Universidad de Granada

Ingeniería Informática

Computación y Sistemas Inteligentes

Cuestionario 1

Autor: José Antonio Ruiz Millán

Asignatura: Visión por Computador

27 de octubre de 2018

1. **Diga en una sola frase cuál cree que es el objetivo principal de la Visión por Computador. Diga también cuál es la principal propiedad de cara a los algoritmos que está presente en todas las imágenes.**

   La Vision por Computador tiene como principal objetivo extraer información de las imágenes para posteriormente ser procesada. A día de hoy, este procesamiento se aplica a través de un computador, por lo que podemos decir que se centra en utilizando un computador, ser capaz de extraer la información de una imagen, tratarla y procesarla para posteriormente tomar alguna decisión sobre ella.

   Por otra parte, la principal propiedad que poseen las imágenes es que todas tienen el mismo formato, es decir, ya sea una imagen a color, una imagen en grises, todas las imágenes estan representadas de la misma forma, lo que facilita mucho la lectura y el entendimiento sobre ellas a los algoritmos.

2. **Expresar las diferencias y semejanzas entre las operaciones de correlación y convolución. Dar una interpretación de cada una de ellas que en el contexto de uso en visión por computador.**

   Hablando en términos de aplicación, una convolución y un correlación son lo mismo, ya que el funcionamiento de las dos consiste en lo mismo, es una máscara 2D que va recorriendo una imagen sobre el conjunto de pixeles del mismo tamaño que la máscara, aplicando operaciones sobre la imagen y obteniendo nuevos valores para los píxeles, hasta recorrer la imagen completa.

   Una de las caracteristicas de estas operaciones y que las diferencian, es que dada una máscara H no simétrica, si aplicamos la correlación $G = H \bigotimes I$ donde I es una imagen donde todos los píxeles son 0 menos en el origen que tenemos un 1, la salida de esta operación será la propia H **reflejada**, mientras que si aplicamos la convolución $G = H * I$ donde I es la misma que antes y H está reflejada tanto en eje x como en el eje y de la anterior, obtendremos como resultado la propia H. Como he especificado, esto ocurre cuando no son simétricas, ya que si la máscara es simétrica, tanto la convolución como la correlación son exactamente lo mismo.

   Las fórmulas que se aplican tanto en la correlación como en la convolución son:

   - **Correlación:**

     $$G(i,j) = \sum_{u=-k}^{k} \sum_{v=-k}^{k} H[u,v]F[i+u, j+v]$$

     Que como he mostrado en el apartado anterior, se expresa como $G = H \bigotimes I$

   - **Convolución:**

     $$G(i,j) = \sum_{u=-k}^{k} \sum_{v=-k}^{k} H[u,v]F[i-u, j-v]$$

     Que como he mostrado en el apartado anterior, se expresa como $G = H * I$

   Ahora, tanto la convolucion como la correlacion cumplen la caracteristica de *linear Shift-Invariant (LSI)*, lo que nos dice que cumplen tanto el principio de superposición.

   $$h * (f_0 + f_1) = (h * f_o) + (h * f_1)$$

   como el principio de cambio invariante *(shift-invariant)*

   $$g(i,j) = f(i+u, j+v) \Leftrightarrow (h * g)(i,j) = (h * f)(i+u, j+v)$$

Además, tenemos que la convolución tiene una serie de propiedades interesantes que son la asociatividad, la conmutatividad y es distributiva respecto a la adición.

Una relacion entre la transformada de Fourier y estas operaciones es que la propia transformada de Fourier de una correlación es el producto de la primera transformación por el conjugado de la segunda, mientras que la transformada de Fourier de dos señales convolucionadas es el producto de sus transformaciones.

Por último, comentar que solemos utilizar las correlaciones para búsquedas de patrones dentro de una imagen y obtener así información de la posicion donde se puede encontrar ese patron, mientras que las convoluciones se pueden expresar como filtros o transoformaciones sobre las imágenes y deteccion de fronteras, como por ejemplo el paso de un filtro Gaussiano a una imagen para alisarla.

3. **¿Los filtros de convolución definen funciones lineales sobre las imágenes? ¿y los de mediana? Justificar la respuesta.**

   Los filtros de convolución sí definen una funcion lineal sobre la imagen ya que para el calculo del pixel se realiza utilizando una conbinacion de operaciones con los pixeles vecinos utilizando unos pesos y ese valor sería el calculado para el pixel mientras que usando los de mediana, se utiliza una información en la que no se compone como funcion lineal del resto de píxeles.

   Para demostrarlo, vamos a comprobar si cumplen las propiedades de una funcion lineal:

   - $f(x) + f(y) = f(x + y)$
   - $f(kx) = k(fx)$

   Por lo que ahora vamos a demostrar cada uno de los filtros:

   - **Convolución:**

     Para ello, utilizaré la media como filtro de convolución. Dados dos conjuntos de valores $X = \{x_1, x_2, ..., x_n\}$ e $Y = \{y_1, y_2, ..., y_n\}$.

     Si calculamos la media de X tenemos

     $$media(X) = \frac{1}{N} \sum_{i=1}^{N} x_i$$

     Si calculamos la media de Y tenemos

     $$media(Y) = \frac{1}{N} \sum_{i=1}^{N} y_i$$

     Por lo que tenemos que

     $$media(X) + media(Y) = \frac{1}{N} \sum_{i=1}^{N} x_i + \frac{1}{N} \sum_{i=1}^{N} y_i = \frac{1}{N} \sum_{i=1}^{N} x_i + y_i = media(X + Y)$$

     Con lo que cumple la primera propiedad.

     Ahora, para la segunda propiedad tenemos

     $$k \cdot media(X) = k \cdot \frac{1}{N} \sum_{i=1}^{N} x_i = \frac{1}{N} \sum_{i=1}^{N} k \cdot x_i = media(k \cdot X)$$

     Con lo que cumple la segunda propiedad y por tanto, podemos decir que **una convolución define funciones lineales sobre una imagen.**

- **Mediana:**

  Para este caso, mostraré un ejemplo en el que no se cumple la propiedad y por lo tanto al existir un ejemplo en el que no se cumple, demostramos que no lo es.

  Supongamos $X = \{9, 4, 5, 6, 7\}$ e $Y = \{5, 5, 4, 2, 1\}$ por lo tanto, $X + Y = \{14, 9, 9, 8, 8\}$

  $$mediana(X) = 6$$

  $$mediana(Y) = 4$$

  Por lo que tenemos que

  $$mediana(X) + mediana(Y) = 10$$

  $$mediana(X + Y) = 9 \neq 10$$

  Por lo que con este contraejemplo podemos decir que **los filtros de mediana no definen funciones lineales sobre una imagen.**

4. **Una operación de máscara que tipo de información usa, ¿local o global? Justificar la respuesta**

   La diferencia entre que una operacion use información local o global es que si utiliza información local, el valor de cada pixel se calcula respecto a un vencindario colindante con el propio pixel dado un tamaño de máscara y aplicando una combinacion de operaciones con todos los elementos de la máscara. Sin embargo, si la información es global, se aplica una transformación sobre un pixel sin necesidad de consultar o saber los valores de pixeles vecinos a éste.

   Esto significa que dada una imagen, si cambiamos los pixeles de posición, aplicamos una operacion que utilice información local y volvemos a poner los pixeles donde estaban, no conseguiremos un resultado correcto ya que al deformar la imagen, hemos perdido la conexión que había entre los pixeles y el calculo de un pixel no será el correcto. Sin embargo, si cambiamos los pixeles de la imagen, aplicamos una operación que utilice información global y volvemos a colocar los pixeles en su posición, el calculo de cada uno de los píxeles es el correcto, ya que no utiliza la informacion de los pixeles que están cerca a él.

   Por lo tanto, tenemos que **una operación de máscara utiliza información local** ya que sabemos que el funcionamiento de la misma se corresponde al explicado en el parrafo anterior, que dado una máscara de un tamaño concreto se va aplicando las correspondientes operaciones con la imagen recorriendo cada pixel y calculando su valor respecto al valor de los pixeles de su alrededor.

5. **¿De qué depende que una máscara de convolución pueda ser implementada de forma separable por filas y columnas? Justificar la respuesta**

   Para comprobar si una máscara de convolución se puede separar por filas y columnas se suele hacer por inspección, mirando la forma analítica de la máscara *([1]Freeman y Adelson 1991).* Sin embargo, un método más directo es tratar la máscara 2D como una matriz 2D K y tomar su descomposición en valores singulares (SVD)

   $$K = \sum_i \sigma_i u_i v_i^T$$

Si el primer valor singular σ_0 no es cero, la máscara es separable y $\sqrt{\sigma_0}u_o$ y $\sqrt{\sigma_0}v_0^T$ proporcionan las máscaras verticales y horizontales *([2]Perona 1995).*

Por otra parte, inspeccionando la máscara como he comentado, una forma de ver si es separable es viendo si el rango de la matriz es 1, por lo que en ese caso podríamos descomponerla. Por ejemplo, dada una matriz de rango 1 como la siguiente:

$$\begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix} \begin{pmatrix} 1 & 2 & 1 \end{pmatrix}$$

6. **Para implementar una función que calcule la imagen gradiente de una imagen cabe plantearse dos alternativas:**

   a) **Primero alisar la imagen y después calcular las derivadas sobre la imagen alisada**

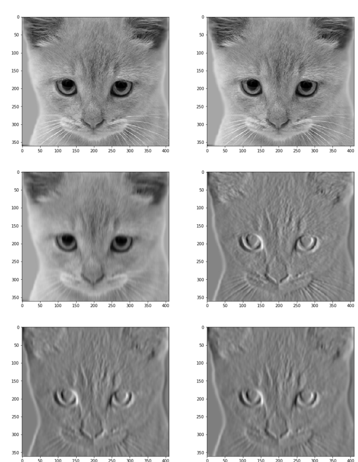
   b) **Primero calcular las imágenes derivadas y después alisar dichas imágenes.**

   Discutir y decir cuál de las estrategias es la más adecuada, si alguna lo es, tanto en el plano teórico como en el de la implementación. Justificar la decisión.

   Como he comentado en ejercicios anteriores, sabemos que las convoluciones cumplen una serie de caracteristicas, entre ellas, la conmutatividad. Por lo que sabiendo esto, podemos decir que **tanto la opcion *a* como la opción *b* serían correctas**, ya que al ser asociativa, nos da lo mismo aplicar primero un filtro y luego otro que alrevés, ya que finalmente como resultado obtendremos la misma imagen.

   Por otra parte, desde el punto de vista de la implementación para el caso *a)*, necesitaríamos hacer una convolución y luego dos derivadas, sin embargo, para el caso *b)* necesitaríamos hacer dos derivadas y luego dos convoluciones, asique en ese sentido nos conviene más la opción *a)*.

   Como ejemplo final, en la imagen siguiente podemos ver dos columnas, donde la primera columna muestra la imagen original, en segundo lugar la imagen con el filtro Gaussiano y la tercera, la imágen derivada de ésta última. Por otra parte, en la segunda columna tenemos la imagen original, en segundo lugar la imágen derivada de la imagen original y por último la imagen aplicando la Gaussiana a ésta última. Como podemos ver, al final de las operaciones, tanto por un lado como por otro, obtenemos el mismo resultado.



7. **Verificar matemáticamente que las primeras derivadas (respecto de x e y) de la Gaussiana 2D se puede expresar como núcleos de convolución separables por filas y columnas. Interpretar el papel de dichos núcleos en el proceso de convolución.**

   Dada la fórmula de la Gaussiana:

   $$G(x,y;\sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

   Ahora calculo las derivadas de respecto a x y respecto a y

   $$\frac{d}{dx} G(x,y;\sigma) = -\frac{1}{2\pi\sigma^4} x e^{-\frac{x^2+y^2}{2\sigma^2}}$$

   $$\frac{d}{dy} G(x,y;\sigma) = -\frac{1}{2\pi\sigma^4} y e^{-\frac{x^2+y^2}{2\sigma^2}}$$

   Voy a calcular respecto a la derivada, una separación de las variables.

   $$\frac{d}{dx} G(x,y;\sigma) = -\frac{1}{2\pi\sigma^4} x e^{-\frac{x^2+y^2}{2\sigma^2}}$$

   $$= -\frac{1}{2\pi\sigma^4} x e^{-(\frac{x^2}{2\sigma^2}+\frac{y^2}{2\sigma^2})}$$

   $$= -\frac{1}{2\pi\sigma^4} x e^{-\frac{x^2}{2\sigma^2}} e^{-\frac{y^2}{2\sigma^2}}$$

   $$= -\sqrt{\frac{1}{2\pi\sigma^4}}^2 x e^{-\frac{x^2}{2\sigma^2}} e^{-\frac{y^2}{2\sigma^2}}$$

   $$= -\sqrt{\frac{1}{2\pi\sigma^4}}\sqrt{\frac{1}{2\pi\sigma^4}} x e^{-\frac{x^2}{2\sigma^2}} e^{-\frac{y^2}{2\sigma^2}}$$

   $$= (-\sqrt{\frac{1}{2\pi\sigma^4}} x e^{-\frac{x^2}{2\sigma^2}})(\sqrt{\frac{1}{2\pi\sigma^4}} e^{-\frac{y^2}{2\sigma^2}})$$

   $$= (-\frac{1}{\sigma^2\sqrt{2\pi}} x e^{-\frac{x^2}{2\sigma^2}})(\frac{1}{\sigma^2\sqrt{2\pi}} e^{-\frac{y^2}{2\sigma^2}})$$

   Como podemos ver, he conseguido tener la derivada como un producto en el que por una parte sólo de pende de x y por otra sólo de y. Por lo que con esto demostramos que pueden ser separables.

   Por otra parte, para el cálculo pero en este caso de la derivada respecto y, tenemos que seguir exactamente los mismos pasos, y como tenemos que las dos funciones son simétricas, tenemos que:

   $$\frac{d}{dy} G(x,y;\sigma) = -\frac{1}{2\pi\sigma^4} y e^{-\frac{x^2+y^2}{2\sigma^2}}$$

   $$= (\frac{1}{\sigma^2\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}})(-\frac{1}{\sigma^2\sqrt{2\pi}} y e^{-\frac{y^2}{2\sigma^2}})$$

   Que como vemos, tenemos lo mismo que para el caso de la derivada en x, por lo que la conclusión es la misma.

   Por último, esta separación lo que nos permite es conseguir resaltar los bordes que se encuentran en direccion vertical para el caso de la derivada en x, mientras que conseguiremos resaltar los bordes horizontales para el caso de la derivada en y.

8. **Verificar matemáticamente que la Laplaciana de la Gaussiana se puede implementar a partir de núcleos de convolución separables por filas y columnas. Interpretar**

el papel de dichos núcleos en el proceso de convolución.

Sabemos que la fórmula de la Laplaciana de la Gaussiana es la siguiente:

$$\nabla^2 G(x,y;\sigma) = \frac{\partial^2 G(x,y;\sigma)}{\partial x^2} + \frac{\partial^2 G(x,y;\sigma)}{\partial y^2}$$

Lo que nos dice que necesitamos calcular las segundas derivadas de $G(x,y;\sigma)$ para posteriormente sumarlas.

$$\frac{\partial^2 G(x,y;\sigma)}{\partial x^2} = \frac{x^2-\sigma^2}{2\pi\sigma^6} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

$$\frac{\partial^2 G(x,y;\sigma)}{\partial y^2} = \frac{y^2-\sigma^2}{2\pi\sigma^6} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Por lo que finalmente tenemos:

$$\begin{aligned}
\nabla^2 G(x,y;\sigma) &= \left(\frac{x^2-\sigma^2}{2\pi\sigma^6} e^{-\frac{x^2+y^2}{2\sigma^2}}\right) + \left(\frac{y^2-\sigma^2}{2\pi\sigma^6} e^{-\frac{x^2+y^2}{2\sigma^2}}\right) \\
&= \frac{e^{-\frac{x^2+y^2}{2\sigma^2}}(x^2-\sigma^2+y^2-\sigma^2)}{2\pi\sigma^6} \\
&= \frac{e^{-\frac{x^2}{2\sigma^2}} e^{-\frac{y^2}{2\sigma^2}}(x^2-\sigma^2+y^2-\sigma^2)}{2\pi\sigma^6} \\
&= \left(\frac{e^{-\frac{x^2}{2\sigma^2}}(x^2-\sigma^2)}{2\pi\sigma^6} e^{-\frac{y^2}{2\sigma^2}}\right) + \left(\frac{e^{-\frac{y^2}{2\sigma^2}}(y^2-\sigma^2)}{2\pi\sigma^6} e^{-\frac{x^2}{2\sigma^2}}\right)
\end{aligned}$$

Esta expresión la podríamos expresar de la siguiente forma:

$$(L_1(x) \cdot L_2(y)) + (L_3(y) \cdot L_4(x))$$

Donde:

- $L_1(x) = \frac{e^{-\frac{x^2}{2\sigma^2}}(x^2-\sigma^2)}{2\pi\sigma^6}$
- $L_2(y) = e^{-\frac{y^2}{2\sigma^2}}$
- $L_3(y) = \frac{e^{-\frac{y^2}{2\sigma^2}}(y^2-\sigma^2)}{2\pi\sigma^6}$
- $L_4(x) = e^{-\frac{x^2}{2\sigma^2}}$

Como vemos, $L_1(x)$ y $L_3(y)$ podrían ser implementadas como máscaras 1D por columnas, mientras que $L_2(y)$ y $L_4(x)$ podrían ser implementadas como máscaras 1D por filas, o viceversa.

9. **¿Cuáles son las operaciones básicas en la reducción del tamaño de una imagen? Justificar el papel de cada una de ellas.**

La reduccion de tamaño de una imagen tiene dos operaciones básicas, **el alisado** y la propia reducción de la imagen, llamado **submuestreo**.

La aplicación del submuestreo es reducir el tamaño de la imagen a $f/2$ y $c/2$ donde f son las filas y c son las columnas. Lo que hace para ello es eliminar los pixeles que se encuentran en las filas y columnas pares y quedarse con las impares. Esta operación realiza correctamente el submuestreo pero tiene el gran problema de que perdemos la información de los pixeles que hemos eliminado y añade un contraste entre los nuevos pixeles más grande, es decir, aumenta las frecuencias altas. Es por ello, por lo que antes de hacer un submuestreo, se hace un alisamiento.

El alisamiento nos permite recalcular el valor de los pixeles respecto a sus vecinos, eliminando las altas frecuencias y así podemos tener en los pixeles que se van a quedar tras el submuestreo algo de información de los pixeles que se van a eliminar, y al haber eliminado las altas frecuencias, tener una mejor visualización de la imagen.

10. **¿Qué información de la imagen original se conserva cuando vamos subiendo niveles en una pirámide Gaussiana? Justificar la respuesta.**

    Se conserva únicamente la información que ya tenemos, es decir, **las bajas frecuencias**, si aumentamos una imagen de 30x30 a 60x60, la información que tenemos en la imagen de 60x60 son los pixeles que teniamos en la imagen de 30x30 ya que los otros pixeles que se crean nuevos, se interpolan utilizando distintos métodos para calcularlos.

    Esto es porque la pirámide Gaussiana, al bajar los niveles, elimina toda las informacion que va retirando *(altas frecuencias)* al disminuir el tamaño y no hay forma ninguna de recuperarla, por lo que si luego queremos recomponer la imagen a su tamaño original, no obtendremos el resultado esperado.

11. **¿Qué información podemos extraer de la pirámide Gaussiana y la pirámide Laplaciana de una imagen? ¿Qué nos aporta cada una de ellas? Justificar la respuesta.**

    La información que podemos extraer de cada uno de los métodos esta basado directamente en su funcionamiento.

    Por lo que de la **piramide Gaussiana**, la información que vamos obteniendo de la imagen en cada paso son las **frecuencias bajas**, que las vamos almacenando como he comentado en ejercicios anteriores, lo que nos permite una mayor calidad en la reduccion de la imagen, ya que si sólo reduciesemos el tamaño sin más, obtendriamos una imagen con muchas frecuencias altas y poco visualizable.

    Por otra parte, la información que obtenemos de la **pirámide Laplaciana** es justo lo contrario, lo que vamos almacenando en cada paso es las **frecuencias altas** que eliminamos, ya que lo que nos interesa es poder hacer una perfecta reconstruccion de la imagen cuando vayamos subiendo los niveles. Para ello, en cada uno de los pasos cuando estamos descendiendo, se aplica exactamente el mismo paso que en la piramide Gaussiana, pero en este caso una vez reducida la imagen, se calcula la diferencia entre la imagen original y la aproximación del aumento de la imagen disminuida, lo que hace que lo que almacenemos sean las frecuencias altas que hemos eliminado. Con estas diferencias, mas la última imagen de la pirámide, podemos reconstruir perfectamente la imagen original.

12. **¿Podemos garantizar una perfecta reconstrucción de una imagen a partir de su pirámide Laplaciana? Dar argumentos y discutir las opciones que considere necesario.**

    Como he comentado en el ejercicio anterior, la respuesta es **sí** *(asumiendo que tenemos la última imagen submuestreada)*. El funcionamiento de la pirámide Laplaciana es justamente ese, nos permite que cuando vayamos a subir los niveles de una pirámide Laplaciana, la reconstrucción de la imagen sea exactamente la correcta. Esto se debe esencialmente al funcionamiento de la misma que se basa en guardar en cada paso las frecuencias altas que hemos eliminado en cada uno de los pasos al reducir. Por lo que cuando subimos los niveles, solo tenemos que sumarle a la imagen aumentada sus correspondientes frecuencias altas que fueron eliminadas al reducir y ya tendremos la imagen original correctamente representada.

13. **En OpenCV solo se pueden calcular máscaras Sobel de hasta dimensión 7x7 ¿Por qué? De una explicación razonable a este hecho y diga cómo influye en el cálculo con máscaras de mayor tamaño. Justificar la respuesta**

    La razón principal para poner un máximo de tamaño de la máscara en Sobel es **la eficiencia**. Sobel es una función de OpenCV que directamente le aplica el filtro a la imagen y devuelve el resultado de la aplicación y no la máscara. La forma que tiene de trabajar es con la máscara completa y no separable, esto hace que la eficiencia del cálculo sea más elevada y por ello han decidido ponerle un máximo de tamaño de máscara.

    La eficiencia de una máscara que no se aplica separable es $O(n^2m^2)$ siendo n el tamaño de la imagen y m el tamaño de la máscara, mientras que para un filtro que se aplica de forma separable tenemos $O(n^2m)$

14. **Cuales son las contribuciones más relevantes del algoritmo de Canny al cálculo de los contornos sobre una imagen?. ¿Existe alguna conexión entre las máscaras de Sobel y el algoritmo de Canny? Justificar la respuesta**

    El algoritmo Canny tiene unas contribuciones mas relevantes frente a otros algoritmos que se preocupan de hacer lo mismo que Canny, ya que el algoritmo Canny no sólo calcula la intensidad del vector grandiente en cada uno de los píxeles, sino que también se encarga de:

    - **Supresión de no máximos:**

      Se encarga de que por cada píxel de la imagen gradiente se establece una región para comparar la intensidad de los píxeles vecinos que tienen la misma dirección gradiente y si es mayor, se conserva el píxel del borde y en caso contrario lo elimina.

    - **Histéresis:**

      En este proceso, una vez ya se a completado la supresión de no máximos, se establecen dos umbrales, un umbral máximo y otro mínimo, se vuelve a analizar la imagen, y se comprueba para cada pixel, si se encuentra por encima, por debajo o entre los umbrales seleccionados anteriormente. Si el pixel está por debajo del umbral bajo, este se elimina, en caso de estar por encima del máximo, este pixel pertenece al límite, y si se encuentra entre los dos umbrales, el algoritmo dice que pertenece a la unión entre dos límites en la dirección del gradiente. Entonces, si se detectan dos límites puede detectarse también la forma de unirlos.

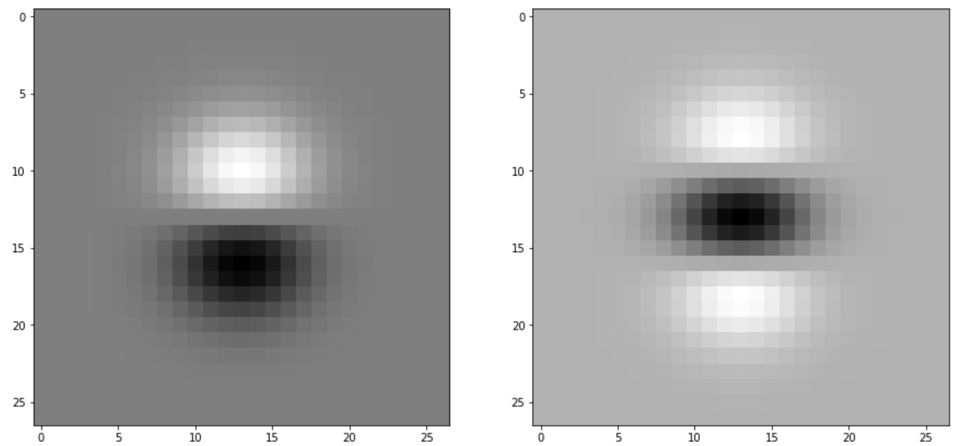
    Respecto a la conexión que hay entre Canny y Sobel, Canny utiliza Sobel para el cálculo del gradiente, y a partir de ahí trabaja con ese resultado. Incluso en OpenCV, el algoritmo Canny está limitado en el tamaño de la máscara por el algoritmo Sobel, ya que el máximo tamaño de máscara para el algoritmo Sobel es el máximo tamaño de máscara que se le puede asignar al algoritmo Canny.

15. **Suponga que le piden implementar un algoritmo para el cálculo de la derivada de primer y segundo orden sobre una imagen usando un filtro gaussiano cualesquiera. Enumere y explique los pasos necesarios para llevarlo a cabo.**

Para este proceso, lo haría de la siguiente manera:

- Se calcula la Gaussiana.
- Se crea una matriz para aplicar la convolución *[[-1,0,1]]*
- Para la primera derivada se aplica la convolución sobre la Gaussiana aplicandole la convolución anterior.
- Para la segunda derivada se aplica la misma convolución sobre el resultado del paso anterior.

Con este proceso conseguimos una aproximación hacia la primera y segunda derivada de la Gaussiana, obteniendo:



Referencias

[1] Freeman, W. T. and Adelson, E. H. (1991). The design and use of steerable filters. IEEE Transactions on Pattern Analysis and Machine Intelligence, 13(9):891–906. Accedido el 27 de octubre de 2018.

[2] Perona, P. (1995). Deformable kernels for early vision. IEEE Transactions on Pattern Analysis and Machine Intelligence, 17(5):488–499. Accedido el 27 de octubre de 2018.

[3] Richard Szeliski (2010). Computer Vision: Algorithms and Applications Accedido el 27 de octubre de 2018.