, cmGreen et cmBlue » dont les valeurs suivent la syntaxe suivante : « min max [fct] »[28] - *min* est la plus petite valeur de pixel à prendre en compte, *max* la plus grande et *fct* une fonction de transfert parmi : log, sqrt, linear (défaut), asinh, pow2. Dans le cas où le HiPS couleur à générer utilise uniquement 2 HiPS en niveaux de gris, la troisième composante couleur sera obtenue en moyennant les deux autres.

```
Ex: inRed=hips1 cmRed="10 1000 log" inBlue=hips3 cmBlue="40 800 log" …
```

Méthode Lupton : Hipsgen propose également l'alternative de la méthode Lupton[29]. Elle utilise 2 paramètres pour chacune des composantes couleurs : luptonM (minimum) et luptonS (strecht). Ces paramètres suivent la syntaxe « *valRed/valGreen/valBlue* » et un coefficient global « luptonQ=*x* » (quality). La méthode Lupton met un oeuvre un algorithme accroissant les différentes entre les composantes rouge, verte et bleue. Le « minimum » détermine le début de la plage des pixels concernés, le « stretch » agit sur l'étendue des pixels concernées, et « Q » régit la sensibilité de l'algorithme.

```
Ex: … luptonM="10/30/45" inBlue=hips3 luptonS="1.1/1.3/1.2" luptonQ=20 …
```

[27] Il est fortement déconseillé d'effectuer les APPEND en parallèle (risque de conflits d'accès).

[28] A noter que vous pouvez également ajouter une valeur intermédiaire entre le « min » et le « max » pour reprendre exactement le même fonctionnement qu'utilise Aladin Desktop pour son affichage (cf. le manuel d'utilisation d'Aladin Desktop)

[29] https://ui.adsabs.harvard.edu/abs/2004PASP..116..133L

Astuce : *Le réglage de ces paramètres est assez délicat. Vous trouverez une aide précieuse en utilisant Aladin Desktop (menu : Tools -> Generate a HiPS based on -> a image collection -> Generate RGB). Vous pourrez ainsi visualiser immédiatement le résultat qui sera obtenu.*

Métadonnées d'un HiPS

A chaque HiPS est associé un fichier de « properties » permettant de décrire le HiPS : l'identifier, en connaitre sa provenance, décrire les droits afférents, etc. Ces informations sont particulières importantes si vous souhaitez distribuer votre HiPS, voire le rendre public à travers le « HiPS network » de l'International Virtual Observatory Alliance. Ce fichier « properties » contient à la fois des données techniques (ordre max du HiPS, format des tuiles, …) automatiquement renseignées par Hipsgen, mais également des métadonnées de description qui vous est demandé de mettre à jour manuellement. La liste des champs du fichier « properties » sont décrits dans le standard IVOA HiPS. Quelques-unes d'entre elles peuvent être directement spécifiées lors du lancement de Hipsgen : id – l'identificateur du HiPS, creator – la personne et/ou l'institut à l'origine de la création du HiPS, status – le status de votre HiPS (private|public clonable|clonableOnce|unclonable)[30], target – les coordonnées et la dimension d'un champ de vue par défaut pour la visualisation du HiPS, title – le titre (ou label) du HiPS.

```
Ex: in="…" out="…" id=CDS/P/myhips1 creator="Dupont [CDS]"
```

Les autres métadonnées doivent être éditées manuellement dans le fichier de « properties » :

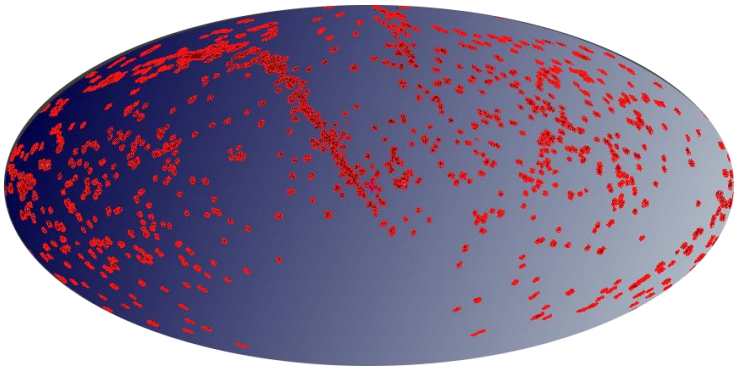
```
hips_copyright      = Copyright mention of the HiPS
obs_collection      = Dataset collection name
obs_description     = Dataset text description
obs_ack             = Acknowledgement mention
prov_progenitor     = Provenance of the original data (free text)
bib_reference       = Bibcode for bibliographic reference
obs_copyright       = Copyright mention of the original data
t_min               = Start time in MJD ( =(Unixtime/86400)+40587
t_max               = Stop time in MJD
obs_regime          = Waveband keyword (Radio Infrared Optical UV X-ray Gamma-ray)
em_min              = Start in spectral coordinates in meters
em_max              = Stop in spectral coordinates in meters
                      ( =2.998E8/freq in Hz, or =1.2398841929E-12*energy in MeV )
```

Moc – la couverture spatiale d'un HiPS

Le standard HiPS recommande l'ajout d'un fichier « Moc.fits » décrivant la couverture spatiale du HiPS. Ce fichier suit le standard MOC de l'IVOA[31], il permet aux clients HiPS de connaitre les zones du ciel concernées par le HiPS sans avoir à vérifier si les tuiles existent ou non.

[30] Le status private empêchera la publication de votre HiPS si vous l'avez ajouté au HiPS network IVOA, clonableOnce indique que vous autorisez que votre HiPS soit copié et redistribué, mais cette copie ne pourra pas elle-même être recopiée. Le défaut est « public clonableOnce ».
[31] https://ivoa.net/documents/MOC/

Hipsgen peut explicitement générer ou regénérer le fichier « Moc.fits » au moyen de l'action « MOC ». Dans la version actuelle de Hipsgen, cette action est toujours exécutée systématiquement à la fin de l'étape de génération des tuiles (action « TILES ») il est donc inutile de la lancer spécifiquement sauf à vouloir modifier les paramètres de génération par défaut.



Ainsi le paramètre « mocOrder=*nn* » permet d'imposer l'ordre du Moc à générer pour éventuellement fournir une couverture plus détaillée que ce qu'aurait fait Hipsgen par défaut. Hipsgen détermine l'ordre du Moc pour minimiser le temps de génération. Plus le HiPS sera petit, plus la résolution du Moc sera précise. En revanche, pour des grands HiPS, l'ordre peut être ramené à celui du HiPS lui-même (mais jamais en-deçà).

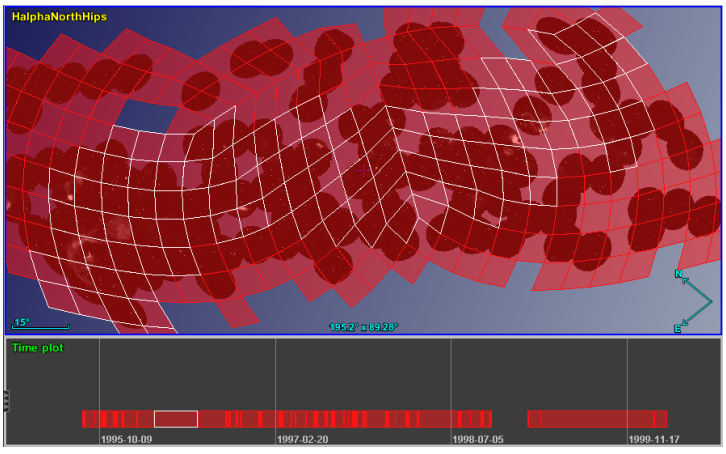
```
Ex: out=/data/hips mocOrder=12 MOC
```

A noter que le calcul du MOC d'ordre identique au HiPS est bien plus rapide car Hipsgen n'a pas besoin d'explorer le contenu de chaque tuile mais se contente de vérifier qu'elle existe.

HiPS non équatorial : Les standards IVOA HiPS recommande qu'un MOC céleste soit équatorial. Dans le cas où le HiPS n'est pas lui-même équatorial (cf. le paramètre frame) - et ne couvre pas tout le ciel - cela va poser un souci de cohérence ne permettant plus aux clients de connaître à l'avance les indices des tuiles HiPS. Pour pallier à cela, l'action « MOC » génèrera d'une part un fichier « Moc.fits » dans le même système de référence que le HiPS, et limité à l'ordre du HiPS (ex : HiPS galactique, order 9 => Moc.fits galactique, order 9). Et d'autre part, il produira un fichier « SMoc.fits » donnant la couverture spatiale en système équatorial et si besoin à une meilleure résolution que l'ordre du HiPS.

STMoc – couverture spatio-temporelle ?

Hipsgen est également capable de générer un fichier « STMoc.fits » fournissant la couverture, cette fois-ci non pas uniquement spatiale, mais spatio-temporelle des images d'origines. L'action « STMOC » va ainsi reprendre l'index spatial créé à la toute première étape de la génération du HiPS (action INDEX), et va utiliser la date d'observation et le temps d'exposition propre à chaque image pour en générer cette couverture spatio-temporelle. Bien évidemment, si ces caractéristiques font défaut, ou n'ont pas été reconnues lors de l'étape d'indexation, ou encore non spécifiées via le paramètre « fitsKeys = *xxx* », la génération du STMOC ne sera pas réalisée. A noter que la résolution spatiale et temporelle du STMOC généré sont fixes et ne peuvent être modifiées.




```
Ex: out=/data/hips STMOC
```

Ce fichier « STMoc.fits » ne fait pas partie du standard IVOA HiPS 1.0, mais certains clients HiPS sont déjà capables d'en tirer parti.

Rq : La version 12.118 et suivantes d'Hipsgen va également mettre à jour les champs « tmin » et « tmax » du fichier « properties » avec les bornes temporelles minimale et maximale du STMOC généré.

Allsky ou non ?

Le standard HiPS offre la possibilité d'ajouter un fichier « Allsky » lors de la génération d'un HiPS. Ce fichier est une mosaïque à faible résolution des 768 tuiles d'ordre 3, assemblées bord à bord. Ce fichier - de fait, ces fichiers « Allsky.*ext* » car associés à chaque format de tuile - sont localisés dans le répertoire « Norder3 ». Ils sont destinés aux clients de visualisation HiPS qui ne savent pas afficher les tuiles d'ordres 0, 1 et 2 sans introduire des distorsions rédhibitoires. Ces clients étant de plus en plus rares, cette option s'avère de moins en moins nécessaire.

Hipsgen peut explicitement générer ou regénérer les fichiers « Allsky » au moyen de l'action « ALLSKY ». Dans la version actuelle de Hipsgen, cette action est toujours exécutée systématiquement à la fin de l'étape de génération des tuiles (action « TILES ») il est normalement inutile de la lancer spécifiquement.

```
Ex: out=/data/hips ALLSKY
```

Page de garde : index.html

Le standard HiPS requière une page de garde sous la forme d'un fichier Web « index.html ». Hipsgen génère cette page automatiquement à la fin de l'étape de production des tuiles. Hipsgen fournit le code HTML pour une visualisation du HiPS au moyen du client Web Aladin lite. Cette page peut être visualisée localement, ou à distance via un serveur Web. Suivant les versions de Hipsgen, le rendu de cette page a évolué. N'hésitez pas à effectuer une action « UPDATE » sur votre HiPS pour bénéficier de la dernière mouture proposée par Hipsgen.

Notez que vous pouvez éditer et modifier cette page selon vos besoins, notamment si la présentation par défaut proposée par Hipsgen ne vous est pas adaptée. Après cela, Hipsgen inhibera toutes modifications automatiques ultérieures afin de conserver votre version personnalisée. Si vous changiez d'avis il vous sera nécessaire de supprimer au préalable votre fichier « index.html » afin qu'il soit re-généré (par exemple via une action « UPDATE »).

Génération d'un HiPS de poids

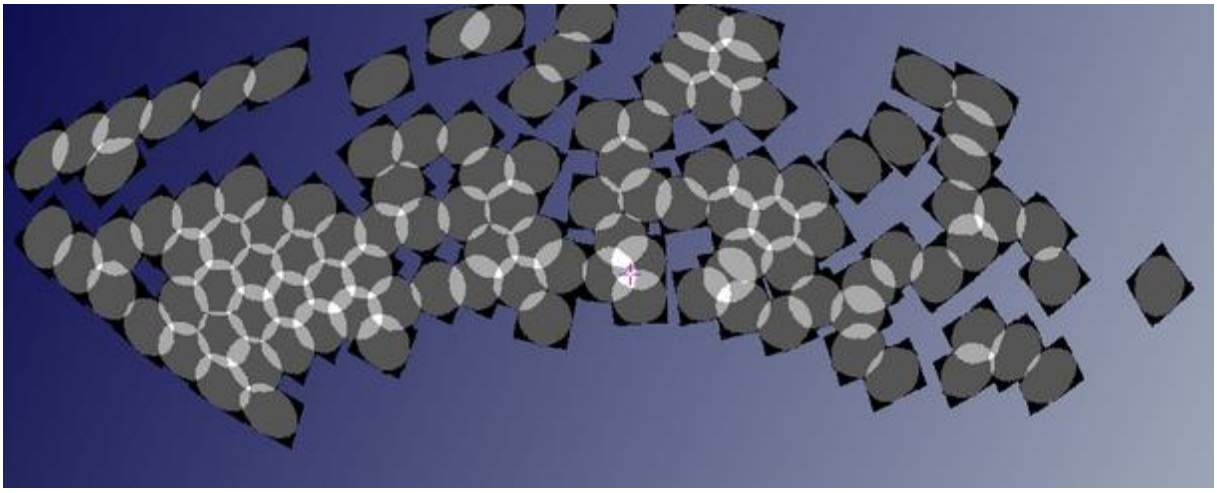
L'action « COUNT » (Hipsgen version 12.134 et suivantes) ajoute la possibilité de générer un HiPS de poids, en addition du HiPS classique. Il s'agit de fournir un HiPS dont chaque pixel de chaque tuile indique, non pas la valeur du pixel final, mais le nombre d'images originales ayant contribuées à son calcul.

Si le paramétrage d'Hipsgen nécessite une pondération sur le calcul des pixels, typiquement une moyenne pondérée grâce au paramètre « exptime = », ou encore un mode de recouvrement en

« fondu-enchainé » par le paramètre « mode=overlayFading », le HiPS de poids fournira le facteur de poids associé à chaque pixel HiPS.

Un tel HiPS permet de connaître individuellement les caractéristiques des pixels du HiPS final. Il peut être utile pour sélectionner/réduire le jeu des images originales requises, ainsi que de déterminer le meilleur paramétrage d'Hipsgen.

Ce « HiPS de poids » est généré dans un sous-répertoire dédié « HpxCounter ».

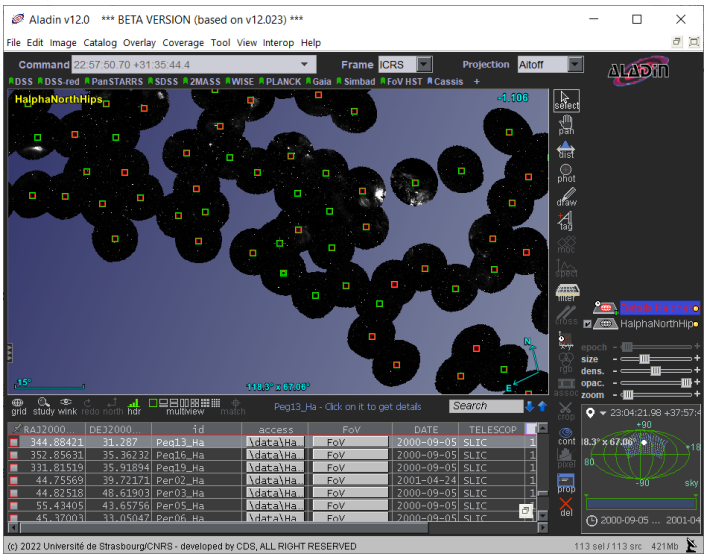


La suppression d'un tel HiPS – typiquement après génération du HiPS final – se fait par l'action « CLEANCOUNT ».

Informations et liens vers les Progenitors d'un HiPS

Si vous le souhaitez, Hipsgen peut compléter les informations d'indexation spatiale créées à la toute première étape (INDEX) afin d'offrir un moyen pratique d'accéder à des informations descriptives des images originales (« progenitors ») ainsi qu'à des liens directs permettant d'y accéder. Hipsgen va générer dans le répertoire HpxFinder de l'index spatial une arborescence HiPS de tuiles ne contenant pas des pixels mais les informations requises. Les clients de visualisation compatibles avec ce mécanisme, tel Aladin Desktop, pourront dès lors offrir à leurs utilisateurs un moyen d'accéder directement aux informations et aux liens des images originales ayant servi à générer le HiPS.

Les informations descriptives des images originales sont par défaut le nom de l'image, le champ de vue ainsi que tous les descriptifs qu'Hipsgen trouvera dans les entêtes FITS : le télescope, la date du cliché, le temps d'exposition, etc. Pour cela, Hipsgen utilise une liste de mots clés habituellement utilisés pour ces informations. Cependant, si vous souhaitez modifier cette liste, vous devrez renseigner le paramètre « fitsKeys=key1 key2 … » lors de la toute première étape INDEX. Si vous ne l'avez pas fait à ce moment-là, vous pouvez bien évidemment relancer cette action a posteriori.




```
Ex: in=/data/img out="…" fitsKeys="DATE TELESCOP" INDEX
```

Mots clés FITS détectés par défaut : DATE, MJD_OBS, UTC, LST, DATE-OBS, MJD-OBS, MJD-END, OBS-DATE, DATE-END, DATEOBS1, DATEOBS2, MIDOBS, ORDATE, TIMESYS, MJDREF, JD, EXPTIME, TEXPTIME, OBSTIME, TIME-OBS, WAVELMIN, WAVELMAX, WAVELEN, TELESCOP, TELNAME

Format des tuiles de l'index spatial HpxFinder : Le format utilisée pour les tuiles de l'index spatial est compatible JSON, avec la contrainte supplémentaire d'un enregistrement par ligne. Il fournit dans ce format un ensemble de couples « clé : valeur » pour chaque caractéristique des images d'origine qui intersectent la tuile spatiale.

Exemple de tuiles HpxFinder


```
{ "name": "And09_Ha", "path": "\data\HalphaNorth\And09_Ha.fits", "ra": "14.941473011561637",
"dec": "40.25323764636017", "cellmem": "592900", "stc": "POLYGON J2000 21.244649286615324
34.981714101299595 8.737279657225253 34.91732519744614 7.6327614938394825 45.11716085475206
22.16170676427507 45.199693481863804", "DATE": "2000-09-05", "TELESCOP": "SLIC", "DATE-OBS":
"1996-11-03T03:31:00", "EXPTIME": "360.0" }
{ "name": "And12_Ha", "path": "\data\HalphaNorth\And12_Ha.fits", "ra": "7.634980856435132",
"dec": "36.17319535985604", "cellmem": "592900", "stc": "POLYGON J2000 13.64854679555983
30.92460475628456 1.7069870688210023 30.863603081811345 0.8016234536391366 41.07006051893838
14.389415836239602 41.14458120239245", "DATE": "2000-09-05", "TELESCOP": "SLIC", "DATE-OBS":
"1996-11-04T04:30:00", "EXPTIME": "360.0" }
```


Pour mettre en forme l'index spatial il est nécessaire d'exécuter l'action « DETAILS ».

```
Ex: in="…" out=/data/hips DETAILS
```

Cette action, non seulement génère l'arborescence HpxFinder nécessaire, mais met en place un fichier « metadata.xml » dans le répertoire HpxFinder. Ce fichier utilise un mécanisme de substitution et de macros qui va permettre aux clients de visualisation de convertir le contenu des tuiles d'indexation spatiale en une table au format VOTable (standard IVOA[32]).

Exemple de fichier « metadata.xml »

L'entête du fichier « metadata.xml » décrit les colonnes qui doivent apparaître dans le client HiPS lorsque l'utilisateur souhaite consulter les caractéristiques d'une image originale particulière (cf. illustration Aladin Desktop page précédente). Cette entête est conforme au document IVOA VOTable, et peut utiliser les options de ce standard pour générer des liens vers l'image d'origine, que ce soit au

[32] https://www.ivoa.net/documents/VOTable

moyen d'un « LINK » VOTable (exemple ci-dessous ligne 15), ou de champs prévus par le standard IVOA Obscore / DATALINK[33].

```
1 <VOTABLE>
2 <RESOURCE>
3   <TABLE>
4     <FIELD name="RA" datatype="double" precision="5" unit="deg">
5       <DESCRIPTION>Right ascension</DESCRIPTION>
6     </FIELD>
7     <FIELD name="DE" datatype="double" precision="5" unit="deg">
8       <DESCRIPTION>Declination</DESCRIPTION>
9     </FIELD>
10    <FIELD name="id" datatype="char" arraysize="13*">
11      <DESCRIPTION>Dataset name</DESCRIPTION>
12     </FIELD>
13    <FIELD name="access" datatype="char" arraysize="9*">
14      <DESCRIPTION>Display original image</DESCRIPTION>
15       <LINK content-type="image/fits" href="${access}"/>
16    </FIELD>
17    <FIELD name="FoV" datatype="char" arraysize="12*">
18       <DESCRIPTION>Field of View</DESCRIPTION>
19    </FIELD>
```

La section des données du fichier « metadata.xml » se réduit à un unique enregistrement VOTable. Celui-ci va permettre d'extraire les informations souhaitées des tuiles HpxFinder en fonction des consultations de l'utilisateur. Pour cela, chaque champ de cet enregistrement unique contient une « macro » qui fait référence aux valeurs des tuiles. Par exemple la macro « $[name] » va être remplacée par la valeur associée au champ « name » dans les tuiles du HpxFinder (cf. exemple de tuile HpxFinder à la page précédente).

```
20    <DATA>
21     <TABLEDATA>
22       <TR>
23       <TD>$[ra]</TD>
24       <TD>$[dec]</TD>
25       <TD>$[name]</TD>
26       <TD>$[path:([^\[]*).*]</TD>
27       <TD>$[stc]</TD>
28       </TR>
29     </TABLEDATA>
30    </DATA>
31   </TABLE>
32 </RESOURCE>
33 </VOTABLE>
```

Syntaxe des macros « metadata.xml »

Les macros utilisées dans le fichier « metadata.xml » mettent en œuvre un mécanisme sophistiqué de substitutions dont le fonctionnement de base a été illustré dans le paragraphe précédent. Lorsque la simple substitution n'est pas adaptée, une syntaxe plus complexe peut être déployée.

En premier lieu, il est possible d'utiliser plusieurs macros pour un même champ et d'ajouter de part et d'autre du texte libre qui apparaitra toujours à l'identique. Seules les macros seront substituées par les valeurs correspondants issues des tuiles HpxFinder.

`Texte1 $[macro1] texte2 … $[macro2]…`

D'autre part, si pour une macro donnée, la substitution prévue ne doit pas reprendre la totalité de la valeur, mais uniquement une portion de celle-ci, il est possible de désigner cette portion au moyen d'un suffixe utilisant la syntaxe des expressions régulières Unix avec « groupes de capture ».

[33] https://www.ivoa.net/documents/DataLink

$[macro:expres(sion)reguliere]

Si par exemple, les tuiles HpxFinder contiennent un champ indiquant la date de l'observation (ex : *{"DATE-OBS" : "1996-11-04T04:30:00"}*) et que vous souhaitez voir s'afficher dans le client HiPS uniquement l'année avec un préfixe constant (ex : *Year 1996*) il vous sera nécessaire d'utiliser la macro suivante qui va extraire du champ « DATE-OBS » tous les chiffres jusqu'au premier tiret, précédé du préfixe « Year »:

```
Year $[DATE-OBS:^([0-9]+)-.*]
```

L'entête du fichier « metadata.xml » déclarera un champ additionnel[34] :

```
20      <FIELD name="Info"/>
```

Et la section donnée du fichier « metadata.xml » fournira le champ supplémentaire :

```
29          <TD>Year $[DATE-OBS:^([0-9]+)-.*]</TD>
```

Ainsi en fonction de l'image original désignée, le client HiPS obtiendra par exemple le résultat suivant :

Mise en œuvre : La mise au point du fichier « metadata.xml » est grandement facilitée en appliquant la méthode suivante : 1 – vérifier avant tout que la syntaxe VOTable est correcte. Pour cela, soit utiliser un validateur VOTable, soit charger directement le fichier « metadata.xml » dans Aladin Desktop pour s'assurer qu'il parvient à le lire ; 2 – Utiliser des expressions régulières les plus simples possibles en les complexifiant par étapes jusqu'à obtenir la chaine souhaitée ; 3 – Ne pas hésiter à créer des colonnes VOTable « cachées » (attribut « type=hidden » dans le FIELD VOTable) et à se servir de ces colonnes pour construire le LINK ou DATALINK approprié.

Génération d'un HiPS à partir d'une unique image couvrant tout le ciel

Certaines missions fournissent leurs observations sous la forme d'une carte en représentation cartésienne couvrant tout le ciel. Elle est souvent codée en format jpeg sans aucune calibration associée. Avec l'option « -hhhcar » et l'indication de l'image unique concernée en paramètre « in », Hipsgen générera automatiquement le fichier de calibration « .hhh » associé à l'image, et lancera le processus de génération du HiPS.

```
-hhhcar in=/data/skymission.jpg out=/data/hips …
```

Image non équatoriale : Dans le cas où l'image utilise un système de référence spatiale non équatorial, il sera nécessaire de l'indiquer par le paramètre frame=*xxx* où *xxx* peut prendre les valeurs « galactic » ou « ecliptic ». Attention ce paramètre va également contraindre le système de référence du HiPS final. Ainsi, dans le cas où vous souhaitez générer un HiPS équatorial à partir d'une image non équatoriale, il faudra procéder en deux étapes. D'abord créer le fichier « hhh » dans le système de référence de l'image source (l'ajout de l'option –n accélérera cette étape en inhibant les actions effectives), puis par une deuxième commande Hipsgen, générer le HiPS équatorial.

```
Etape 1 : -n -hhhcar in=/data/skymission.jpg out=/data/hips frame=galactic
```

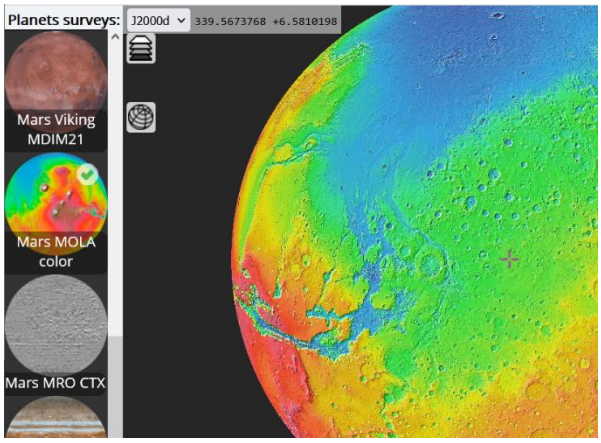
[34] Le nom de la colonne (ici « Info ») est indépendant du nom des macros.

```
Etape 2 : in=/data/skymission.jpg out=/data/hips …
```

Image volumineuse : Dans le cas d'une très grosse image et sur une machine n'ayant pas beaucoup de ressource mémoire, il sera probablement nécessaire de procéder manuellement, par découpage préalable de l'image d'origine en portions plus aisément manipulables. Les fichiers « *hhh* » associés devront être adaptés en fonction.

Génération d'un HiPS planétaires

Originalement prévu pour des relevés du ciel, Hipsgen peut être également utilisé pour générer des HiPS planétaires tant que le corps concerné est raisonnablement sphérique. Cette extension d'usage des HiPS est conforme au standard HiPS 1.0 de l'IVOA même s'il n'est pas décrit spécifiquement. L'utilisation de Hipsgen est similaire à son usage céleste, mais requière quelques adaptations mineures décrites ci-dessous. De même les clients HiPS doivent être légèrement adaptés (cf. Aladin Lite planétaire ci-contre).



Les images de surfaces planétaires utilisent généralement des formats spécifiques, par exemple PDS[35], qui ne sont pas reconnus par Hipsgen. Il est donc nécessaire de convertir ces images dans un format compatible : FITS, JPEG ou PNG. Cette étape peut être très complexe ou très simple suivant la nature des données originales. Le cas le plus simple décrit dans ce document consiste à disposer d'une image cartésienne couvrant la totalité de la surface planétaire. Dès lors la méthode décrite dans le paragraphe précédent : « *Génération d'un HiPS à partir d'une unique image couvrant tout le ciel* » peut immédiatement s'appliquer. Une fois le HiPS généré, il sera nécessaire d'éditer manuellement le fichier « properties » d'une part pour indiquer la planète concernée via un nouveau paramètre « hips_body = *xxx* » et de remplacer le système de référence « hips_frame = equatorial » par celui de la planète. Dans l'attente d'une évolution du document de standardisation IVOA HiPS prenant en compte les planètes, l'usage conseillé est d'utiliser le nom du corps (en anglais, et en caractères minuscules) aussi bien pour la désignation de la planète/corps que de son système de référence spatiale (***sun***, ***mercury***, ***venus***, ***earth***, *moon*, ***mars***, ***ceres***, ***saturn***, *titan, dione, enceladus, iapetus, mimas, rhea, tethys,* ***jupiter***, *callisto, europa, ganymede, io,* ***uranus***, *ariel, miranda, oberon, titania, umbriel,* ***neptun***, *triton,* ***pluto***, *charon…*)

```
10 hips_frame          = mars
11 hips_body           = mars
12 hips_order          = 5
```

Génération d'un HiPS à partir d'une « carte HEALPix »

Certaines missions d'observations, telle PLANCK, fournissent des relevés du ciel sous la forme d'un gros fichier FITS contenant directement les valeurs des pixels en projection HEALPix. C'est ce qu'on appelle une « carte HEALPix ». Hipsgen peut prendre en entrée une telle « carte » à la place d'images classiques. L'action « MAPTILES » est dédiée à ce traitement. L'étape de localisation spatiale

[35] https://pds.nasa.gov

(« INDEX ») est inutile, et le choix du système de coordonnées est contraint par celui utilisé dans la carte HEALPix. Tout comme un HiPS classique, il est possible de générer en plus un jeu de tuiles compressées via les actions « PNG » ou « JPEG ».

```
e.g.: in=/data/map.fits out=/data/hips MAPTILES PNG
```

Il est important de noter que dans un tel cas, Hipsgen conserve strictement les valeurs de la carte d'origine, sans aucun rééchantillonnage ni interpolation.

Génération d'une carte HEALPix à partir d'un HiPS

L'action « MAP » de Hipsgen permet de générer une carte HEALPix à partir d'un HiPS. Une carte HEALPix est un unique fichier FITS qui contient explicitement chaque valeur des pixels HEALPix. C'est un format très similaire au HiPS à cela près qu'il est limité par la taille du fichier (et la RAM nécessaire pour le manipuler). La résolution de la carte HEALPix, déterminée par le NSIDE, est par défaut de 1024. Le paramètre Hipsgen « mapNside=*xxx* » permet d'indiquer une valeur de NSIDE particulière. Pour que cette résolution soit identique au HiPS d'origine elle devrait correspondre à la formule : *mapNside = tileWidth x 2^order*.

```
in="/data/hips" out="/data/map.fits" mapNside=2048 MAP
```

Cette opération n'a d'intérêt que si la carte HEALPix produite conserve une taille raisonnable pour être manipulable. Un NSIDE supérieur à 4096 est en pratique assez peu utilisable.

Génération d'un HiPS à partir des tuiles de plus bas niveau

La construction d'un HiPS peut éventuellement être partiellement réalisée au moyen de code dédié (python, java, C, Rust…). Hipsgen ne sera utilisé que pour finir le travail. Ainsi, si vous êtes en mesure de générer l'ensemble des tuiles de plus bas niveau d'un HiPS en les stockant comme attendu suivant le standard HiPS IVOA dans un répertoire Norder***x*** et dans les sous-répertoires désignés Dir***NN000*** suivant la nomenclature des tuiles Npix***NNNNN***.fits (respectivement .png ou .jpg), vous pourrez reconstruire le reste du HiPS en utilisant l'action « TREE » d'Hipsen. Celle-ci va (re)créer toute la hiérarchie manquante, puis les fichiers additionnels requis.

Notez que par défaut, le HiPS généré utilisera le référentiel galactique[36]. Vous devrez indiquer spécifiquement un autre référentiel via le paramètre « frame=xxx ».

```
Ex: out=/data/hips frame=equatorial TREE
```

Vous pouvez également utiliser cette méthode pour appliquer un traitement externe sur toutes les tuiles d'ordre le plus profond (ex : une convolution). Puis régénérer la hiérarchie des tuiles d'ordres inférieurs au moyen de cette action « TREE ». Dans ce cas vous pouvez accélérer le processus, et contraindre Hipsgen de ne pas construire l'arborescence HiPS la première fois en indiquant dans le paramètre « minOrder » la même valeur que l'ordre le plus profond[37]. Puis après avoir opéré les modifications sur les tuiles, relancer Hipsgen avec cette fois-ci un « minOrder=0 ».

[36] Raison historique liée à la genèse des HiPS avant standardisation IVOA.

[37] Cette astuce va vous faire gagner 10% de temps de génération. En revanche, si vous ne comptez pas modifier les tuiles d'ordre le plus profond, ce procédé en 2 étapes est à éviter car, au contraire, il augmenterait de 6% le temps de génération.

```
Etape1: in=/data/img out=/data/hips minOrder=7 order=7 INDEX TILES
Etape2: traitement des tuiles d'ordre 7
Etape3: out=/data/hips minOrder=0 TREE
```

Génération d'un HiPS cube à partir de plusieurs HiPS

Hipsgen peut être utilisé pour générer un HiPS cube à partir de plusieurs HiPS déjà générés. Pour une telle opération, il est indispensable que tous vos HiPS soient compatibles. Ils doivent partager les mêmes encodages (bitpix, bscale, bzero), le même ordre (order), les mêmes formats et tailles de tuiles.

Pour réaliser une telle opération, vous devez utiliser l'action « CUBE », en indiquant en paramètre « in » la liste des HiPS **séparés par des points-virgules**. Par défaut, les tuiles des HiPS seront dupliquées. Cependant en ajoutant le paramètre « mode=link »[38] vous pouvez demander à Hipsgen de créer des « liens symboliques relatifs » vers les tuiles des HiPS d'origine. Cette méthode est bien plus rapide, mais elle vous oblige à conserver les HiPS d'origines pour que le HiPS cube soit valide. L'identificateur d'un HiPS cube utilise par convention la hiérarchie « …/C/… » en lieu et place du « …/P/… » (ex : id=CDS/**C**/mycube).

```
Ex: in=/data/hips1;…;/data/hipsN out=/data/hipscube id=CDS/C/mycube CUBE
```

Génération d'un HiPS cube à partir de cubes FITS

Hipsgen peut être également utilisé pour générer un HiPS cube directement à partir d'un ensemble de cubes FITS. La génération s'effectue exactement de la même manière que pour un HiPS pixel classique, avec les mêmes actions et les mêmes paramètres. Comme indiqué dans le paragraphe précédent, l'identificateur d'un HiPS cube utilise par convention la hiérarchie « …/C/… ».

Prérequis : L'ensemble des cubes FITS doivent impérativement partager la même profondeur (même nombre de plans) [39].

```
Ex: in=/data/fitscubes out=/data/hipscube id=CDS/C/mycube INDEX TILES …
```

Lorsqu'il s'agit de cubes pointés (dispersés) et non d'une mosaïque de cubes, la part des tuiles « creuses »[40] peut devenir conséquentes car multipliée par la profondeur du cube. Le recours à un mode de compression des tuiles FITS est à envisager si vous souhaitez les conserver (cf. section « Compression des tuiles FITS »).

Notez que les clients HiPS capables de visualiser les HiPS cubes sont encore rares. Contrairement à Aladin Desktop, un client non compatible affichera uniquement le premier plan du HiPS cube.

[38] Sous certaines versions de l'OS Windows, la génération de liens symboliques n'est autorisée que si vous disposez des droits d'administration.
[39] Les HiPS cubes définis par le standard HiPS 1.0 ne considère que le numéro du plan, et non sa correspondance physique (longueur d'onde, échelle temporelle, …). La mosaïque cubique n'effectue pas de rééchantillonnage dans la 3e dimension.
[40] Une tuile « creuses » contient une part prépondérante de pixels non significatifs (valeur « blank »).

Duplication d'un HiPS

La duplication d'un HiPS peut être effectuée par n'importe quel outil dédié à la copie, par exemple « *cp –R /dir .* » pour une copie locale ou à travers le réseau via un « *rsync –Hav host:/dir .* ». Si le HiPS a été installé sur un serveur http, il est également possible d'utiliser une commande « *wget* » sans avoir besoin de s'authentifier sur la machine distante et avec les paramètres requis pour parcourir et copier l'ensemble de l'arborescence.

Pour une copie partielle, par exemple en ne s'intéressant qu'à certains formats de tuiles, ou en écartant les ordres les plus profonds, les outils standards peuvent devenir complexes à paramétrer. L'action « MIRROR » de Hipsgen permet d'effectuer de telles opération de copies partielles ou totales[41]. Pour cette opération, le paramètre « in » renseignera l'URL de base du HiPS disponible sur le serveur http, ou, dans le cas d'une duplication locale le répertoire du HiPS. Le paramètre « format=*xxx* »[42] vous permettra d'indiquer la liste des formats de tuiles souhaités (fits, png et/ou jpeg). Le paramètre « order=*nn* » permettra de limiter la profondeur des ordres HiPS recopiés. Il est également possible de ne prendre en compte qu'une zone du ciel en renseignant le paramètre « region=… » (cf. section « Reprise du calcul de HiPS » ci-avant).

A la fin de l'opération de copie, Hipsgen mettra à jour le fichier « properties » afin d'ajuster les paramètres concernés par la copie (hips_order, hips_format, hips_status) et vérifiera la conformité des clés de vérification pour s'assurer que la copie est conforme.

```
Ex1: in="/data/hips" out="/data/myhips" format=png MIRROR
Ex2: in="http://server/hips" out="/data/myhips" order=4 MIRROR
```

Reprise d'une duplication : En cas d'interruption de la duplication d'une copie distante, Hipsgen MIRROR peut être relancé pour compléter la copie. Par défaut, Hipsgen vérifie uniquement que la taille des tuiles déjà copiées est raisonnable, et si besoin relance la copie de ces tuiles. Le paramètre « fastCheck=false » peut être positionné pour vérifier en plus que la date et la taille est bien conforme à la tuile distance. Cependant, ce test ralentit considérablement la reprise de la copie[43]. A noter que dans le cas d'une copie complète, la clé numérique du HiPS est vérifiée à la fin de la copie ce qui détectera une copie possiblement erronée.

Copie répartie sur plusieurs répertoires : La duplication d'un HiPS peut nécessiter beaucoup d'espace disque. Si le répertoire d'arrivée n'a pas la taille suffisante il est possible de demander à Hipsgen de répartir la copie sur plusieurs répertoires d'arrivée. Ainsi le paramètre « mirrorSplit=size;altpath … » permet d'indiquer un ou plusieurs répertoires alternatifs nécessaires à la copie si le répertoire par défaut ne dispose pas de la taille requise. Par exemple « *mirrorSplit="10g;/data/hips-ext1 200g;/data/hips-ext2"* » entrainera la copie des 10 premiers Go dans le répertoire par défaut « *out* », puis 200Go dans le répertoire alternatif */data/hips-ext1*, et tout le reste dans */data/hips-ext2*.

```
Ex: in=http://server/hips out=/data/myhips mirrorSplit="100g;/data1/althips" MIRROR
```

[41] Hipsgen gère plusieurs threads de copie en parallèle dont il ajuste le nombre en fonction des performances instantanées du réseau. Cette technique donne de bons résultats à la fois sur des réseaux rapides ou sur des réseaux peu fiables.

[42] Précédemment nommé « mirrorFormat » pour les versions Hipsgen antérieures à 12.135

[43] La vérification de la taille et date sur le serveur distant prend souvent autant de temps que la copie elle-même.

Vérification de conformité d'un HiPS

Un HiPS représente un grand nombre de fichiers qui doivent être conformes au standard édicter par l'International Virtual Observatory Alliance.

Conformité au standard : Un premier niveau de vérification d'un HiPS va consister à valider sa conformité, ou non, au standard IVOA. L'action « LINT » d'Hipsgen vérifie chaque élément requis du document standard et produit un rapport de conformité.

```
Ex: out=/data/hips LINT
```

Si votre HiPS a été généré antérieurement au standard HiPS de l'IVOA, l'action « UPDATE » d'Hipsgen vous permettra de le mettre à jour afin qu'il soit conforme au standard en vigueur. Cette opération mettra également en place – si absentes - les clés numériques (checkcodes et DATASUM) que les versions récentes d'Hipsgen utilisent pour vérifier la conformité d'un HiPS (cf. paragraphes suivants).

```
Ex: out=/data/hips UPDATE
```

D'autre part, suite à une copie ou à un déplacement, il peut être utile de pouvoir vérifier la conformité de la copie. Deux techniques complémentaires sont possibles, l'une très rapide, l'autre plus lente mais plus complète.

Vérification rapide : La première méthode consiste à vérifier que l'ensemble des fichiers sont présents, et ont conservé leur taille d'origine. L'action « CHECKCODE », exécutée par défaut par Hipsgen à la fin de la génération d'un HiPS va ajouter le mot clé « hips_check_code » dans le fichier properties en lui associant une clé numérique spécifique à chaque jeu de tuiles (ex : « *png:3137881707 fits:3716611556* »)[44]. Ces clés numériques sont liées au nombre et à la taille de l'ensemble des tuiles de chaque format. Dès lors, après une copie, l'exécution de l'action « CHECK » va recalculer ces clés en fonction des tuiles présentes et vérifier si le résultat correspond bien aux clés enregistrées dans le fichier « properties ». Si ce n'est pas le cas, la copie n'est pas conforme. Cette méthode a l'avantage d'être très rapide, mais elle n'est pas totalement sûre dans le sens où un fichier aurait pu être altéré sans en changer sa taille. Cette éventualité est peu probable dans le cas d'une copie ou d'un transfert, mais envisageable dans d'autres cas d'altérations (permutations involontaires de fichiers FITS de même taille, virus crypteur, …)

```
Ex: out=/data/hips CHECK
```

Vérification complète : La deuxième méthode offre une plus grande fiabilité de vérification car elle va calculer une clé numérique directement liée au contenu des tuiles, et non pas simplement sur leur taille. Lors de la génération des tuiles et suivant le standard FITS, Hipsgen génère une clé DATASUM associée au contenu de chaque tuile FITS, et la conserve dans son entête. Ainsi, après une copie, l'exécution de l'action « CHECKDATASUM » va recalculer chaque DATASUM de chaque tuile FITS et vérifier sa conformité avec ce qui avait été mémorisé au préalable. Si ce n'est pas le cas, la copie des tuiles FITS n'est pas conforme, et ce, même si le nombre et la taille des tuiles sont correctes. Cette méthode est très sûre[45] mais malheureusement une centaine de fois plus lente que la méthode précédente. D'autre part, elle ne prend en compte que les tuiles FITS.

---

[44] L'action CHECKCODE va également mettre à jour dans le fichier « properties » le nombre total de tuiles (hips_nb_tiles) ainsi que la taille totale du HiPS (hips_estsize) fournie en Ko.

[45] La méthode DATASUM même si elle est très fiable, n'est pas pour autant absolument certaine. Par exemple, elle ne peut détecter une permutation de valeurs (2 pixels de 4 octets inversés). Mais une telle probabilité est très faible (sauf à le faire exprès).

```
Ex: out=/data/hips CHECKDATASUM
```

Pour des HiPS générés avec les versions précédentes d'Hipsgen, ou par d'autres outils n'utilisant pas de clé numérique FITS, l'action « UPDATEDATASUM » vous permettra de calculer et mémoriser a posteriori les DATASUM sans pour autant devoir régénérer les tuiles FITS. Cette opération prend approximativement 25% de temps de plus que la vérification elle-même (et approximativement la moitié du temps qu'il aurait fallu pour régénérer toutes les tuiles FITS).

Vérification aléatoire : Afin de lutter contre les virus « rançongiciel », Hipsgen permet également de « sonder » un HiPS au moyen de l'action FASTCHECK pour vérifier rapidement s'il n'aurait pas été altéré ou crypté. La méthode consiste à vérifier aléatoirement la cohérence de certains éléments du HiPS afin d'éviter d'effectuer une sauvegarde d'un HiPS ayant été vérolé. En cas de souci, Hipsgen affiche le message suivante « *ERROR: HiPS is corrupted »

```
Ex: out="/data/hips" FASTCHECK
```

Paquetage des tuiles HiPS (version Hipsgen 12.135 et suivantes)

Le standard HiPS IVOA décrit les méthodes d'accès aux éléments d'un HiPS (tuiles, métadonnées, couverture, etc.) au moyen d'une API HTTP. Dès lors, il peut sembler naturel de stocker physiquement ces éléments sont forme de fichiers individuels dans le système de fichiers du serveur. Cette simplicité de stockage est l'un des atouts de la technologie HiPS. Cependant, il est tout à fait envisageable d'utiliser d'autres moyens de stockages de ces éléments HiPS. Ce peut être une base de donnée chargée de fournir les éléments demandés, ou encore un logiciel spécifique dédié à la génération à la volée des tuiles et autres éléments demandés.

Ainsi, afin de fournir une alternative au stockage des tuiles individuelles et réduire considérablement le nombre de fichiers physiques nécessaires au stockage[46], une fois le HiPS généré, l'action « PACK » remplacera toutes les tuiles de chaque sous-répertoire « **Dir*NNNN*** » par un unique fichier binaire contenant l'agrégation des tuiles concernées et cela pour chaque format de tuiles disponibles.

```
Ex: out=/data/hips PACK
```

Si besoin est, l'action « UNPACK » permettra de rétablir la distribution originale[47].

```
Ex: out=/data/hips UNPACK
```

Ces deux actions peuvent être contrôlées par le paramètre « format=… » pour préciser quels types de tuiles sont concernées. La valeur de ce paramètre prendra la forme d'une suite de mots séparés par des espaces : *fits* - les tuiles FITS, *png* - les tuiles PNG, *jpeg* - les tuiles JPEG, *index* - les tuiles de l'index HpxFinder. A défaut, tous les formats représentés seront pris en compte.

```
Ex: out=/data/hips format="jpeg index" PACK
```

Temps d'exécution et reprise : Les actions « PACK » et « UNPACK » requière la lecture et l'écriture complète de l'ensemble des tuiles concernées. Il s'agit d'une opération assez longue même si Hipsgen débutera le processus par une estimation du nombre de threads le plus adapté. En cas

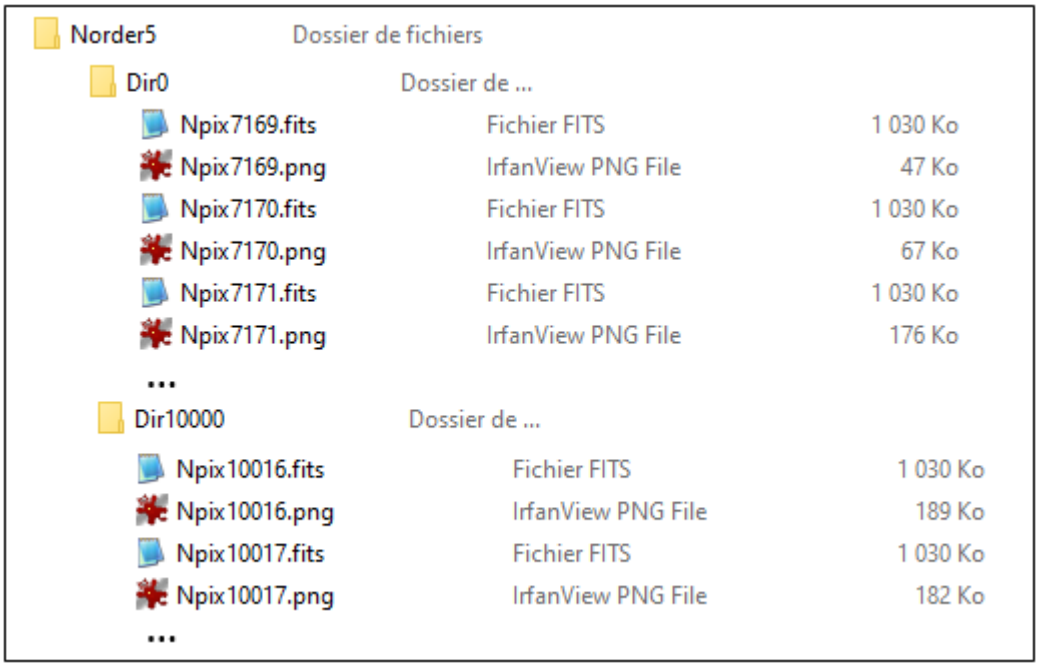
[46] Un HiPS « packé », notamment lorsqu'il concerne un très grand nombre de tuiles, offre une plus grande souplesse de manipulation système (copie, déplacement, sauvegarde, etc) et réduit la place perdue engendrée par les blocs non totalement utilisés du système de fichiers (de l'ordre de 4 à 6% sur la totalité du stockage).
[47] Un HiPS « packé » ne peut être mise-à-jour directement.

d'interruption avant terme il est toujours possible de relancer l'action, ou d'annuler le travail déjà effectué en utilisant l'action complémentaire.

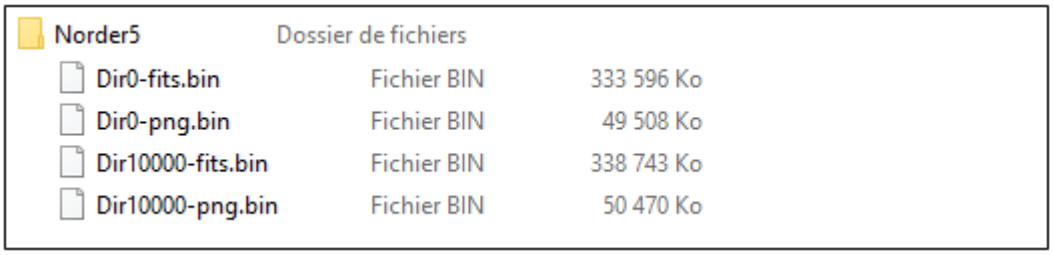
Nom des fichiers binaires : Les fichiers binaires générés par « PACK » reprennent en préfixe le nom du répertoire auquel est ajouté le suffixe « -ext.bin » où « ext » correspond au format (fits, png, jpg).

```
Ex: Dir10000-png.bin
```

Exemple : Structuration des répertoires DirNNNN avant l'action « PACK »



Le même HiPS après l'action « PACK » :



Format des fichiers binaires : Les fichiers binaires sont constitués de deux parties : « l'index » et « les tuiles ». « L'index » fait exactement 80 000 octets. Il mémorise les positions des 10 000 tuiles potentiellement concernées qui vont suivre, codées sur des entiers longs (8 octets, big endian). Chacun de ces index mémorise la position de l'octet qui suit immédiatement la tuile concernée. A la suite de l'index, les tuiles sont stockées, telles que, les unes après les autres dans l'ordre numérique croissant des indices des tuiles.

Extraction d'une tuile : L'algorithme pour extraire une tuile **M** de format **ext** d'un fichier binaire utilisera les fonctions suivantes pour localiser et extraire la tuile en question :

```
BinFile = "Dir".(M/10000)*10000. "-ext.bin"        48
NumIndex = M % 10000                               49
FirstByte = IndexValue(NumIndex-1)                 50
TileSize = IndexValue(NumIndex) - FirstByte        51
```

Publication/distribution : La distribution d'un HiPS « *packé* » nécessitera[52] l'utilisation d'un petit CGI (ou équivalent) pour que le serveur http puisse extraire les tuiles demandées des fichiers binaires[53]. Le code Perl ci-dessous en propose une implémentation basique.

```perl
#!/bin/perl

# hipstile.pl
# v1.0 - May 2024 - P.Fernique [CDS]
#
# Extracts the specified tile from a packed HiPS on the standard output
# A packed HiPS is generated thanks to "Hipsgen PACK" command, by grouping HiPS tiles in aggregated packed files.
#
# Command line ex: hipstile.pl path/Norder5/Dir10000/Npix10025.fits


($order,$dir,$npix,$ext)= $ARGV[0]=~/.*Norder(\d+)\/Dir(\d+)\/Npix(\d+)\.?(.*)/;
($path)= $ARGV[0]=~/^(.*Dir\d+)\/.*/;

# Open associated packed file
$file = "$path-$ext.bin";
sysopen($f, $file, 0) or die "HiPS packed file not found [$file]!\n";

# Read start and end tile positions from the packed file index
$n = $npix % 10000 -1;
$start = 80000;
if( $n>=0 ) {
   sysseek($f,$n*8,0);
   sysread($f,$start,8);
   $start=unpack("Q>",$start);
   $n++;
}
sysseek($f,$n*8,0);
sysread($f,$end,8);
$end=unpack("Q>",$end);
$size = $end-$start;

# Read the tile, and write it on the stdout
exit 0 if $size<0;
sysseek($f,$start,0);
sysread($f,$buf,$size);
print $buf;
```

Les "traces" de Hipsgen

Hipsgen est un outil très bavard qui fournit une foultitude de renseignements lors du traitement. Appréhender les grandes étapes du processus est un vrai plus pour utiliser efficacement cet outil.

Lancement

[48] Division entière
[49] Modulo
[50] *IndexValue(n)* retourne la valeur de l'index à la position *n*
[51] Une tuile non présente aura une taille nulle *(TileSize=0)*
[52] Un HiPS « packé », stocké localement reste directement visualisable par le client Aladin Desktop (v12.135 et suivantes). Ce client HiPS sait extraire les tuiles directement des fichiers binaires.
[53] Il faudra compter une vingtaine de millisecondes en plus pour extraire et publier une tuile comparativement à un accès fichier traditionnel.

```
INFO  : Starting HipsGen 20/12/22 19:46:21 (based on Aladin v12.023)...
OPTION: in=HalphaNorth
OPTION: out=hips
INFO  : Action => INDEX: Build spatial index (in HpxFinder directory) + MOC index
INFO  : Action => TILES: Build all true value pixel tiles + Allsky + MOC
INFO  : Action => PNG: Build all preview tiles (PNG) + Allsky.png
INFO  : Action => CHECKCODE: Compute+store the check codes (and the size) associated to the target HiPS
INFO  : Action => DETAILS: Adapt HiPS index for supporting the "detail table" facility
```

Indexation spatiale – INDEX

```
RUN   : ============================================== INDEX  ==============================================
INFO  : Partitioning large original image files in blocks of 4096x4096 pixels
INFO  : Output directory: .\hips
INFO  : Pre-existing HpxFinder index => will add new images only...
INFO  : Use this reference image => HalphaNorth\And03_Ha.fits
INFO  : Max Order=3 => Pixel angular resolution=51.53"
INFO  : Tile Order=9 => tile size: 512x512 pixels
INFO  : MEF strategy => extension 0, otherwise 1
INFO  : Extended metadata extraction based on FITS keys: DATE, TELESCOP, DATE-OBS, EXPTIME
INFO  : HiPS coordinate frame => equatorial
.
STAT  : 26 files in 893ms => 3,67Mpix using 14,85MB => biggest: [385x385 x4]
...
INFO  : Max original image overlay estimation (4096x4096 pixel blocks from original images): 16 => may required 1
GB per thread
INFO  : MOC Index done in 15ms: mocOrder=3  frame=C size=624B
STAT  : 113 files in 4s 751ms => 23.8/s => 15,97Mpix using 64,56MB => biggest: [385x385 x4]
DONE  : INDEX done (in 4s 753ms)
```

Génération des tuiles FITS – TILES

```
RUN   : ============================================== TILES  ==============================================
INFO  : Output directory: .\hips
INFO  : Reference image: HalphaNorth\And03_Ha.fits
INFO  : Set default target => 345.73718 +37.6812
INFO  : BITPIX found in the reference image => -32
INFO  : Data range [-601.8 .. 1783], pixel cut [-4.691 .. 402.8]
INFO  : Overlay mode (progenitors): OVERLAY_MEAN: Mean of the progenitor pixel values
INFO  : Merge mode (tiles): MERGE_OVERWRITE: Replace existing pixel values if the new value is not BLANK
INFO  : Hierarchy mode (tree): TREE_MEAN: The mean of the 4 sublevel pixel values
INFO  : Frame (HiPS coordinate reference frame) => equatorial
INFO  : Creating 257 .fits tiles (order=3)...
INFO  : BITPIX = -32 (no conversion)
INFO  : BLANK=NaN
INFO  : Building tiles thanks to 8 threads
INFO  : Available RAM: 7,89GB => Cache size: 300 items / 5,26GB
.........................
STAT  : 160+10/257 tiles + 77 nodes in 26s 536ms (66.1%) by 8/8 threads
STAT  : Cache: 190 items/300 using 108,11MB/5,26GB freeRAM=7,38GB (opened=190 reused=806 released=0)
......................
STAT  : Tile overlay stats : max overlays=16, 254 in one step, 0 in multi steps
STAT  : Tiles trim method saves 0KB
STAT  : Pixel times: Original images=15,97Mpix => 333,81Kpix/s  Low tiles=61,25Mpix (x3.834) => 1,25Mpix/s
INFO  : MOC generation based on fits tiles (deep resolution mocOrder=7)...
...
INFO  : MOC done in 2s 174ms: mocOrder=7 size=25,38KB
INFO  : Creating Allsky...
INFO  : properties & index.html files created or updated
DONE  : ALLSKY file done
DONE  : TILES done (in 52s 402ms)
```

Génération des tuiles PNG

```
RUN   : ================================================ PNG  ================================================
INFO  : Output directory: .\hips
INFO  : Hierarchy mode (tree): TREE_MEAN: The mean of the 4 sublevel pixel values
INFO  : Map pixel cut [-4.691 .. 402.8] to [1..255] (linear)
INFO  : Building tiles thanks to 8 threads
.....
STAT  : 245/257 tiles in 2s 423ms (95.3%)  endsIn:118ms
INFO  : properties & index.html files created or updated
STAT  : 245/257 tiles in 2s 423ms (95.3%)  endsIn:118ms
DONE  : PNG done (in 4s 839ms)
```

Génération des codes de sécurité – CHECKCODE

```
RUN   : ============================================== CHECKCODE  ==============================================
INFO  : Output directory: .\hips
INFO  : Scanning png tiles...
INFO  : 356 png files for 28,68MB
INFO  : Check code for png tiles: 3137881707
INFO  : Scanning fits tiles...
INFO  : 356 fits files for 369,1MB
INFO  : Check code for fits tiles: 3716611556
INFO  : Full HiPS size: 397,78MB
INFO  : Check codes: png:3137881707 fits:3716611556
INFO  : Check codes and HiPS metrics stored/updated in properties file
STAT  : 712 files scanned
DONE  : CHECKCODE done (in 80ms)
```

Génération des tuiles des liens vers les progenitors – DETAILS

```
RUN   : ============================================== DETAILS  ==============================================
INFO  : Output directory: .\hips
INFO  : Order retrieved from HpxFinder => 3
INFO  : Detail table min order determined by "minOrder" parameter => 0
INFO  : Mapping hpxFinder/metadata.xml file has been generated
STAT  : 257/257 tiles in 117ms (100.0%)  endsIn:
DONE  : DETAILS done (in 127ms)
INFO  : Tip: Edit the "properties" file for describing your HiPS (full description, copyright, ...)
DONE  : ================================== THE END (done in 1m 2s)  ==================================
```

Le fichier « properties » à la loupe

Le fichier « properties » contient toutes les métadonnées du HiPS : l'origine de ces images, des éléments techniques liés à la génération du HiPS, sa description, l'historique des traitements opérés ainsi que des informations destinées aux clients. Certains champs ont un vocabulaire contrôlé à consulter dans le document de standardisation HiPS IVOA, d'autres sont libres. Voici un exemple de fichier properties assez complet, commenté…

Descriptif et origine : Les champs regroupés ci-dessous donnent l'identificateur du HiPS, son titre (une ligne), la collection de données d'où il est issu, une description assez complète (un petit paragraphe), le titulaire des droits, une suggestion de mention de crédit, la longueur d'onde, une référence bibliographique, et enfin le type de données.

```
creator_did          = ivo://CDS/P/2MASS/K
obs_title            = 2MASS K (2.16um)
obs_collection       = The Two Micron All Sky Survey - K band (2MASS K)
obs_description      = 2MASS has uniformly scanned the entire sky in three ...
obs_copyright        = University of Massachusetts & IPAC/Caltech
obs_copyright_url    = http://www.ipac.caltech.edu/2mass/
obs_ack              = IPAC/NASA
prov_progenitor      = Caltech
bib_reference        = 2006AJ....131.1163S
bib_reference_url    = http://simbad.u-strasbg.fr/simbad/sim-ref?bibcode=2006A
dataproduct_type     = image
```

Couvertures et résolutions originales : Les lignes suivantes regroupent les champs de couvertures : la longueur d'onde, la plage temporelle et la surface du ciel concernés. Les deux champs suivants donnent la taille angulaire d'un pixel original et son mode de codage.

```
obs_regime          = Infrared
em_min              = 2.015E-6
em_max              = 2.303E-6
t_min               = 50600
t_max               = 51941
moc_sky_fraction    = 1
s_pixel_scale       = 2.777E-4
data_pixel_bitpix   = -32
```

Champs de l'origine et des droits du HiPS : Les champs commençant par « hips_ » concernent spécifiquement le HiPS associé au fichier de « properties ». Ainsi sont indiqués en premier lieu les informations de créations : outils, numéro du standard, dates, personnes, titulaire des droits.

```
hips_builder        = Aladin/HipsGen v11.023
hips_version        = 1.4
hips_creation_date  = 2013-01-14T09:45Z
hips_release_date   = 2021-02-23T01:29Z
hips_creator        = CDS (Oberto A.)
hips_copyright      = CNRS/Unistra
hips_status         = public master clonableOnce
```

Champs techniques du HiPS : Viennent ensuite les éléments techniques décrits en détail dans ce présent manuel.

```
hips_pixel_bitpix   = -32
hips_frame          = equatorial
hips_order          = 9
hips_order_min      = 0
hips_tile_width     = 512
hips_tile_format    = jpeg fits
```

Champs des modes de génération : les champs suivants fournissent les informations sur les méthodes employées pour générer le HiPS.

```
hips_sampling       = bilinear
hips_overlay        = overlayMean mergeOverwriteTile treeMean
hips_skyval_method  = SKYVAL
hips_skyval_value   = -0.5 100.0 -2660.728 9046.069
hips_pixel_cut      = -0.5 40
hips_data_range     = -2661 9046
```

Champs de mesures : les champs suivants fournissent la taille angulaire du pixel HiPS, le volume total du HiPS (en Ko), le nombre de tuiles, les codes numériques de validation et un champ de vue suggéré par défaut au client.

```
hips_pixel_scale    = 2.236E-4
hips_estsize        = 4706425658
hips_nb_tiles       = 315200120
hips_check_code     = fits:3568836873 jpg:4537812981
hips_initial_fov    = 58.63230142835039
hips_initial_ra     = 266.40499479
hips_initial_dec    = -28.936173970
```

Historique du traitement : A chaque utilisation d'Hipsgen sur ce HiPS, la date et les paramètres de la ligne de commande – fichier de paramètres inclus - sont mémorisés pour fournir un historique des traitements.

```
46 hipsgen_date          = 2020-08-04T10:56Z
47 hipsgen_params        = cache=CACHE-TODEL-K cacheRemoveOnExit=true in=2MASSk
48 hipsgen_date_1        = 2020-10-04T11:03Z
49 hipsgen_param_1        = out=Hips-K3 creator_did=ivo://CDS-test/P/2MASS/K UPD
```

Champs destinés aux clients : Le fichier « properties » peut contenir des champs à l'attention des clients, voire d'un client spécifique. Ci-dessous, le champ « client_category » est destiné à Aladin Desktop afin qu'il puisse présenter le HiPS dans la branche adéquate de son arbre des ressources.

```
50 client_category       = Image/Infrared/2MASS
```

Les options, actions et paramètres d'Hipsgen

Les options

```
   -clean  : Delete previous computations
   -n      : Just print process information, but do not execute it
   -color  : Colorize console log messages
   -nocolor: Uncolorize console log messages
   -nice   : [MIRROR] Slow download for avoiding to overload remote http server
   -notouch: Do not touch the hips_release_date
   -hhhcar : [INDEX] Generate hhh file for an all sky image
   -trim   : [TILES,CONCAT,APPEND] Trim FITS tiles if possible
   -gzip   : [TILES,CONCAT,APPEND] Gzip FITS tiles
   -d      : Debug messages
   -h      : Inline help
   -man    : Full inline man (may be followed by a parameter or an
             action for a full explanation)
```

Les actions

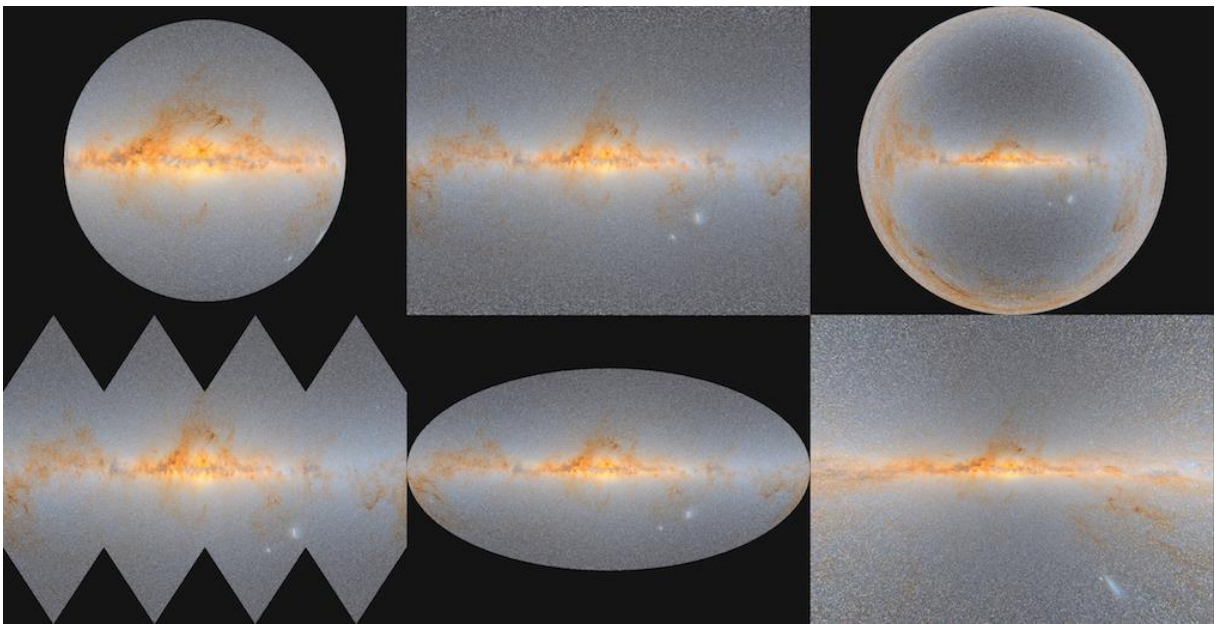
```
(by default: "INDEX TILES PNG CHECKCODE DETAILS"):
   INDEX        : Build spatial index
   TILES        : Build all true value pixel tiles
   PNG          : Build PNG preview tiles
   JPEG         : Build JPG preview tiles
   MOC          : Regenerate the HiPS spatial coverage (MOC)
   MAP          : Build an HEALPix map from the HiPS tiles
   ALLSKY       : (Re)build all Allsky files
   CLEAN        : Delete all HiPS files (except properties file)
   CLEANINDEX   : Delete spatial index
   CLEANFITS    : Delete all FITS tiles
   CLEANJPEG    : Delete all JPEG tiles
   CLEANPNG     : Delete all PNG tiles
   CLEANWEIGHT  : Delete all WEIGHT tiles
   CLEANCOUNT   : Delete counting tiles (HpxCounter dir)
   TREE         : (Re)build HiPS hierarchy from existing tiles
   APPEND       : Append new images to an already existing HiPS
   CONCAT       : Concatenate two HiPS
   CUBE         : Create a HiPS cube based on a list of HiPS
   DETAILS      : Extends HiPS spatial index for supporting 'progenitor' facility
   STMOC        : Build a STMOC.fits based on HpxFinder tile descriptions
   UPDATE       : Upgrade HiPS metadata additionnal files to the last HiPS standard
   CHECKCODE    : Compute and store the check codes
   MIRROR       : Duplication of a HiPS (local or remote)
   RGB          : Build and RGB HiPS based on 2 or 3 other HiPS
   CHECK        : Basic HiPS integrity check
   CHECKDATASUM : HiPS FITS tiles full integrity check
   CHECKFAST    : Fast HiPS integrity test
   LINT         : Check HiPS IVOA 1.0 standard compatibility
   MAPTILES     : Build HiPS tiles from a HEALPix FITS map
   COUNT        : Build progenitor counting HiPS
```

Les paramètres

```
   in=dir              : Source directory
   out=dir             : Ouput directory
   order=nn            : HiPS order
   minOrder=nn         : HiPS min order
   frame=equatorial|galactic|ecliptic: HiPS coordinate frame
   tileWidth=nn        : HiPS tile width
   bitpix=8|16|32|64|-32|-64: HiPS bitpix
   dataRange=min max   : Original pixel range
   pixelCut=min max [fct]: 8 bits pixel mapping
   img=filename        : Reference image for default initializations
```

```
hdu=n1,n2-n3,...|all: List of FITS HDU numbers (original images)
blank=nn|key        : Alternative BLANK value (or alternate BLANK fits keyword)
validRange=min max  : Range of valid pixels
skyVal=key|auto|%info|%min %max: Background removal method
expTime=key         : Method of adjusting the exposure time
maxRatio=nn         : Image source pixel ratio test
fov=true|x1,y1..    : Masks on the original images.
border=nn|N W S E   : Edge removal
shape=rectangle|ellipse: Image FoV
mode=m1,m2..        : Coadd pixel modes
incremental=false|true: Incremental HiPS
region=inline moc|moc.fits: Working region
partitioning=false|nnn: Splitting large original images into blocks
maxThread=nn        : CPU thread limitation
fastCheck=true|false: Mirror check method
fitsKeys=key1,key2...: FITS keywords for image characteristics extraction
id=AUTH/P/...       : HiPS identifier
title=title         : HiPS title
creator=name        : HiPS creator
target=ra dec [rad] : Default HiPS target
status=private|public [clonable|clonableOnce|unclonable]: HiPS status
color=jpeg|png      : Tile format of a colour HiPS
inRed=hipspath      : Red HiPS path
inGreen=hipspath    : Green HiPS path
inBlue=hipspath     : Blue HiPS path
cmRed=min [mid] max [fct]: Red color mapping
cmGreen=min [mid] max [fct]: Green color mapping
cmBlue=min [mid] max [fct]: Blue color mapping
luptonQ=x           : Q coef Lupton RGB builder
luptonS=x/x/x       : Scale coefs Lupton RGB builder
luptonM=x/x/x       : M coefs Lupton RGB builder
cache=dir           : Alternative cache directory
cacheSize=nnMB      : Alternative cache size limit
cacheRemoveOnExit=true|false: Removing cache disk control
mocOrder=nn [<nnMB] : Specifical MOC order an/or size limit
mapNside=nn         : HEALPix map NSIDE
format=fmt1 fmt2 ...: Tile formats
mirrorSplit=size;altPath ...: Multi disk partition split
pilot=nn            : Pilot HiPS for testing
```

Table des matières



Hipsgen - Manuel de l’utilisateur
Version de janvier 2023 – augmentée