

Visión por computador. Cuestionario 1.

José Carlos Martínez Velázquez

21 de octubre de 2016

1.- ¿Cuáles son los objetivos principales de las técnicas de visión por computador? Poner algún ejemplo si lo necesita

El objetivo principal que persiguen las técnicas de visión por computador es extraer significado de los píxeles de una imagen. Esto debe hacerlo un computador, por lo que se trata de dotar a los computadores de habilidades análogas a la visión humana, esto es, hacer que los computadores sean capaces de tomar decisiones en función del análisis y tratamiento de imágenes.

Ejemplos claros de lo que se ha conseguido con visión por computador son la detección de caras o sonrisas en el software de cámaras de fotos/video, reconocimiento de objetos/personas en robots, etc. Una aplicación que podría ser una realidad cotidiana en poco tiempo gracias a las técnicas de visión por computador, entre otras, son los vehículos autotripulados. Todo esto se hace, lógicamente, manipulando y analizando imágenes captadas mediante una cámara.

2.- ¿Una máscara de convolución para imágenes debe ser siempre una matriz 2D? ¿Tiene sentido considerar máscaras definidas a partir de matrices de varios canales como p.e. el tipo de OpenCV CV_8UC3? Discutir y justificar la respuesta.

Una máscara de convolución para imágenes no siempre debe ser una matriz 2D, aunque hay casos en los que no hay más remedio, pues no se pueden descomponer en vectores 1D o no se conoce cómo. Una máscara de convolución Gaussiana para alisado no tiene por qué aplicarse como matriz. Como sabemos, las funciones Gaussianas son simétricas respecto de su media. Sabiendo esto, un núcleo de convolución Gaussiano siempre puede ser descompuesta en dos vectores que convolucionan en una dirección cada uno: uno en horizontal (eje X o convolución por filas) y otro en vertical (eje Y o convolución por columnas).

Para la segunda cuestión, imaginemos una máscara de varios canales con el fin de modificar cada canal de la imagen por separado (forzosamente la imagen debería tener más de un canal, en caso contrario no nos plantearíamos esta cuestión). En la siguiente figura, como ejemplo, una máscara y una imagen de tres canales (red,green,blue).

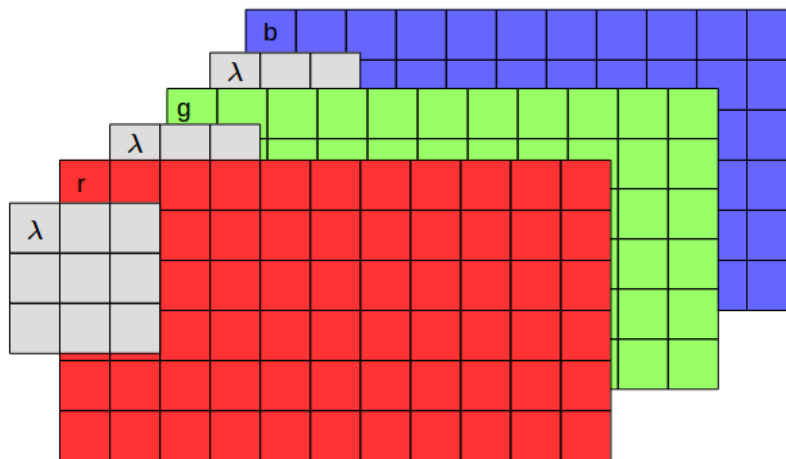


Figura 1. Representación gráfica de una máscara de tres canales.

Si las capas de la máscara no fuesen iguales, los píxeles resultantes de aplicar los diferentes canales de la máscara a cada canal de la imagen, sufrirían operaciones distintas. Al superponer las capas resultantes, obtendríamos una deformación de la imagen original. ¿Qué podría salir si un canal de una máscara 3x3x3 aplica convolución, otro un filtro de desplazamiento y otro un filtro de Sobel?. En definitiva, **no tiene sentido** tener una máscara de varios canales, pues en cada canal de la imagen deberían realizarse los mismos cálculos. Entonces, si quisiéramos multiplicar un píxel de una imagen multicapa (en el ejemplo de la figura, un vector de valores (r,g,b)) por un pixel de la máscara, lo que tendría sentido es que la máscara tuviera un solo canal, así, multiplicar un pixel de la imagen por un pixel de la máscara, sería equivalente a multiplicar un vector por un escalar $\lambda \cdot (r, g, b)$, donde λ es el valor del pixel de la máscara correspondiente al píxel de la imagen y todos los canales de la imagen se verían modificados por igual.

3.- Expresar y justificar las diferencias y semejanzas entre correlación y convolución. Justificar la respuesta.

Correlación y convolución son la misma operación en el sentido en que dada una máscara 2D y una imagen, imaginamos la máscara como una ventana sobre la imagen que se va “paseando” por la imagen haciendo ciertos cálculos. Cuando la máscara es simétrica, no hay diferencia entre correlación y convolución, pero cuando no lo es existen grandes diferencias.

La razón de las diferencias entre correlación y convolución residen en que la operación de convolución es asociativa mientras que la de correlación, en general, no lo es. Vamos a ver las expresiones de ambas operaciones y demostraremos que la convolución es asociativa mientras que la operación de correlación no lo es.

- Operación de correlación:

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

Esta operación se denota como $G = H \otimes F$.

- Operación de convolución:

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

Esta operación se denota como $G = H \star F$. Hay que darse cuenta que $F[i - u, j - v]$ indica que la operación se hace volteando la máscara en x y en y.

Vamos a suponer para ambas demostraciones que nuestros datos son tres señales 1D: F,H,I.

Demostración de que la correlación, en general, no es asociativa.

Para demostrar esto basta con encontrar un contraejemplo. Supongamos $F = (2, 3)$, $H = (-1, 0)$, $I = (-1, 2)$. Vamos a aplicar las operaciones en órdenes distintos:

- 1. $F \otimes (H \otimes I)$

Realicemos las operaciones. Sin pérdida de generalidad, rellenaremos las señales donde sea necesario con ceros a izquierda y derecha.

Primero calculamos $H \otimes I$:

$$(0, -1, 2, 0) \star (-1, 0) = (0, 1, -2)$$

Ahora realizamos $F \otimes (H \otimes I)$:

$$(0, 0, 1, -2, 0) \star (2, 3) = (0, 3, -4, -4)$$

- 2. $(F \otimes H) \otimes I$

Primero calculamos $(F \otimes H)$:

$$(0, -1, 0, 0) \star (2, 3) = (-3, -2, 0)$$

A continuación calculamos $(F \otimes H) \otimes I$:

$$(0, -3, -2, 0, 0) \star (-1, 2) = (-6, -1, 2, 0)$$

Como vemos, los resultados obtenidos en 1 y 2 son diferentes y queda demostrado por contraejemplo que la correlación, en general, no es asociativa.

Demostración de que la convolución es asociativa.

Vamos a hacer lo mismo que en la demostración anterior y veremos que en 1 y 2 sale lo mismo. Recordemos que en convolución hay que voltear la máscara en x e y (como en este ejemplo trabajamos en 1D, volteamos en x, la única dimensión) Así una máscara (a, b, c, \dots, z) para aplicar convolución quedaría como (z, \dots, c, b, a) .

- 1. $F \star (H \star I)$

Primero calculamos $H \star I$, volteando H :

$$(0, -1, 2, 0) \star (0, -1) = (1, -2, 0)$$

Ahora realizamos $F \star (H \star I)$, volteando F :

$$(0, 1, -2, 0, 0) \star (3, 2) = (2, -1, -6, 0)$$

- 2. $(F \star H) \star I$

Primero calculamos $(F \star H)$, volteando F :

$$(0, -1, 0, 0) \star (3, 2) = (-2, -3, 0)$$

A continuación calculamos $(F \star H) \star I$:

$$(0, -2, -3, 0, 0) \star (2, -1) = (2, -1, -6, 0)$$

Como vemos, 1 y 2 devuelven el mismo resultado, pero esto no es suficiente para demostrar que la convolución es asociativa en todos los casos. Vamos a demostrarlo analíticamente.

Para generalizar, partiremos de la expresión de convolución 2D:

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

Si H es una **máscara de convolución** de tamaño $m \times n$ puede descomponerse en dos máscaras 1D: $H1$ de tamaño $m \times 1$ y $H2$ de tamaño $1 \times n$, entonces la expresión queda:

$$G[i, j] = \sum_{u=-k}^k \sum_{v=-k}^k H1[u, 1] H2[1, v] F[i - u, j - v]$$

Como $H1$ no depende de la segunda sumatoria, puede sacarse. Del mismo modo, $H2$ no depende de la primera y podría sacarse, entonces tendríamos que:

$$\begin{aligned} G[i, j] &= \sum_{u=-k}^k \sum_{v=-k}^k H1[u, 1] H2[1, v] F[i - u, j - v] \\ &= \sum_{u=-k}^k H1[u, 1] \left(\sum_{v=-k}^k H2[1, v] F[i - u, j - v] \right) \\ &= \sum_{v=-k}^k H2[1, v] \left(\sum_{u=-k}^k H1[u, 1] F[i - u, j - v] \right) \end{aligned}$$

Con esta expresión vemos cómo da igual aplicar una máscara de convolución primero que otra, el resultado sería el mismo.

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

Los filtros de **convolución** **sí** definen funciones lineales, pues se basan en una media ponderada, que es una función lineal, sin embargo los de **mediana** **no**, pues la mediana no es una función lineal. Vamos a ver en primer lugar qué es una función lineal. Una función es lineal si y sólo si cumple las siguientes características:

- La suma de los valores de la función en varios puntos es igual a la función de la suma de dichos valores: $f(x) + f(y) = f(x + y)$
- El valor de la función en un punto multiplicado por una constante es igual a la constante multiplicada por el valor de la función en dicho punto: $f(kx) = kf(x)$

Vamos a demostrar que la función media cumple las dos propiedades. Imaginemos dos conjuntos de valores $A = \{a_1, a_2, \dots, a_n\}$ y $B = \{b_1, b_2, \dots, b_n\}$. Vamos a comprobar la primera propiedad:

Si calculamos la media de A , tenemos:

$$media(A) = \frac{1}{n} \sum_{i=1}^n a_i$$

Del mismo modo, la media de B es:

$$media(B) = \frac{1}{n} \sum_{i=1}^n b_i$$

Si sumamos ambas, tenemos:

$$\begin{aligned} media(A) + media(B) &= \frac{1}{n} \sum_{i=1}^n a_i + \frac{1}{n} \sum_{i=1}^n b_i \\ &= \frac{1}{n} \sum_{i=1}^n a_i + b_i \\ &= media(A + B) \end{aligned}$$

Ahora, vamos a comprobar la segunda propiedad:

$$\begin{aligned} k \cdot media(A) &= k \cdot \frac{1}{n} \sum_{i=1}^n a_i \\ &= \frac{1}{n} \sum_{i=1}^n k \cdot a_i \\ &= media(k \cdot A) \end{aligned}$$

Vemos cómo la operación media es una función lineal, entonces, **un filtro de convolución es lineal** porque hace una media (ponderada con distintos pesos). Dicho de otro modo, reemplaza cada píxel por una combinación lineal de su entorno.

Por contra, un filtro de mediana no define una operación lineal, pues no cumple las propiedades del principio. Para demostrarlo, vamos a dar valores concretos a los conjuntos anteriores, pues para demostrar que la mediana no es una operación lineal, basta con mostrar un contraejemplo.

Supongamos $A = \{2, 5, 1, 4\}$ y $B = \{5, 1, 3, 7\}$, entonces $mediana(A) = 3$ y $mediana(B) = 4$. Vamos a ver si se cumplen las propiedades que la convierten en función lineal. $mediana(A) + mediana(B) = 3 + 4 = 7$. Ahora consideramos el conjunto $A + B = \{7, 6, 4, 11\}$, cuya mediana es $\frac{6+7}{2} = 6.5$. Como vemos $mediana(A) + mediana(B) = 7 \neq 6.5 = mediana(A + B)$. Viendo que la función mediana incumple la primera propiedad para un ejemplo concreto, no es necesario seguir. **La mediana no es una operación lineal y por tanto tampoco pueden serlo los filtros que implementen esta operación.**

5.- ¿La aplicación de un filtro de alisamiento debe ser una operación local o global sobre la imagen? Justificar la respuesta

Siempre local. Si fuese una operación global sólo cambiaría el píxel central, en caso de haberlo. Aún cuando la máscara tiene el mismo tamaño que la imagen, es una operación local, pues la máscara se centra en un sólo píxel y sólo para el píxel central de la imagen original (si lo hubiera), se consideraría la imagen completa y podríamos hablar de una operación global, esto es, que el píxel central estaría influenciado por la imagen completa.

6.- Para implementar una función que calcule la imagen gradiente de una imagen dada pueden 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 que estrategia es la más adecuada, si alguna lo es, tanto en el plano teórico como de implementación. Justificar la decisión.

Me quedaría con la opción a).

El objetivo del gradiente es indicar un cambio de intensidad o color en una imagen. Como es un vector, podemos hablar de su módulo (intensidad de la gradiente) y su dirección (hacia dónde se produce el cambio). Si aplicamos a una imagen un filtro de alisado, eliminamos altas frecuencias (detalles), o similarmente estamos suavizando los cambios bruscos en las tonalidades que permitirán, una vez calculadas las derivadas, calcular las propiedades del gradiente en un punto. Teóricamente, si calculamos las derivadas y posteriormente alisamos, estaríamos contaminando el valor del gradiente en un punto con el valor del gradiente en su entorno (promediamos el valor gradiente). Si, por el contrario, alisamos y posteriormente calculamos las derivadas, estaremos extrayendo los cambios que realmente son prominentes, por lo que me quedaría con esta segunda opción.

Dado que este proceso puede ser implementado, vamos a realizar las derivadas en X e Y de una imagen antes y después de alisar con un valor de $\sigma = 3$.

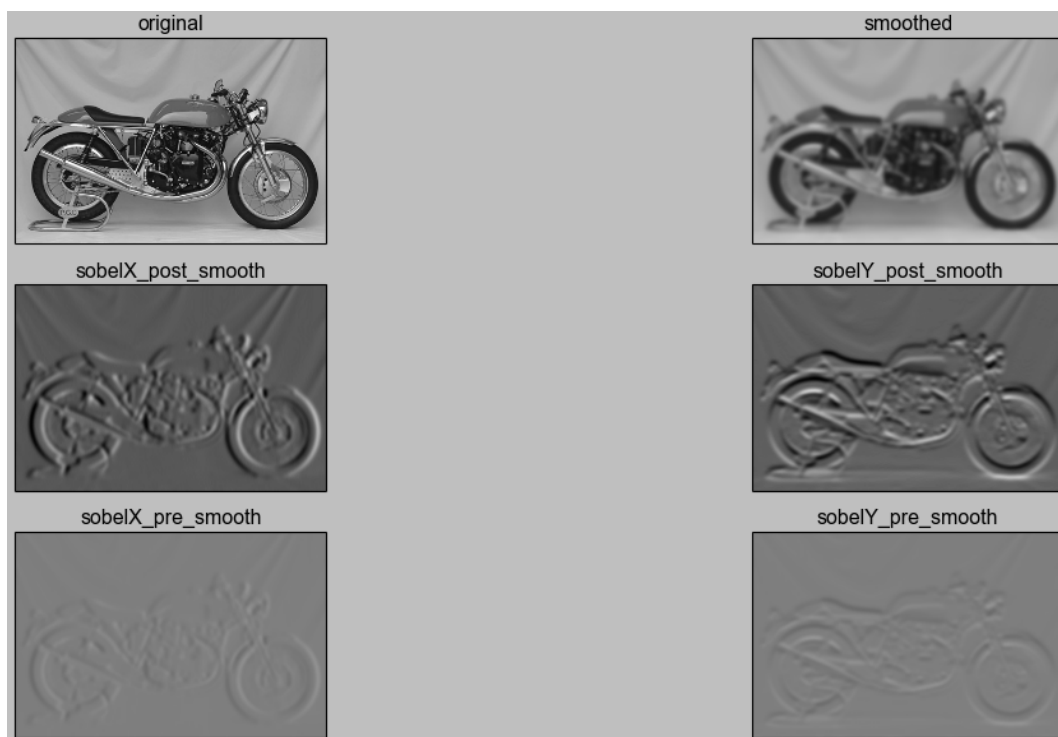


Figura 2. Derivadas en X e Y antes (pre) y después (post) del alisado.

Fijémonos cómo al realizar las derivadas después de alisar, las imágenes resultantes destacan los saltos importantes en la imagen. Si alisamos después de calcular las derivadas, estamos reduciendo las diferencias

entre los valores del gradiente en toda la imagen, justo lo contrario de lo que buscamos, tal y como había supuesto.

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.

Una función se puede representar como un núcleo de convolución 1D si y sólo si la función que representa el filtro depende de una única variable. La función Gaussiana 2D depende de dos variables:

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

Donde σ^2 es la varianza (σ es la desviación típica que decide la amplitud de la campana). Es una constante en lo que a derivadas se refiere.

Vamos a separar ambas dimensiones, de forma que $G(x,y)$ se pueda descomponer en dos funciones 1D independientes entre sí.

$$\begin{aligned} G(x, y) &= \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \\ &= \frac{1}{2\pi\sigma^2} e^{-\left(\frac{x^2}{2\sigma^2} + \frac{y^2}{2\sigma^2}\right)} \\ &= \frac{1}{2\pi\sigma^2} e^{-\frac{x^2}{2\sigma^2}} e^{-\frac{y^2}{2\sigma^2}} \\ &= \sqrt{\frac{1}{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}} e^{-\frac{y^2}{2\sigma^2}} \\ &= \sqrt{\frac{1}{2\pi\sigma^2}} \sqrt{\frac{1}{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}} e^{-\frac{y^2}{2\sigma^2}} \\ &= \left(\sqrt{\frac{1}{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}} \right) \left(\sqrt{\frac{1}{2\pi\sigma^2}} e^{-\frac{y^2}{2\sigma^2}} \right) \\ &= \left(\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}} \right) \left(\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{y^2}{2\sigma^2}} \right) \\ &= G_x(x) \cdot G_y(y) \end{aligned}$$

Ahora G_x y G_y son dos funciones completamente independientes, cuyo producto da lugar a G . Parece razonable que, si derivamos G_x y G_y por separado, las derivadas de ambas funciones sean también independientes. Vamos a calcularlas:

$$\begin{aligned} G'_x(x) &= \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}} \left(\frac{-2x2\sigma^2}{4\sigma^4} \right) \\ &= \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}} \left(\frac{-4x\cancel{\sigma^2}}{4\cancel{\sigma^2}\sigma^2} \right) \\ &= \frac{e^{-\frac{x^2}{2\sigma^2}}(-x)}{\sigma\sqrt{2\pi}\sigma^2} = -\frac{xe^{-\frac{x^2}{2\sigma^2}}}{\sigma^3\sqrt{2\pi}} \end{aligned}$$

Del mismo modo:

$$G'_y(y) = -\frac{ye^{-\frac{y^2}{2\sigma^2}}}{\sigma^3\sqrt{2\pi}}$$

Ahora que ya sabemos cuáles son las derivadas de $G_x(x)$ y $G_y(y)$, vamos a calcular las derivadas parciales de la función $G(x, y)$. Recordemos su forma descompuesta:

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

Entonces:

$$\begin{aligned} \frac{\partial G(x, y)}{\partial x} &= \left(-\frac{xe^{-\frac{x^2}{2\sigma^2}}}{\sigma^3\sqrt{2\pi}} \right) \left(\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{y^2}{2\sigma^2}} \right) + \left(\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}} \right) \cancel{\frac{\partial G_y(y)}{\partial x}} \xrightarrow{0} \\ &= -\frac{xe^{-\frac{x^2}{2\sigma^2}} e^{-\frac{y^2}{2\sigma^2}}}{2\pi\sigma^4} \end{aligned}$$

$$= \overbrace{-\frac{xe^{-\frac{x^2}{2\sigma^2}}}{2\pi\sigma^4}}^{\text{Núcleo 1D: depende de } x} \underbrace{e^{-\frac{y^2}{2\sigma^2}}}_{\text{Núcleo 1D: depende de } y}$$

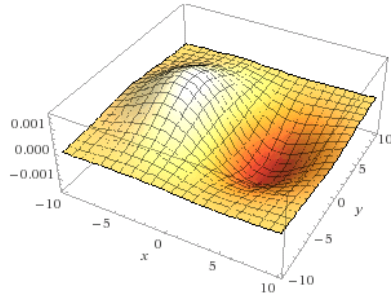
Vemos como se descompone la derivada de $G(x, y)$ con respecto a x en dos funciones independientes que podrían ser implementados como núcleos de convolución 1D independientes. De la misma manera, la derivada con respecto a y de $G(x, y)$ puede descomponerse como:

$$\frac{\partial G(x, y)}{\partial y} = \overbrace{-\frac{ye^{-\frac{y^2}{2\sigma^2}}}{2\pi\sigma^4}}^{\text{Núcleo 1D: depende de } y} \underbrace{e^{-\frac{x^2}{2\sigma^2}}}_{\text{Núcleo 1D: depende de } x}$$

Fijémonos en que son la misma excepto cambiando x por y , lo cual tiene sentido, pues calcular la derivada con respecto a x o y , dada la simetría de la Gaussiana, no es más que cambiar el sentido de los ejes. Con $\sigma = 4$, este es el aspecto que toman las primeras derivadas:

plot	$-\frac{x \exp\left(-\frac{x^2}{2 \times 4^2}\right)}{2 \pi \times 4^4} \exp\left(-\frac{y^2}{2 \times 4^2}\right)$	$x = -10 \text{ to } 10$ $y = -10 \text{ to } 10$
plot	$-\frac{y \exp\left(-\frac{y^2}{2 \times 4^2}\right)}{2 \pi \times 4^4} \exp\left(-\frac{x^2}{2 \times 4^2}\right)$	$x = -10 \text{ to } 10$ $y = -10 \text{ to } 10$

3D plot:



3D plot:

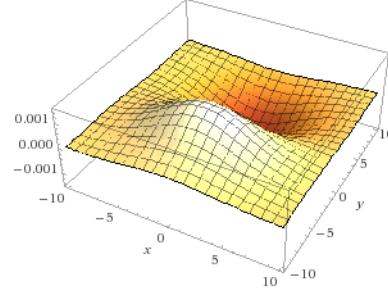


Figura 3. Derivadas con respecto a x e y de G.

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.

En el ejercicio anterior, veíamos que las primeras derivadas de la Gaussiana eran las siguientes:

$$\frac{\partial G(x, y)}{\partial x} = -\frac{x e^{-\frac{x^2}{2\sigma^2}}}{2\pi\sigma^4} e^{-\frac{y^2}{2\sigma^2}} \quad \frac{\partial G(x, y)}{\partial y} = -\frac{y e^{-\frac{y^2}{2\sigma^2}}}{2\pi\sigma^4} e^{-\frac{x^2}{2\sigma^2}}$$

La función Laplaciana de la Gaussiana se define como:

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

Es decir, la suma de las segundas derivadas de G . El plan es claro, para obtener la función Laplaciana de la Gaussiana, vamos a volver a derivar cada una de las primeras derivadas de la Gaussiana y las sumaremos. Estas segundas derivadas son costosas de calcular así que sólo pondré el resultado, pues el proceso ocuparía demasiado espacio:

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

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

Entonces, desarrollamos:

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

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

Cada uno de los sumandos anteriores, para $\sigma = 4$, tiene el siguiente aspecto:

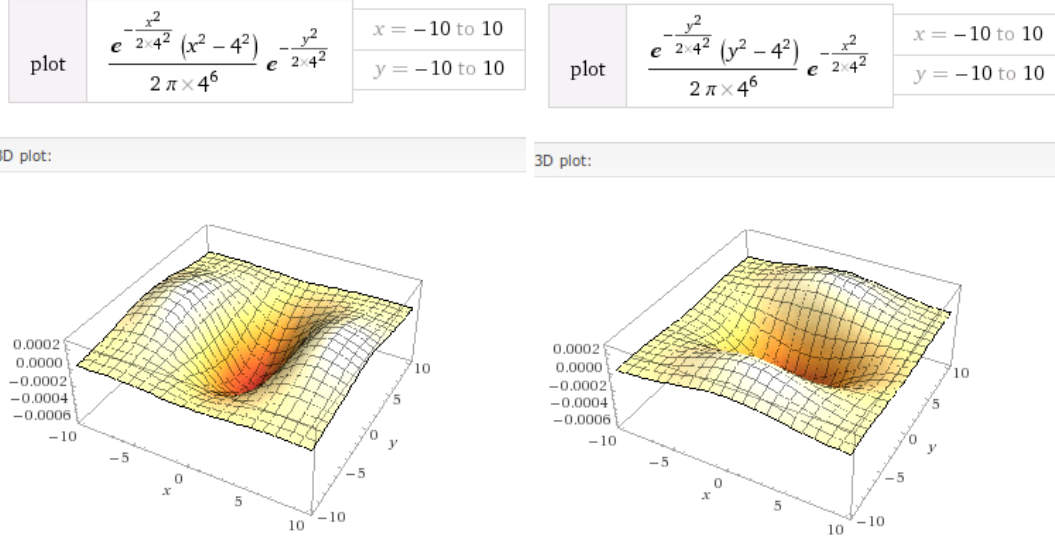


Figura 4. Sumandos de la función Laplaciana de la Gaussiana.

La expresión anterior podríamos expresarla así:

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

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.

La función resultante, para $\sigma = 4$ tiene el siguiente aspecto:

$$\text{plot} \quad \frac{e^{-\frac{x^2}{2 \cdot 4^2}} (x^2 - 4^2)}{2 \pi \times 4^6} e^{-\frac{y^2}{2 \cdot 4^2}} + \frac{e^{-\frac{y^2}{2 \cdot 4^2}} (y^2 - 4^2)}{2 \pi \times 4^6} e^{-\frac{x^2}{2 \cdot 4^2}}$$

3D plot:

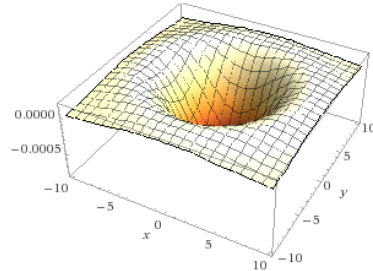


Figura 5. Función Laplaciana de la Gaussiana.

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

Las operaciones básicas son el suavizado y la eliminación de filas y columnas (submuestreo). El papel que juega el submuestreo es claro, pues si queremos reducir el tamaño de la imagen, hay que hacerlo eliminando información y la información de que disponemos son píxeles dispuestos en filas y columnas. El problema es que cuando hacemos esto, nos quedarían “saltos” en la imagen producidos por la pérdida de relación de los píxeles que se quedan con los que componían su entorno, es decir, quedan muchas altas frecuencias de la imagen original. Corregir este problema es el papel jugado por el suavizado.

El procedimiento que se sigue es en primer lugar suavizar la imagen con el fin de eliminar algunas de esas altas frecuencias que van a sobrar y una vez hecho, submuestrear. Vamos a ver cómo quedaría una imagen submuestreada sin presuavizado junto con una imagen presuavizada y submuestreada.



Figura 6. Pasos de la reducción: original, sólo submuestreo y presuavizado+submuestreo.

La primera imagen es la original. En la segunda, se han eliminado la mitad de las filas y la mitad de las columnas, lo que conlleva a que se observen saltos en la continuidad de la imagen o “dientes de sierra”. Para calmar esa sensación se realiza el suavizado, que aplaca estos saltos, como se puede observar en la tercera imagen, volviendo a relacionar cada píxel con su entorno mediante dicha operación.

El submuestreo es la principal causa de *aliasing*, que ocurre cuando el tamaño de la imagen no es capaz de capturar todo el detalle que hay en ella. Existe todo un conjunto de sistemas dedicados a solucionar este problema denominados, como no, *antialiasing*.

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

Sólo podemos conservar la única información de la que disponemos, es decir, **las bajas frecuencias**. Al subir niveles en una pirámide Gaussiana, el resto de información es interpolada de alguna manera (inventada) o calculada, siendo posible esto último sólo si hemos usado una pirámide Laplaciana, la cual va guardando las altas frecuencias en cada reducción y recuperándolas en cada ampliación. De este modo, las altas frecuencias que se pierden al bajar niveles, pueden recuperarse al irlos subiendo, así que la información nunca se pierde.

11.- ¿Cuál es la diferencia entre una pirámide Gaussiana y una Pirámide Laplaciana? ¿Qué nos aporta cada una de ellas? Justificar la respuesta. (Mirar en el artículo de Burt-Adelson)

La diferencia es básica: la pirámide Laplaciana nos permite subir niveles sin pérdida de información, es decir, se puede recuperar la información verdadera de la imagen que se pierde al reducir. En la pirámide Gaussiana, la información se pierde de verdad en la reducción y en caso de querer ampliar, la información debería obtenerse interpolando y nunca podría llegar a ser real, tal y como se ha comentado en la cuestión anterior.

La información es recuperable en el caso de la pirámide Laplaciana porque en cada nivel se guardan las altas frecuencias, esto es, la diferencia entre la imagen que corresponde a dicho nivel y la misma imagen suavizada.

12.- Cual es la aportación del filtro de Canny al cálculo de fronteras frente a filtros como Sobel o Robert. Justificar detalladamente la respuesta.

Los operadores de Sobel o Robert se limitan a calcular la intensidad del vector gradiente en cada píxel, mientras que el filtro de Canny, además de realzar y suavizar con la primera derivada de la Gaussiana, implementa varias etapas para usar este dato:

- Supresión de no máximos: comprueba si el valor del píxel es máximo local en la dirección del gradiente del píxel. Si no lo fuese, queda con el valor 0.
- Histéresis: este es el proceso más interesante. Una vez completado el proceso de supresión de no máximos, se establecen dos umbrales, uno mayor u_M y uno menor u_m . Se analiza la imagen de nuevo. Si el valor del píxel analizado supera u_M , dicho píxel pertenece a un límite (edge), si no supera a u_m , el píxel no pertenece al límite. Si el valor de dicho píxel queda entre ambos 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.

Los filtros de Sobel o Robert quedaban cortos en cuanto a detección de límites, lo que el algoritmo de Canny trata de solucionar con las dos fases nuevas que aporta. Eso sí, presenta dificultades. Según el valor de σ elegido en el proceso, los límites detectados no son los mismos. Para valores de σ altos, se detectan los límites muy evidentes, los fuertes. Para valores de *sigma* bajos se detectan además, límites débiles que podrían considerarse ruido, por lo que la elección del valor de σ debe hacerse con cuidado.

13.- Buscar e identificar una aplicación real en la que el filtro de Canny garantice unas fronteras que sean interpretables y por tanto sirvan para solucionar un problema de visión por computador. Justificar con todo detalle la bondad de la elección.

Una aplicación para la que se usa el algoritmo de Canny con buenos resultados es en reconocimiento de objetos por descriptores de forma. El problema trata de identificar qué objeto es el que estamos analizando, con un

sistema previamente entrenado. Para ello deben extraerse los bordes, donde entra en juego el algoritmo de Canny y se obtiene una imagen dividida en regiones. El resto del sistema se encarga de analizar cada región por separado, guardando una referencia a cada región que la rodea. A continuación se aplica el algoritmo k-NN (k vecinos más cercanos) para decidir qué objeto es. Este problema usa la visión por computador para solucionar un problema de clasificación en aprendizaje automático. (Ver referencias)

14.- Suponga que necesita implementar un detector de distintas características de la imagen, p.e. bordes y cruces por cero, usando filtros gaussianos. Explique cómo abordaría la implementación de los filtros para garantizar su perfecto funcionamiento en todos los casos.

Habría que considerar las funciones con las que vamos a filtrar. En primer lugar, eliminaría posible ruido con una función Gaussiana, cuyo valor de σ debería ser elegido según la imagen. Para detectar los bordes, elegiría el filtro de Canny pero filtrando con la segunda derivada de la Gaussiana. Esto es una ventaja de cara a calcular los cruces por cero, pues los cruces por cero en la primera derivada de la Gaussiana serían mínimos en la segunda derivada, tal y como muestra la figura:

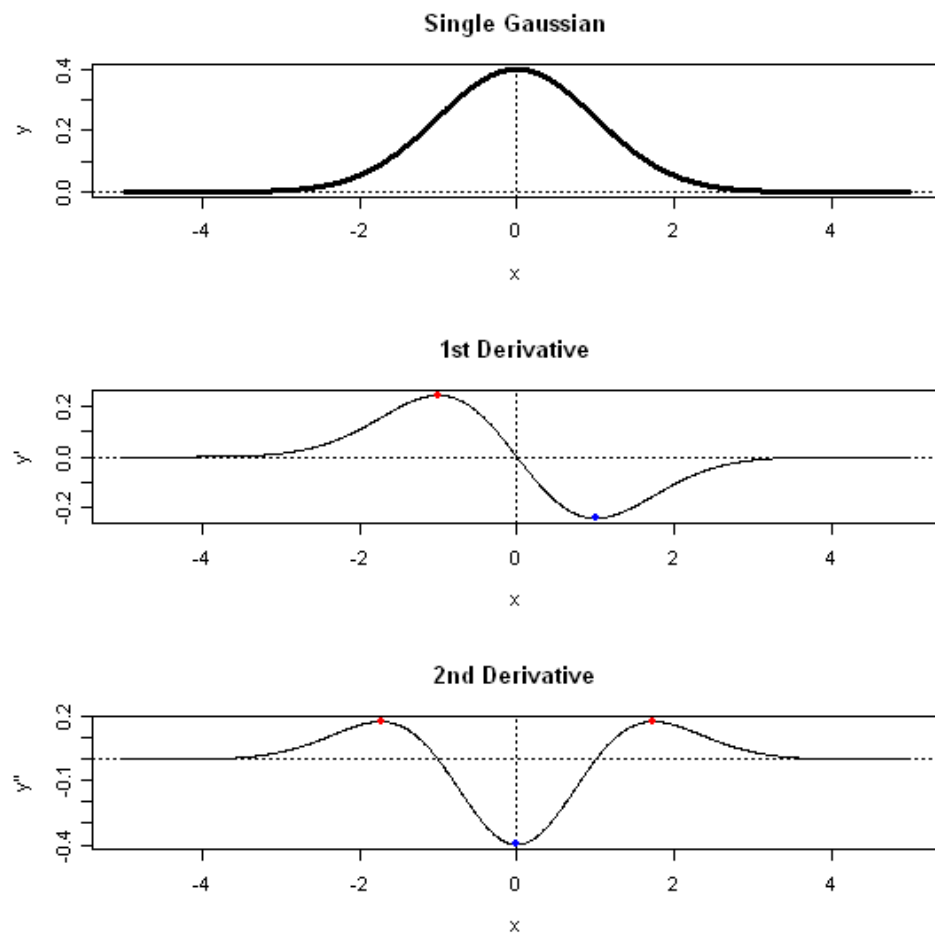


Figura 7. Funcion Gaussiana, primera y segunda derivada
(*Mixtures of Gaussians. Glynn*).

Referencias

- *Computadores paralelos y evaluación de prestaciones. Pág.87*
Francisco J. Quiles Flor, Antonio Garrido del Solo. 1996
<https://books.google.com.cu/books?id=QXNTf4ePkGQC>
- *Introduction to Computer Vision*
Michael J. Black. Sept. 2009
<http://cs.brown.edu/courses/cs143/2009/lecture9.pdf>
- *Correlation and Convolution*
David Jacobs. 2005
<http://www.cs.umd.edu/~djacobs/CMSC426/Convolution.pdf>
- *The Laplacian Pyramid as a Compact Image Code*
Peter J. Burt, Edward H. Alderson. 1983
http://persci.mit.edu/pub_pdfs/pyramid83.pdf
- *Reconocimiento de objetos por descriptores de forma*
Julián Sanz Cuenca. Junio 2008
<http://www.maia.ub.es/~sergio/linked/julian08.pdf>
- *Mixture of Gaussians*
Earl F. Glynn. Febrero 2007
<http://research.stowers-institute.org/efg/R/Statistics/MixturesOfDistributions/>