



## Filtros Lineales

Ariel Plasencia Díaz

[ARIELPLASENCIA00@GMAIL.COM](mailto:ARIELPLASENCIA00@GMAIL.COM)

# Índice

<b>1</b>	<b>Introducción</b>	<b>3</b>
<b>2</b>	<b>Filtro Bilateral</b>	<b>3</b>
2.1	Filtrado gaussiano . . . . .	3
2.2	Filtrado bilateral . . . . .	3
2.3	Fórmulas del filtrado bilateral . . . . .	4
<b>3</b>	<b>Filtrado bilateral en Python</b>	<b>5</b>
3.1	Sintaxis . . . . .	5
3.2	Parámetros . . . . .	5
<b>4</b>	<b>Ejemplo de filtrados usando OpenCV</b>	<b>5</b>
4.1	Desenfocar usando un filtro de mediana . . . . .	5
4.2	Desenfocar usando un filtro gaussiano . . . . .	6
4.3	Desenfocar usando un filtro bilateral . . . . .	7
<b>5</b>	<b>Conclusiones</b>	<b>7</b>

## 1

## Introducción

En esta pequeña investigación, hablaremos sobre el filtrado bilateral sobre imágenes en escala de grises. Además, profundizaremos en la sintaxis de la función `cv2.bilateralFilter()` utilizada por la librería OpenCV provista por el lenguaje de programación python y llevaremos a cabo diferentes ejemplos para una mejor y mayor comprensión. Como estudiamos en conferencias, para realizar el suavizado de imágenes se pueden utilizar tres tipos de técnicas de filtrado: desenfoque promedio, desenfoque gaussiano y desenfoque medio. Estos métodos difuminan o suavizan todo, independiente de si se trata de ruido o bordes, por lo que existe una pérdida importante de información en las imágenes. Para superar estos contratiempos proponemos el método de filtrado bilateral.

## 2

## Filtro Bilateral

Un filtro bilateral es un filtro de suavizado no lineal, que conserva los bordes y reduce el ruido para las imágenes. Reemplaza la intensidad de cada píxel con un promedio ponderado de los valores de intensidad de los píxeles cercanos.

La principal propiedad del filtrado bilateral es que no promedia entre los bordes. Es por eso que también se le llama filtro de preservación de bordes. A continuación, examinaremos algunas propiedades matemáticas detrás de este método de filtrado, pero antes de eso, haremos un rápido e importante resumen del filtrado gaussiano ya que estos filtros poseen características comunes sobre todo en el plano matemático.

### 2.1 Filtrado gaussiano

El filtrado gaussiano es un promedio ponderado de la intensidad de las posiciones adyacentes con un peso que disminuye con la distancia espacial a la posición central. Matemáticamente, la imagen filtrada con desenfoque gaussiano (GB) viene dada por:

$$GB[I]_p = \sum_{q \in S} G_\sigma(\|p - q\|) I_q \quad (1)$$

En la ecuación anterior,  $p$  y  $q$  corresponden con la posición de los píxeles de la imagen y  $G_\sigma(x)$  denota el núcleo gaussiano bidimensional:

$$G_\sigma(x) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2}{2\sigma^2}\right) \quad (2)$$

Básicamente,  $G_\sigma$  es un gaussiano espacial que disminuye la influencia de píxeles distantes. La distancia está definida por  $G_\sigma(\|p - q\|)$ , donde  $\sigma$  es un parámetro que define la extensión o el tamaño de la vecindad.

### 2.2 Filtrado bilateral

Como hemos visto anteriormente, en el filtro gaussiano solo se consideran los píxeles cercanos durante el filtrado. No se consideran los píxeles con la misma intensidad ni si un píxel es constituido por un borde. Por lo tanto, el filtrado gaussiano difumina los bordes, lo que no queremos hacer ya que elimina detalles cruciales de la imagen. Por otra parte, el filtrado bilateral también requiere un filtro gaussiano en el espacio, pero además considera otro filtro de tipo gaussiano que es una función que calcula la diferencia de píxeles. La función gaussiana del espacio asegura que solo los píxeles cercanos se consideren para desenfoque, mientras que la función gaussiana de diferencia de intensidad asegura que solo aquellos píxeles con intensidades similares al píxel central se consideren para desenfoque. Por lo tanto, conserva los bordes, ya que los píxeles en los bordes tendrán una gran variación de intensidad. El punto importante que se considera en el filtrado bilateral es que los dos píxeles están cerca uno del otro no solo si ocupan ubicaciones espaciales cercanas, sino también si tienen alguna similitud en el rango fotométrico. Estas propiedades del filtrado bilateral superan el inconveniente de otras técnicas

como desenfoque promedio, desenfoque gaussiano y desenfoque medio, ya que es capaz de preservar los bordes.

### 2.3 Fórmulas del filtrado bilateral

Matemáticamente, el filtro bilateral (BF) viene dado por la siguiente ecuación:

$$BF[I]_p = \frac{1}{W_p} \sum_{q \in S} G_{\sigma_s}(\|p - q\|) G_{\sigma_r}(I_p - I_q) I_q \quad (3)$$

donde  $W_p$  es un factor de normalización

$$W_p = \sum_{q \in S} G_{\sigma_s}(\|p - q\|) G_{\sigma_r}(I_p - I_q) \quad (4)$$

Entonces, como vimos en la ecuación 3, se agregan dos nuevos términos con respecto a la ecuación 1 (correspondiente al filtro gaussiano) para convertirse en el filtro bilateral. El primero de ellos es:

$$G_{\sigma_r}(I_p - I_q) I_q \quad (5)$$

y el segundo:

$$\frac{1}{W_p} \quad (6)$$

Básicamente,  $\sigma_s$  es el parámetro de espacio y  $\sigma_r$  es el parámetro de rango. Es decir,  $\sigma_s$  representa la extensión espacial del tamaño del núcleo de la vecindad considerada y  $\sigma_r$  representa la amplitud mínima de un borde. Esto significa que los parámetros  $\sigma_s$  y  $\sigma_r$  medirán la cantidad de filtrado para la imagen "I".

La idea principal detrás de estas fórmulas es que:

1. Cada píxel se reemplaza por un promedio ponderado de sus vecinos.
2. Cada vecino está ponderado por un componente espacial que penaliza los píxeles distantes y un componente de rango que penaliza los píxeles con una intensidad diferente.
3. La combinación de ambos componentes asegura que solo los píxeles similares cercanos contribuyan al resultado final.

Como hemos visto con anterioridad el filtro bilateral está controlado por los parámetros  $\sigma_s$  y  $\sigma_r$ , los cuales tienen una estrecha relación. A medida que el parámetro de rango  $\sigma_r$  aumenta, el filtro bilateral se acerca al desenfoque gaussiano, sin embargo, con el aumento del parámetro espacial  $\sigma_s$  se suavizan las características mas grandes de la imagen.

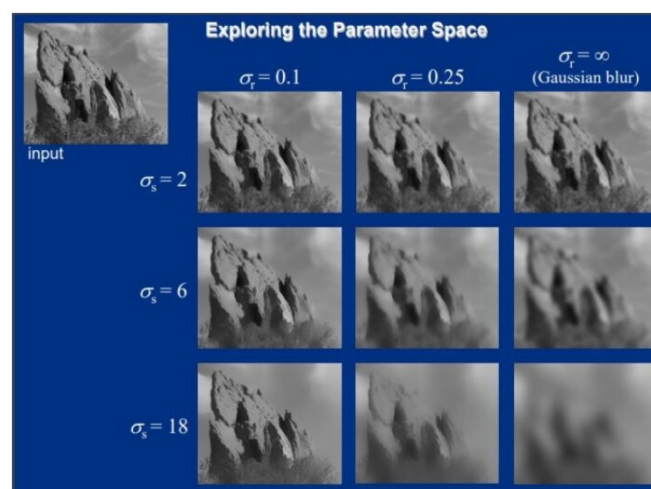


Figura 1: Relación entre los parámetros  $\sigma_s$  y  $\sigma_r$

## 3

### Filtrado bilateral en Python

Para realizar el filtrado bilateral en el lenguaje de programación python utilizamos la librería OpenCV, en la cual existe una función llamada `bilateralFilter()`. A continuación presentaremos su sintaxis.

#### 3.1 Sintaxis

```
cv2.bilateralFilter(src, dst, d, sigmaColor, sigmaSpace, borderType = BORDER_DEFAULT)
```

#### 3.2 Parámetros

- **src**: Es la imagen que se va a difuminar.
- **dst**: Es la imagen destino del mismo tamaño y tipo que la imagen anterior.
- **d**: Diámetro de cada vecindad de píxeles que se utiliza durante el filtrado. Si no es positivo, se calcula a partir de *sigmaSpace*.
- **sigmaColor**: Filtro sigma en el espacio de los colores. Un valor mayor que el parámetro significa que los colores más lejanos dentro de la vecindad de píxeles se mezclarán, dando como resultado áreas más grandes de color semi igual.
- **sigmaSpace**: Filtro sigma en el espacio de las coordenadas. Un valor mayor que el parámetro significa que los píxeles más lejanos influirán entre sí siempre que sus colores estén lo suficientemente cerca. Cuando  $d > 0$ , se especifica el tamaño de la vecindad independientemente de *sigmaSpace*. De lo contrario,  $d$  es proporcional a *sigmaSpace*.
- **borderType**: Modo de borde utilizado para extrapolar píxeles fuera de la imagen.

## 4

### Ejemplo de filtrados usando OpenCV

En esta sección, aplicaremos una serie de algoritmos para el filtrado de imágenes en python usando OpenCV y matplotlib en una imagen de ejemplo. Haremos una comparación entre los distintos métodos de suavizado de imágenes con el objetivo de apreciar cómo el filtrado bilateral conserva los bordes durante el proceso. Cabe mencionar que adjuntamos otras imágenes, a las cuales les aplicamos el mismo procedimiento.

#### 4.1 Desenfocar usando un filtro de mediana

La técnica del filtro de mediana es muy similar a la técnica del filtro promedio. La única diferencia es que se calcula la mediana de todos los píxeles debajo de la ventana del kernel y el píxel central se reemplaza por este valor en lugar del valor promedio.

```
import cv2
from image import Image
import matplotlib.pyplot as plt

if __name__ == '__main__':
    image = Image.load_image('../img/image1.png')
    median_blur_filter_image = cv2.medianBlur(image, 5)
    Image.show_images([image, median_blur_filter_image])
    Image.plt_show()
```

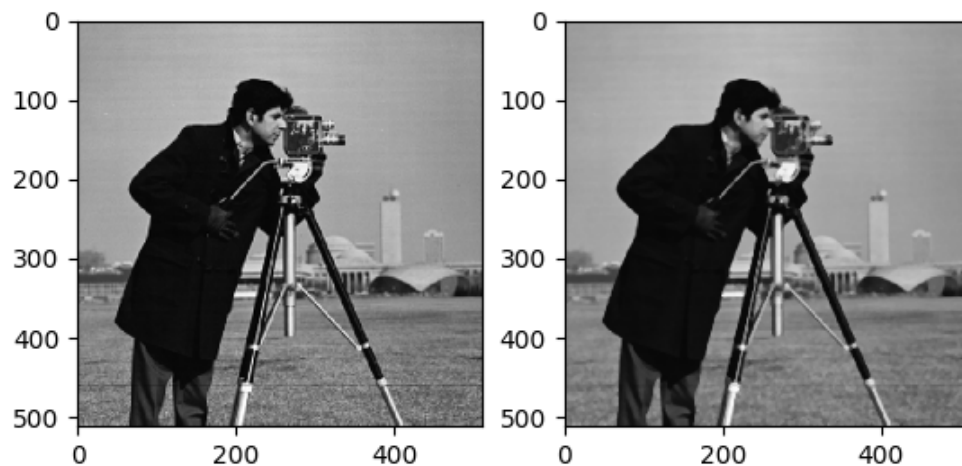


Figura 2: Imagen original y resultado usando filtrado de mediana (de izquierda a derecha)

#### 4.2 Desenfocar usando un filtro gaussiano

El desenfoque de imágenes mediante filtros gaussianos se realiza según las ecuaciones explicadas en el epígrafe 2.1, en esencia, la imagen se convoluciona con un filtro gaussiano en lugar de un cuadro o filtro normalizado.

```
import cv2
from image import Image
import matplotlib.pyplot as plt

if __name__ == '__main__':
    image = Image.load_image('./img/image1.png')
    gaussian_blur_filter_image = cv2.GaussianBlur(image, (5,5), sigmaX=0)
    Image.show_images([image, gaussian_blur_filter_image])
    Image.plt_show()
```

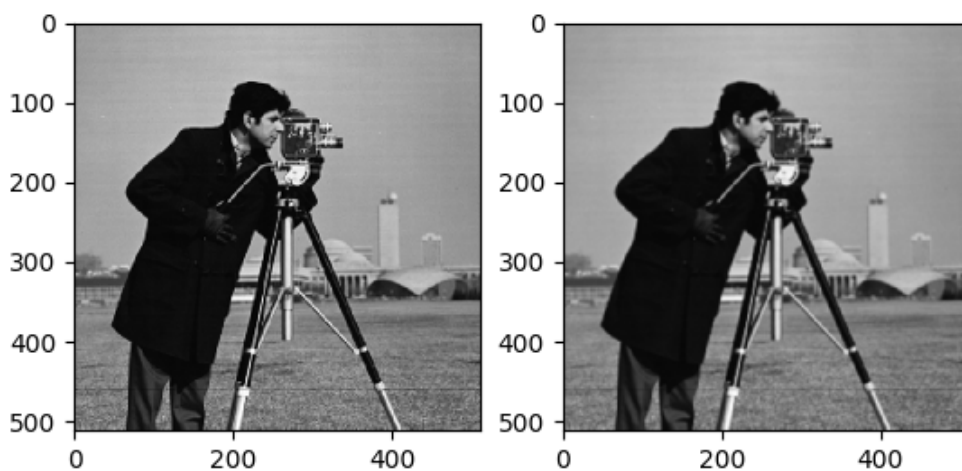


Figura 3: Imagen original y resultado usando filtrado gaussiano (de izquierda a derecha)

### 4.3 Desenfocar usando un filtro bilateral

```
import cv2
from image import Image
import matplotlib.pyplot as plt

if __name__ == '__main__':
    image = Image.load_image('../img/image1.png')
    bilateral_filter_image = cv2.bilateralFilter(image, 15, 80, 80)
    Image.show_images([image, bilateral_filter_image])
    Image.plt_show()
```

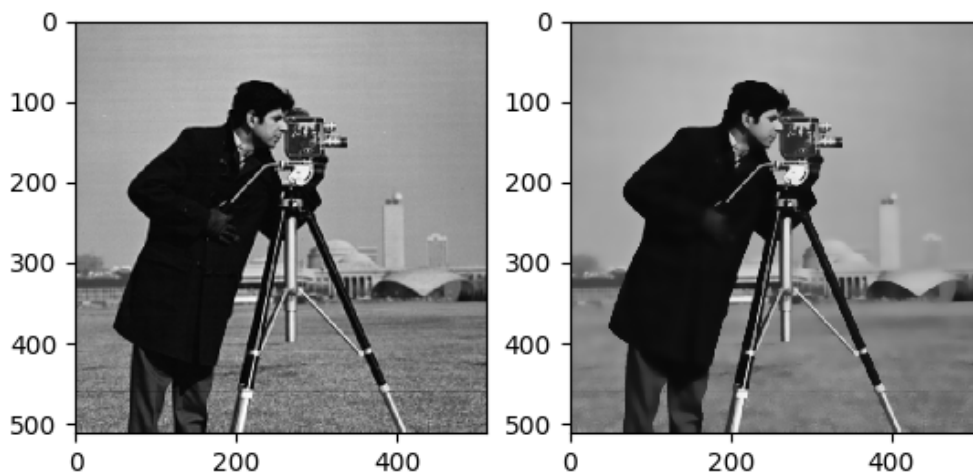


Figura 4: Imagen original y resultado usando filtrado bilateral (de izquierda a derecha)

Como podemos apreciar, el efecto de suavizado o desenfoque de la imagen logrado por el filtrado bilateral ha conservado los bordes de manera hermosa y distintiva en comparación con las restantes técnicas de filtrado.

## 5

### Conclusiones

A lo largo de esta pequeña investigación, aprendimos cómo podemos suavizar una imagen mediante el filtrado bilateral. Cubrimos los conceptos fundamentales en detalle y también vimos un ejemplo de filtrado bilateral con la función `cv2.bilateralFilter()` provista por el lenguaje de programación Python y por la librería OpenCV. También hicimos una comparación entre las diferentes técnicas estudiadas donde, sin dudas, hay una diferencia marcada por el filtro bilateral.

### Referencias

- [1] Ángela León Mecías. *Curso Optativo Procesamiento de Imágenes*.
- [2] Charles L. Epstein. *Introduction to the Mathematics of Medical Imaging*.
- [3] Chris Solomon, Toby Breckon. *Fundamentals of Digital Image Processing*.