



INTRODUCTION AU TRAITEMENT DES IMAGES

Rapport TP01 IMA201

Réalisé par:

Adnane EL BOUHALI

Encadré par:

GOUSSEAU Yann, LECLAIRE Arthur, LESNÉ Gwilherm, ROUX
Michel

Etablissement :

Télécom Paris

Filière :

Image

Année universitaire:

2023 - 2024

Table des matières

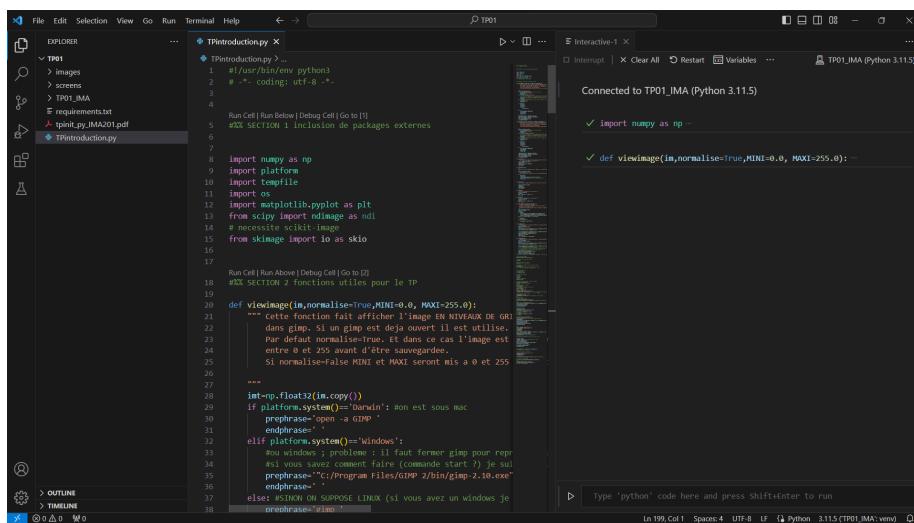
INTRODUCTION AU TRAITEMENT DES IMAGES.....	0
1 Préliminaires.....	2
2 Visualisation et utilisation de gimp.....	2
2.1 ZOOMS.....	2
2.2 Espace couleurs.....	4
3 Niveaux de gris, histogrammes et statistiques.....	6
3.1 Histogramme.....	6
3.2 Changement de contraste.....	10
3.3 Égalisation d'histogramme.....	12
3.4 Prescription d'histogramme.....	13
3.5 Dithering.....	16
3.6 Différences de niveaux de gris voisins.....	17
4 Spectre des images et transformation de Fourier.....	18
4.1 Visualisation de spectres.....	18
4.2 Ringing.....	21

1 Préliminaires

Tout d'abord, nous préparons l'environnement de développement en installant les librairies nécessaires.

```
PS C:\Users\adnan\Desktop\Documents\telecom Paris\Etudes\IMA\IMA201\TPs\TP01> pip freeze > requirements.txt
PS C:\Users\adnan\Desktop\Documents\telecom Paris\Etudes\IMA\IMA201\TPs\TP01> pip install -r requirements.txt
```

Une fois cette étape terminée, nous pouvons passer à l'exécution des deux premières sections, ce qui nous permettra de commencer le TP.

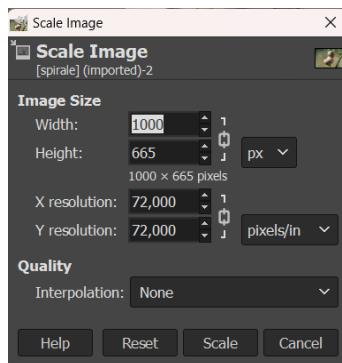


The screenshot shows a Jupyter Notebook interface with two tabs: 'TP01introduction.py' and 'Interactive-1'. The 'TP01introduction.py' tab contains Python code for image processing, including imports for numpy, platform, tempfile, os, matplotlib, and scikit-image. It defines a function 'viewimage' that takes an image and optional parameters for normalization and color limits. The 'Interactive-1' tab shows the output of the code execution, indicating it is connected to 'TP01_IMA (Python 3.11.5)'. The code cell for 'viewimage' has been run, and the resulting image is displayed in the notebook.

2 Visualisation et utilisation de gimp

2.1 ZOOMS

Avant de commencer, on s'assure qu'il n'y a aucune méthode d'interpolation sélectionnée, comme ça les pixels originaux sont simplement agrandis sans ajout d'informations supplémentaires.



On ouvre l'image **spirale.jpeg** dans gimp. On la zoom d'un facteur de quatre pour la visualiser en plus grand.



Que fait gimp pour afficher l'image en plus grand?

Le logiciel GIMP agrandit simplement les pixels de l'image pour les afficher à une taille plus grande à l'écran.

Ensuite, on ouvre l'image **maison.tif**, et on réduit sa taille d'un facteur deux (avec aucune méthode d'interpolation).



On la zoom pour l'afficher en deux fois plus grand.



On ouvre l'image **maison_petit.tif** et on la zoom d'un facteur 2 également.

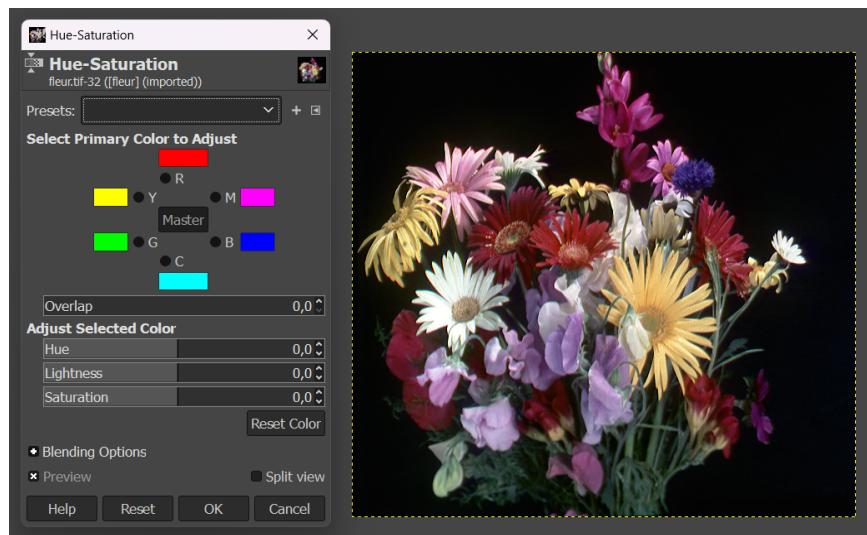


Comparez le résultat par rapport au zoom de la petite image que vous avez produite. Quelle hypothèse pouvez-vous faire sur la génération de maison_petit.tif?

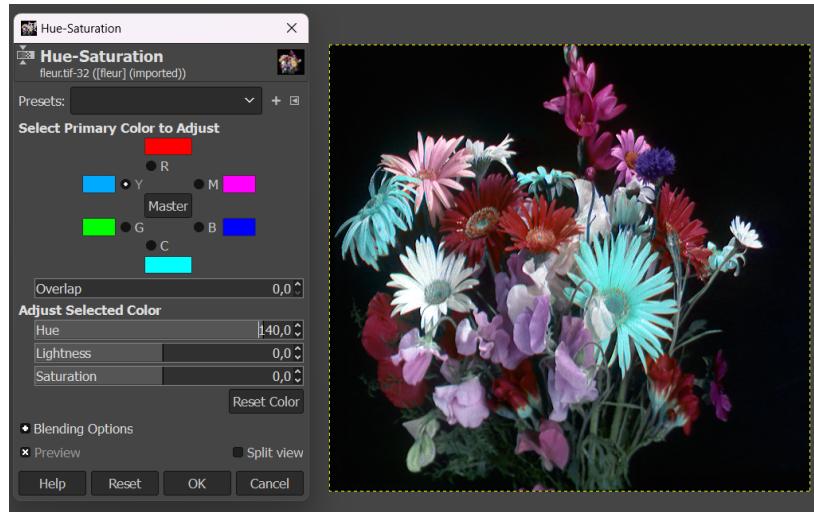
On remarque que la première image est beaucoup plus nette après un zoom que la deuxième image, cela peut indiquer que la deuxième image avait une résolution plus faible ou était déjà floue. Il est également possible qu'il y ait du bruit dans la deuxième image.

2.2 Espace couleurs

On ouvre l'image **fleur.tif** et le dialogue "Couleurs->Teinte-Saturation".



On transformer les fleurs jaunes en fleurs bleues avec le bouton Hue.

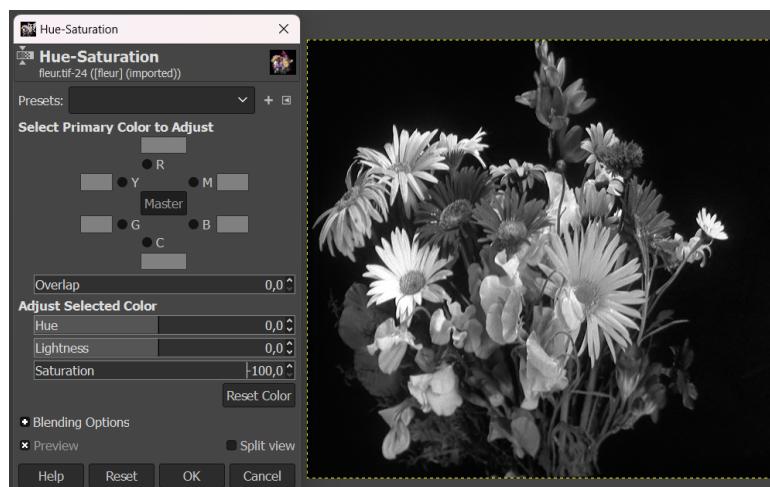


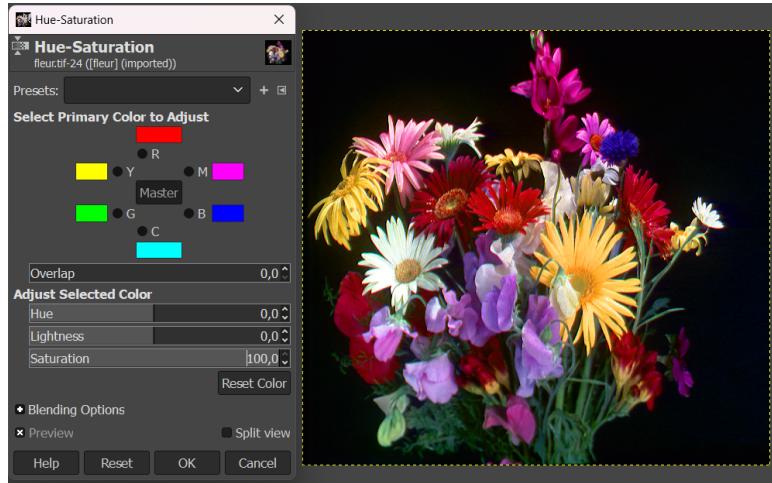
Comprenez-vous pourquoi les deux positions extrêmes de ce bouton font, en fait, la même transformation?

Le bouton Hue permet de modifier la teinte d'une couleur. La teinte est la position d'une couleur dans le cercle chromatique. Les deux positions extrêmes du bouton Hue sont 0° et 360° . À 0° , la couleur n'est pas modifiée. À 360° , la couleur est déplacée d'un tour complet sur le cercle chromatique, ce qui revient à la même chose qu'à ne pas la modifier.

A quoi correspond la saturation (essayez -100% et +100%)?

La saturation est la pureté d'une couleur. Une couleur saturée est une couleur vive et contrastée. Une couleur non saturée est une couleur terne et fade.

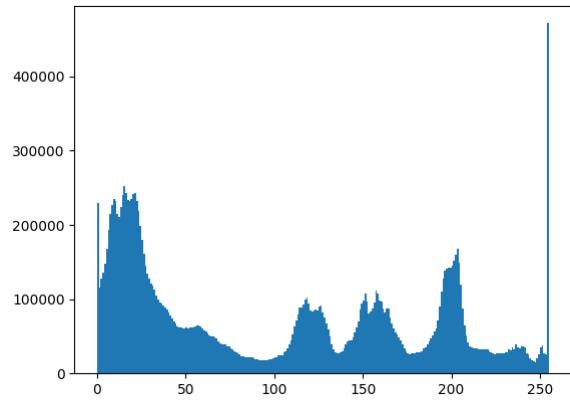
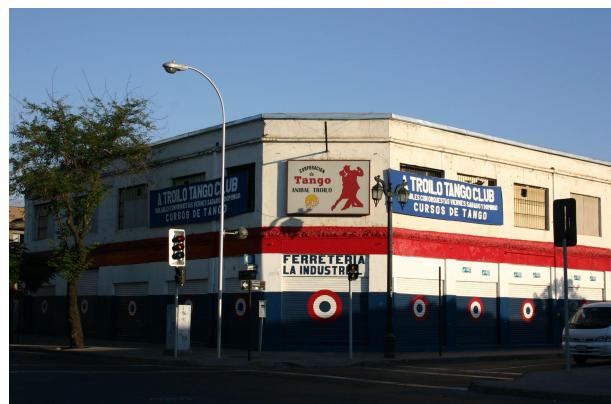
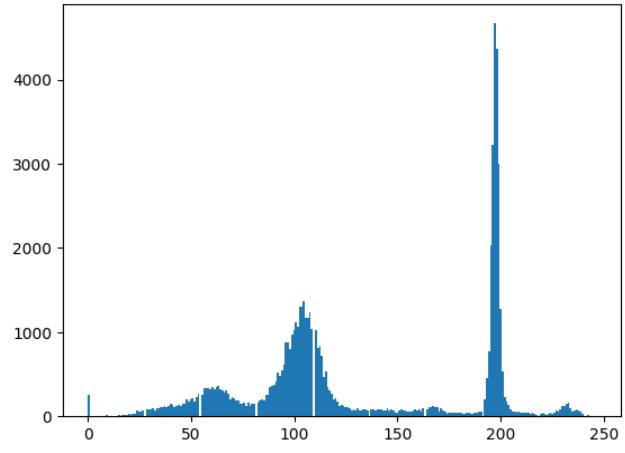


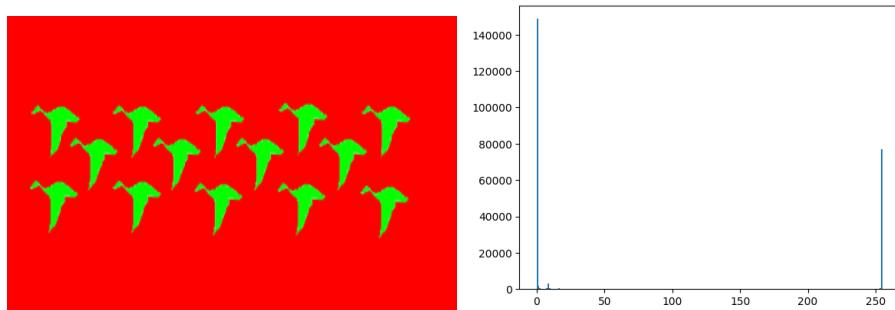
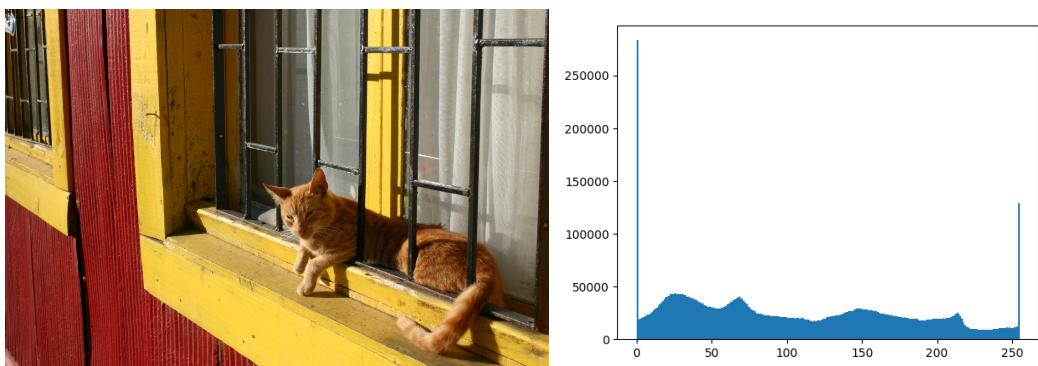
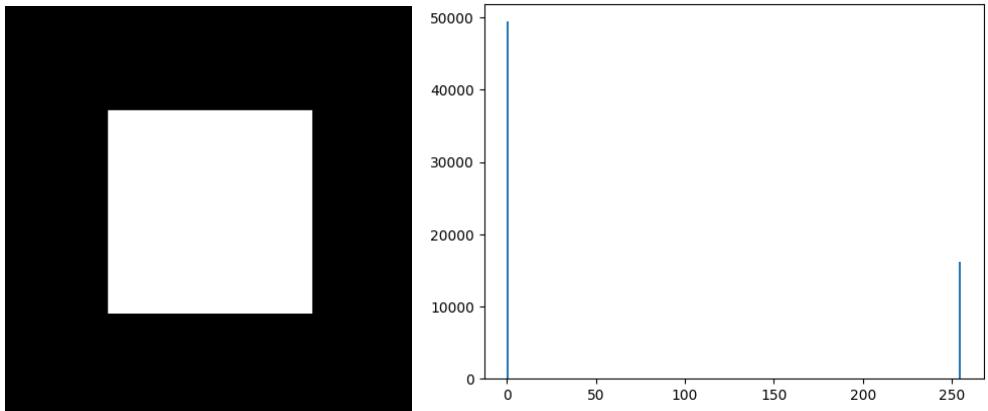


3 Niveaux de gris, histogrammes et statistiques

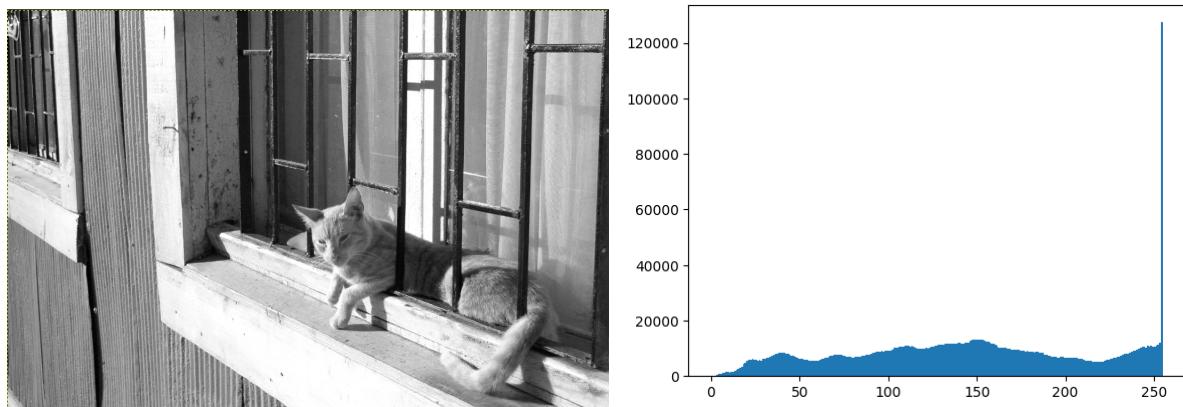
3.1 Histogramme

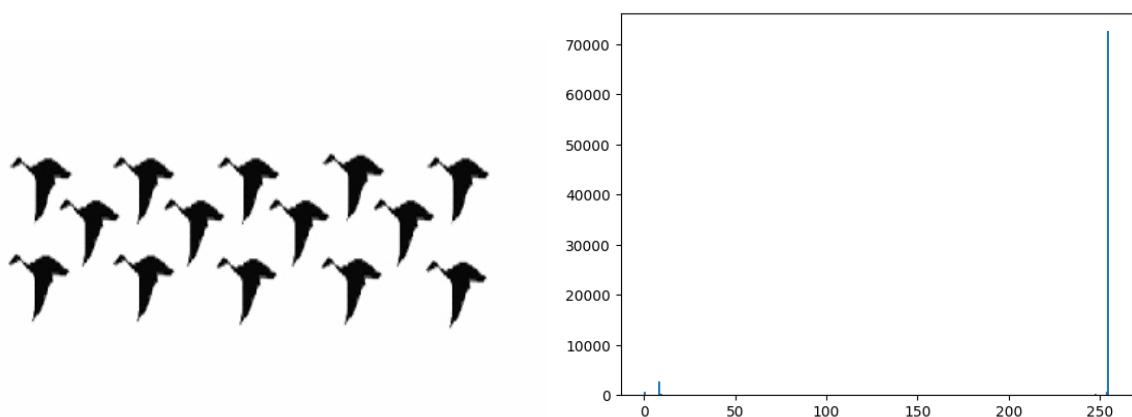
Commençons par visualiser quelques histogrammes d'images.



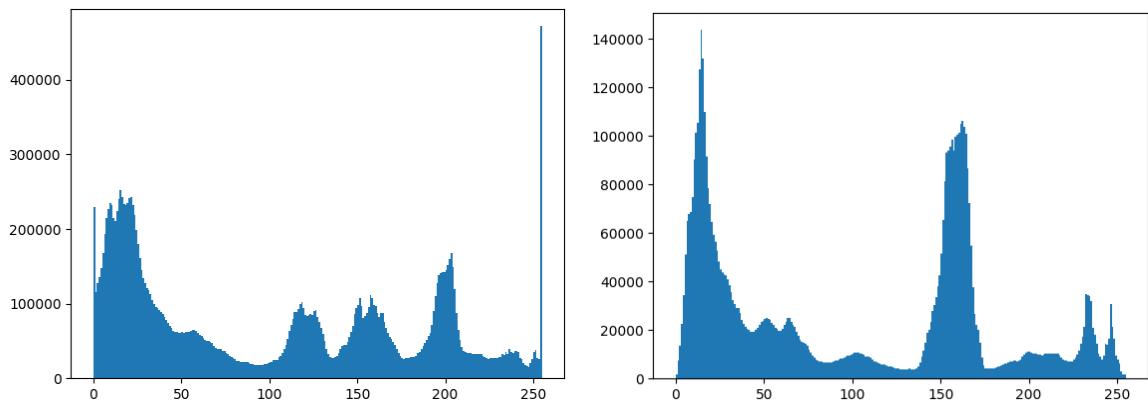


Maintenant, on va visualiser l'histogramme du canal rouge des images **after.jpg** et **chat.tif**.

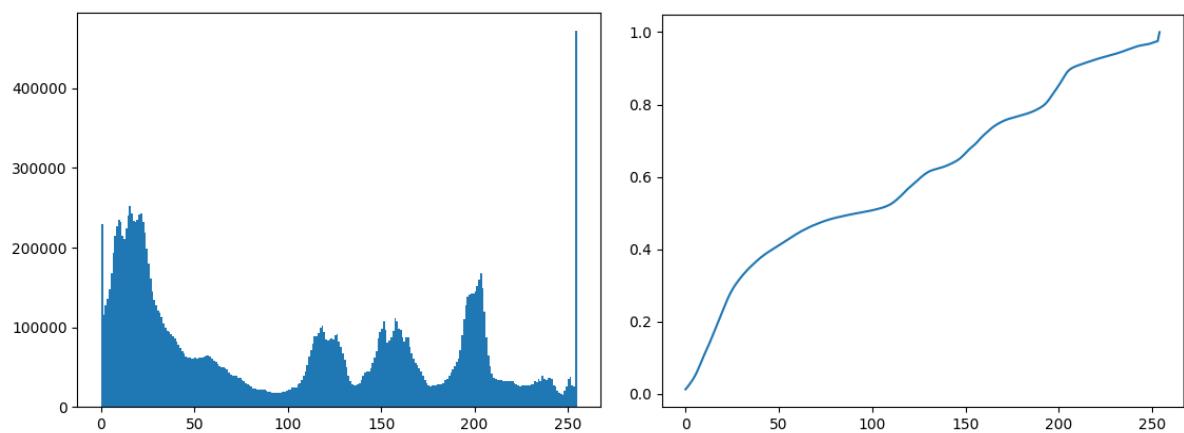




Ensuite, on va visualiser l'histogramme de l'image obtenue en moyennant les canaux couleur de l'image **chile.jpg** (à droite), et on le compare à l'histogramme classique de la même image (à gauche).



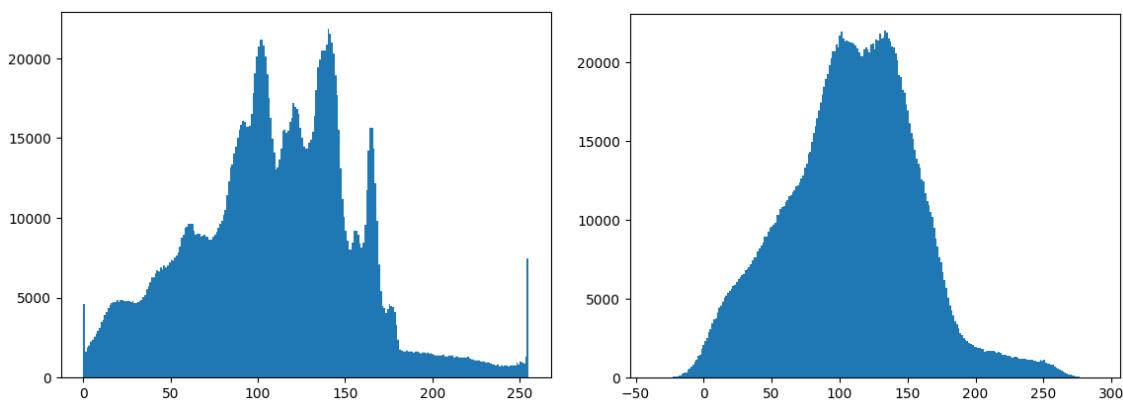
Finalement, on visualise l'histogramme cumulé de la même image (à gauche), en comparaison avec l'histogramme normal (à droite).



Ajoutez un bruit gaussien à une image et étudiez l'effet produit sur son histogramme.

Étudions maintenant l'effet de dégradations simples des images sur leur histogramme.

A l'aide de la fonction **noise** , ajoutons un bruit gaussien à l'image **spirale.jpg** et étudions l'effet produit sur son histogramme (à droite) en comparaison avec son histogramme normal (à gauche).



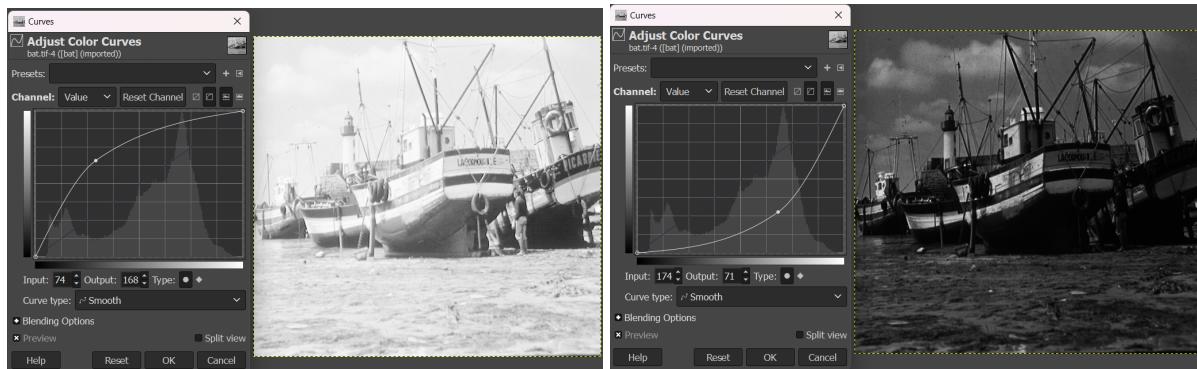
On remarque que l'ajout d'un bruit gaussien cause l'histogramme à s'étaler (une sorte de moyennisation des valeurs), c'est-à-dire que les pics de l'histogramme se réduisent et s'étalement.

En considérant les niveaux de gris d'une image comme la réalisation d'une variable aléatoire dont la loi est l'histogramme de l'image, interprétez le résultat.

Soit X la valeur d'un pixel de l'image tiré uniformément, X est donc une v.a à loi discrète, ajouter du bruits à l'image revient à ajouter du bruit à tous les pixels, X devient alors $X + Z$ où Z suit $N(0, \sigma^2)$ avec X et Z des v.a indépendantes. Par ailleurs, $X + Z$ est une variable aléatoire à densité (donc ajouter du bruit est une sorte de discréétisation).

3.2 Changement de contraste

Prenons l'image suivante, et appliquons différentes fonctions de changement de contraste.

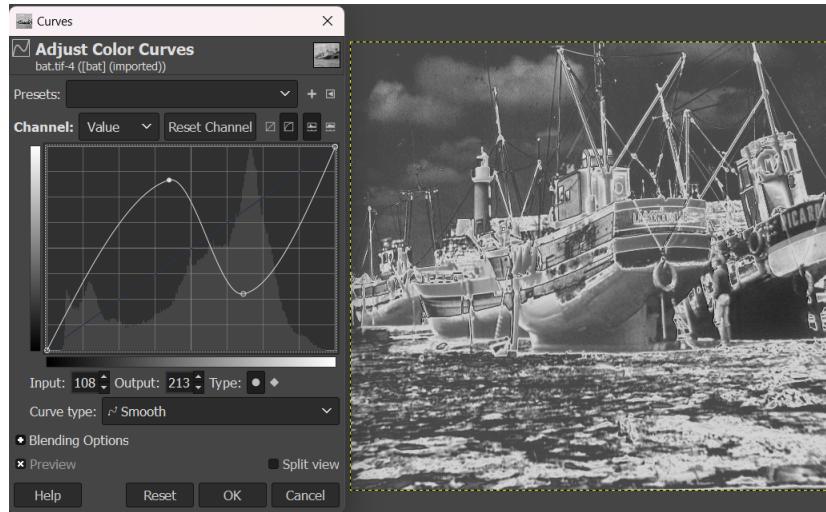


L'aspect global de l'image est-il modifié par l'application de fonctions croissantes ?

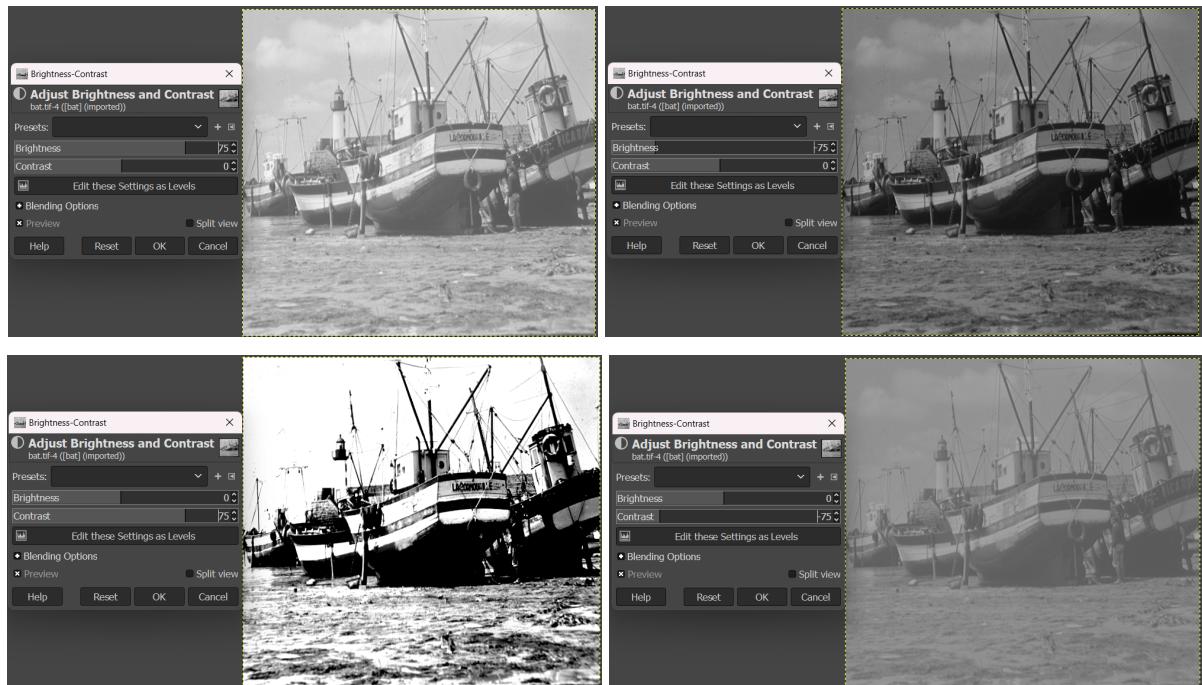
L'effet sur l'aspect global de l'image dépend de la forme de la courbe que l'on applique. En effet, les fonctions croissantes rendent l'image plus claire en augmentant la différence de luminosité entre les pixels plus sombres et les pixels plus clairs, ce qui augmente le contraste. Tandis que les fonctions décroissantes rendent l'image plus sombre en réduisant la différence de luminosité entre les pixels plus sombres et les pixels plus clairs, ce qui diminue le contraste.

Que se passe-t-il si l'on applique une transformation non-croissante des niveaux de gris?

Cela peut provoquer des résultats inattendus, tels que des parties de l'image qui deviennent identiques ou qui s'inversent en termes de luminosité, ce qui peut être utile pour des effets artistiques comme l'exemple ci-dessous.

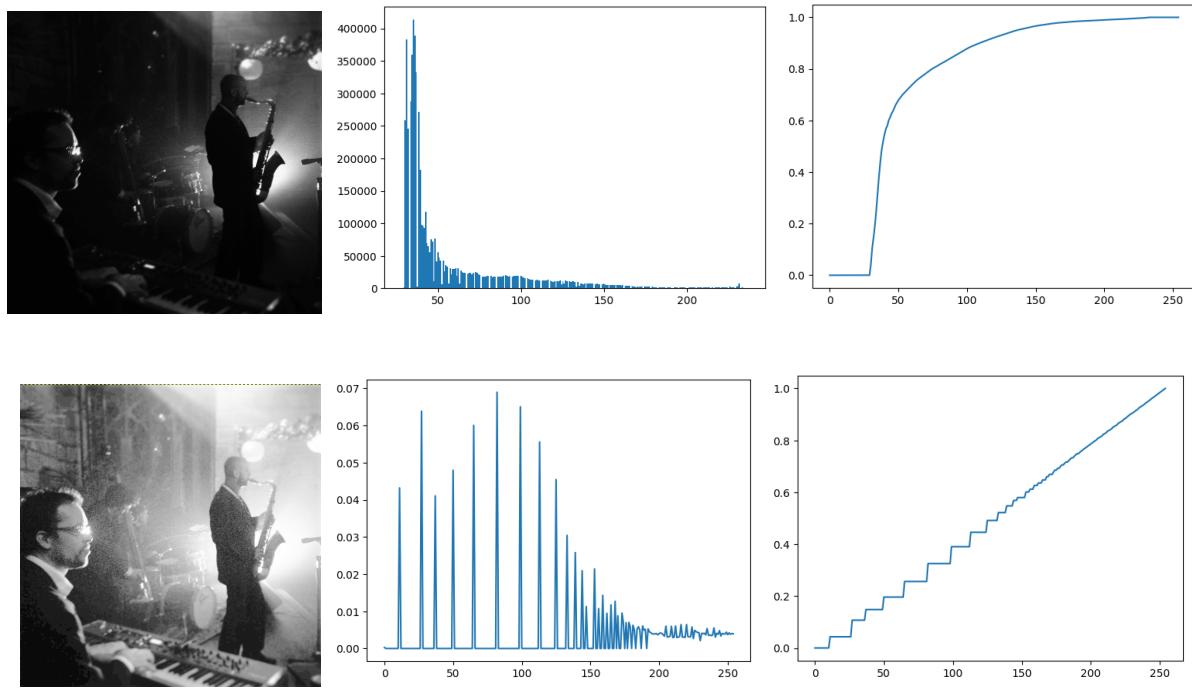


Finalement, on va voir l'effet de l'ajout et la diminution de la luminosité et le contraste sur la même image.



3.3 Égalisation d'histogramme

Visualisons l'effet de l'égalisation de l'histogramme sur l'image elle-même, ainsi que son histogramme normal et cumulatif.



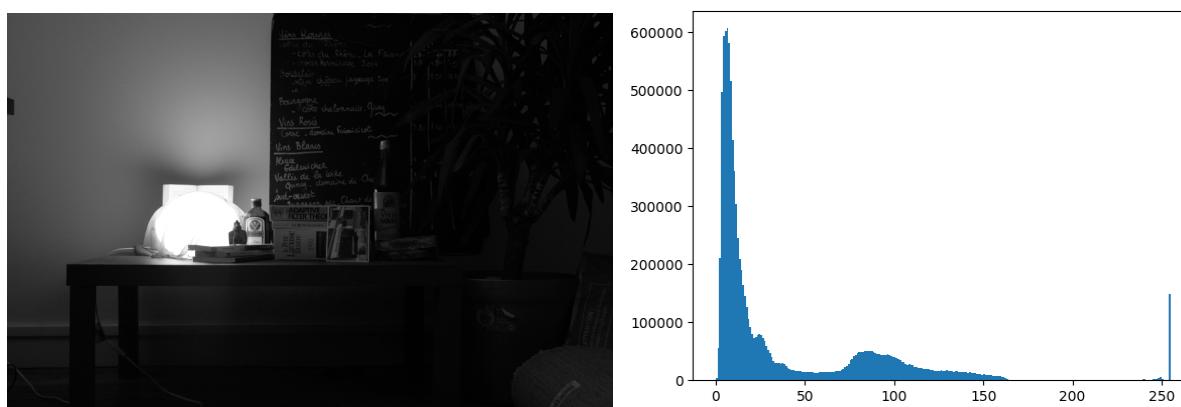
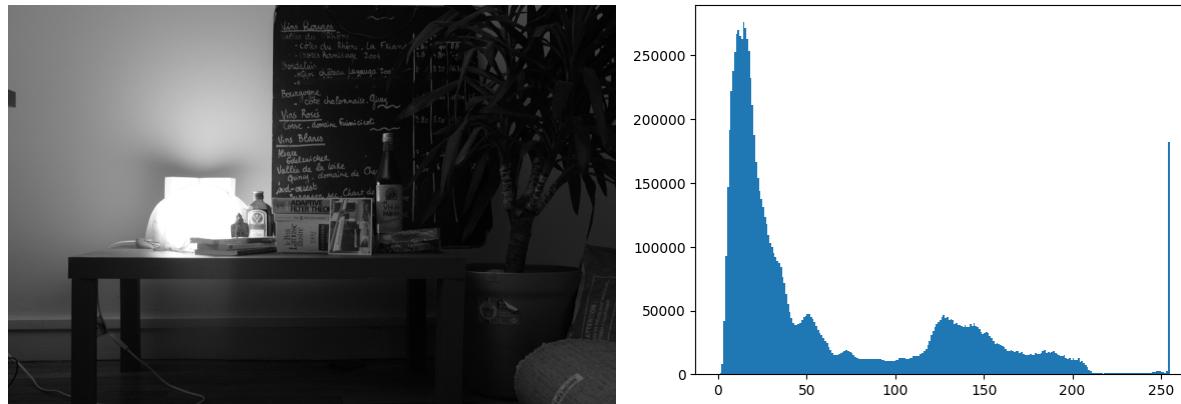
Où observez-vous sur `inequal`, sur son histogramme et sur son histogramme cumulé?

L'égalisation d'histogramme est une technique qui vise à modifier la distribution des niveaux de gris ou des valeurs d'occurrence dans un histogramme d'une image. Son objectif est de réduire les disparités dans la répartition des niveaux de gris, notamment dans le contexte des images, afin d'améliorer la visibilité des détails qui étaient moins perceptibles auparavant.

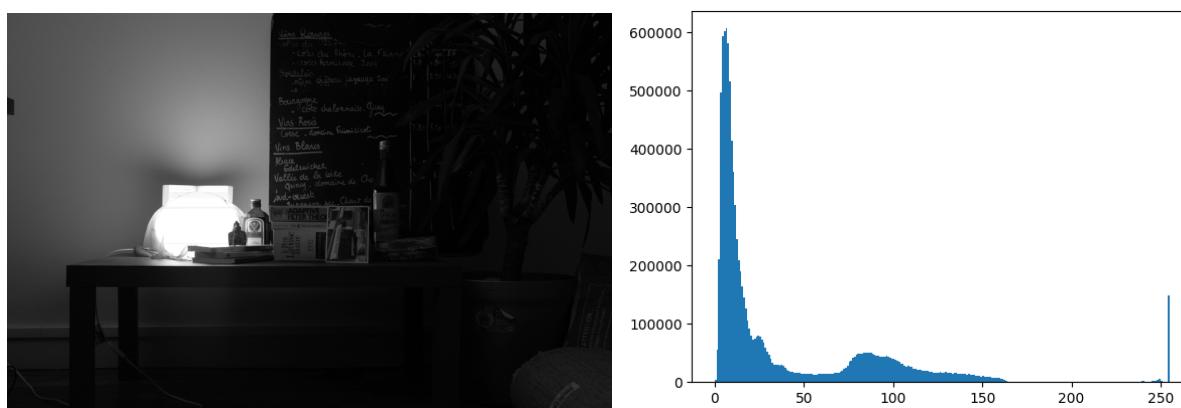
En pratique, cette opération rend l'histogramme de l'image initial ainsi que l'histogramme cumulé plus uniformes. Cela se traduit par une image avec un contraste accru et une meilleure clarté visuelle, car les valeurs de niveaux de gris sont mieux réparties sur toute la plage des valeurs possibles. En conséquence, les détails qui étaient auparavant difficiles à distinguer deviennent plus apparents et la photo paraît plus nette, avec davantage de détails visibles.

3.4 Prescription d'histogramme

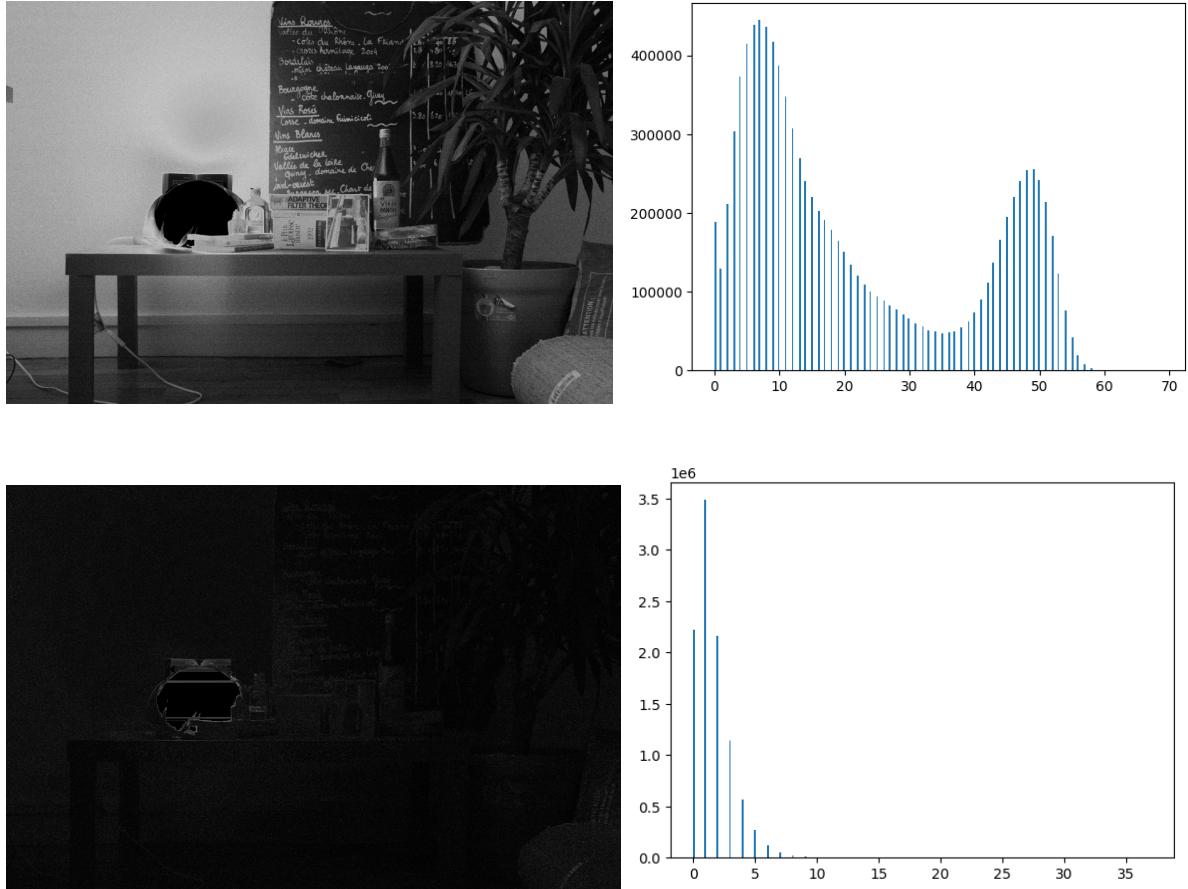
Visualisons les images de **vue1.tif** et **vue2.tif** et leur histogrammes respectifs.



Après, on visualise l'image résultante après la prescription d'histogramme (**vue1.tif** avec l'histogramme de **vue2.tif**).



Maintenant, on visualise la valeur absolue de la différence des images. (entre **vue1.tif** et **vue2.tif** dans un premier temps, puis entre **vue2.tif** et la nouvelle image prescrite)



On remarque qu'on gagne en informations dans la première image, la clarté de l'image a augmenté, et les détails qui étaient auparavant difficiles à distinguer sont devenus plus apparents et la photo paraît plus nette.

Pour la deuxième image, on ne voit presque rien, car il n'y a presque aucune différence entre les deux images, d'où la valeur absolue de leur différence est quasiment nulle partout.

Sachant que ces images ont été obtenues à partir d'images RAW (donc avec une réponse proportionnelle à la quantité de photon mesurée), a-t-on un moyen plus simple d'obtenir le même résultat (donner le même histogramme aux deux images) ?

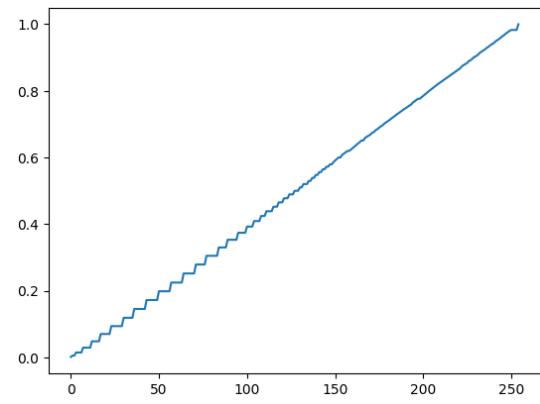
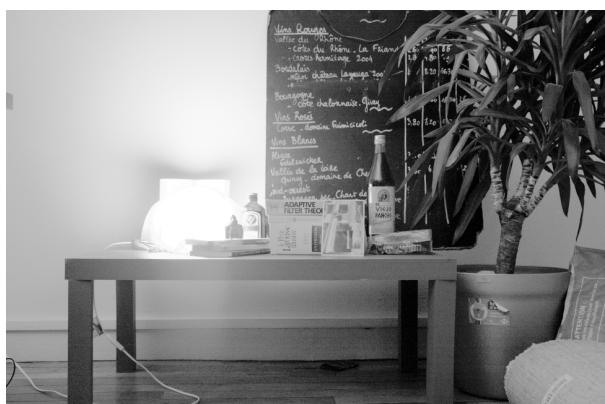
On commence par obtenir la forme (dimensions) de l'image u. Ensuite, on aplatis l'image u en un tableau 1D (ligne). On fait de même avec l'image v. On utilise argsort pour obtenir les indices qui trieraient uligne. On crée un tableau vide unew avec la même forme que uligne et le même type de données. Puis on

remplace les valeurs de `unew` en utilisant les valeurs triées de `vligne`. Enfin, on remet `unew` à la bonne forme (celle de l'image `u`).

```
300 ushape=u.shape
301 uligne=u.reshape((-1,)) #transforme en ligne
302 vligne=v.reshape((-1,))
303 ind=np.argsort(uligne)
304 unew=np.zeros(uligne.shape,uligne.dtype)
305 unew[ind]=np.sort(vligne)
306 # on remet a la bonne taille
307 unew=unew.reshape(ushape)
308 viewimage(unew)
```

En vous inspirant du code proposé pour la prescription d'histogramme, donnez un code simple permettant d'égaliser l'histogramme d'une image (le rendre aussi proche que possible d'une fonction constante).

```
379 from skimage import io, exposure
380 image = io.imread('images/vue1.tif')
381 image_equalized = 255*exposure.equalize_hist(image)
382 viewimage(image_equalized)
383 (histo,bins)=np.histogram(image_equalized.reshape((-1,)),np.arange(0,256))
384 histo=histo/histo.sum()
385 histocum=histo.cumsum()
386 plt.plot(histocum)
```



3.5 Dithering

Visualisons l'effet d'une quantification de plus en plus brutale sur l'image **lena.tif** (0, 2, 4, 6 ...).



Comme anticipé, en augmentant la quantification, nous pouvons observer une amélioration de l'image.

Maintenant, on va seuiller une l'image **lena.tif**, puis on la compare avec le résultat de seuillage de son image bruitée.



L'image avec du bruit semble manifestement plus similaire à l'image originale que celle sans bruit.

En considérant un pixel de niveau x dans l'image initiale, donnez la probabilité pour que ce pixel soit blanc après ajout de bruit et seuillage.
Pourquoi l'image détramée ressemble-t-elle plus à l'image de départ que l'image simplement seuillée?

La probabilité qu'un pixel de niveau x dans l'image initiale devienne blanc après ajout de bruit et seuillage dépendra du niveau de seuil, de la distribution du bruit et du modèle de seuillage utilisé. En général, si le bruit est proche du seuil, la probabilité que le pixel devienne blanc sera plus élevée.

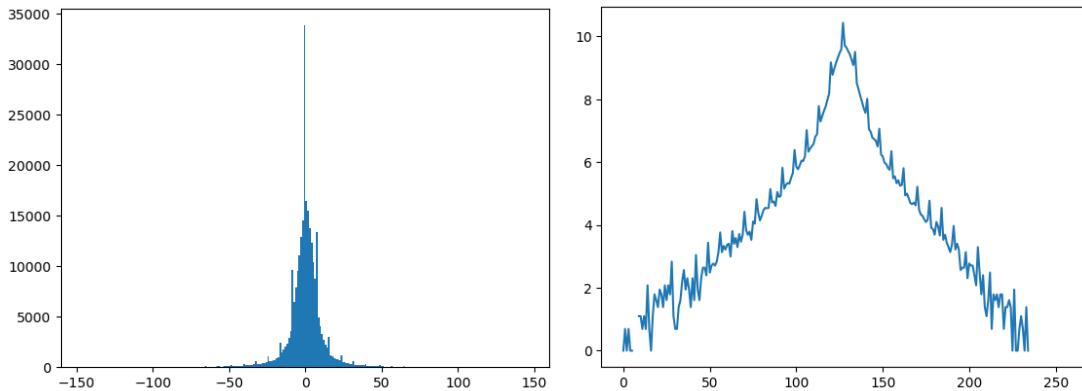
L'image détramée ressemble plus à l'image de départ que l'image simplement seuillée parce que le processus de débruitage tente de rétablir les valeurs des pixels à leurs valeurs d'origine autant que possible, tout en réduisant l'impact du bruit. En revanche, l'image simplement seuillée prend une décision binaire sur chaque pixel, ce qui peut entraîner une perte d'information et une sensibilité accrue au bruit.

3.6 Différences de niveaux de gris voisins

Grâce à la commande **gradx** on peut obtenir une image de la différence entre pixels adjacents.



Visualisons l'histogramme d'une telle image, et le logarithme de l'histogramme.



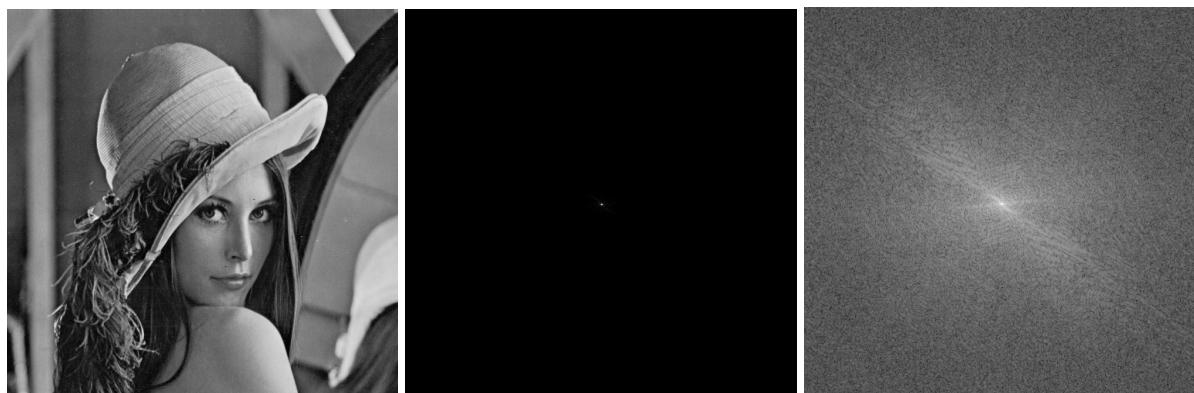
La distribution des différences vous semble-t-elle obéir à une loi gaussienne ? Pourquoi ? Quelle aurait été la forme de l'histogramme si l'on avait considéré la différence entre pixels plus éloignés?

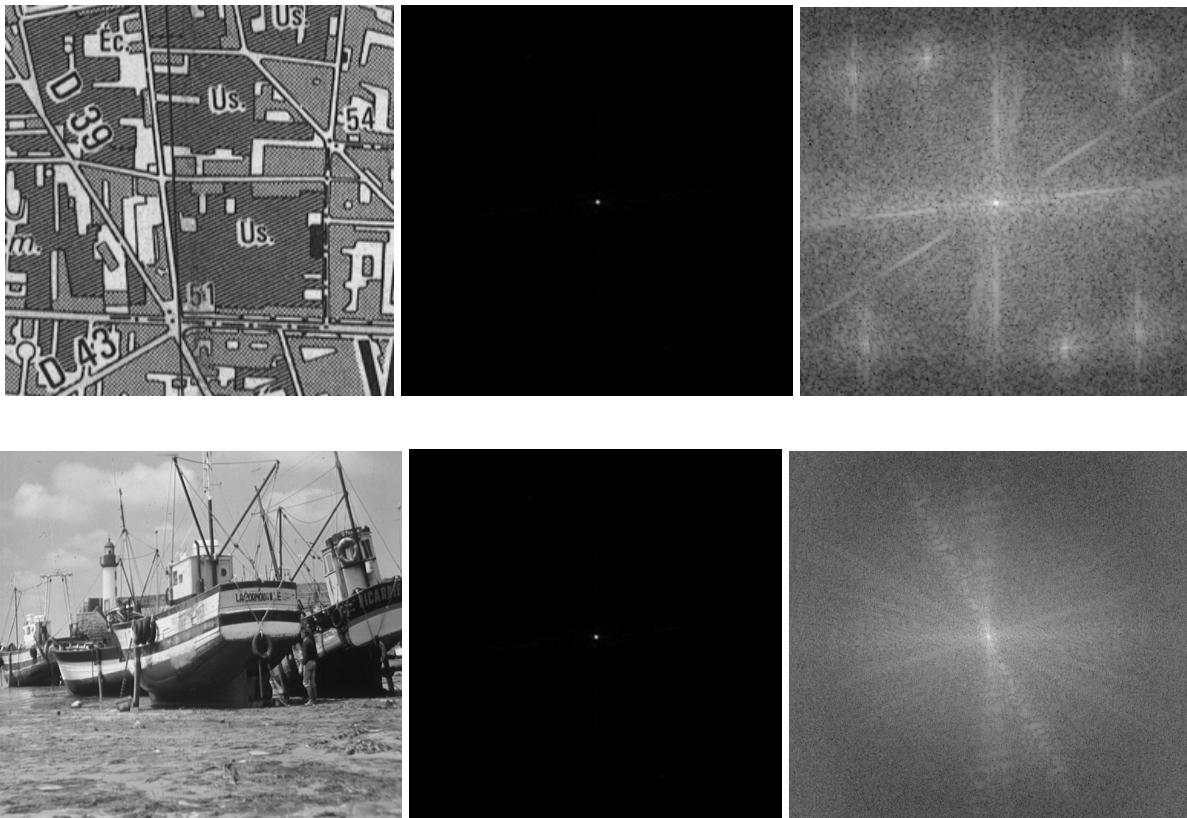
Oui, c'est vrai. La principale différence entre les voisins proches se situe dans le contour de l'image. En général, ils ont une différence de moyenne de 0, ce qui correspond au contour. Je pense que cela suit également une distribution gaussienne, mais avec un écart-type plus grand.

4 Spectre des images et transformation de Fourier

4.1 Visualisation de spectres

Visualisons le spectre de quelques images grâce à la fonction `view_spectre` avec les options 1 et 2 (respectivement).



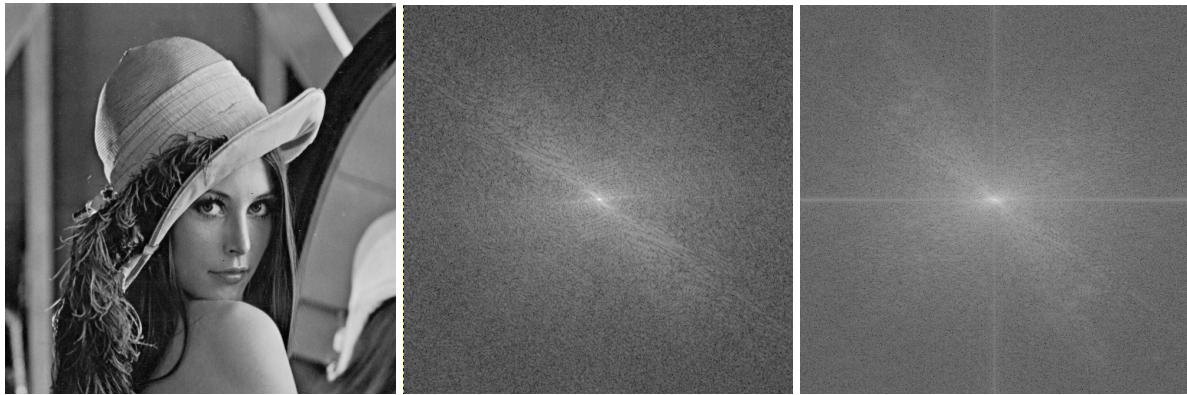


Que constatez-vous? Qu'en déduisez-vous par rapport au spectre d'une image?

Comment influe l'option hamming sur le spectre de l'image? (multiplication de l'image originale par une fonction très lisse qui s'annule aux bords de l'image)

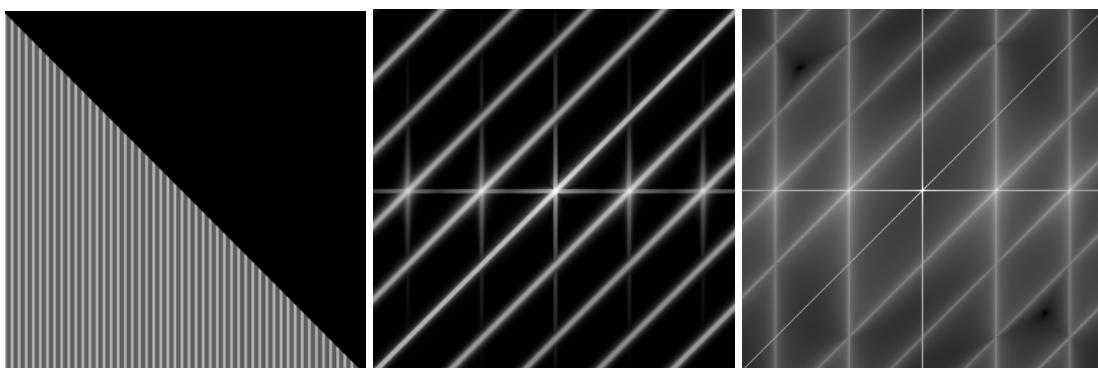
L'option Hamming, lorsqu'elle est activée, applique une fenêtre de Hamming à l'image avant de calculer son spectre. Cela atténue les effets des hautes fréquences et réduit les artefacts dans le spectre en lissant les bords de l'image. En conséquence, le spectre obtenu sera moins bruité et plus lisse aux bords, ce qui peut améliorer la qualité de l'analyse fréquentielle de l'image.

Voiçi une comparaison entre l'option Hamming=True (à gauche) et Hamming=False (à droite) de l'image **lena.tif**.

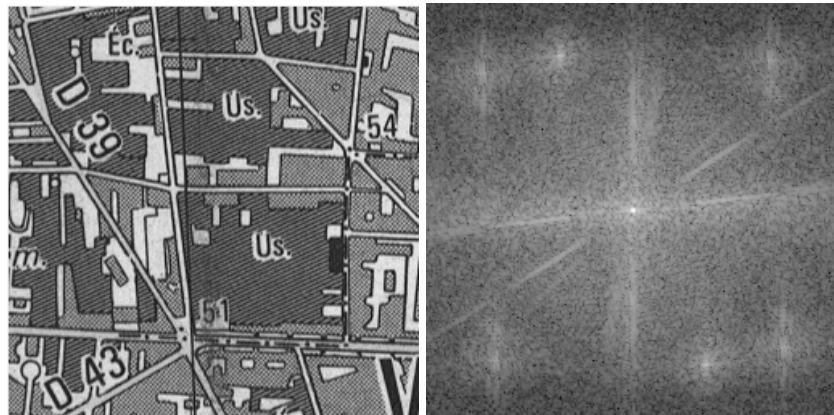


On peut observer que les rayons sont plus prononcés et plus contrastés par rapport à l'utilisation de l'option Hamming.

Visualisez le spectre de l'image synthétique rayures.tif. Que constatez-vous?
Peut-on retrouver les caractéristiques des rayures de l'image à partir de son spectre? Expliquez la différence entre la 3 visualisation avec et sans l'option hamming? Quel effet a le sous-échantillonnage sur le spectre (on peut utiliser une image synthétique et une image naturelle (Par exemple carte_nb.tif)).



On constate que cette fois-ci, les deux spectres avec l'option 1 et 2 se ressemblent, et ces derniers reflètent beaucoup d'informations de l'image. Oui, il est possible de retrouver les caractéristiques des rayures d'une image à partir de son spectre. Le spectre contient des informations sur les fréquences spatiales présentes dans l'image. Dans notre cas, les caractéristiques telles que la largeur, l'espacement et l'orientation des rayures se reflètent dans le spectre. Lorsqu'on ne fait pas usage de l'option Hamming, l'image apparaît moins nette, montrant moins de distinction entre les fréquences. En cas de sous-échantillonnage de l'image, on remarque une image plus nette, ce qui traduit une augmentation de la fréquence globale de l'image.



4.2 Ringing

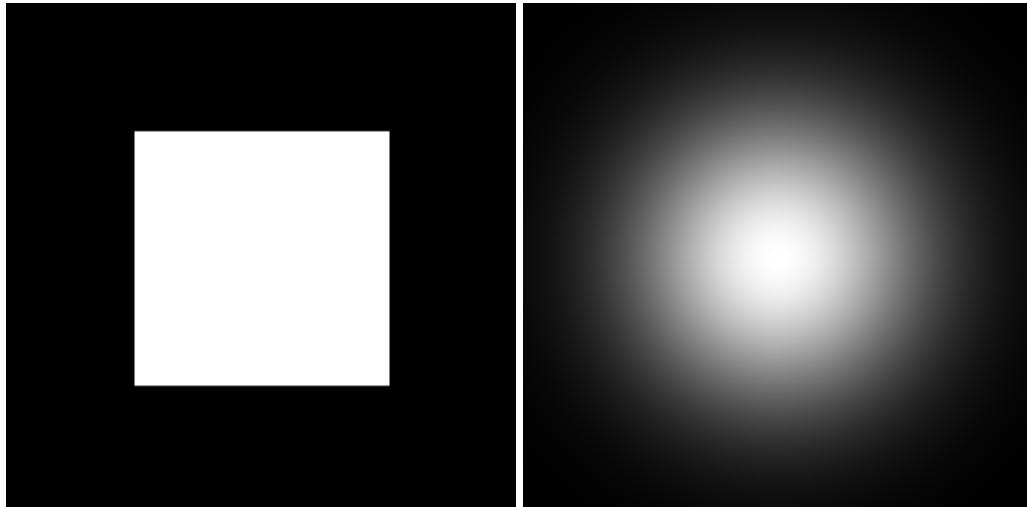
A l'aide de la fonction filterlow, appliquez un filtre passe bas parfait à une image. Visualisez l'image résultante, ainsi que son spectre. Que constatez-vous? Mêmes questions en utilisant la commande filtergauss.



Lorsque nous appliquons un filtre passe-bas, nous éliminons les fréquences élevées de l'image, ce qui les supprime également du spectre. En ce qui concerne le filtre de Gauss, on peut observer que le spectre devient plus

régulier, ce qui est logique puisqu'un filtre capable de réduire le bruit dans un signal d'entrée, pour diminuer la distorsion d'une image, diminue également la variation de fréquence.

Visualisez les deux masques (sous le répertoire images) masque_bas_centre.tif (pour le filtrage passe-bas parfait) et masque_gauss_centre.tif (pour le filtrage gaussien).



Quelle différence constatez-vous, en particulier quelle conséquence a la discontinuité de la transformée de Fourier sur la vitesse de décroissance du filtre spatial correspondant?

Le **masque_bas_centre.tif** démontre l'élimination des fréquences plus élevées, tandis que le **masque_gauss_centre.tif** montre une réduction semblable à une fonction gaussienne.

La discontinuité entraîne l'apparition d'effets de rebond, se manifestant par la duplication des contours, comme on peut le voir près du chapeau de la femme. Interprétation de ringing : la transformée de fourier change la régularité et introduit des oscillations à l'infini.