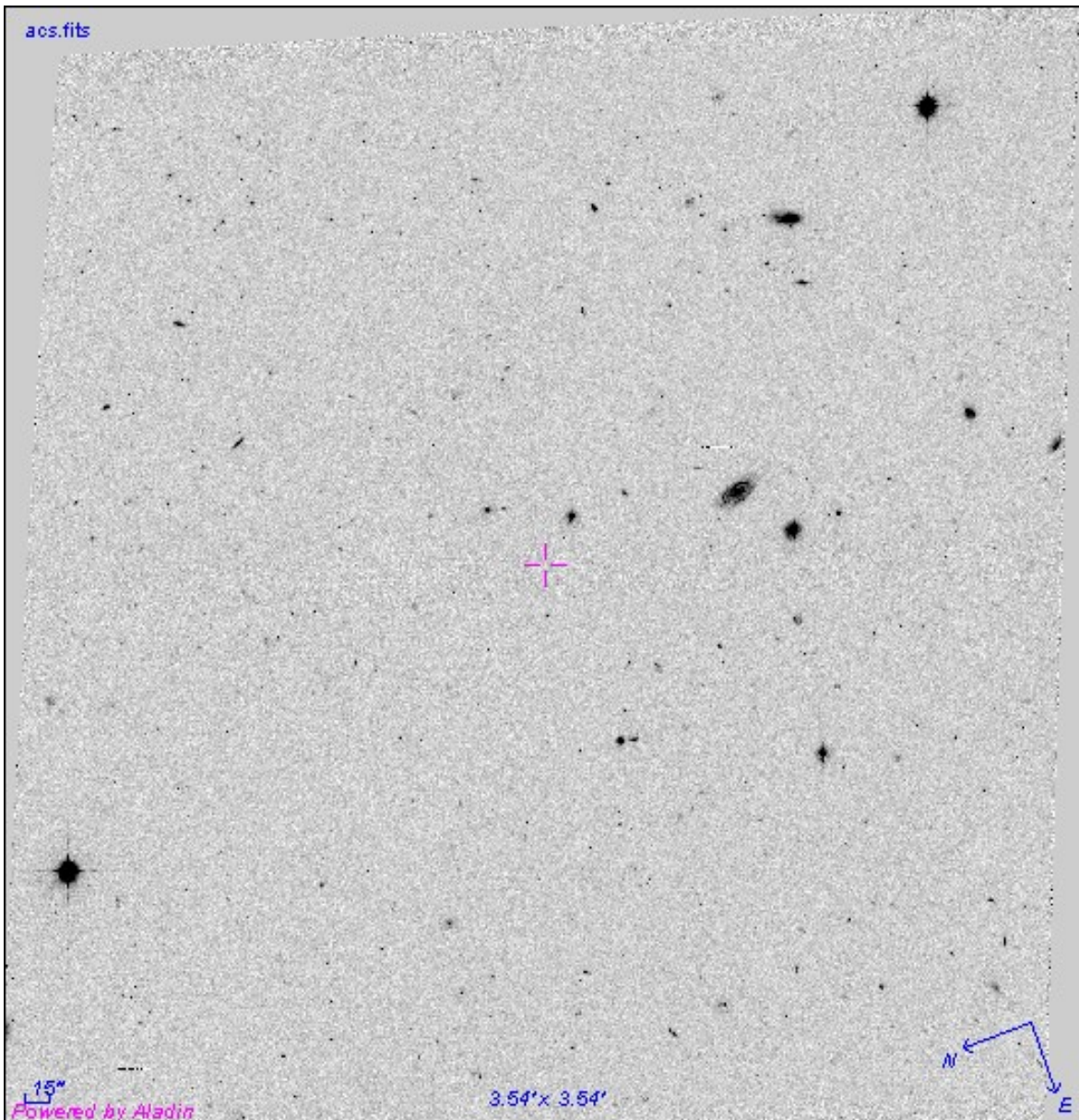


Etat d'avancement – Août 2007

Présentation de l'image pour les tests :

4238 x 4354 pixels (32 bits)

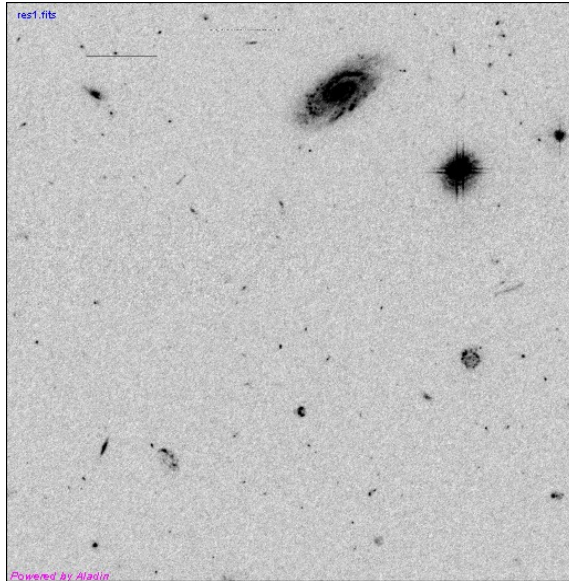


1 - Comparaison des méthodes

Conditions de la mise en page : affichage en zoom x4 pour mettre en avant les différences, application du filtre automatique d'Aladin pour la répartition des niveaux de gris.

L'image en pleine résolution en 8 bits fait 18M. Les images en plus basse résolution générées sont de longueur et largeur originales / 8 = 529 x 544, soit 285K.

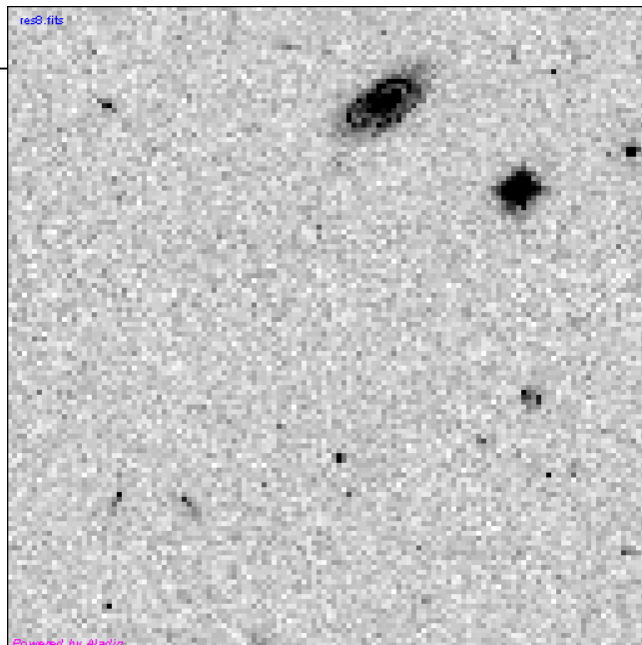
Image en pleine résolution, transformée en 8 bits
(4238 x 4354)



Images basses résolutions : 529 x 544

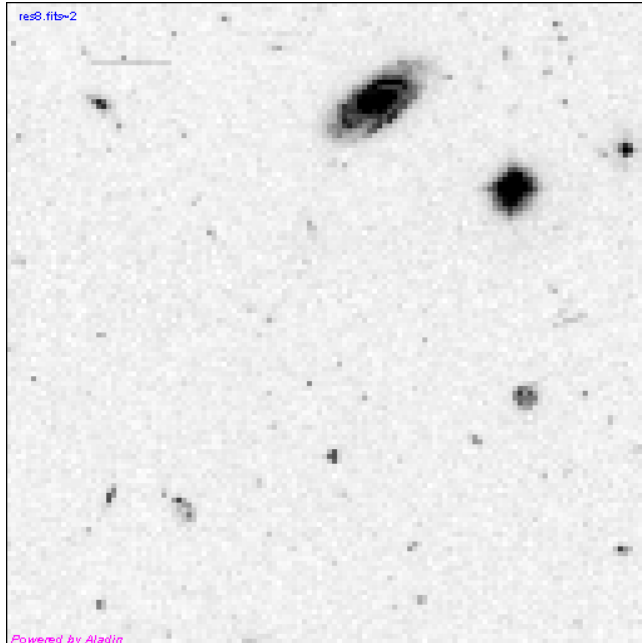
Méthode 1 : 1 pixel sur n
(garde le premier)

On reconnaît grossièrement les
grosses structures.



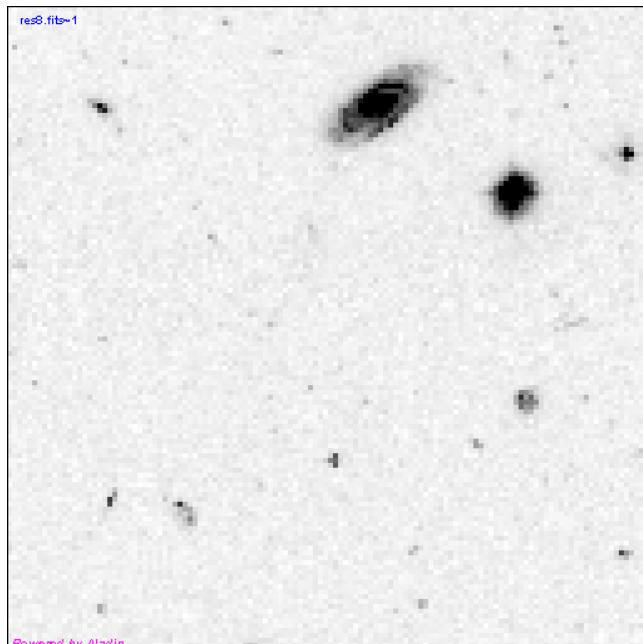
Méthode 2 : moyenne

L'image est globalement plus claire, car le bruit de fond de l'image (ce qui rend l'image originale plus « grise ») disparaît avec le moyennage.



Méthode 3 : médiane

Cette méthode donne un résultat très proche de la moyenne. On peut quand même noter la disparition (ou atténuation) des étoiles les plus petites.



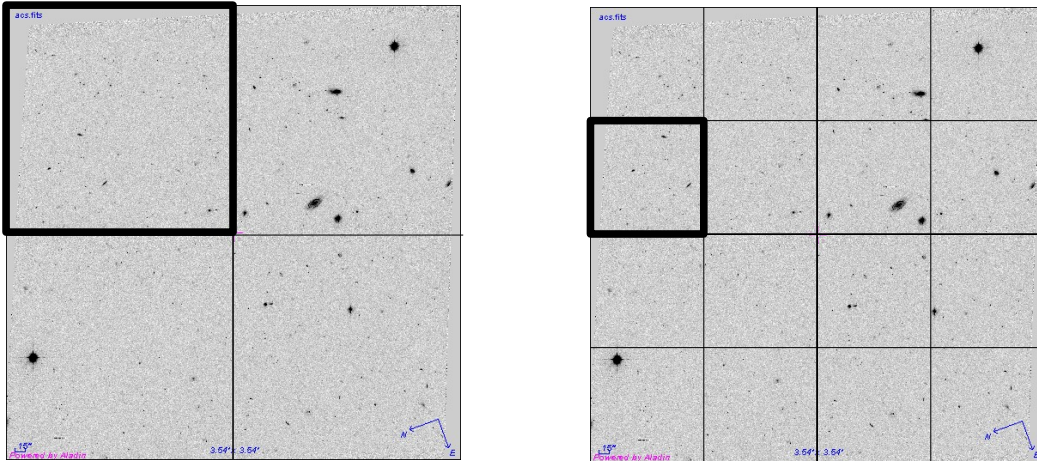
Conclusion :

La méthode 1 correspond mieux aux besoins, dans le sens où elle est simple et rapide à générer, et offre une répartition des niveaux de gris assez similaires (ce qui permet d'éviter d'avoir à recalculer la fonction de répartition pour un affichage optimal à chaque nouvelle résolution).

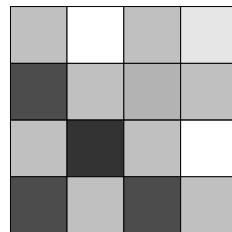
2 - Description des méthodes :

Conversion de l'image originale en 8 bits

Découpe des pixels correspondants à la zone voulue (*exemple : pour les images en résolution 4 : on découpe l'image en 4 => on obtient une image de 2119 x 2177*)



Sélection des pixels utilisés pour calculer le nouveau pixel (*exemple : zone de pixels 4 x 4 transformée en 1 valeur => on obtient une image de 529 x 544*). La méthode choisit soit le premier pixel, soit de faire une moyenne des pixels de cette zone, soit de garder la médiane.

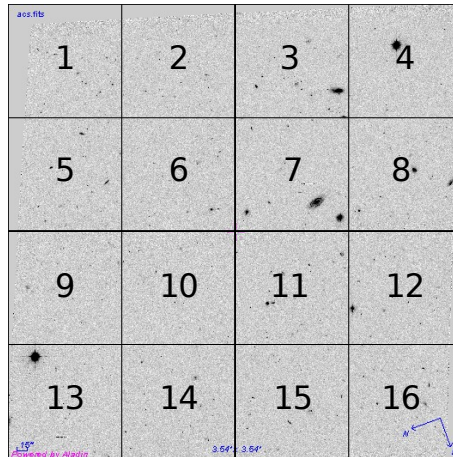


3 - Client / Serveur

Le client demande au serveur son image à une position donnée dans le ciel.

Pour la démo, dès que l'utilisateur demande à zoomer, le client calcule la position de la souris dans l'image et calcule le numéro de zoom (ou de résolution) cherché et le numéro de l'imagette sous la souris.

Exemples : zoom x4



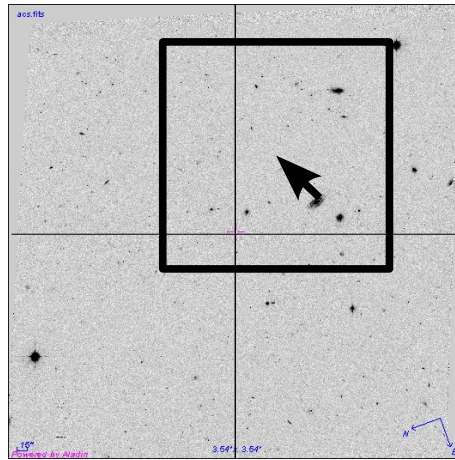
La base de données contient des images dont les noms sont composés du nom du fichier original + numéro de résolution + numéro de l'imagette dans l'image globale.

Ce système permet d'éviter au client de redemander au serveur les numéros d'imagettes dans la vue courante, mais de le savoir lui même.

Dans le cas où l'on change d'image originale, il faut à nouveau demander au serveur son nom.

4 - Evolution

Si l'on veut centrer là où l'utilisateur demande -> zoom, mais cela donne une mauvaise qualité



2 solutions :

=> prévoir une qualité supérieure à celle affichée pour avoir une liberté (calcul à ajuster selon chaque résolution des images originales)

=> charger toutes les imageries (ici les 4 concernées). A ce moment, c'est le visualiseur (le client) qui s'occupe de les recoller et d'en afficher que la zone demandée autour de la souris.

Nous choisissons la deuxième solution et en même temps, il serait utile de prévoir un sous découpage (en 4) pour charger des imageries plus petites (donc avec un temps de transfert plus rapide) mais également plus souvent.

On peut imaginer proposer un zoom de mauvaise qualité en attendant le chargement des imageries.

Nous soulevons le problème de la génération d'une image plus large que celles originales (par exemple pour afficher une vue de 6 minutes, soit 2 images). Nous avons donc le problème de savoir quels pixels choisir sur les bords des images et comment les projeter pour avoir un affichage sans distorsions.

Une solution serait d'utiliser des formats de fichiers déjà proposés en astronomie pour s'abstraire de ces problèmes. La prochaine étape est donc de tester l'utilisation du format HTM et Healpix.