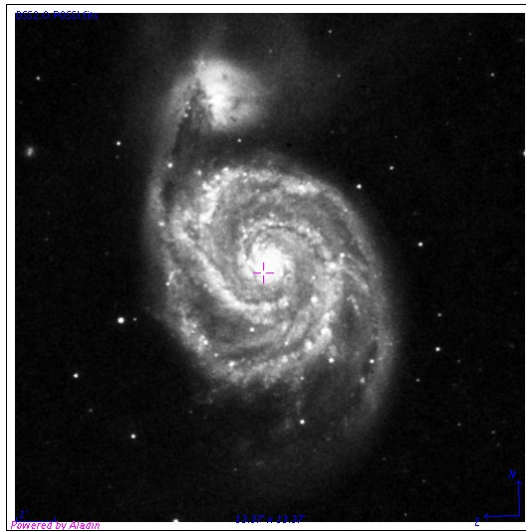


Compte rendu 19 décembre

J'ai écrit une partie du code permettant de générer notre base de données d'images au format « pseudo-healpix » que nous avons défini à la réunion précédente.

Supposons que nous ayons l'image suivante 768*768 avec environ 1pixel = 1seconde (13.37arcmin de coté) :



Analyses préalables pour le format Healpix :

Dans un premier temps nous allons déterminer la « taille » sur le ciel du fichier healpix voulu.

A un NSIDE fixé à 2^8 nous aurons 12×256 fichiers qui quadrilleront tout le ciel, et dont le coté fera 0.46 degrés (soit 27arcmin).

Grâce aux méthodes fournies dans la librairie healpix nous pouvons connaître le fichier qui contiendra la position centrale de notre image (appelé N_file).

Dans ce fichier définissant une zone (de forme losange), nous allons fixer la résolution que nous souhaitons : 1024×1024 . Ce qui nous amènerait à des pixels healpix avec un $NSIDE = 2^{18}$ (car $2^8 \times 1024 (=2^8 \times 2^{10}) = 2^{18}$).

Malheureusement, les librairies healpix ne nous permettent pas de dépasser $NSIDE = 2^{13}$, puisqu'au delà nous aurions à manipuler des valeurs au dessus de 4 milliards (le nombre maximum de pixels contenu dans une sphère healpix est de : $12 \times 4^{\text{ORDER}}$, avec ORDER l'ordre auquel on élève 2 dans le NSIDE). L'évolution de la librairie vers la possibilité de manipuler des valeurs plus grande n'est pas du tout prévue par les développeurs de la librairie. Nous allons donc nous limiter pour l'instant avec 2^{13} tout en gardant à l'esprit que nous ne voulons pas nous y contraindre.

Nous pouvons rajouter, qu'actuellement ces valeurs sont limitées à cause de la taille

des conteneurs à 4 octets, mais passer à 8 octets, ne fait que repousser cette limite à un NSIDE=2³⁰.

Pour le reste des explications nous supposons que NSIDE=2¹³ est la valeur optimale pour l'affichage que nous souhaitons faire.

Description des développements côté serveur :

```
// Pour chaque pixel du losange sélectionné sur la sphère healpix
// (théoriquement nous en aurions 1024*1024, mais ici nous sommes limités
// soit 213/28 = 1024 pixels)
array[] losange;
for (index in healpix_file) {
    // nous calculons ses coordonnées grâce aux librairies healpix
    coordinates = Healpix.pix2ang(index);
    // nous recherchons la valeur de l'image originale pour ces coordonnées
    // (si elle existe)
    if (fits_file.contains(coordinates)) {
        value = fits_file.get(coordinates);
    } else { value = 0; }
    // Ces pixels sont stockés dans une structure temporaire de la taille du losange
    // healpix et dans l'ordre défini par les librairies healpix.
    losange[index] = value;
}

// Toutes ces valeurs de pixels sont ensuite enregistrées dans un fichier FITS, où l'on
// enregistre également le numero du fichier healpix calculé (N_file) et le
// NSIDE associé (ici 8).
writeFits(losange, healpix_file.NSIDE , healpix_file.N_file);
```

Maintenant, nous nous plaçons au moment où le serveur soit relire sa base de données pour envoyer au client la zone intéressante. A un zoom donné (ici on va supposer que nous avons besoin d'une résolution de fichier à 2⁸), il cherche alors quel fichier contient les coordonnées demandées et doit reprojeter l'image stockée.

Bien entendu, il va souvent correspondre 4 fichiers healpix pour une vue donnée, nous devront donc effectuer le traitement suivant pour ces 4 images healpix.

```
// Pour chaque pixel de l'image de la vue demandée par le client
for (y < height) {
```

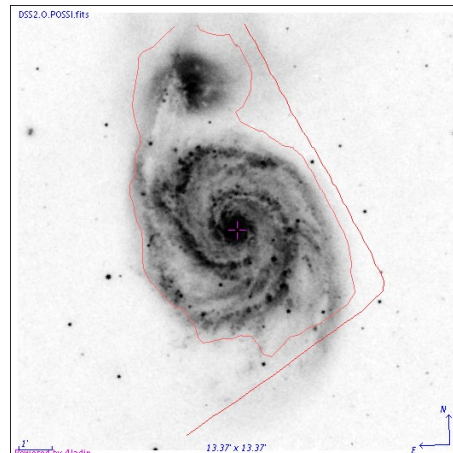
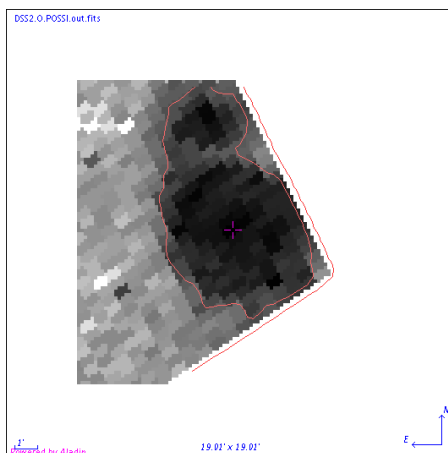
```

for (x < width) {
    // nous calculons ses coordonnées
    coordinates = view.getCoordinates(x,y);
    // grâce aux librairies healpix nous retrouvons l'index du pixel associé
    index = Healpix.ang2pix(coordinates);
    if (healpix_file.contains(index)) {
        value = healpix_file.get(index);
    } else { value = 0; }
    fits_file.setValue(x,y,value);
}
}

```

Sur les figures suivantes, on peut voir à gauche, la reprojexion d'un fichier image healpix (en forme de losange) sur une zone voulue (rectangulaire) ; On reconnaît la galaxie sur le bord du losange.

Sur l'image de droite, on voit en superposition en rouge la forme de la galaxie provenant de l'image healpix et du bord du fichier image.



Evolutions :

L'idée pour la suite, est depuis la couverture d'un survey, on sélectionne les fichiers healpix. Puis on cherche à remplir chacun des fichiers healpix de façon indépendante.

Afin d'optimiser le traitement, nous allons chercher l'image contenant un premier pixel, traiter toute cette image pour la remplir dans ce fichier, et ensuite pour tous les pixels non encore définis, de rechercher s'ils existent dans une autre image, etc.... Ainsi, dans le cas où des images se recouvriraient, on ne garderait que la première valeur trouvée.