

ex5

September 18, 2024

1 ex5: High Boost

```
[81]: import cv2
import numpy as np
import matplotlib.pyplot as plt
```

```
[82]: def plot_images(img1, title1, img2, title2, img3, title3, img4, title4):
    fig = plt.figure(figsize=(6, 7))

    fig.add_subplot(2, 2, 1)
    plt.imshow(img1)
    plt.axis('off')
    plt.title(title1)

    fig.add_subplot(2, 2, 2)
    plt.imshow(img2)
    plt.axis('off')
    plt.title(title2)

    fig.add_subplot(2, 2, 3)
    plt.imshow(img3)
    plt.axis('off')
    plt.title(title3)

    fig.add_subplot(2, 2, 4)
    plt.imshow(img4)
    plt.axis('off')
    plt.title(title4)

    plt.subplots_adjust(wspace=0, hspace=0.17)
    plt.show()
```

`high_boost_filter` aplica un filtro que realza los detalles de una imagen aumentando los componentes de la alta frecuencia de una imagen. El parámetro `A` controla la cantidad de dicha amplificación.

```
[83]: def high_boost_filter(image, A):  
    kernel = np.array([[ -1,    -1,    -1],  
                       [-1,   A + 8,  -1],  
                       [-1,    -1,    -1]])  
    return cv2.filter2D(image, ddepth=-1, kernel=kernel)
```

Si aumentamos el valor de A:

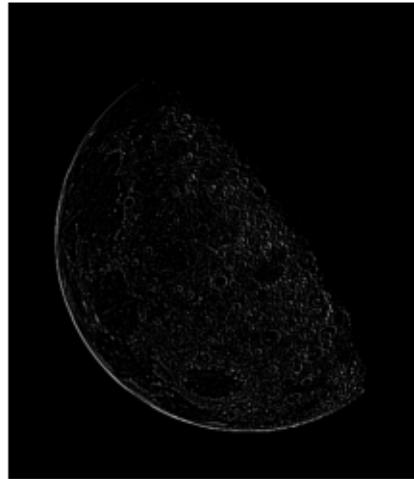
```
[86]: def show_high_boost(image):  
    A = 0  
    filtered_image = high_boost_filter(image, A)  
    A1 = 1.0  
    filtered_image1 = high_boost_filter(image, A1)  
    A2 = 1.5  
    filtered_image2 = high_boost_filter(image, A2)  
    plot_images(image, "Original",  
                filtered_image, f"Laplaciano A={A}",  
                filtered_image1, f"High Boost A={A1}",  
                filtered_image2, f"High Boost A={A2}")
```

```
[87]: image = cv2.imread('assets/blurry_moon.tif')  
show_high_boost(image)
```

Original



Laplaciano A=0



High Boost A=1.0



High Boost A=1.5



En principio, incrementar el valor de A resulta en una mejora pronunciada de los bordes y detalles ya que suelen ser los componentes con mayor valor de frecuencia espacial de una imagen, pero pasado cierto valor se vuelven a perder.