

Realce de imagen usando máscara de enfoque (Sharp mask)

Objetivos de la práctica:

- Aprender a mejorar una imagen aplicando un realce por máscara de enfoque (sharp mask).
- Aprender aplicar una convolución a la imagen..
- Aprender a construir filtros paso banda (Laplaciano y DoG).
- Aprender a utilizar la GUI de openCV usando controles con barra de deslizamiento.

Descripción (hasta 6p.):

Recuerda que una señal digital (una imagen por ejemplo) puede descomponerse en versiones de la misma en distintas frecuencias y en general podríamos simplificar diciendo que $I = I_l + I_h$ donde I_l , I_h son las versiones de baja y alta frecuencia respectivamente de la imagen. Las bajas frecuencias representan las estructuras mayor escala de la imagen y las altas frecuencias los detalles de menor escala.

La técnica de realce denominada “sharpening” consiste en realizar la imagen I usando la siguiente operación:

$$G = gI + I_h,$$

es decir, sumamos a la imagen amplificada una versión de las altas frecuencias y por lo tanto los detalles de las misma (también el ruido) ganan peso por lo que la imagen resultante G tendrá realzados estos detalles.

Para obtener la versión de alta frecuencia de la imagen podremos utilizar un filtro que aproxime el operador Laplaciano de tal forma que la máscara de enfoque (“sharp mask”) $I_h = -\nabla^2 I$. Si además al valor central del filtro le sumamos el factor de ganancia todo el proceso se hará en un solo paso al realizar la convolución ya que es una operación lineal.

<table><tr><td>0</td><td>-1</td><td>0</td></tr><tr><td>-1</td><td>g+4</td><td>-1</td></tr><tr><td>0</td><td>-1</td><td>0</td></tr></table>	0	-1	0	-1	g+4	-1	0	-1	0	<table><tr><td>-1</td><td>-1</td><td>-1</td></tr><tr><td>-1</td><td>g+8</td><td>-1</td></tr><tr><td>-1</td><td>-1</td><td>-1</td></tr></table>	-1	-1	-1	-1	g+8	-1	-1	-1	-1
0	-1	0																	
-1	g+4	-1																	
0	-1	0																	
-1	-1	-1																	
-1	g+8	-1																	
-1	-1	-1																	
(a) Filtro Sharp con Laplaciano de 5 puntos.	(b) Filtro Sharp con Laplaciano de 9 puntos.																		

Los requisitos básicos de la práctica realizar son crear un operador “sharp” donde:

- Se implementa dos funciones:

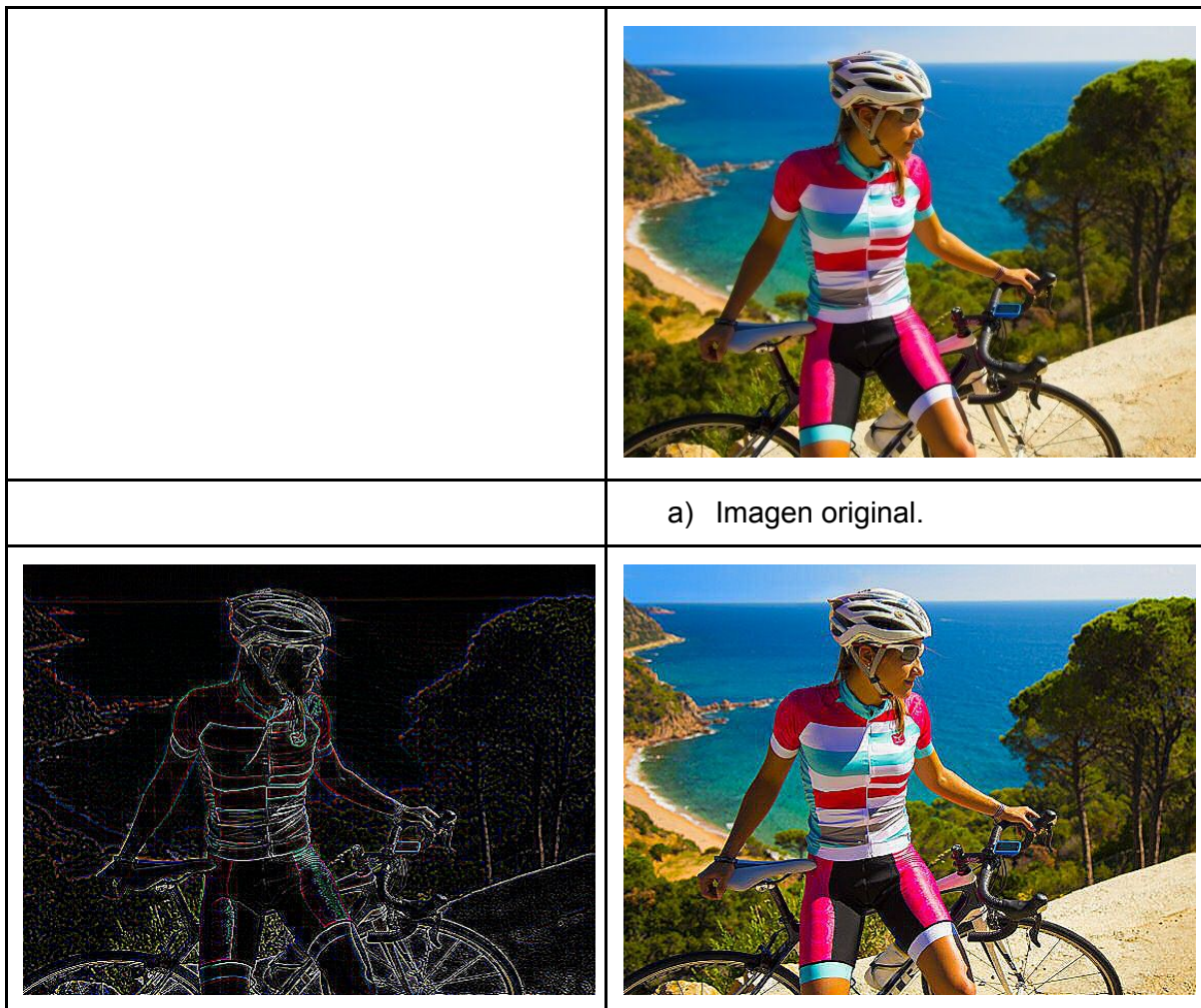
```
cv::Mat create_sharp_filter(int tipo, float g); //Crea un filtro
Sharp 5 puntos (tipo==0) o 9 puntos (tipo==1) con una ganancia
g>=0.0. El tipo de matriz será CV_32FC1.
```



```
void convolve(const cv::Mat& in, const cv::Mat& filter, cv::Mat&
out, int border_type=0); //calcula la convolución digital. La
imagen de salida tendrá la mismas dimensiones y tipo de la
imagen de entrada. La imagen original se extiende rellenando con
ceros para poder realizar la convolución en todos los puntos de
la imagen original. Precondiciones: in.type()==CV_32FC1 &&
filter.type()==CV_32FC1.
```

- Implementa un programa 'sharpen' con la siguiente CLI:

```
sharpen [-f <int=0>] [-g <float=1.0>] <input img> <output img>
```

- La opción 'f' controla el tipo de filtro: 0 para Laplaciano de 5 puntos, 1 para Laplaciano de 9 puntos, y 'g' la ganancia del filtro.



b) Laplaciano 4p.	c) Relace con laplaciano 4 puntos y ganancia 1.0
	
d) Laplaciano 9 puntos.	e) Relace con laplaciano 9 puntos y ganancia 1.0.

Opcional (hasta 4p.):

- (1.p) Podemos crear filtros paso banda para realzar sólo una banda de frecuencias, para ello podemos utilizar el operador DoG (Difference of Gaussians) que consiste en construir la máscara de la forma:

$$\nabla^2 I \approx DoG(\sigma_1, \sigma_2) = G_{\sigma_2} * I - G_{\sigma_1} * I = (G_{\sigma_2} - G_{\sigma_1}) * I, \sigma_1 < \sigma_2,$$



donde G sería un filtro gaussiano. Recuerda que para aplicar el proceso en un sólo paso, el proceso a realizar sería:

$$G = I * [g + (-DoG)]$$

siendo 'g' la ganancia que se sumaría al elemento central del filtro DoG en negativo. Así se trata de codificar una función que genera un filtro DoG ('-f=2') conocidas las dos sigmas y ganancia.

```
cv::Mat create_DoG_filter(float sigma1=0.5, float sigma2=1.0,
float g=1.0); //crea un filtro DoG.
```

Además añadir a la CLI dos opciones más '-s' para indicar sigma1, '-S' para sigma2.

	
<p>a) DoG con $\sigma_1=1.0$ y $\sigma_2=2.0$ (escalado para verlo bien)</p>	<p>b) Realce DoG (a) con $g=0.5$</p>

- (1.p) Añade el flag '-c' para activar que la función 'convolve' para realice convolución circular (la imagen se extiende de forma periódica en vez de rellenar con zeros.). Esto se corresponderá con un valor border_type=1 de la función convolve.
- (1.p) Añadir soporte en color. En una imagen en color se debe pasar a un espacio de color HSV y aplicar el realce al plano de iluminación (V) para deshacer el cambio al espacio RGB al final para almacenar.
- (1.p) Añadir modo interactivo para seleccionar los valores de los parámetros. El flag [-i] indica modo interactivo. Se muestra las imágenes original, la máscara de enfoque y la imagen procesada. Si el usuario pulsa la tecla "ESC" se sale sin guardar los resultados. Si pulsa la tecla "Intro" se sale guardando el resultado.

Recursos:

[Operador DoG.](#)

Para crear deslizadores en una ventana gráfica: [cv::createTrackBar](#).

Para comprobar tu función de convolución, puedes utilizar la función de opencv [cv::filter2D](#), que aplica un filtro (sin reflejarlo).