SAÉ S3.C2 : Réduction d'étoiles en astrophotographie

Rémi Cozot, Samuel Delepoulle et Claire Guilloteau

Décembre 2025

1 Introduction et mise en situation

Félicitations ! Après votre BUT, vous avez été sollicité par l'unité de traitement d'images d'un observatoire astronomique.

Leurs dernières captures de nébuleuses souffrent d'un problème esthétique et scientifique : les étoiles les plus brillantes sont "gonflées" (voir Figure 1) par la diffusion atmosphérique. Ce phénomène masque les structures ténues des nuages de gaz environnants. De plus, les traitements habituels utilisé pour augmenter le contraste ne font qu'empirer les choses.

Votre mission est de développer un outil de **Star Reduction** (réduction d'étoiles). L'objectif est de diminuer le diamètre apparent des étoiles sans altérer la texture des nébuleuses présentes en arrière-plan (voir en Figure 2).

(a) M45 detail

(b) Sadr (γ Cygni)

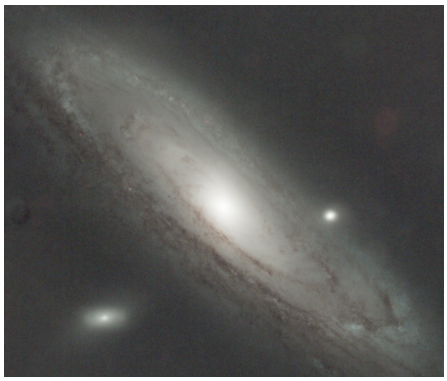
FIGURE 1 – Phénomène d'étalement des étoiles : ici sur les étoiles de M45 et Sadr

2 Problématique :

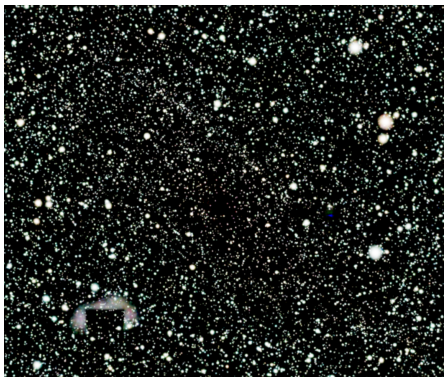
2.1 la PSF et la Diffusion

En astronomie, compte tenu de sa distance, une étoile est considérée une source ponctuelle. Cependant, à cause de la turbulence atmosphérique (le seeing) et de la diffraction optique de l'ins-

(a) Étoiles fortement visibles



(b) Image "star-less"



(c) Masque d'étoiles

(d) Image recomposé faisant mieux appraître la galaxie

FIGURE 2 – Principe de la réduction d'étoiles sur la galaxie d'Andromède (M31) par SD.

trument, elle apparaît sur le capteur comme une tache étalée.

Ce phénomène est décrit par la PSF (Point Spread Function).

Le problème : Pour révéler les détails sombres d'une nébuleuse, on applique souvent des étirements d'histogramme (courbes en S). Malheureusement, ces traitements font "exploser" les hautes lumières, transformant de petits points brillants en disques blancs massifs qui saturent l'image et masquent les filaments de gaz.

2.2 L'enjeu Scientifique vs Esthétique

Votre outil doit naviguer entre deux exigences :

L'esthétique : Permettre à l'œil de se concentrer sur les structures de la nébuleuse (piliers de poussière, ondes de choc) sans être distrait par un champ d'étoiles trop dense.

L'intégrité des données : Contrairement à un filtre classique, votre algorithme ne doit pas créer d'artefacts (halos noirs autour des étoiles) ni lisser les textures réelles de la nébuleuse, car ces textures sont des données scientifiques précieuses.

2.3 Votre Rôle dans l'Observatoire

En tant que développeur, vous n'êtes pas seulement là pour coder un filtre, mais pour créer un outil de précision.

Votre logiciel devra probablement :

- Détecter automatiquement les étoiles (distinguer un point brillant d'un nœud de gaz dans la nébuleuse).
- Générer un masque de protection pour l'arrière-plan.
- Appliquer une réduction proportionnelle (réduire davantage les grosses étoiles que les petites pour garder un aspect naturel).

2.4 Format des données

Dans le cadre de votre mission à l'observatoire, vous ne travaillerez pas avec des formats classiques comme le JPEG ou le PNG, qui compressent et détruisent l'information. Le standard universel en astronomie est le format FITS (Flexible Image Transport System).

Voici ce qu'il faut savoir sur ce format :

- C'est un format "Scientifique" : Contrairement aux images standards, le FITS stocke les données brutes du capteur (souvent en 16 ou 32 bits), garantissant une précision mathématique totale pour vos calculs de réduction d'étoiles.
- Il contient des Métadonnées (Header) : Chaque fichier contient un en-tête textuel détaillé incluant les coordonnées célestes, le temps de pose, la température du capteur et le télescope utilisé.
- Il est multi-couche : Un seul fichier FITS peut contenir plusieurs "extensions" (images de différents filtres, spectres ou tables de données).

Outils de manipulation

Pour intégrer ce format dans votre outil de Star Reduction, vous disposez de ressources : comme la bibliothèques de programmation : Astropy (Python) est la référence, mais il existe aussi des interfaces en C++ et Java.

3 Phase 1 : Preuve de concept (La méthode "Out of the box")

Avant de coder votre propre algorithme, vous devez valider la faisabilité sur des données réelles en utilisant des fonctions de traitement d'image standard.

3.1 Données de test

Vous utiliserez les bibliothèques `astropy` et `astroquery` pour récupérer des images de référence. Vous utiliserez aussi la bibliothèque `opencv` pour réaliser des opérations simple de traitement d'images.

Cible conseillée : Deux fichiers .fits de départ sont fournis

— HorseHead.fits : nébuleuse de la tête de cheval, c'est un fichier monocanal (noir et blanc)

— M31.fits : galaxie d'andromède, en couleur (multi-canal)

3.2 Algorithme de base

Le moyen le plus simple de réduire une étoile est d'appliquer un opérateur de mathématique : l'érosion. L'idée est de remplacer chaque pixel par la valeur minimale de son voisinage. Pour cela, on "déplace" une petite grille (appelée noyau) devant chaque pixel, on enregistre la valeur minimale des pixels environnant et le pixel est remplacé par cette valeur.

Cette opération peut être réalisée en utilisant open-cv.

Listing 1 – Exemple de réduction rapide avec OpenCV

```
import cv2
import numpy as np

# On suppose 'img' la matrice de l'image FITS
kernel = np.ones((3,3), np.uint8)
img_eroded = cv2.erode(img, kernel, iterations=1)
```

Référez vous à la documentation pour davantage de précision sur les érosions et les dilatations : `https://docs.opencv.org/3.4/db/df6/tutorial_erosion_dilatation.html`

3.3 Travail à réaliser

Chargez une image FITS, appliquez une érosion simple et observez le résultat.

Testez plusieurs paramètres (taille de noyau, itérations).

Quels sont les défauts de cette méthode appliquée à l'image entière ?

Une fois arrivé à cette étape, faites valider cette première étape par un enseignant.

4 Phase 2 : Développement de l'algorithme sélectif

Une érosion globale "écrase" les détails de la nébuleuse. Il faut donc travailler avec un **masque**.

4.1 Étape A : Création du masque d'étoiles

Utilisez un seuillage adaptatif ou la fonction `DAOStarFinder` de la bibliothèque `photutils` pour localiser les étoiles. Le masque doit être une image binaire où les pixels "blancs" correspondent aux étoiles.

4.2 Étape B : Réduction localisée

Implémentez la logique suivante :

1. Créer une version érodée de l'image originale (I_{erode}).
2. Créer un masque d'étoiles (M) dont les bords sont adoucis par un léger flou gaussien (pour éviter les transitions brutales).
3. Calculer l'image finale I_{final} par une interpolation : $I_{final} = (M \times I_{erode}) + ((1 - M) \times I_{original})$

Après cette étape, faites à nouveau valider votre travail par un enseignant.

5 Phase 3 : Prolongements (Défis experts)

Voici une liste de prolongements possibles, choisissez au minimum un. Cette liste n'est pas exhaustive et vous pouvez également proposer vos propres pistes.

— **Astrométrie via API :** Utilisez l'API de *Astrometry.net* pour envoyer votre image. Récupérez le catalogue d'étoiles détectées pour générer un masque parfait, basé sur des données astronomiques réelles.

— **Réduction Multitaille :** Les grosses étoiles nécessitent une érosion plus forte que les petites. Proposez un algorithme qui adapte la taille du noyau d'érosion en fonction de la magnitude de l'étoile.

— **Interface Utilisateur :** Développez une interface permettant de charger un fichier FITS et d'ajuster en temps réel la force de la réduction (taille du noyau et seuil du masque).

— **Détection par Machine Learning** : Entraînez un petit réseau de neurones (ou utilisez un modèle pré-entraîné comme StarNet) pour segmenter les étoiles et les séparer totalement du fond de ciel avant traitement.

— **Comparateur Avant/Après** : Créez un outil de visualisation qui permet de superposer rapidement l'image originale et l'image traitée (fonction "clignotement" ou soustraction) pour détecter les pertes de détails dans la nébuleuse.

— **Batch Processing** : Développez un mode "ligne de commande" permettant d'appliquer automatiquement les mêmes paramètres de réduction à une série de 100 images FITS d'un même catalogue.

6 Livrables et Évaluation

6.1 Partie code

— Le code source commenté (Python).

— Un rapport au format **texte** (README.md) qui comportera les noms des auteurs et un descriptif des méthodes choisies, des difficultés rencontrées ainsi que des résultats.

— Deux images (jpg) présentant un comparatif "Avant/Après".

— L'ensemble regroupé dans un fichier .zip (pas d'autre format)

6.2 Exposé oral

Slides au format pdf à déposer pour le jeudi 8 janvier 2026 à 19h.