

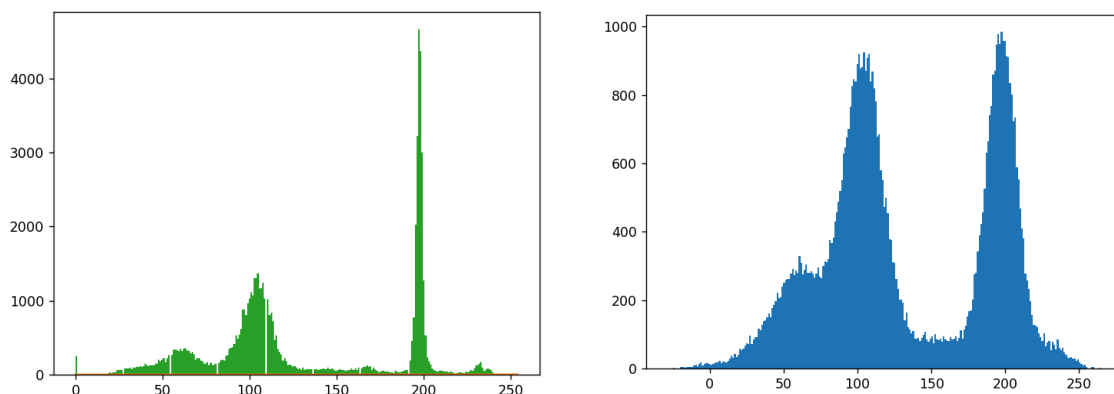
TP01 - Miguel Angel RODRIGUEZ FUENTES

2.1. Lorsqu'une image plus grande est réduite sans aucune méthode d'interpolation, on peut observer une perte d'informations dans les pixels résultants, en particulier au niveau des bords. En effet, en réduisant l'image de moitié, seuls un pixel sur deux est conservé, ce qui résulte de l'aliasing. L'hypothèse est qu'en agrandissant l'image, le nouveau pixel généré doit être la moyenne des pixels voisins.

2.2. Parce que nous prenons la palette de couleurs sur une circonférence avec un domaine de -180° à 180° , avec les valeurs originales à 0° , et les couleurs primaires et secondaires séparées de 60° . Au moment de changer la teinte, nous faisons pivoter la palette de couleurs et réattribuons de nouvelles valeurs pour chacune des couleurs primaires et complémentaires. Pour cette raison, lorsque nous sommes aux positions extrêmes de -180° à 180° , nous sommes théoriquement au même angle de rotation de la palette de couleurs.

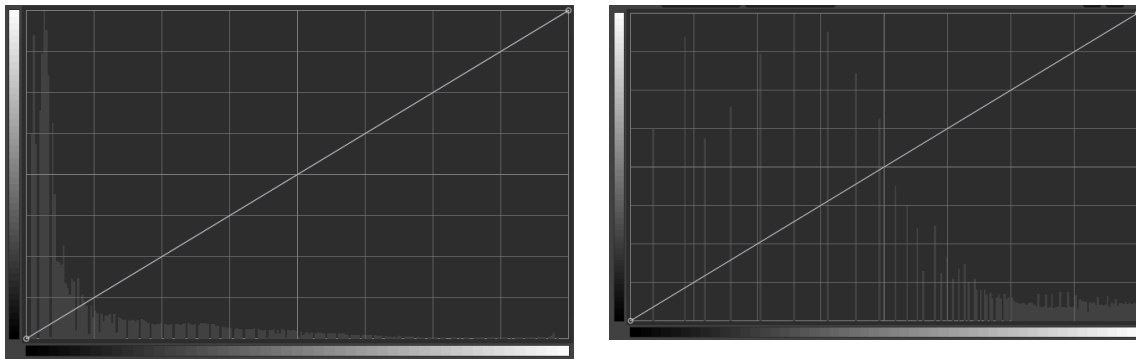
Cela se réfère à l'intensité de la couleur. Lorsque nous regardons le cube RGB, la saturation à 100 % correspond à envoyer la valeur RGB vers son coin le plus proche. À -100 %, cela correspond à la projection de cette valeur sur le coin vers la diagonale du centre du cube, acquérant ainsi sa valeur en échelle de gris.

3.1. Dans les histogrammes, nous pouvons voir que le bruit gaussien ajoute de l'intensité à tous les pixels de l'image. Ainsi, en réalisant l'histogramme correspondant, nous observons une augmentation du nombre de pixels avec des tons plus gris ou plus foncés par rapport à l'image originale, éliminant presque totalement les pixels clairs ou ceux situés à l'extrémité supérieure de l'histogramme.



3.2. De manière générale, avec des fonctions croissantes, seule la relation entre l'entrée et la sortie du ton des pixels change, c'est-à-dire que nous augmentons ou réduisons simplement leur intensité de gris. Cependant, lorsqu'il s'agit d'une fonction décroissante, nous travaillons sur le négatif de l'image, car, par exemple, dans un cas extrême, nous pourrions avoir un pixel d'entrée à 0 et une sortie à 255, et vice versa. Pour conclure, les fonctions décroissantes sur la tonalité nous amènent à travailler sur le négatif de l'image originale.

3.3.



En modifiant le contraste à partir de l'histogramme, on peut voir que la distribution des valeurs des pixels devient plus uniforme, ce qui se traduit par une image plus claire, notamment si celle-ci est composée de nombreux tons sombres.

3.4. Ce code a pour objectif d'égaliser l'histogramme d'une image, ce qui signifie redistribuer les valeurs des pixels de manière à rendre son histogramme le plus uniforme possible. Tout d'abord, l'image est chargée en utilisant la fonction ``skio.imread`` de la bibliothèque ``skimage``. Ensuite, l'égalisation de l'histogramme est effectuée grâce à la fonction ``exposure.equalize_hist``. Cette fonction ajuste les valeurs d'intensité des pixels pour qu'elles se distribuent de façon plus uniforme, ce qui améliore généralement le contraste de l'image.

Ensuite, le code visualise à la fois l'image originale et l'image avec l'histogramme égalisé en utilisant la bibliothèque ``matplotlib``. Deux sous-graphiques sont créés pour afficher côte à côte les deux versions de l'image, facilitant ainsi la comparaison visuelle des résultats.

Enfin, le code affiche les histogrammes des deux images dans deux autres sous-graphiques. Cela permet de visualiser comment le processus d'égalisation modifie la distribution des niveaux d'intensité. Dans l'image originale, l'histogramme présente généralement des pics et des vallées, tandis que dans l'image égalisée, l'histogramme a tendance à être plus plat, distribuant les intensités des pixels de manière plus uniforme.

```
# Charge l'image
u = skio.imread('images/vue1.tif')

# Faire correspondre l'histogramme
u_equalized = exposure.equalize_hist(u)

# Afficher l'image originale et l'image correspondant à l'histogramme
fig, ax = plt.subplots(1, 2, figsize=(12, 6))
ax[0].imshow(u, cmap='gray')
ax[0].set_title('Imagen Original')
ax[0].axis('off')

ax[1].imshow(u_equalized, cmap='gray')
ax[1].set_title('Imagen con Histograma Igualado')
ax[1].axis('off')
```

```
plt.show()
```

```
# Afficher l'histogramme de l'image originale et l'image correspondant à l'histogramme
fig, ax = plt.subplots(1, 2, figsize=(12, 6))
ax[0].hist(u.ravel(), bins=256, histtype='step', color='black')
ax[0].set_title('Histograma de la Imagen Original')

ax[1].hist(u_equalized.ravel(), bins=256, histtype='step', color='black')
ax[1].set_title('Histograma de la Imagen Igualada')
```

```
plt.show()
```

3.5. L'image binarisée sans ajout de bruit (seuil simple) a tendance à montrer des artefacts visuels et peut sembler très abrupte dans les transitions entre le blanc et le noir. L'image binarisée avec dithering (bruit + seuil) paraît généralement plus proche de l'image originale, car le bruit introduit des variations dans les niveaux de gris, créant un effet visuel qui simule des dégradés.

La probabilité qu'un pixel de niveau x devienne blanc après l'ajout de bruit et l'application du seuil dépend à la fois du niveau de gris x et du type et du niveau du bruit ajouté. Si le bruit est gaussien et le seuil est fixe, la probabilité peut s'exprimer comme l'intégrale de la fonction de distribution du bruit au-dessus du seuil. En termes plus simples, si le niveau x est proche du seuil, le dithering rend la décision de savoir si le pixel sera blanc ou noir plus aléatoire, ce qui adoucit la perception de l'image finale.

3.6. La distribution des différences de niveau de gris entre pixels adjacents suit généralement une forme proche d'une distribution gaussienne centrée sur zéro, surtout dans les images naturelles. Cela s'explique par le fait que, dans la plupart des images, les changements entre pixels voisins sont petits, en raison de la continuité présente dans la majorité des scènes.

Si nous considérons les différences entre des pixels plus distants, l'histogramme s'aplatirait et aurait une variance plus élevée. En effet, à mesure que la distance entre les pixels augmente, les différences ont tendance à être plus grandes et moins prévisibles.

4.1. Le spectre d'une image fournit des informations clés sur ses composantes de fréquence spatiale. En visualisant les spectres de différentes images, nous remarquons que les images naturelles, comme `maison.tif`, présentent un spectre avec la majorité de leur énergie concentrée dans les basses fréquences, reflétant les variations douces et continues de ces scènes. En revanche, les images avec des motifs répétitifs, comme `rayures.tif`, montrent des pics dans les hautes fréquences, indiquant la périodicité et l'orientation des éléments dans l'image.

L'option de Hamming affecte le spectre en appliquant une fenêtre qui adoucit les bords de l'image avant d'effectuer la transformation. Cela réduit les artefacts et le "leakage" dans le spectre, ce qui donne une représentation plus claire et précise des caractéristiques de

l'image. Sans la fenêtre de Hamming, le spectre peut présenter plus de bruit et d'artefacts, rendant l'interprétation des motifs plus difficile.

Dans l'image rayures.tif, les pics définis dans le spectre permettent d'identifier l'orientation et la périodicité des rayures. Cela démontre que les caractéristiques des motifs périodiques dans l'image se reflètent clairement dans son spectre.

Enfin, si une image est sous-échantillonnée, le spectre changera, pouvant introduire le phénomène d'aliasing, qui déforme la représentation des fréquences présentes. Cela souligne l'importance d'un échantillonnage adéquat pour maintenir l'intégrité de l'information spectrale.

4.2. Lorsque nous appliquons un filtre passe-bas parfait (`filterlow``) à une image, les hautes fréquences sont éliminées, ce qui adoucit l'image en réduisant les détails. Cependant, ce type de filtrage provoque un effet connu sous le nom de "ringing" ou de bandes artificielles autour des objets. Cela se produit en raison de la discontinuité abrupte dans le domaine de Fourier, car le filtre idéal coupe les fréquences de manière soudaine, causant des artefacts visuels dans l'image résultante.

En appliquant un filtre gaussien (`filtergauss``), l'image est également adoucie, mais de manière plus progressive. Le spectre résultant montre une transition plus douce entre les fréquences permises et bloquées. Grâce à cette douceur, l'effet de "ringing" est beaucoup moins prononcé, car il n'y a pas de discontinuité aussi marquée dans le domaine de Fourier.

En comparant les deux masques (`masque_bas_centre.tif`` pour le filtre idéal et `masque_gauss_centre.tif`` pour le gaussien), on observe que le masque du filtre idéal présente une transition brusque, tandis que celui du filtre gaussien est plus doux. Cette douceur dans le filtre gaussien implique une atténuation plus progressive des hautes fréquences, ce qui évite le "ringing" et offre une image lissée avec moins de distorsions. En résumé, la discontinuité du filtre idéal provoque des artefacts, tandis que le filtre gaussien évite ces problèmes en réduisant les hautes fréquences de manière plus douce.