
Curso: 2019 - 2020

Cuestionario 1

TEORÍA

José Javier Alonso Ramos



**UNIVERSIDAD
DE GRANADA**

Índice

Ejercicio 1	3
Ejercicio 2	3
Ejercicio 3	4
Ejercicio 4	5
Ejercicio 5	5
Ejercicio 6	6
Ejercicio 7	6
Ejercicio 8	7
Ejercicio 9	7
Ejercicio 10	8
Ejercicio 11	8
Ejercicio 12	9
Ejercicio 13	10
Ejercicio 14	10
Ejercicio 15	10
Fuentes:	10

Ejercicio 1

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 las imágenes de cara a la creación algoritmos que la procesen.**

El objetivo es ser capaces de extraer información de las imágenes ya que una sola puede contener mucha información relevante y nos evita tener que buscar en distintas fuentes e incluso contener información imposible de conseguir de otro modo.

La principal característica de cara a la creación de algoritmos es su representación como matriz numérica. Simplifica el tratamiento de la imagen y la extracción de sus características en un formato entendible y computable.

Ejercicio 2

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.**

■ **Semejanzas:**

Ambas operaciones aplican filtros lineales. Recorren una imagen con una máscara 2D, modificando los píxeles que coincidan con el tamaño de la máscara y dando como resultado una nueva imagen.

Las dos operaciones poseen las propiedades de **shift invariant** (el valor de salida depende del patrón de los píxeles vecinos y no de la posición de los mismos) y **superposición** (a la hora de aplicar una máscara a una combinación de imágenes, da el mismo resultado aplicar la máscara a la combinación o aplicar la máscara a cada imagen por separado y después combinarlas).

■ **Diferencias:**

Mientras que a la hora de aplicar la máscara en la correlación ésta se aplica “superponiéndola” a la imagen, es decir, el píxel [n,m] de la máscara afecta al píxel [n,m] de la imagen (representación de la zona local de la imagen afectada por la máscara); en la convolución la máscara se encuentra volteada en el eje horizontal y vertical o, lo que es lo mismo, el píxel [0,0] de la máscara afecta al píxel [n,m] de la imagen y viceversa, así como el [0,m] afecta al [n,0] y al contrario.

Esta diferencia se ve reflejada en sus fórmulas:

Convolución:

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

Correlación:

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

Sin embargo, aunque esto diferencie a estas dos operaciones, no será así cuando la máscara se trate de una máscara simétrica en ambos ejes ya que al voltearla no cambiarán la disposición de sus valores.

Otra diferencia es que solo la convolución tiene propiedades como la conmutativa, asociativa, distributiva en la suma, extracción de factores escalares e identidad. Por último, mencionar que solemos usar la correlación para la búsqueda de patrones y la posición de este patrón en la imagen mientras que la convolución es más usada para la aplicación de filtros o transformaciones como el filtro de Gaussiana.

Ejercicio 3

3. **¿Cuál es la diferencia “esencial” entre el filtro de convolución y el de mediana? Justificar la respuesta.**

La diferencia esencial entre estos dos filtros es que el filtro de convolución define una función lineal y el de mediana no. La convolución combina los píxeles de la máscara aplicándoles unos pesos para dar como resultado el nuevo píxel; hallar la mediana de los píxeles vecinos no es una función lineal. Vamos a demostrarlo. Para que una función sea lineal debe cumplir dos propiedades:

$$1 - f(x) + f(y) = f(x + y) \quad 2 - f(k \cdot x) = k \cdot f(x)$$

Suponemos dos imágenes genéricas $I = \{i_1, i_2, i_3, \dots, i_n\}$ y $J = \{j_1, j_2, j_3, \dots, j_n\}$. de igual tamaño.

■ **Convolución:**

Vamos a usar el filtro *media*.

- $$f(x) = \text{media}(x) = \frac{1}{N} \sum_{i=1}^N x_i$$

$$f(I) = \frac{1}{N} \sum_{i=1}^N i_i$$

$$f(J) = \frac{1}{N} \sum_{i=1}^N j_i$$

$$f(I) + f(J) = \text{media}(I) + \text{media}(J) = \frac{1}{N} \sum_{i=1}^N i_i + \frac{1}{N} \sum_{i=1}^N j_i = \frac{1}{N} \sum_{i=1}^N i_i + j_i$$

$$f(I + J) = \frac{1}{N} \sum_{i=1}^N i_i + j_i$$

$$f(I) + f(J) = f(I + J)$$

$$2. \quad k \cdot f(I) = k \cdot \text{media}(I) = k \cdot \frac{1}{N} \sum_{i=1}^N i_i = \frac{1}{N} \sum_{i=1}^N k \cdot i_i$$

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

■ **Mediana:**

Para demostrar que no define una función lineal vamos a buscar un contraejemplo:

Suponemos dos imágenes genéricas $I = \{1, 2, 4, 5, 6\}$ y $J = \{9, 9, 9, 11, 12\}$. de igual tamaño.

$$\text{mediana}(I) = 4$$

$$\text{mediana}(J) = 9$$

$$\text{mediana}(I) + \text{mediana}(J) = 13 \quad I + J = \{10, 11, 12, 16, 18\}$$

$$\text{mediana}(I + J) = 12$$

$$\text{mediana}(I) + \text{mediana}(J) \neq \text{mediana}(I + J)$$

Ejercicio 4

4. **Identifique el “mecanismo concreto” que usa un filtro de máscara para transformar una imagen.**

El mecanismo al que nos referimos es que la máscara utiliza información local para transformar un píxel en concreto. Esta información local es el valor de los píxeles que se encuentran dentro de la propia máscara que son los que conocemos como el vecindario del píxel que estamos tratando.

Para calcular el nuevo valor del píxel tenemos que tener en cuenta los valores que lo rodean y por tanto deducimos que el orden de los píxeles es importante. Si desordenamos los píxeles de una imagen, después aplicamos un filtro local y por último volvemos a ordenar los píxeles no obtendremos el mismo resultado que si sólo aplicamos el filtro local sin desordenar. Esto no es así si aplicamos un filtro global en el que no se tiene en cuenta los valores vecinos.

Ejercicio 5

5. **¿De qué depende que una máscara de convolución pueda ser implementada por convoluciones 1D? Justificar la respuesta.**

El poder implementar la máscara de convolución como máscaras 1D depende de si la máscara 2D es separable o no. Esto lo podemos saber si la matriz que forma la máscara 2D es de rango 1. Si lo es, podremos descomponer el filtro de convolución 2D en dos filtros 1D.

Sabiendo que un filtro 2D separable es tal que:

$$h(u, v) = h_1(u) \cdot h_2(v) \quad \text{Tendríamos una convolución de la siguiente manera:}$$

$$h(u, v) \cdot f(x, y) = h_1(u) \cdot h_2(v) \cdot f(x, y)$$

Gracias a las propiedades conmutativa y asociativa de la convolución sabemos que da igual aplicar primero el filtro sobre las columnas o sobre las filas, o multiplicar ambos filtros y luego aplicarlos, que sería lo mismo que aplicar la máscara 2D.

Ejercicio 6

6. **Identificar las diferencias y consecuencias desde el punto de vista teórico y de la implementación entre:**

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.

Justificar los argumentos.

Teóricamente, como ya hemos visto anteriormente, gracias a la propiedad conmutativa y asociativa da igual si realizamos las operaciones en el orden propuesto por a) o por b) ya que ambas nos deben dar (y darán) los mismos resultados.

En cambio en la práctica si que nos encontramos diferencias:

Si seguimos el orden a) tendremos que hacer tres operaciones: primero alisar la imagen original y después derivar dos veces (una vez en x y otra en y) la imagen alisada.

Si seguimos el orden b) tendremos que realizar 4 operaciones: las dos derivadas de la imagen original (en x y en y) y un alisamiento a cada una de ellas.

Por lo tanto en la práctica es más conveniente seguir el orden propuesto por a) ya que nos ahorramos una operación.

Ejercicio 7

7. **Identifique las funciones de las que podemos extraer pesos correctos para implementar de forma eficiente la primera derivada de una imagen. Suponer alisamiento Gaussiano.**

Para extraer los pesos correctos para derivar usando alisamiento Gaussiano lo más eficiente sería derivar la función Gaussiana y extraer de esa función los pesos ya que nos da igual aplicar el filtro Gaussiano y después derivar; que derivar el filtro gaussiano (o hallar directamente el filtro derivado) y aplicarlo a la imagen (Teorema derivativo de la convolución). Los resultados son los mismos:

$$\frac{\partial}{\partial x}(h \star f) = \left(\frac{\partial}{\partial x}h\right) \star f$$

$h := \text{filtro Gaussiano}$

$f := \text{imagen}$

Por lo tanto la función de la que podemos extraer los pesos para el filtro es la derivada de la Gaussiana:

$$\frac{\partial g(x, \sigma)}{\partial x} = -\frac{x}{\sigma^3 \sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}}$$

Ejercicio 8

8. **Identifique las funciones de las que podemos extraer pesos correctos para implementar de forma eficiente la Laplaciana de una imagen. Suponer alisamiento Gaussiano.**

Al partir de una imagen con alisamiento Gaussiano lo que estamos calculando es la Laplaciana de la Gaussiana que es igual a la segunda derivada de la Gaussiana. Por lo tanto, con la explicación dada en el ejercicio anterior sabemos que obtenemos los mismos resultados aplicando los filtros y derivando después, que derivando la función que nos proporciona los filtros y aplicando estos a la imagen. Así pues, la función que nos proporcionará los pesos correctos para obtener la Laplaciana de una imagen aplicando previamente un filtro Gaussiano en la segunda derivada de la Gaussiana:

$$\frac{\partial^2 g(x, \sigma)}{\partial^2 x} = -\frac{\sigma^2 - x^2}{\sigma^5 \sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}}$$

Ejercicio 9

9. **Suponga que le piden implementar de forma eficiente un algoritmo para el cálculo de la derivada de primer orden sobre una imagen usando alisamiento Gaussiano. Enumere y explique los pasos necesarios para llevarlo a cabo.**

El algoritmo a llevar a cabo es el descrito en el ejercicio 7 que consta de los siguientes pasos:

- 1. Calcular la derivada de la Gaussiana (o partir de ella directamente)
- 2. Fijar un valor de σ y el tamaño del kernel ($[-3\sigma, +3\sigma]$ por ejemplo)
- 3. Descomponer la máscara o kernel 2D obtenido en kernels 1D usando la descomposición en valores singulares (SVD)

$$G(x, y) = \sum_{i=1}^n \sigma_i u_i v_i^T = \sigma_1 u_1 v_1^T$$

Podemos hacer esa igualdad ya que solo será separable si y solo si $\forall i > 1 \rightarrow \sigma_i = 0$. Como sabemos que la función Gaussiana es separable podemos afirmar esto.

- 4. Aplicar por convolución los kernels 1D obtenidos (u y v) a la imagen

Ejercicio 10

10. **Identifique semejanzas y diferencias entre la pirámide gaussiana y el espacio de escalas de una imagen, ¿cuándo usar una u otra? Justificar los argumentos.**

■ **Semejanzas:**

En ambos casos aplicamos iterativamente filtros de Gaussiana y realizamos un subsampling a las imágenes.

■ **Diferencias:**

La pirámide Gaussiana aplica un subsampling cada vez que aplica el filtro Gaussiano mientras que en el espacio de escalas es más habitual aplicar varias veces el filtro de suavizado antes de bajar una octava.

Otra diferencia es para qué se usa cada concepto. Por un lado, la pirámide Gaussiana se utiliza para la reconstrucción de imágenes: en el caso de realizar un upsampling realiza una interpolación de los píxeles vecinos y cuando realizamos un subsampling aplica anti-aliasing (suaviza los bordes de la imagen). Por otro lado el espacio de escalas se utiliza para la detección de características (zonas de interés en la imagen) que se mantienen constantes sin importar la escala (distancia desde la cámara al objeto en cuestión, por ejemplo).

Ejercicio 11

11. **¿Bajo qué condiciones podemos garantizar una perfecta reconstrucción de una imagen a partir de su pirámide Laplaciana? Dar argumentos y discutir las opciones que considere necesario.**

En primer lugar hay que aclarar que nunca conseguiremos una reconstrucción 100 % perfecta ya que a la hora de reducir y aumentar la resolución de las imágenes estamos perdiendo información de los píxeles originales ya sea por supresión o interpolación de los mismos. Ahora sabiendo esto, si es cierto que podemos obtener una imagen bastante parecida a la que teníamos en un inicio. Podemos reconstruir una imagen si el último nivel de la pirámide Laplaciana corresponde con la imagen suavizada (filtro de Gaussiana aplicado) que se restaría a la imagen en ese nivel de la Laplaciana y seguimos una serie de pasos. Partiendo desde el último nivel hacia arriba:

- 1. Hacer upsampling del nivel al tamaño del nivel superior
- 2. Sumar estas dos imágenes
- 3. Continuar ascendiendo en los niveles partiendo ahora de la imagen obtenida en 2.

La reconstrucción funciona debido a que el último nivel se trata de un filtro de Gaussiana y no de uno de Laplaciana y podemos obtener así las frecuencias altas y bajas de la imagen.

La reconstrucción podría ser perfecta si a la hora de restaurar la imagen Gaussiana (upsampling) no hiciéramos una interpolación de los píxeles que insertamos y supiéramos a ciencia cierta cuáles eran los valores que tenían antes de hacer subsampling.

Ejercicio12

12. . **¿Cuáles 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**

Lo más importante es que Canny desarrolló una teoría computacional acerca de la detección de bordes. Es un algoritmo de detección de bordes más completo que el de Sobel, Roberts o Prewitt ya que es una extensión de los mismos. El algoritmo de Canny sigue el siguiente procedimiento:

- 1. Aplica un fultro Gaussiano para reducir el ruido de la imagen ya que este filtro se ve muy afectado por el mismo.
- 2. Encuentra los gradientes de intensidad de la imagen. Aquí es donde entra en juego el filtro de Sobel. Se aplica sobel para obtener un valor de la primera derivada en la dirección horizontal y vertical. A partir de esto se puede determinar el gradiente del borde y su dirección. Canny usa cuatro filtros para deteectar bordes horizontales, verticales y diagonales.
- 3. Supresión de no máximos. Una vez calculados los gradientes y sus direcciones se comparan los gradientes de los píxeles de manera local. Si el gradiente del píxel es el mayor de entre todos los demás gradientes locales con la misma dirección el valor se mantiene, de lo contrario, es borrado (puesto a 0).
- 4. Aplicar doble umbral para determinar posibles bordes. Tras la supresión de no máximos los píxeles restantes proporcionan una representación más precisa de los bordes reales de la imagen, sin embargo quedan algunos bordes causados por ruido en la imagen o variación de color. Para tratar de corregir esto definiremos dos umbrales: gradiente débil y rgadiente fuerte. Si el gradiente de un borde es menor que el umbral débil será eliminado, si un gradiente es menor que el umbral fuerte será marcado como píxel de borde débil, y si el gradiente es mayor que el umbral fuerte será marcado como píxel de borde fuerte.
- 5. Seguimiento de bordes por histéresis. Los píxeles de borde débil están en duda de si deben ser eliminados (han sido causados por ruido en la imagen) o no (han sido formados por un borde real de la imagen). Para decidirlo debemos comprobar que haya un piíxel de borde fuerte conectado al píxel de borde débil, es decir, mediante análisis de blobs debemos comprobar que en la vecindad (8 píxeles alrededor) del píxel débil se encuentra un píxel de

borde fuerte. Si efectivamente se encuentra uno en el vecindario el píxel de borde débil permanece de lo contrario se elimina.

Ejercicio 13

13. **Identificar pros y contras de k-medias como mecanismo para crear un vocabulario visual a partir del cual poder caracterizar patrones. ¿Qué ganamos y que perdemos? Justificar los argumentos**

Ejercicio 14

14. **Identifique pros y contras del modelo de “Bolsa de Palabras” como mecanismo para caracterizar el contenido de una imagen. ¿Qué ganamos y que perdemos? Justificar los argumentos.**

Ejercicio 15

15. **Suponga que dispone de un conjunto de imágenes de dos tipos de clases bien diferenciadas. Suponga que conoce como implementar de forma eficiente el cálculo de las derivadas hasta el orden N de la imagen. Describa como crear un algoritmo que permita diferenciar, con garantías, imágenes de ambas clases. Justificar cada uno de los pasos que proponga.**

Fuentes:

Ejercicio 10: Wikipedia - Scale Space