

## Práctica 1: Filtrado y Muestreo

Ignacio Aguilera Martos

### Ejercicio 1

#### Apartado A

Para aplicar la convolución con una máscara Gaussiana 2D en OpenCV nos valemos de la función `GaussianBlur` que toma como parámetros (los que nos interesan) la imagen, el tamaño del kernel y la varianza en el eje X y en el eje Y. Para esta prueba he movido la varianza entre 0 y 5 (en ambos ejes) y el tamaño del núcleo de 3 a 11 haciendo kernels cuadrados y rectangulares.

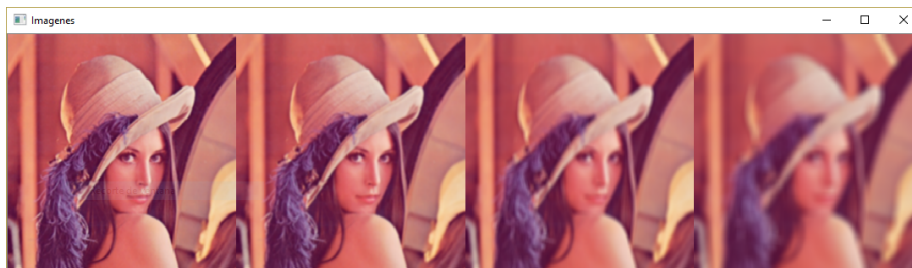


Figure 1: GaussianBlur

Podemos observar que el efecto se hace más visible de izquierda a derecha. Si nos fijamos en la imagen podemos ver que al hacer más grande el tamaño del núcleo podemos percibir de forma más nítida los rectángulos o cuadrados en la imagen. Además se observa que al aumentar los valores de sigma el suavizado se hace más visible.

#### Apartado B

En este caso el ejercicio nos pide que tomemos mediante la función `getDerivKernels` las máscaras 1-dimensionales que permiten calcular la convolución 2D con máscaras derivadas. Para ello la función `getDerivKernels` toma como parámetros el orden de la derivada tanto en el eje X como en el Y y el valor de sigma. Observemos el ejemplo para obtener conclusiones:

```
nacheteam@nacheteam-Equipo: ~/MEGA/Doble Grado DGIIM/5to Curso/Primer Cuatrimest...
Archivo Editar Ver Buscar Terminal Ayuda
Convolución gaussiana.
Kernel de primera derivada con varios tamaños.
Tamaño 3
Orden X 1
Orden Y 1
dx: [-1.  0.  1.]
dy: [-1.  0.  1.]
Tamaño 5
Orden X 2
Orden Y 2
dx: [ 1.  0. -2.  0.  1.]
dy: [ 1.  0. -2.  0.  1.]
Tamaño 7
Orden X 3
Orden Y 4
dx: [-1.  0.  3.  0. -3.  0.  1.]
dy: [ 1. -2. -1.  4. -1. -2.  1.]
Tamaño 11
Orden X 4
Orden Y 5
dx: [ 1.  2. -3. -8.  2. 12.  2. -8. -3.  2.  1.]
dy: [-1.  0.  5.  0. -10.  0. 10.  0. -5.  0.  1.]
Convolución laplaciana.
```

Figure 2: getDerivKernels

Debemos saber que la función `getDerivKernels` utiliza por debajo el operador de Sobel para hacer los cálculos. Interpretemos ahora los vectores obtenidos.

Podemos observar que el orden de la derivada en cada eje nos varía cada uno de los vectores correspondientes. Debemos recordar que este operador (Sobel) es empleado para reconocimiento de bordes en las imágenes, o lo que es lo mismo, un cambio muy notable de color en el entorno de un píxel. La función nos devuelve dos vectores que corresponden a la máscara, ya que es separable.

Miremos el primer caso. Tenemos los vectores  $(-1,0,1)$  y  $(-1,0,1)$  tanto para filas como columnas. Esto lo que va a hacer es dar como resultado la diferencia de valores de color que hay entre el píxel a la izquierda y derecha del que estamos teniendo en cuenta y la diferencia entre el píxel superior e inferior. En otras palabras, si hallamos una diferencia muy grande de color entre los mencionados es que tendremos un borde en la figura y por tanto al obtener la diferencia en valor absoluto tendremos un valor muy alto próximo al blanco en escala de grises, con lo que se resaltarán en blanco los bordes.

Veamos los ejemplos tres y cuatro en los que los órdenes de las derivadas no coinciden en el eje X e Y. En estos casos podemos observar que la posición central es cero cuando tenemos que el orden de derivación es impar y en estos casos cuanto mayor es el orden de derivación mayores son los números estrictamente anterior y posterior al central, con lo que mayor relevancia se les está dando en la resta a los mismos. Esto quiere decir que si los píxeles estrictamente al lado

del que estamos estudiando tienen valores muy diferentes el valor asignado al pixel será muy elevado y por tanto se le tomará como borde. Si el orden de la derivada es par podemos observar que la posición central deja de ser 0 para ser un número directamente proporcional al orden de derivación con lo que cuanto más alto sea el mismo mayor importancia se le asignará al color del pixel que estamos teniendo en cuenta. Además podemos observar que no realizan una resta de los píxeles que quedan a los lados del que estamos considerando de forma simétrica si no que lo que tiene en cuenta es de alguna manera como van haciendo el gradiente los píxeles colindantes hacia el centro.