VC P1: Cuestionario de Teoría 1

Alejandro Alcalde, *Universidad de Granada*

21 de octubre de 2015

Índice

1. Cuestión 1 **2**

2. Cuestión 2 **3**

3. Cuestión 3 **4**

4. Cuestión 4 **4**

5. Cuestión 5 **5**

6. Cuestión 6 **5**

7. Cuestión 7 **6**

8. Cuestión 8 **7**

9. Cuestión 9 **8**

10. Cuestión 10 **9**

11. Cuestión 11 **9**

12. Cuestión 12 **10**

13. Cuestión 13 **12**

14. Cuestión 14 **12**

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

Extraer información con significado del un conjunto de píxeles. Se intenta describir el mundo observado en una imagen (o varias) y reconstruir sus propiedades. Propiedades como la forma, iluminación, distribución de color etcétera.

Con la información extraida a partir de una imagen, se pueden llevar a cabo tareas como:

- Inspeccionar los objetos en una cadena de montaje para verificar que están correctamente.
- Reconocer en una carretera la cantidad de coches circulando para gestionar el tráfico en tiempo real.
- Efectos especiales, como el CGI.
- Solapar varias fotos del mismo paisaje en una a modo de panorama.
- Reconocimiento facial. Además de reconocer la cara, también es interesante reconocer el estado de ánimo de la persona mediante sus microgestos faciales.

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.*

No necesariamente, ya que la convolución es un filtro lineal y un operador de vecindad, lo cual significa que determina el valor de un pixel en función de sus vecinos asignándoles un peso a cada uno. En el caso de un filtro de alisamiento, por ejemplo, la máscara debe cumplir una serie de propiedades, entre ellas que sume 1. Mientras la máscara cumpla esta propiedad, la convolución 1D de la máscara con la imagen 2D alisará la imagen. Siendo cierto por contra, que no estamos siendo justos en cálculo del valor del pixel, ya que estamos obviando ciertos píxeles vecinos de la imagen. Es decir, si tenemos una máscara $\frac{1}{4}\begin{bmatrix}1 & 2 & 1\end{bmatrix}$ y la aplicamos a una imagen 2D, estamos ignorando los valores de los píxeles de arriba, abajo y las diagonales en el cálculo. Para aplicar un filtro correctamente, se deben tener en cuenta todos los píxeles vecinos, por ejemplo con esta máscara:

$$\frac{1}{12} \cdot \begin{bmatrix}1 & 1 & 1\\1 & 4 & 1\\1 & 1 & 1\end{bmatrix} \tag{1}$$

Es frecuente aplicar filtros a imágenes mediante dos máscaras 1D, una para el eje x y otra para el y, ya que el efecto al final será el mismo, porque realmente se está haciendo la multiplicación de dos vectores, que da lugar una matriz.

No tiene sentido definir máscaras para varios canales, ya que una máscara de alisamiento, por ejemplo, alisa la imagen a la que se le aplica. Por tanto las máscaras son una matriz/vector de un solo canal. Para aplicarla a una imagen con más de un canal, se separan los canales en matrices distintas, se les aplica el filtro con la máscara de alisamiento por separado y se vuelven a juntar en una sola matriz con todos los canales.

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

La correlación viene dada por

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

mientras que la convolución:

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

lo cual implica que la máscara se rota 180 grados. Las consecuencias dependen del tipo de máscara, para una máscara simétrica el resultado de la convolución y la correlación es el mismo. Sin embargo, cuando no es simétrica, la convolución alisa, agudiza y detecta bordes mientras que la correlación indica cómo es de similar el trozo analizado con respecto a cualquier punto.

Ambas tienen en común las propiedades *shift invariant* (El operador se comporta de igual manera en todos lados, es decir, la salida depende del patrón en la vecindad de la imagen, no en su posición) y la superposición.

$$h \cdot (f_1 + f_2) = (h \cdot f_1) + (h \cdot f_2)$$

Sin embargo, la convolución es commutativa y asociativa, mientras que la correlación no lo es.

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

Sí, la convolución es una función lineal, ya que su respuesta a la suma de dos señales es la misma que la suma de cada respuesta individual. Los de mediana no son lineales, ya que requieren de ordenar los valores de los píxeles de menor a mayor y quedarse con el del medio. El resultado de aplicar filtros siguiendo este criterio es eliminar píxeles que de algún modo han tomado un valor muy distinto al de sus vecinos, y por tanto introducen ruido.

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

Local, los filtros de alisamiento se aplican a un conjunto de vecinos a un pixel. Ya que lo que se pretende es suavizar la imagen, las operaciones hay que realizarlas de forma local porque dichos píxeles tendrán valores aproximados entre ellos, y se calcula una media. No tendría sentido aplicar un filtro de alisamiento de forma global, pues los valores de los píxeles distarían mucho en sus valores.

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. Justificar la decisión.

La función gradiente

$$\nabla I(x) = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}\right)(x)$$

calcula la dirección del cambio de mayor intensidad en una superficie. Esto puede usarse para detectar bordes en una imagen, ya que un borde no es más que un cambio brusco del valor de los píxeles. Aplicar esta función acentúa las altas frecuencias en una imagen, como resultado se amplifica el ruido y los bordes no se detectarán correctamente. Como solución a este problema, la opción adecuada es la *a*, aplicar primero un filtro de alisamiento circular y simétrico, normalmente el Gaussiano. Aplicando antes el filtro eliminamos las altas frecuencias/ruido, y la función gradiente detectará mejor los bordes.

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.*

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

Esta es la forma no separable, si partimos de la forma separable de la Gaussiana:

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

y derivamos la parte izquierda con respecto a x:

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

y la derecha con respecto a y

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

Como vemos en el resultado, la primera derivada de la Gaussiana es separable también. Ya que se podría poner como sigue:

$$\begin{aligned}\nabla G(x,y) &= \left(\frac{\partial G(x,y)}{\partial x}, \frac{\partial G(x,y)}{\partial y}\right) \\ &= \left(-\frac{xe^{-\frac{x^2}{2\sigma^2}}}{\sqrt{2\pi}\sigma^3}\right)\left(-\frac{ye^{-\frac{y^2}{2\sigma^2}}}{\sqrt{2\pi}\sigma^3}\right)\end{aligned}$$

El papel de los núcleos en las derivadas es calcular la pendiente entre dos puntos, y la dirección del gradiente, con esto, se sabe dónde comienza un borde, ya que un borde es simplemente un cambio abrupto de la intensidad en una región de píxeles. Las derivadas de magnitudes grandes, positivas o negativas, son elementos de bordes verticales. La derivada en la x detectará bordes verticales, mientras que la y bordes horizontales.

Que los nucleos sean separables implica que el cálculo es más eficiente, ya que la operación requiere de $\mathcal{O}(K^2)$ por píxel, mientras que aplicandola con los núcleos seperados, como vectores 1D vertical, y luego otro 1D horizontal, requiere de $\mathcal{O}(2K)$.

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.

Tomemos $G(x,y) = \left(c \cdot e^{-\frac{x^2+y^2}{2\sigma^2}}\right)$ siendo c una constante. En el ejercicio anterior vimos que:

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

La segunda derivada, que corresponde con la Laplaciana sería (Se omiten en este caso c para simplificar los cálculos), comenzamos derivando la x:

$$\begin{aligned}\frac{\partial^2 G(x,y)}{\partial^2 x} &= \left(-\frac{x}{\sigma^2}\right)\left(e^{-\frac{x^2+y^2}{2\sigma^2}}\right)\\ &= \left(-\frac{x}{\sigma^2}\right)\left(-\frac{x}{\sigma^2}\right)\left(e^{-\frac{x^2+y^2}{2\sigma^2}}\right) - \left(\frac{1}{\sigma^2}e^{-\frac{x^2+y^2}{2\sigma^2}}\right)\\ &= e^{-\frac{x^2+y^2}{2\sigma^2}}\left(\frac{x^2}{\sigma^4}-\frac{1}{\sigma^2}\right)\\ &= e^{-\frac{x^2+y^2}{2\sigma^2}}\left(\frac{x^2-\sigma^2}{\sigma^4}\right)\end{aligned}$$

Para la y quedaría igual:

$$\begin{aligned}\frac{\partial^2 G(x,y)}{\partial^2 y} &= \left(-\frac{y}{\sigma^2}\right)\left(e^{-\frac{x^2+y^2}{2\sigma^2}}\right)\\ &= \left(-\frac{y}{\sigma^2}\right)\left(-\frac{y}{\sigma^2}\right)\left(e^{-\frac{x^2+y^2}{2\sigma^2}}\right) - \left(\frac{1}{\sigma^2}e^{-\frac{x^2+y^2}{2\sigma^2}}\right)\\ &= e^{-\frac{x^2+y^2}{2\sigma^2}}\left(\frac{y^2}{\sigma^4}-\frac{1}{\sigma^2}\right)\\ &= e^{-\frac{x^2+y^2}{2\sigma^2}}\left(\frac{y^2-\sigma^2}{\sigma^4}\right)\end{aligned}$$

Escribiendolo todo junto, con la constante quedaría:

$$\begin{aligned}\nabla^2 G(x,y) &= \left(\frac{\partial^2 G(x,y)}{\partial^2 x}, \frac{\partial^2 G(x,y)}{\partial^2 y}\right)\\ &= \left(c\cdot e^{-\frac{x^2+y^2}{2\sigma^2}}\left(\frac{x^2-\sigma^2}{\sigma^4}\right), c\cdot e^{-\frac{y^2+y^2}{2\sigma^2}}\left(\frac{y^2-\sigma^2}{\sigma^4}\right)\right)\end{aligned}$$

El papel que juegan dichos núcleos es el de representación espacial a escala.

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

El proceso de reducción de tamaño de una imagen se llama decimación. Para poder realizarla, primero se aplica una convolución a la imagen con un filtro de paso bajo, eliminando así las altas frecuencias (*aliasing*) y luego quedarnos con cada r-ésima muestra. A r se le llama el ratio de decimación. Sólo se realiza la convolución a cada r-ésima muestra:

$$g(i,j) = \sum_{k,l} f(k,l)h(ri-k, rj-l)$$

El papel que juega cada una de ellas es el siguiente:

- La convolución con un filtro de paso bajo elimina las altas frecuencias, de este modo, se elimina el ruido existente en la imagen. Si no se aplica este filtro para eliminar altas frecuencias, a medida que se reduce la imagen, el ruido se irá haciendo más notable en la imagen.
- Quedándonos con cada r-ésima muesta, estamos eliminando píxeles de la imagen, y por tanto reduciendo su tamaño.

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

Se conservan las bajas frecuencias, ya que para cada nivel de la pirámide que se sube, se aplica un alisado y luego se remuestrea. Es fácil comprobar que son las bajas frecuencias las que se mantienen. Por ejemplo, tomemos la imagen de partida, llamémosla f, a la imagen con un filtro de alisado la llamaremos f', si restamos ambas $(f - f')$ nos quedará g, y esta g son las altas frecuencias, que se van perdiendo. Lo que la pirámide Gaussiana hace con f' es remuestrearla a un ratio determinado, para reducir su escala, ya sin frecuencias altas y por tanto conservando las bajas. El motivo de eliminar las altas frecuencies es evitar el *aliasing*.

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

La principal diferencia reside en que la Laplaciana está formada por pirámides Gaussianas.

La Gaussiana nos aporta una estimación multi escala, cuando queremos reconocer un patrón, normalmente no se sabe a qué escala se encuentra, realizando una pirámide Gaussiana se podrá buscar el patrón a distintas escalas. La Laplaciana sin embargo, realiza la operación vista en el ejercicio anterior para querdarse únicamente con las frecuencias altas. Esto tiene un rango amplio de aplicaciones, como codificación y compresión de imágenes (Ya que almacenar solo las altas frecuencias requiere de poco espacio, y al tener muchos ceros, se puede comprimir), reconstrucción de la imagen, realizar composiciones de imagenes (mezclar dos objetos en uno).

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 filtros de *Sobel* y *Robert* son más simples, se describen brevemente a continuación:

Robert realiza una operación 2D simple para calcular el gradiente, acentuando regiones con frecuencias espaciales altas que a menudo se corresponden con bordes. Usa dos máscaras de 2x2 y las convoluciona, una máscara es la rotacion a 90 grados de la otra. Es muy rápido de calcular, pero al usar un kernel tan pequeño es muy sensible al ruido. Para solucionar el problema del ruido se puede establecer un umbral y eliminar el valor de los píxeles que lo sobrepasen.

Sobel realiza un gradiente 2D al igual que *Robert*. Usa dos kernels 3x3 para convolucionarlos, un kernel es el otro rotado 90 grados. Más que el de *Robert*, pero al tener un kernel mas grande alisa la imagen y es menos sensible al ruido. Aunque también produce ruido como *Robert*, es más fácil de quitar mediante la imposición de umbrales.

Canny, sin embargo, trabaja en un proceso multi etapa:

1. Alisa.
2. Realiza el gradiente.
3. Supresión de No-Máximos.
4. Doble umbral.
5. Rastreo de bordes mediante histéresis.

Las dos primeras etapas son similares a los dos filtros anteriores. A continuación se describen las etapas:

1. El alisamiento se realiza con una convolución Gaussiana.

2. Se aplica la primera derivada a la imagen alisada para resaltar regiones de la imagen con bordes.

3. *Canny* intenta convertir los bordes detectados por el gradiente en bordes nítidos, conservando los maximos locales en la imagen gradiente y borrando lo demas. Para ello se redondea la dirección del gradiente θ a 45 grados con un vecindario 8-conectado. Luego se compara la intensidad del pixel analizado con la intensidad de los píxeles en la dirección positiva y negativa de su dirección. Por ejemplo, si el pixel analizado apunta al norte, se compara con el pixel de arriba y el de abajo. Por último, si la intensidad del pixel analizado es mayor que la de los otros dos, se conserva, de lo contrario se elimina. es decir, si tenemos

$$\begin{bmatrix} 2 & 4 & 1 \end{bmatrix}$$

   y la dirección del pixel del medio es hacia la izquierda, al compararlo con los otros dos, su intensidad es mayor y por tanto se conserva.

4. Tras aplicar la supresión de no máximos, algunos de los bordes que quedan pueden no ser bordes reales, y que se hayan confundido con ruido. Para elminarlos se usa un umbral para quedarse únicamente con los que lo sobrepasen. *Canny* usa un umbral doble, los píxeles mayores que el primer umbral se marcan como "fuertes", los menores que el umbral bajo se borran y los que quedan entre ambos umbrales se marcan como "débiles". A los bordes fuertes se les asigna una intensidad mayor que a los débiles.

5. Ahora, los bordes marcados como fuertes se interpretan como "bordes certeros", y se incluyen en la imagen final como bordes. Lo débiles se incluyen sí y solo sí están conectados a bordes fuertes. De este modo, eliminamos los bordes débilies que probablemente se hayan detectado como consecuencia de ruido, sin ser bordes reales.

El operador *Canny* por tanto, se determina por 3 parámetros, la anchura del kernel Gaussiano y los dos umbrales.

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.*

En entornos industriales, donde la iluminación está controlada, y el objeto a analizar está siempre en una posición determinada, por ejemplo en una cinta transportadora, el filtro de *Canny* podría usarse para verificar que el objeto analizado está correctamente ensamblado y no le falta ninguna pieza.

14. *BONUS: Usando la descomposición SVD (Singular Value Decomposition) de una matriz, deducir la complejidad computacional que es posible alcanzar en la implementación de la convolución 2D de una imagen con una máscara 2D de valores y tamaño cualesquiera (suponer la máscara de tamaño inferior a la imagen).*