* Stack utilisé
  + Dépendances
* Présentation de la première image originale
  + Présentation de la première image en black and gray
  + Explication du bruit utilisé : salt and pepper
  + Résultats d’application des 5 filtres
    - Description du filtre utilisé
    - Comparaison avec l’image initiale
    - Appréciation du résultat
* Présentation de la deuxième image originale
  + Présentation de la deuxième image en black and gray
  + Explication du bruit utilisé : Gaussian
  + Résultats d’application des 5 filtres
    - Description du filtre utilisé
    - Comparaison avec l’image initiale
    - Appréciation du résultat

# Utilisation de filtres afin de « traiter » une image

## Technologies utilisées

Pour le traitement des images, nous avons utilisé le langage de programmation Python ainsi que l’environnement web interactif Jupyter Lab. Cela nous a permis de rapidement créer un environnement de développement pour pouvoir tester différents types de filtres et d’être en mesure de partager notre travail sans nécessiter une gestion fastidieuse des dépendances. Le code a pu ainsi être subdivisé en étape importante et chaque section est commentée pour faciliter la compréhension du déroulement de l’exécution.

La librairie *Numpy*, qui est réputée pour la manipulation massive de données, nous permet de traiter les images en tableau de valeurs pour les canaux de couleurs. La librairie *Matplotlib* aide à l’affichage des résultats de traitement et à l’enregistrement des données. Et finalement, *opencv* rend possible l’application de différents types de filtres d’une façon simple et efficace.

Chaque image est redimensionnée au besoin, puis convertie en teintes de gris et transformée en tableau *Numpy*. Un bruit quelconque est ensuite appliqué à l’image en noir et blanc. Cinq filtres sont appliqués à cinq copies différentes de l’image prétraitée puis le résultat est affiché.

## Traitement de la première image

La première image utilisée est celle de la nébuleuse M1-67, qui est une expulsion de matériel autour de l’étoile de type Wolf-Rayet nommée WR124, découverte en 1938 par Paul W. Merrill.



Figure : Image originale colorisée

. L’image originale est du type non compressé *png* et possède une taille de 400x400 pixels. Notre questionnement était de savoir s’il était possible de filtrer l’image tout en conservant les étoiles capturées dans la nébuleuses et qu’elles ne soient pas considérée comme du bruit.



Figure : Image originale en teintes de gris

Nous avons choisi d’appliquer un bruit de type *salt and pepper* qui est caractérisé par des perturbations soudaines et vives dans le signal d’une image. Ainsi, seulement quelques pixels sont bruités, mais ils le sont fortement. Visuellement, cela s’apparente à un étalement clairsemé de particules blanches et noires qui pourraient apparaître lors d’une transmission sur un lien digital subissant un bruit ambiant. Le filtre tout désigner pour traiter ce genre de bruit est le filtre médian ou le filtre morphologique.



Figure 3: M1-67 avec bruit salt and pepper

Ce type de bruit est intéressant pour l’image que nous tentons de traiter puisque les particules noires sont dissimulées dans le cosmos entourant la nébuleuse et viendront impacter l’ombrage au centre. Quant à elles, les particules blanches sont confondues avec les étoiles présentes dans et autour du nuage de matériel. Nous avons émis l’hypothèse que certains filtres, possiblement le filtre médian, allaient considérer les particules blanches comme des étoiles et en retirer une grande partie.

### Filtre médian

Le filtre médian est un filtre digital non-linéaire capable de réduire drastiquement le bruit d’un signal tout en préservant les bords, dans des conditions spécifiques. Ces qualités font du filtre médian un excellent outil pour le prétraitement d’images destinées à des applications comme la détection de rebords. Ce filtre fonctionne également avec un masque matriciel, appelé *kernel*, qui est déplacé sur chaque pixel à la manière d’une fenêtre. Cette fenêtre de dimension impaire, se positionne sur chaque pixel et le remplace par une valeur médiane des pixels voisins. Ce calcul sur chaque pixel fait en sorte qu’un algorithme d’application du filtre médian peut s’avérer très lourd en mémoire s’il n’est pas optimisé convenablement. Le filtre médian est efficace avec un haut taux de bruit et est donc plus performant que le filtre de flou Gaussien. Il est particulièrement efficace pour le traitement du bruit de type speckle (moucheté) ou poivre et sel (salt and pepper).

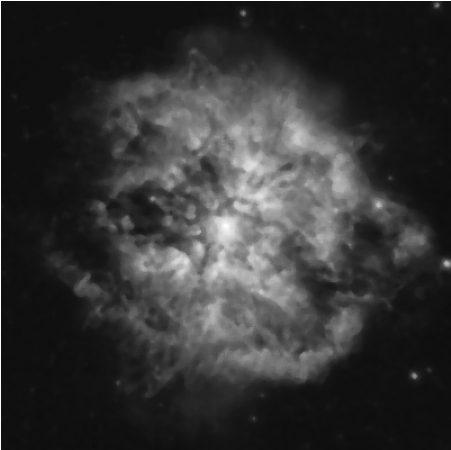


Figure 4: M1-67 traité avec un filtre médian

D’entrée de jeu, il est remarqué que les étoiles autour du nuage de particules sont pratiquement tous estompées. Le filtre médian est très efficace contre le bruit sald and pepper et par le fait même, traite les étoiles comme étant un haut niveau de bruit de type sel. Cependant, les bords de la nébuleuse ont un fort contraste et il est simple de détecter les bords. Cela démontre qualitativement le potentiel d’utiliser ce filtre en prétraitement. Les différents niveaux d’illumination du nuage demeurent distinguable malgré l’introduction d’une certain floue. Un autre facteur qui pourrait expliquer d’une certaine manière la perte d’information est l’utilisation de la technique de la fenêtre pour effectuer la médiane des pixels voisins. Lorsqu’on analyse la première ou la dernière colonne de pixels, certains pixels avoisinant sont manquant puisque l’image possède des limites. Ainsi, l’algorithme se doit de répéter le pixel central afin d’être en mesure d’effectuer les calculs. Cette technique palliative peut injecter dans l’image résultante certaines déformations ou incohérances.

### Box filter

Le Box filter est un filtre linéaire du domaine spatial où chaque pixel de l’image résultante possède une valeur égale à la moyenne des pixels avoisinants dans l’image en entrée. Ce filtre est un type de filtre passe-bas, ce qui veut dire que notre image résultante sera floutée. Ce type de filtre utilise une matrice de convolution ou *kernel* pour permettra d’appliquer un flou à l’aide d’une convolution entre ce masque et l’image à traiter. Nous utilisons la fonction *blur* d’opencv avec une dimension de masque de 5x5, ce qui représente une taille optimale standard.

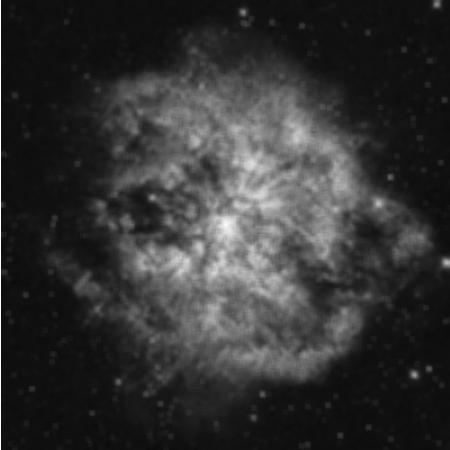


Figure 5: M1-67 traitée avec un box filter

Tel que mentionné précédemment, on remarque que le filtre de type passe-bas applique un flou à notre image résultante. Le bruit de type poivre et sel est visiblement retiré dans son entièreté, cependant il y a une très grande diminution du contraste de l’image. La perspective de profondeur de l’image est pratiquement inexistante et les détails du rayonnement du noyau sont estompés. Les étoiles blanches, devenues très embrouillée, apparaissent comme ayant une faible densité de pixels et celles près les unes des autres apparaissent comme un seul bloc monolithique. Le box filter est généralement reconnu pour atténuer les effets d’un flou gaussien qui est lui-même un bruit qui est appliqué selon une fonction de densité probabiliste apparente à la distribution normale.

En faisant de plus amples recherches, nous avons identifiés certaines recommandations d’expert du domaine de traitement d’images afin d’améliorer les résultats du box filter. Tel que mentionné précédemment, le filtre médian est un excellent outil de prétraitement afin d’améliorer l’efficacité du box filter.

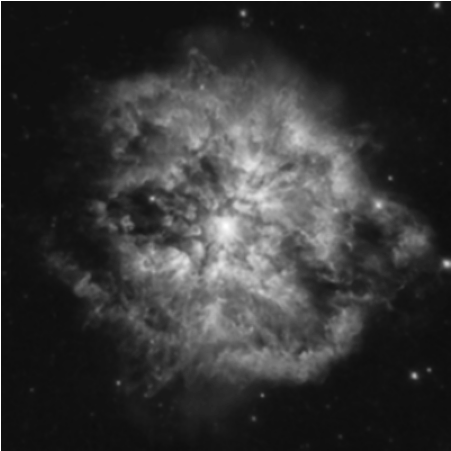


Figure 6: M1-67 traité avec un filtre médian puis box

L’image résultante est moins floue, ce qui permet de mieux distinguer les détails du nuage de matériel. L’effet totale du halo de l’image n’est pas présent, mais est plus contrastée qu’avec le filtre passe-bas uniquement. Le plus gros désavantage de l’utilisation de ces deux filtres est la perte d’information des étoiles entourant la nébuleuse. Seulement les plus grosses étoiles demeurent alors que les moins denses semblent avoir été traitées comme un bruit indésirable. Néanmoins, une analyse qualitative des caractéristiques de la nébuleuse sans considération des corps célestes est possible.

### Filtre Gaussien

Le filtre Gaussien, ou flou Gaussien, est l’application d’un flou sur une image à l’aide d’une fonction Gaussienne. Utilisée dans beaucoup de logiciels de design graphique, le flou Gaussien permet de réduire le bruit et d’atténuer les détails d’une image. De ce fait, le flou Gaussien est un excellent outil de prétraitement des images pour les algorithmes de vision par ordinateur afin d’augmenter la structure d’une image à différentes échelles et d’être en mesure de mieux détecter les rebords. L’application à une image d’un flou Gaussien est équivalente à la convolution de cette image avec une fonction Gaussienne. La transformation de Fourier d’une fonction Gaussienne est une autre fonction Gaussienne, ce qui infère que le flou Gaussien réduit l’influence des composantes de haute fréquence du signal. Le filtre Gaussien est ainsi donc classifié comme un filtre passe-bas.

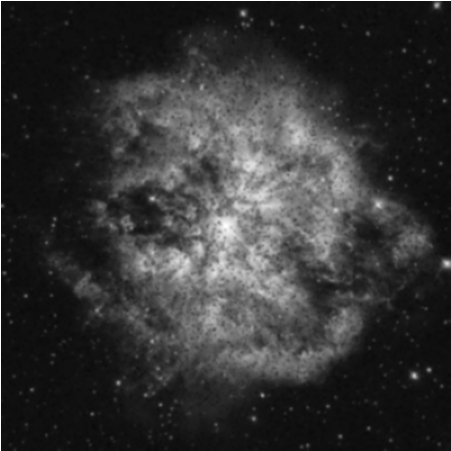


Figure 7: M1-67 traité avec un filtre Gaussien

Nous avons traité l’image bruitée avec un filtre Gaussien possédant un kernel de dimension 3x3, qui est une valeur par défaut proposé par la documentation d’opencv. Tel que prédit, l’image résultante est d’une piètre qualité puisque l’application d’un simple flou sur un bruit de type sel et poivre ne permet de l’éliminer.

Le produit de fonctions Gaussiennes en deux dimensions est défini par l’équation :

|  |  |  |
| --- | --- | --- |
|  |  | () |

Où est le paramètre de déviation standard, que nous avons laissé être inféré par l’algorithme selon la dimension du masque matriciel. À son application, la formule produit une surface aux contours circulaires avec une distribution Gaussienne à partir du point central. Les valeurs de cette distribution sont utilisées pour effectuer une convolution avec l’image originale. Lors de cette convolution, la nouvelle valeur d’un pixel se fait à partir d’une moyenne pondérée des pixels voisins. Le pixel original reçoit un poids plus important alors que les pixels voisins reçoivent un plus petit poids à mesure que leur distance avec le point central augmente.