

Maestría en Inteligencia Artificial

Percepción Computacional

Índice

Esquema. Tema 1	3
Ideas clave. Tema 1	4
1.1. ¿Cómo estudiar este tema?	4
1.2. Objetivos	5
1.3. Funcionamiento del sistema auditivo	5
1.4. Percepción visual	12
Esquema. Tema 2	27
Ideas clave. Tema 2	28
2.1. ¿Cómo estudiar este tema?	28
2.2. Objetivos	28
2.3. Descubriendo los elementos esenciales de un sistema de percepción	29
2.4. Captura de información	32
2.5. Procesamiento de la información	42
2.6. Toma de decisión	47
2.7. Referencias bibliográficas	48
Esquema. Tema 3	51
Ideas clave. Tema 3	52
3.1. ¿Cómo estudiar este tema?	52
3.2. Objetivos	52
3.3. Introducción	52
3.4. Sistemas de conversión analógico-discreto	55
3.4. Muestreo y cuantificación	60
Esquema. Tema 4	68
Ideas clave. Tema 4	69
4.1. ¿Cómo estudiar este tema?	69
4.2. Entropía: concepto y estimación	69
4.3. Fuentes perturbadoras de las señales	76
4.4. Caracterización matemática del ruido: procesos estocásticos	80

Esquema. Tema 5	86
Ideas clave. Tema 5	87
5.1. ¿Cómo estudiar este tema?	87
5.2. Definición de anomalía	88
5.3. Métodos de identificación de anomalías	93
5.4. Eliminación de anomalías	97
5.5. Referencias bibliográficas	104
Esquema. Tema 6	106
Ideas clave. Tema 6	107
6.1. ¿Cómo estudiar este tema?	107
6.2. Ajuste de intensidad	110
6.3. Procesado sistemático del histograma	117
6.4. Suavizado y realce mediante operadores aritméticos	119
6.5. Referencias bibliográficas	123
Esquema. Tema 7	125
Ideas clave. Tema 7	126
7.1. ¿Cómo estudiar este tema?	126
7.2. Filtros paso bajo y paso alto	129
7.3. Detección de bordes	137
7.4. Referencias bibliográficas	140
Esquema. Tema 8	142
Ideas clave. Tema 8	143
8.1. ¿Cómo estudiar este tema?	143
8.2. Introducción a la morfología matemática	143
8.3. Definición de elemento estructural	146
8.4. Erosión y dilatación	149
8.5. Apertura y clausura	154
8.6. Gradiente morfológico	157
8.7. <i>Top Hat</i>	158
8.8. Ejercicio práctico	159
8.9. Referencias bibliográficas	161

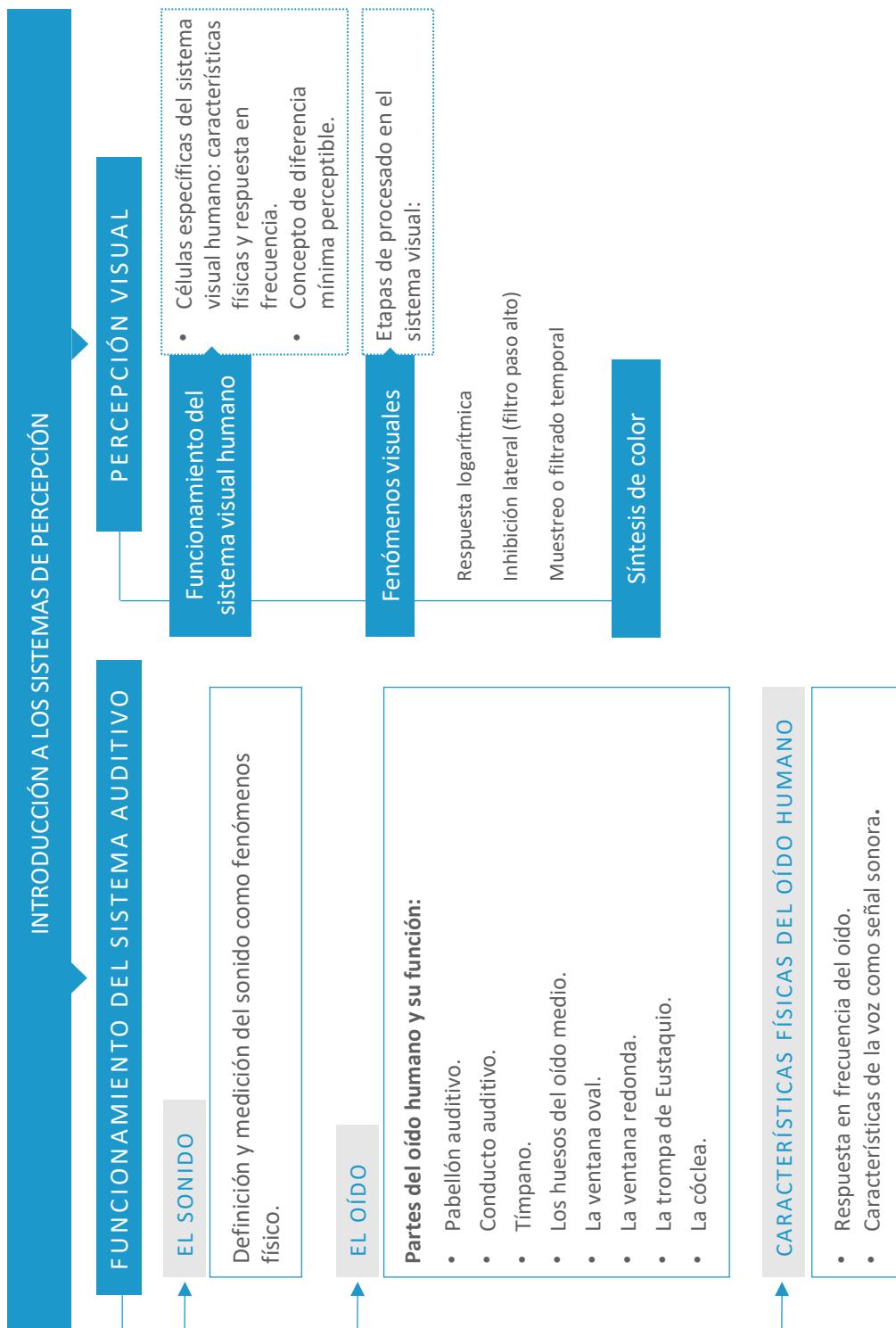
Esquema. Tema 9	164
Ideas clave. Tema 9	165
9.1. ¿Cómo estudiar este tema?	165
9.2. Objetivos	165
9.3. Introducción al análisis en frecuencia	165
9.4. La transformada de Fourier	170
9.5. Transformada discreta de Fourier (DFT) y su implementación mediante <i>Fast Fourier Transform</i> (FFT)	181
Esquema. Tema 10	187
Ideas clave. Tema 10	188
10.1. ¿Cómo estudiar este tema?	188
10.2. Segmentación y crecimiento de regiones	188
10.3. Técnicas empleadas en el crecimiento de regiones	190
10.4. Crecimiento de regiones basado en semillas	192
10.5. Crecimiento de regiones basado en <i>Split and Merge</i>	194
10.6. Crecimiento de regiones basado en <i>Gradient Vector Flow</i> (GVF)	198
10.7. Crecimiento de regiones basado en <i>Watershed</i>	203
10.8. Crecimiento de regiones basado en grafos	206
10.9. Referencias bibliográficas	210
Esquema. Tema 11	212
Ideas clave. Tema 11	213
11.1. ¿Cómo estudiar este tema?	213
11.2. Caracterización de señales en el dominio natural	215
11.3. Características derivadas del análisis en frecuencia	223
Esquema. Tema 12	228
Ideas clave. Tema 12	229

12.1. ¿Cómo estudiar este tema?	229
12.2. Matriz de coocurrencia de Haralick	231
12.3. Patrones binarios locales	234
12.4. Caracterización basada en transformadas unitarias	238
12.5. Referencias bibliográficas	242
Esquema. Tema 13	245
Ideas clave. Tema 13	246
13.1. ¿Cómo estudiar este tema?	246
13.2. Introducción a la extracción de características basada en procesamientos multiescala	246
13.3. Definición de transformada Wavelet	248
13.4. Filtros de Gabor	255
13.5. Transformada SIFT	258
Esquema. Tema 14	265
Ideas clave. Tema 14	266
14.1. ¿Cómo estudiar este tema?	266
14.2. Clasificación y reconocimiento de patrones	268
14.3. Aplicación de técnicas <i>machine learning</i> al procesado de señales	280
Esquema. Tema 15	286
Ideas clave. Tema 15	287
15.1. ¿Cómo estudiar este tema?	287
15.2. <i>Biometrics</i>	287
15.3. <i>Self-driving car</i>	295
15.4. Retos y limitaciones de la percepción computacional	302

Percepción Computacional

Introducción a los sistemas de percepción

Esquema. Tema 1



Ideas clave. Tema 1

1.1. ¿Cómo estudiar este tema?



Accede al vídeo «Introducción a los sistemas de percepción» a través del aula virtual

Para estudiar este tema deberás leer con atención las ideas clave que se desarrollan a continuación.

En el presente tema veremos con detalle los sistemas de percepción del ser humano: el sistema auditivo y el sistema visual.

Sistemas de percepción

Sistema auditivo

- ▶ Definición y medición del sonido como fenómeno físico.
- ▶ Partes del oído humano y su función:
 - Oído interno.
 - Oído medio.
 - Oído externo.
- ▶ Respuesta en frecuencia del oído.
- ▶ Características de la voz como señal sonora.

Sistema visual

- ▶ Células específicas del sistema visual humano: características físicas y respuesta en frecuencia.

- ▶ Concepto de diferencia mínima perceptible.
- ▶ Etapas de procesado en el sistema visual:
 - Respuesta logarítmica.
 - Inhibición lateral (filtro paso alto).
 - Muestreo o filtrado temporal.
- ▶ Síntesis de color.

1.2. Objetivos

El objetivo de este tema es comprender los **mecanismos de captación de señales auditivas y visuales** por parte del ser humano. Estos sistemas sirven de referencia en la implementación de técnicas artificiales de percepción, pues su arquitectura y funcionamiento trata, en muchos casos, de emular el funcionamiento de los sistemas biológicos que encontramos en la naturaleza.

Asimismo, el conocimiento de los procesos de recepción y transformación de estímulos en el ser humano, como la luz o el sonido, facilitará la comprensión de las técnicas de tratamiento de información descritas en temas posteriores de la asignatura.

1.3. Funcionamiento del sistema auditivo



Accede al vídeo «Funcionamiento del sistema auditivo» a través del aula virtual

El sonido

El sonido se define como una **perturbación mecánica del medio** (elástico) por el que se propaga en forma de onda. El sonido resulta de la vibración de ida y vuelta de las partículas del medio a través del cual se mueve dicha onda. Si una onda de sonido se mueve de izquierda a derecha a través del aire, las partículas de aire se desplazarán hacia la derecha y hacia la izquierda a medida que la energía de la onda de sonido lo atraviesa.

La **caracterización física** del sonido permite su medición y evaluación cuantitativa ante diferentes fenómenos:

- ▶ La intensidad acústica de una onda sonora viene dada por la rapidez promedio del flujo de energía acústica a través de un área unitaria, midiéndose en W/m^2 .
- ▶ Los niveles sonoros son la medida de la energía sonora en dB. Se define el nivel de intensidad (NI) de la siguiente forma:

$$NI = 10 \log\left(\frac{I}{I_{ref}}\right)$$

Donde I_{ref} es la **intensidad de referencia**. Esta representa el umbral de audibilidad a 1 KHz en el aire libre y viene dada por:

$$I_{ref} = 10^{-12} \text{ W/m}^2$$

En este apartado se estudiará cómo el cuerpo humano es capaz de captar y procesar estas ondas sonoras. Para ello, en primer lugar, se lleva a cabo una descripción del oído como órgano fundamental en la captación y transducción de estas señales.

El oído

El oído es el órgano encargado de la transmisión de sonidos hasta el cerebro. En él, se distinguen tres partes diferentes: **oído externo, oído medio y oído interno**.

La siguiente imagen muestra la estructura del oído, observándose cada uno de los elementos que lo componen. Además de ser el responsable del sentido del equilibrio, el oído detecta y convierte las señales sonoras en impulsos eléctricos capaces de ser conducidos a través del sistema nervioso.

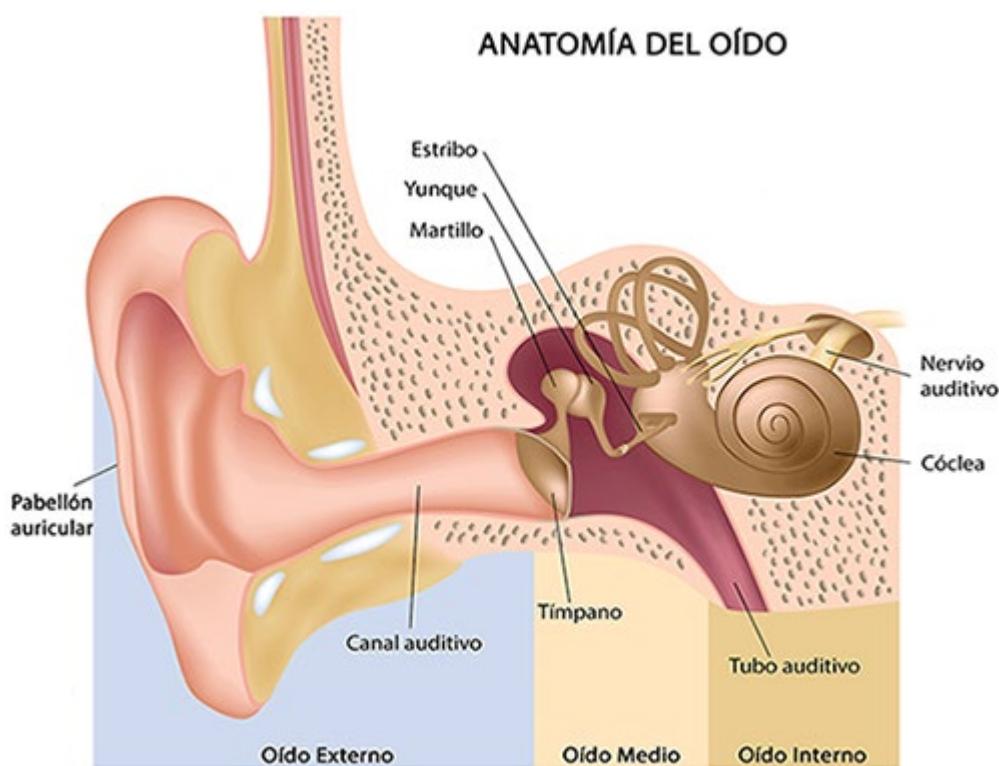


Figura 1. Estructura del oído. Detalle de los elementos que componen oído externo, medio e interno.

Fuente: <https://www.medicaloptica.es/salud-auditiva/como-funciona-el-oido>

Oído externo

La **función** del oído externo es recoger las ondas sonoras procedentes del entorno. Hace las veces de una antena; de hecho, su forma contribuye a reducir la impedancia o resistencia para las ondas, que capta las señales sonoras y las redirige hacia el interior.

Los **elementos** que componen el oído externo son el pabellón auditivo, el conducto auditivo y el tímpano. A continuación, se indica cada una de sus partes y sus correspondientes funciones en el proceso de captación de sonidos:

- ▶ Pabellón auditivo: El pabellón auditivo o, como se conoce comúnmente, **oreja**, se trata de la única parte externa y visible del oído y se caracteriza por su forma helicoidal. Funciona como una antena receptora de señales sonoras, que son captadas y dirigidas hacia el interior del oído.
La **presión en el interior** del oído es mayor que en el exterior, debido a la mayor compresión del aire contenido, por lo que las ondas no penetrarían si no estuviéramos dotados del pabellón auditivo. Este actúa como una interfaz entre el interior y el exterior, disminuyendo la resistencia que la parte interior del oído ofrece a las ondas sonoras y suavizando la diferencia de presión existente. Como resultado, un mayor flujo de sonido es capaz de penetrar en el conducto auditivo.
- ▶ Conducto auditivo: Las ondas captadas por el pabellón auditivo son conducidas hacia el tímpano mediante el conducto auditivo. La longitud de este conducto es de dos a tres centímetros. La forma curvada en su extremo permite proteger el tímpano de, por ejemplo, insectos o impurezas que pudieran llegar hasta él. Además, el conducto auditivo se encuentra recubierto de cera (cerumen), que contribuye a impedir la entrada de materias perjudiciales como polvo o suciedad.

Sin embargo, la protección del tímpano no es su función principal. El conducto auditivo amplifica los sonidos bajos y menos penetrantes de la voz humana, actuando como un **audífono**. De este modo, el oído subsana parte de la debilidad de la señal de voz, facilitando la escucha y comprensión de una conversación normal.

- ▶ Tímpano: La membrana timpánica o tímpano es una fina membrana situada en el extremo final del conducto auditivo. Supone el final del oído externo y el comienzo del oído medio. El tímpano posee una **gran sensibilidad**, de forma que las ondas de sonido que penetran en el oído hacen que vibre. Estas vibraciones, debidas a la

presión de la señal entrante, son transmitidas hacia el interior del oído para su posterior transformación y procesado.

Oído medio

El oído medio es la parte del oído comprendida entre el tímpano y la ventana oval. Lleva a cabo la **transmisión de señales de sonido** desde el oído externo hasta el oído interno. Se compone de tres huesos conectados entre sí: martillo, yunque y estribo; además de la ventana oval, la ventana redonda y la trompa de Eustaquio.

- ▶ Los huesos del oído medio: las vibraciones del tímpano se transmiten al interior del oído a través de los tres huesos del oído medio: **martillo, yunque y estribo**. Estos transmiten las vibraciones del tímpano como consecuencia de la onda sonora entrante hacia el interior del oído. El estribo es el último de esta cadena de huesos y está conectado con la ventana oval, donde tiene lugar la siguiente etapa de percepción del sonido.
- ▶ La ventana oval: es una fina membrana que recubre el extremo de la cóclea expuesto hacia los huesos del oído medio. Actúa como un **amplificador acústico** previo al oído interno para las señales sonoras transmitidas desde el tímpano a la ventana oval. De hecho, se estima que la presión de la señal sonora es aproximadamente veinte veces superior en la ventana oval respecto al tímpano. Este aumento se debe al mayor tamaño de la superficie de la membrana timpánica en comparación con la de la ventana oval.

Una misma fuerza, en este caso, ejercida por la perturbación sonora sobre las partículas del medio, conlleva un aumento de presión cuando se aplica sobre una superficie menor.

- ▶ La trompa de Eustaquio: es parte del oído medio y supone una **vía de comunicación entre el oído y la parte posterior del paladar**. Es la encargada de equiparar el nivel de presión entre las dos caras de la membrana timpánica,

igualando la presión del aire en el interior y exterior del oído. Este conducto se abre cuando realizamos la acción de tragarse, permitiendo la entrada de aire.

Si en el oído se acumula, es decir, el interior del tímpano está sometido a una mayor presión del aire respecto al exterior, se impide que esta membrana vibre adecuadamente. Como consecuencia, se reduce la capacidad auditiva. En escenarios en los que la presión ambiental cambia con rapidez es habitual padecer esta diferencia de presión entre el oído y el exterior. Este es el caso de los vuelos en avión o la travesía por montañas elevadas.

Oído interno

Es la última parte del oído y supone la conexión de este con el cerebro; contiene la cóclea como elemento principal.

- ▶ La cóclea: esta lleva a cabo la transformación de las ondas sonoras en **impulsos eléctricos capaces** de ser transmitidos al cerebro a través de las conexiones nerviosas. Posteriormente, el cerebro traduce estos impulsos en estímulos que podemos reconocer y entender.
 - La cóclea, con forma de concha de caracol, contiene dos membranas cercanas entre sí y está rellena por un fluido conocido como perilinfa. Estas membranas forman la membrana basilar, una especie de pared de separación en la cóclea.
 - La membrana basilar tiene un pequeño orificio, helicotrema, que permite el movimiento del fluido de un lado al otro de la membrana basilar. El helicotrema garantiza que las vibraciones de la ventana oval se transmitan a todo el fluido que rellena la cóclea.
 - A su vez, el interior de la membrana contiene miles de microscópicas fibras pilosas (24 000 fibras aproximadamente). Estas se mueven como consecuencia del movimiento del fluido en el interior de la cóclea. Existen diferentes tipos de fibras, todas ellas conectadas al nervio auditivo. En función de la naturaleza de los movimientos de la perilinfa, diferentes tipos de fibras se ponen en movimiento y generan impulsos eléctricos en el nervio auditivo.

- El nervio auditivo es el canal de comunicación que conecta la cóclea con el cerebro. Cuando las células pilosas que se mencionaban previamente se mueven, el nervio transmite la señal eléctrica resultante hasta la zona del cerebro responsable de la audición. Allí, los sonidos captados se interpretan y asimilan. Por tanto, el rol de estas células pilosas es esencial para nuestra capacidad de audición. Si estas fibras resultaran dañadas, nuestra capacidad auditiva se vería severamente deteriorada.

El oído humano es capaz de captar ondas sonoras en el rango entre los 20 Hz y los 20 KHz de frecuencia. Las ondas por debajo y por encima de estos límites se denominan infrasonidos y ultrasonidos, respectivamente.

Los sonidos pueden distinguirse gracias a las propiedades de resonancia de la **membrana basilar**, que permite identificar ondas sonoras de diferente frecuencia. Esta membrana tiene propiedades mecánicas que varían desde su base hasta el vértice:

- La base es más fina, tiene menor masa y elasticidad.
- En el vértice, al contrario, tanto la masa como la elasticidad de la membrana son mayores. Así, la amplitud del movimiento de la membrana ante la excitación producida por una onda sonora está más cerca de la base para frecuencias de onda mayores.
- Esta amplitud de pico se encuentra más próxima al vértice conforme disminuye la frecuencia de la onda entrante. Por tanto, esta membrana basilar actúa como un banco de filtros paso-banda que dotan al oído de su capacidad selectiva en frecuencia.

Características físicas del oído humano

Por último, el oído está caracterizado por diferentes umbrales de funcionamiento. El **umbral de audibilidad** es el nivel de intensidad mínimo en espacio libre que el oído es capaz de oír. La máxima sensibilidad se tiene a 4 KHz.

A 120 dB, respecto al umbral de referencia, tenemos el **umbral de sensación**. A este nivel de intensidad se produce cierto cosquilleo en el oído. A partir de 140 dB se puede llegar a producir dolor.

Una de las ondas de mayor interés es la propia voz humana. Físicamente, se caracteriza por un rango frecuencial entre 300 Hz y 3.4 KHz. Por tanto, el ancho de banda asignado en telecomunicaciones para un solo canal de transmisión de voz suele ser de 4 kHz. Como se verá más adelante en la asignatura, este ancho de banda permite utilizar una frecuencia de muestreo de 8 kHz como base del sistema de modulación digital.

1.4. Percepción visual



Accede al vídeo «Percepción visual» a través del aula virtual

Funcionamiento del sistema visual humano

La **visión** es el fenómeno resultante de la percepción del color, forma y distancia de los objetos en un espacio. La visión se produce como resultado de la incidencia de la luz, caracterizada como una onda electromagnética, sobre la retina del ojo.

Esta onda debe estar comprendida en un rango frecuencial (luz visible) para producir impresión visual. El color percibido depende de la frecuencia (longitud de onda) de la luz que incide sobre el objeto observado. Asimismo, el estado del propio ojo también influye en la percepción de los colores, como ocurre en el caso del daltonismo.

El comportamiento del ojo humano se asemeja al de una cámara fotográfica. La luz reflejada en los objetos se proyecta sobre un material fotosensible (**retina**) tras pasar a través de una lente (**cristalino**), tal y como se esquematiza en la siguiente imagen.

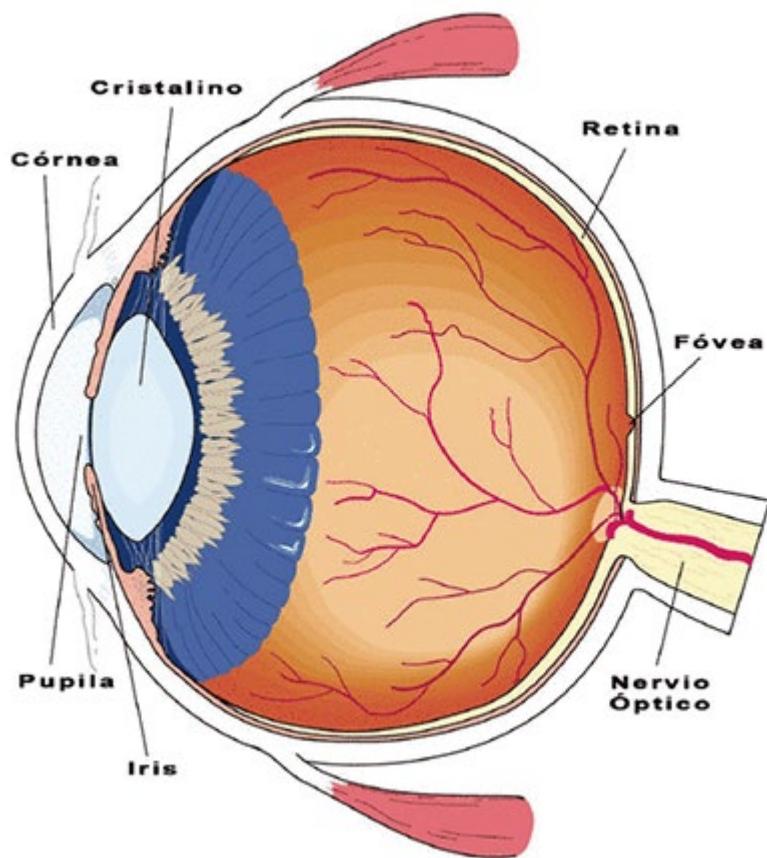


Figura 2. Diagrama simplificado de un corte de ojo humano.

Fuente: <http://www.lucescei.com/estudios-y-eficiencia/extractos-libro-blanco-de-iluminacion/el-sistema-visual-humano/>

La zona visible del cristalino es la **pupila**, que regula el flujo de luz que entra en el ojo. El cristalino refracta los rayos de luz para que estos se focalicen en la zona adecuada de la retina. Gracias al cambio de tamaño de la pupila (cristalino), permite enfocar con precisión objetos cercanos y lejanos. Es decir, se adapta al entorno para producir el resultado de una visión satisfactoria.

La **retina** es una membrana interior y en ella se encuentran las células fotosensibles. Estas células son de dos tipos: conos y bastones.

Conos	Son menos numerosos y muy poco sensibles a la luz. Son las células encargadas de la visión diurna .
Bastones	Precisan de menor cantidad de luz que los conos para su excitación. Por tanto, son las células sensoriales responsables de la visión nocturna .

Figura 3. Tipos de células fotosensibles.

La siguiente gráfica muestra la distribución espacial de estas células en la retina, tomando como referencia el punto central de la retina o **fóvea**. Como puede apreciarse, los bastones se encuentran lejos de la fóvea, principalmente. Mientras, los conos se concentran en torno a esta.

Además, se aprecia la existencia de un **punto ciego en la retina**, carente de células fotosensibles, y que es el lugar desde el que parten los nervios ópticos hacia el cerebro.

La fóvea recibe la luz desde el centro del campo visual, es decir, desde el punto del espacio al que miramos directamente. Los conos, principalmente concentrados en esta zona, nos captan la luz procedente del punto focalizado y son responsables de la visión directa o central. En cambio, los bastones, más alejados de la fóvea, proporcionan visión periférica.

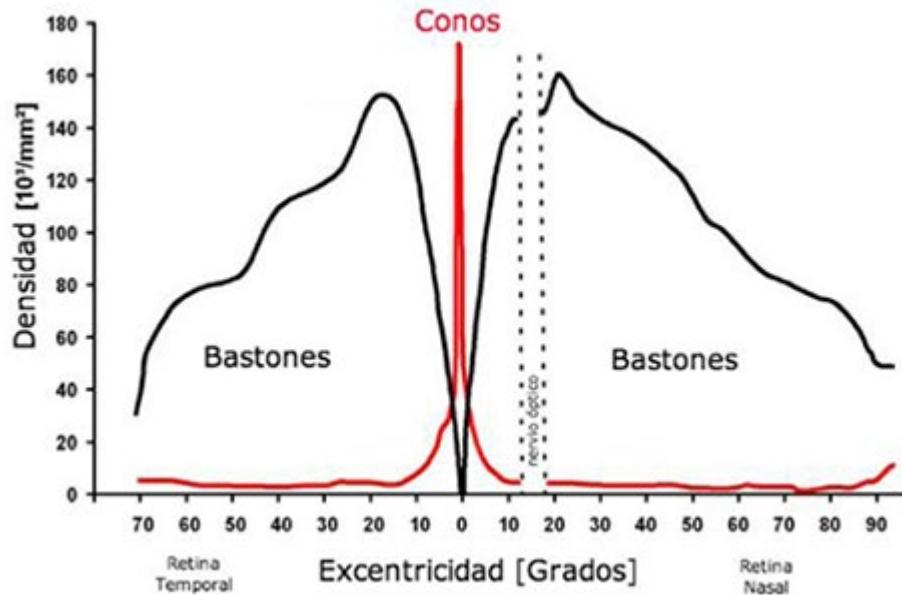


Figura 4. Distribución espacial de conos y bastones en la retina del ojo humano.

Fuente: <https://idus.us.es/xmlui/>

Conos

Los conos actúan con filtros sobre la luz incidente en ellos. Así, los humanos tienen **tres tipos** de conos:

- ▶ El primero responde más a la luz de longitudes de onda largas, alcanzando un máximo a aproximadamente 560 nm. Este tipo de cono se designa mediante la letra **L (long)**.
- ▶ El segundo tipo de cono filtra la luz de longitud de onda media, alcanzando un máximo a 530 nm, y se abrevia mediante la letra **M (medium)**.
- ▶ El tercer tipo responde más a la luz de longitud de onda corta, alcanzando un máximo de 420 nm, y se designa mediante **S (short)**.

Estas longitudes de onda (L, M y S) se corresponden de forma aproximada con los colores **rojo, verde y azul** respectivamente. En el siguiente gráfico vemos la respuesta en frecuencia de los tres tipos de células cono a la luz. Los diferentes tipos de conos permiten percibir distintos colores. Estos se forman a partir de la diferente excitación de los tres tipos de células receptoras.

El color amarillo, por ejemplo, se percibe cuando los conos L se estimulan ligeramente más que los conos M, y el color rojo se percibe cuando los conos L se estimulan significativamente más que los conos M. De manera similar, los tonos azules y violetas se perciben cuando el receptor S se estimula más. Así, mediante la mezcla ponderada de los colores rojo, azul y verde somos capaces de construir la gama cromática perceptible por el ser humano.

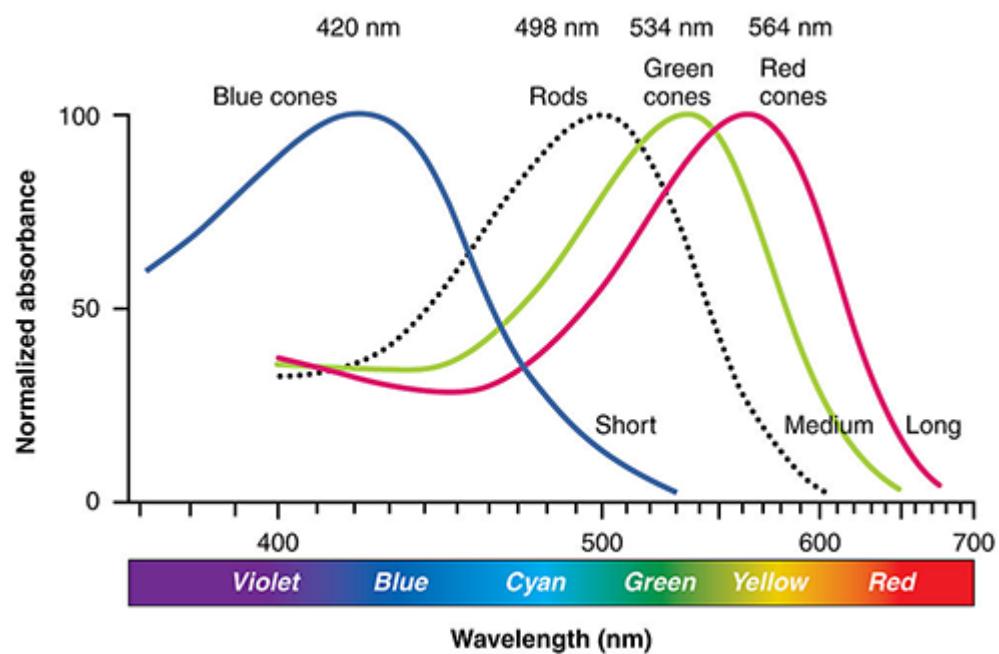


Figura 5. Respuesta en frecuencia de los tres tipos de conos: S (longitud de onda corta, azul), M (longitud de onda media, verde) y L (longitud de onda larga, rojo).

Fuente: https://commons.wikimedia.org/wiki/File:1416_Color_Sensitivity.jpg.

Bastones

Por otro lado, observamos que los bastones **no son sensibles al color** (frecuencia), dado que solo hay un tipo de bastón. Así es que estas células no permiten la formación de colores. Este es el motivo por el que en la oscuridad no somos capaces de distinguir ningún color.

Fenómenos visuales: respuesta logarítmica e inhibición lateral

Respuesta logarítmica

El concepto de *just noticeable difference* (JND) representa, en el ámbito de la psicofisiología, la mínima cantidad de variación ΔI en la magnitud de un estímulo I para que esta sea apreciada.

Por ejemplo, se observa como la siguiente imagen contiene dos áreas de intensidades diferentes. En este caso, dada la clara apreciación de ambas áreas, el incremento de intensidad aplicado sobre el área exterior es mayor a aquel dado por la JND.

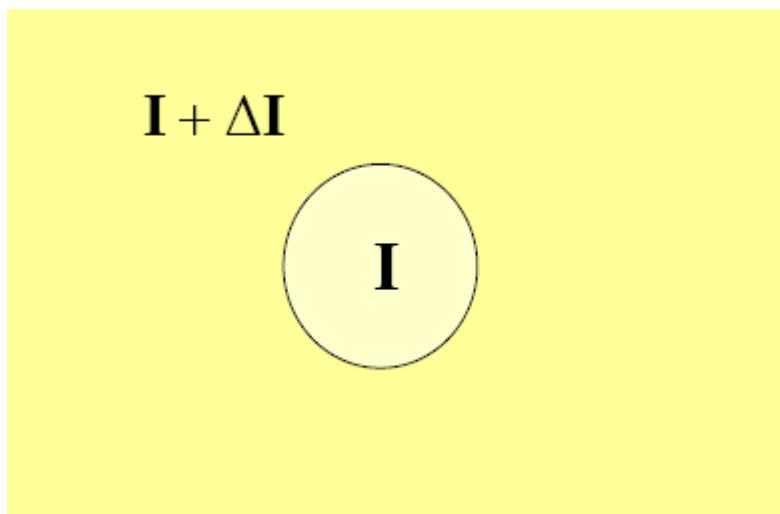


Figura 6. Ilustración del experimento llevado a cabo para la determinación de la JND en una imagen.

Fuente: www.tel.uva.es.

En el siglo XIX, el psicólogo Ernst Weber determinó que existe una relación constante entre ΔI e I en un amplio rango de I . Es decir, la mínima variación necesaria para percibir un cambio en un estímulo se incrementa conforme el estímulo se hace más intenso. Matemáticamente, esta **Ley de Weber** se establece de la siguiente forma:

$$\Delta I/I = \lambda$$

En el marco de la percepción visual, en el que I representaría la cantidad de luz o intensidad percibida, conforme dicha intensidad es mayor, es preciso una variación ΔI más significativa de la misma para que sea apreciable. Por tanto, se concluye que esta **JND es diferente en zonas claras u oscuras de una imagen.**

Así, consideremos:

- ▶ Una imagen Im_0 oscura y su JND asociada, representada por ΔI_0 .
- ▶ Una imagen más clara, Im_1 con una JND representada por ΔI_1 .

De acuerdo a la Ley de Weber, podemos comparar ambas JND y deducir que:

$$\Delta I_0 < \Delta I_1$$

Es decir, los cambios de intensidad en una imagen se aprecian mejor en zonas oscuras. Por tanto, estas zonas serán más sensibles a la presencia de ruido o artefactos de naturaleza aleatoria.

La Ley de Weber muestra, por tanto, que el incremento necesario en la intensidad del estímulo para resultar en una respuesta perceptiva similar aumenta para mayores valores de intensidad. Además, la relación entre la percepción del estímulo y la intensidad de este sigue una relación logarítmica.

Obsérvese el siguiente gráfico. Se ilustra cómo para obtener un incremento similar en la respuesta del ojo humano (en la percepción de la intensidad luminosa recibida), es necesario un mayor incremento de la intensidad de partida para intensidades mayores.

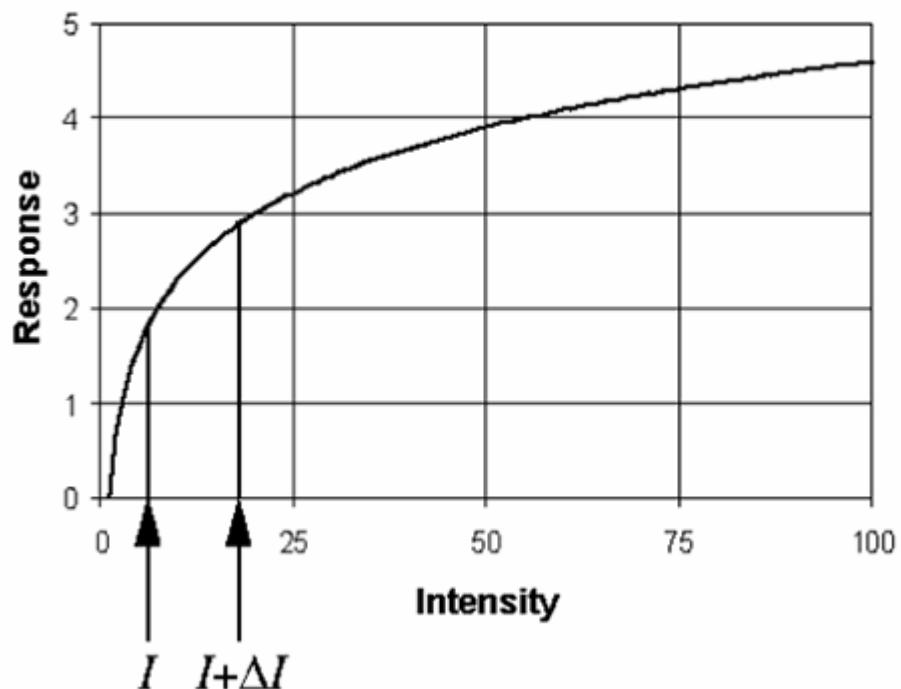


Figura 7. Respuesta logarítmica de la percepción de intensidad de acuerdo a la Ley de Weber.

Fuente: <https://sites.google.com/site/udctopsicologia1/tema-01-introduccion-a-la-percepcion>.

La demostración matemática de este comportamiento puede obtenerse de la siguiente forma. Para una intensidad I , tenemos que el valor de la intensidad mínimamente apreciable es el siguiente:

$$I_{JND} = I + \Delta I = I + \lambda I = (1 + \lambda)I$$

Por tanto, la diferencia entre I_{JND} e I viene dada por:

$$I_{JND} - I = \lambda I$$

Si calculamos esta diferencia en el dominio logarítmico, tenemos la siguiente relación:

$$c_{JND} = \log(1 + \lambda) + \log I$$

$$c = \log I$$

$$c_{JND} - c = \log(1 + \lambda)$$

Es decir, en el dominio logarítmico, los incrementos de intensidad para JND son constantes.

Inhibición lateral

Además de esta transformación logarítmica, el sistema visual humano lleva a cabo un filtrado espacial de la señal luminosa recibida que, como resultado, conlleva el **realce del contraste entre zonas de diferente intensidad**. Las áreas de la imagen donde la intensidad luminosa cambia bruscamente de claro a oscuro, o viceversa, denotan la presencia de bordes.

La conexión de conos y bastones con las células de la retina permite capturar y realizar estos cambios. Tanto conos como bastones están conectados con dos tipos de células (en la segunda y tercera capa de la retina, respectivamente). Estas llevan a cabo un procesado de la señal visual equivalente al que produce el operador laplaciano (diferencial de segundo orden que magnifica las zonas de la señal en los que se aprecian cambios bruscos). Concretamente, obviando la zona central de la retina:

- ▶ Las señales de una gran proporción de conos y bastones vecinos llegan a las células de la tercera capa de la retina, constituyéndose un **agregado** (signo +) de la intensidad recibida.
- ▶ Mientras, las células de la segunda capa retiniana son responsables del envío de **impulsos inhibitorios** (signo -) equivalentes a la substracción del número de impulsos bioeléctricos de otros puntos próximos de la imagen (píxeles).

Este fenómeno constituye la **inhibición lateral**, que, con un comportamiento similar a un filtro de frecuencias altas, nos ayuda a percibir el contraste, facilitando la posterior identificación de límites o fronteras entre regiones de diferente intensidad, así como de contornos o bordes. Este fenómeno fue descrito por Mach, tal y como refleja el experimento mediante bandas de diferente intensidad.

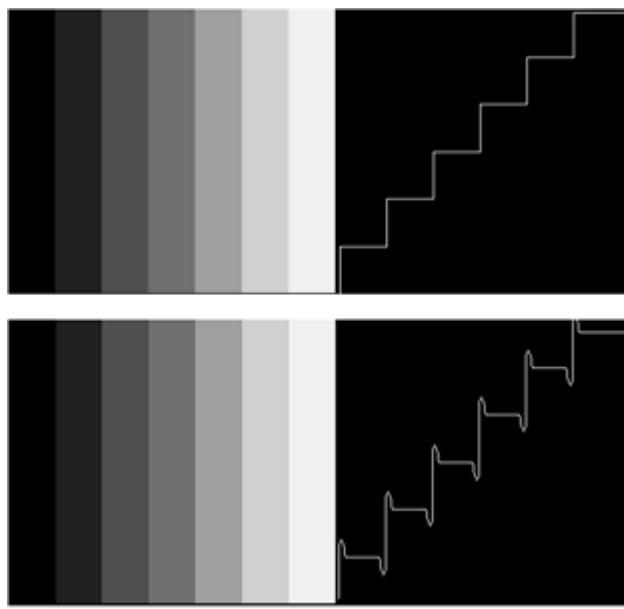


Figura 8. Bandas de Mach y efecto resultante del filtrado espacial llevado a cabo por el ojo humano.

Fuente: <http://www.ub.edu/pa1/node/99>.

Muestreo o filtrado temporal

Por último, el ojo humano también realiza un filtrado temporal de la señal capturada. De hecho, su respuesta temporal es relativamente lenta. Considérese, por ejemplo, el caso en el que se tiene una única fuente de luz que se enciende de forma intermitente. Si la frecuencia de la fuente es baja, se aprecian los períodos en los que no se emite luz, el tiempo entre emisiones de luz consecutivas ha de ser superior a 30 ms. Sin embargo, para frecuencias superiores, no se percibirán los períodos sin emisión de luz, dando la apariencia de una luz continua.

La frecuencia a la que no se percibe la intermitencia de la fuente de luz se denomina **frecuencia de fusión**. Está en torno a 30 Hz, dependiendo del tamaño y del brillo de la fuente. La aplicación práctica de este fenómeno es la **definición de estándares de codificación de vídeo**. Estos no recogen imágenes de forma continua (señal continua), sino que definen la tasa de imágenes necesaria por segundo (frecuencia de muestreo) para que el espectador no perciba las discontinuidades existentes. Así, los antiguos sistemas PAL y NTSC definían una tasa 25 y 30 imágenes por segundo, respectivamente.

Otro ejemplo que refleja la respuesta temporal del ojo humano es el de dos fuentes de luz intermitentes, alternantes (cada una de ellas emitiendo durante 1 ms) y con una separación angular de 1 grado respecto del observador.

- ▶ Si la frecuencia de la intermitencia es superior a 1 s, se perciben como fuentes de iluminación que se alternan.
- ▶ Si la frecuencia se encuentra en torno a 10 ms, la sensación resultante es la de un punto emisor de luz que se traslada desde una fuente a otra.
- ▶ Por último, si la diferencia temporal de encendidos es de aproximadamente 1 ms, la percepción será una iluminación simultánea.

El experimento demuestra que, por tanto, la visión humana está caracterizada por una **frecuencia de rendición del movimiento**. Esta permite crear sensación continua de movimiento a partir de un conjunto de instantáneas. Pero es necesario que la frecuencia a la que se presentan estas instantáneas sea suficientemente alta para que la percepción no refleje discontinuidades.

A partir de los mecanismos descritos, el proceso de captación de imágenes por parte del sistema de visión humana puede esquematizarse en los siguientes pasos:



Figura 9. Representación esquemática de las etapas de procesado realizadas por el sistema visual humano en la percepción de imágenes.

- ▶ Filtrado en frecuencia para seleccionar la parte de la radiación luminosa correspondiente al espectro de **luz visible**.
- ▶ Transformación logarítmica (**Ley de Weber**) del estímulo percibido.
- ▶ Filtrado espacial (realce de bordes y fronteras) de acuerdo al mecanismo de **inhibición lateral**.

- ▶ Filtrado temporal (**muestreo de la señal**) que se refleja en la frecuencia crítica de fusión y la frecuencia de rendición del movimiento.

Síntesis de color

Por último, comentaremos en este tema algunas nociones sobre el color. El color de un objeto viene definido por el **contenido espectral de una determinada radiación**, esto es, por $R(f)$. Así, la variación de color de una señal luminosa respecto a otra está asociada a la diferente frecuencia de ambas señales.

Sin embargo, se puede tener dos radiaciones distintas, $R_1(f)$ y $R_2(f)$; con espectros distintos: $R_1(f) \neq R_2(f)$, pero que se traduzcan en la misma percepción de color. Este resultado se debe a que el color percibido es **función de tres canales no independientes**. Como se ha comentado previamente, existen tres tipos de conos, las células fotorreceptoras responsables de la percepción del color. Y cada uno de ellos actúa como un filtro en frecuencia $S_i(f)$ que selecciona parte de los colores de la radiación entrante.

Matemáticamente, la **potencia de la señal de salida** de cada uno de estos canales α_i puede expresarse de la siguiente forma:

$$\alpha_i(R) = \int_{f_{min}}^{f_{max}} R(f)S_i(f)df$$

$$i = 1, 2, 3$$

Por tanto, dos colores $R_1(f)$ y $R_2(f)$ se percibirán de la misma forma si:

$$\alpha_i[R_1(f)] = \alpha_i[R_2(f)] \text{ para } i = 1, 2, 3$$

Así, un determinado color $R(f)$ puede sintetizarse a partir de la superposición de tres primarios $P_k(f)$ encontrando los coeficientes β_k apropiados en la mezcla, tal y como se esquematiza en la siguiente figura.

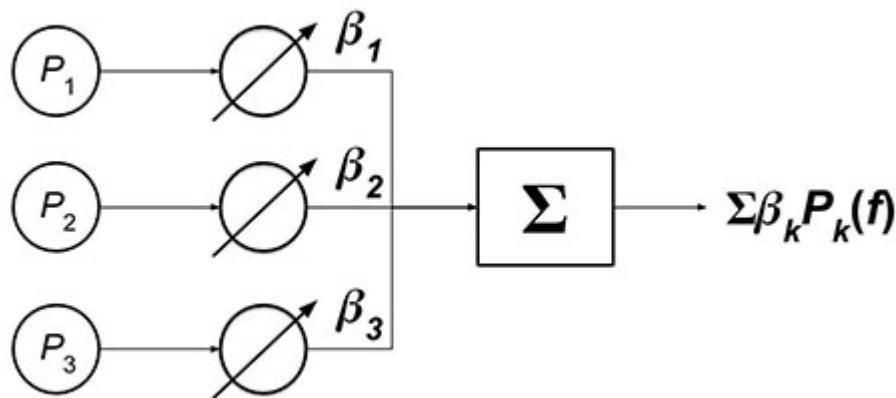


Figura 10. Esquema del proceso de síntesis de color.

Para que el resultado de la síntesis produzca la sensación de color esperada, habrá de cumplirse lo siguiente:

$$\begin{aligned}
 \alpha_i(R) &= \int_{f_{min}}^{f_{max}} R(f) S_i(f) df = \int_{f_{min}}^{f_{max}} \left[\sum_{k=1}^3 \beta_k P_k(f) \right] S_i(f) df \\
 &= \sum_{k=1}^3 \beta_k \int_{f_{min}}^{f_{max}} P_k(f) S_i(f) df = \sum_{k=1}^3 \alpha_{ik} \beta_k
 \end{aligned}$$

Donde α_{ik} es la respuesta de los conos de tipo i al color primario $P_k(f)$. Este viene dado por la siguiente expresión:

$$\alpha_{ik} = \int_{f_{min}}^{f_{max}} P_k(f) S_i(f) df$$

Por tanto, la igualdad anterior demuestra que los coeficientes de mezcla β_k vienen dados por la solución a un sistema de tres ecuaciones y tres incógnitas, dada la respuesta de los tres filtros de conos a los tres colores primarios inicialmente considerados $P_k(f)$.

Por ejemplo, uno de los **sistemas de color** más relevantes es el **RGB** (*red, green, blue*), comúnmente utilizado en pantallas, y que toma el rojo, el verde y el azul como colores primarios de la mezcla.



Accede a los ejercicios de autoevaluación a través del aula virtual

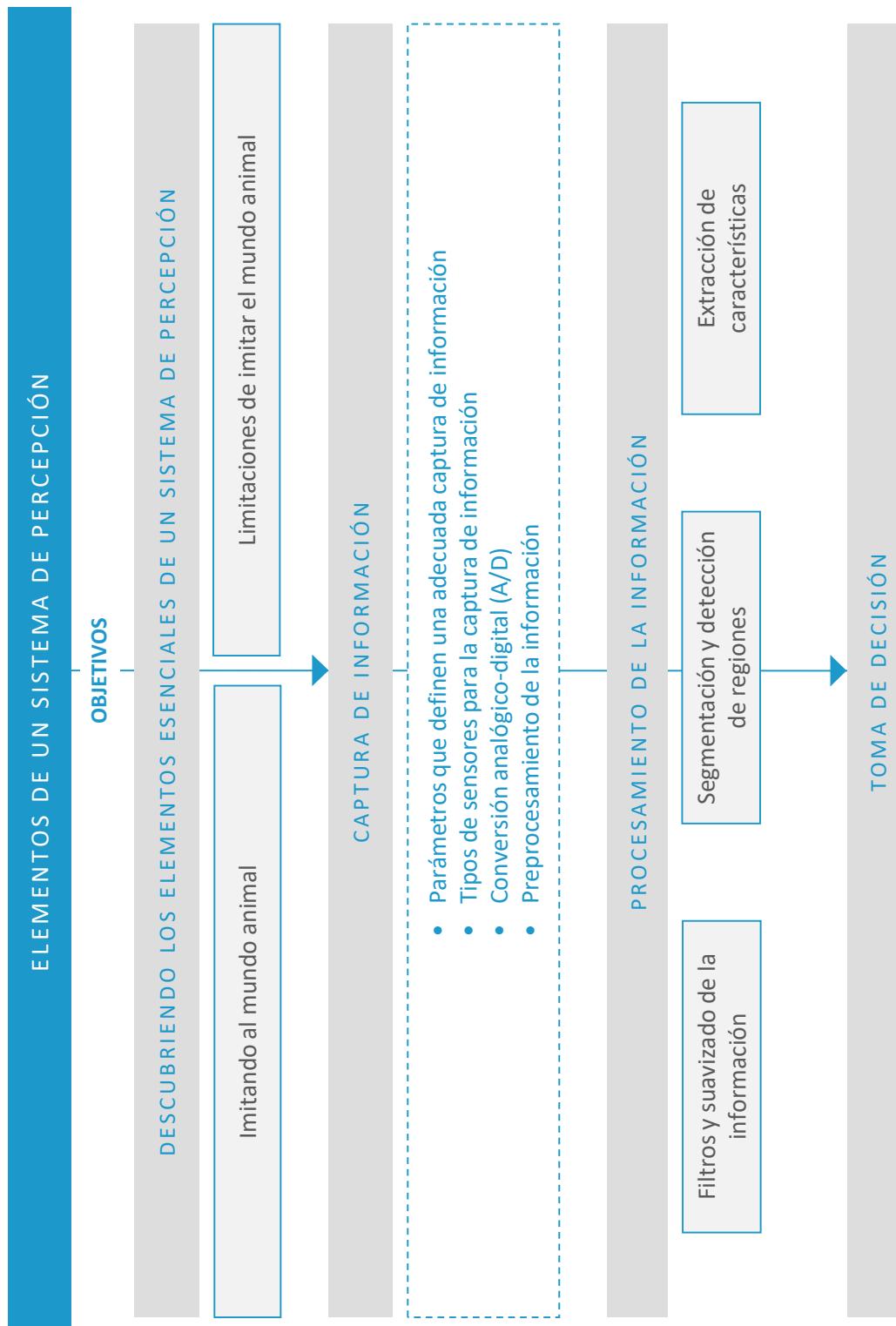


Accede al vídeo «Resumen del tema» a través del aula virtual

Percepción Computacional

Elementos de un sistema de percepción

Esquema. Tema 2



Ideas clave. Tema 2

2.1. ¿Cómo estudiar este tema?

Para estudiar este tema deberás leer con atención las ideas clave que se desarrollan a continuación.

2.2. Objetivos

El objetivo principal de este tema es entender los elementos más importantes de un sistema de percepción y cómo, desde el punto de vista computacional, se desarrolla cada módulo de dicho sistema.

En definitiva, se busca dar una visión general de aquellos ingredientes necesarios para que un sistema de percepción computacional tenga sentido y pueda resolver problemas reales.

En este tema no se busca entender los métodos concretos de cada módulo que integra el sistema de percepción (puesto que esos métodos serán material para los siguientes temas).

2.3. Descubriendo los elementos esenciales de un sistema de percepción



Accede al vídeo «Introducción. Elementos esenciales de un sistema de percepción» a través del aula virtual

Imitando al mundo animal

Existen múltiples maneras de describir los sistemas de percepción. En este tema seguiremos un enfoque que irá del aspecto más genérico al más concreto. Es necesario entender que, en todo momento, los sistemas de percepción intentan imitar a la naturaleza y es, por tanto, lógico pensar que si queremos modelar adecuadamente un sistema, tenemos que empezar por considerar ciertas simplificaciones.

Por ello, sin dejar de imitar a la naturaleza, intentaremos **modelar sistemas que reflejen el comportamiento de organismos sencillos**, como puede ser el de individuos que ni siquiera poseen ojos o elementos concretos para la captura de información visual, como el caso de los moluscos. Estos seres realizan tres funciones de percepción muy sencillas:

Captura de información

- ▶ Este proceso consiste en obtener estímulos del exterior que reflejen qué actividad (movimientos, cambios de temperatura, amenazas, etc.) está teniendo lugar.
- ▶ La captura de información puede considerar múltiples estímulos principalmente físicos, mecánicos o químicos.
- ▶ La información recogida es normalmente mayor que la necesitada para entender el exterior. La adaptación de ese exceso de sensibilidad a las necesidades del entorno es parte de la propia evolución.

Procesamiento

- ▶ La información capturada necesita de un procesamiento posterior para eliminar los datos innecesarios y redundantes; entender si está siendo suficiente y, en caso contrario, redirigir la captura de información hacia otro punto del espacio o bien combinar la información proveniente de una determinada fuente con otras fuentes o con informaciones pasadas.
- ▶ Sin este procesamiento, un ser vivo podría tomar conclusiones erróneas, confundir eventos o incluso conducirlo a su extinción.
- ▶ Una de las características esenciales de este procesamiento es que debe, en todo momento, realizarse a la misma **velocidad** con la que los datos iniciales se están capturando. De esa manera, se asegura de que la información que está llegando al cerebro es únicamente la necesaria.
- ▶ Por el contrario, un animal que almacene, por ejemplo, todo lo que ha oído durante el día y lo analice durante la noche, no podrá esquivar el ataque instantáneo de un depredador.

Toma de decisión y aprendizaje

- ▶ El fin principal de la percepción del exterior no es otra que la toma de decisiones.
- ▶ Una mejor toma de decisiones (basada en más información, contexto o diversidad de fuentes) conducirá, sin lugar a dudas, a una ventaja competitiva mayor.
- ▶ Esa toma de decisiones tiene una consecuencia directa también: el aprendizaje. El ser vivo aprenderá que un determinado estímulo asociado a una determinada decisión tendrá unas consecuencias. Esas consecuencias serán almacenadas para optimizar tanto la toma de decisiones como la captura de información.
- ▶ La **percepción del exterior** es clave para el aprendizaje.

El siguiente esquema capture los puntos anteriormente vistos, enfatizando el hecho de que es un proceso iterativo y constante.



Figura 11. Esquema reducido de percepción del exterior.

Limitaciones de imitar el mundo animal

Los tres puntos anteriormente vistos (captura de información, procesamiento y toma de decisiones) proporcionan una visión muy general de lo que puede ser un sistema de percepción computacional.

Si sustituimos:

- ▶ El módulo de captura de información por una cámara de vídeo.
- ▶ El módulo de procesamiento por un filtrado muy sencillo en color, quedándonos con las componentes en azul, por ejemplo.
- ▶ Y en la toma de decisiones aplicamos una regla que diga si la imagen contiene componentes en azul en más de un 70 % del total.

Podríamos tener un detector de mares o un detector de cielos.

Es un ejemplo muy sencillo, pero muy limitado a una funcionalidad concreta. De hecho, con estos tres módulos es muy complicado hacer funcionalidades más complejas como detectar movimiento, seguir un objeto, analizar la textura de una foto o incluso analizar la forma de un objeto.

Las simplificaciones heredadas de la naturaleza nos ayudan a entender los pasos más esenciales que suceden delante de nuestros ojos, pero el adaptarlo a la resolución de un problema concreto es labor de la inteligencia humana.

Esto es lo que haremos en este tema, entender qué elementos adicionales son necesarios en el esquema de percepción del exterior para desarrollar módulos avanzados de percepción computacional y dar respuesta a los problemas que se plantean actualmente.

2.4. Captura de información



Accede al vídeo «Captura de información» a través del aula virtual

Parámetros que definen una adecuada captura de información

En este apartado nos centraremos en todo lo relativo a la captura de información del exterior. Existen numerosos parámetros que definen cómo es dicha la captura y puede discutirse cuáles son los más importantes. Sin embargo, los más esenciales son los siguientes:

Especificidad

La capacidad de un sistema de captura de información para recoger fielmente los eventos sucedidos es clave. Un ejemplo de especificidad es la **temperatura**.

Un determinado sensor de vídeo, como puede ser una cámara, puede estar en el exterior capturando imágenes de tráfico. Al estar en la calle, dicho dispositivo posee una determinada temperatura. Sin embargo, a pesar de que está sufriendo cambios de temperatura, no puede determinarla ya que no tiene la sensibilidad necesaria para

medirla. Hace falta, claramente, un termómetro que sea sensible a esos cambios. Los sensores han de ser creados específicamente para un propósito.

Precisión

El **error de medida** de un determinado evento es también importante. Continuando con el ejemplo del termómetro, la precisión indicará cómo es el error de la medida proporcionada por el dispositivo. Si el termómetro solo puede indicar frío o calor, tendrá baja precisión. Si es capaz de marcar décimas de grado, será un termómetro muy preciso.

Sensibilidad

Esta propiedad refleja la capacidad que tiene un determinado sensor de recoger las fluctuaciones o cambios del evento que se mide. Si la luz en un determinado punto varía hasta treinta veces al día, pero la cámara no es capaz de detectar más que dos de esos cambios, es porque la cámara no tiene la suficiente sensibilidad.

Es importante indicar que **sensibilidad y precisión van de la mano**, pero la precisión está más orientada a ver cómo difiere la realidad de lo que se ha medido, y la sensibilidad está más orientada a evaluar cómo se adapta a los cambios de lo que se mide.

Consumo y tamaño

La mayoría de dispositivos de captura de información necesitan de una fuente de alimentación para funcionar. En función de la sensibilidad y la precisión (y otros factores como la cantidad de información capturada) requerirá un consumo de potencia eléctrica mayor o menor y poseerá un tamaño diferente.

Es necesario considerar el consumo y el tamaño en el diseño de los sistemas de percepción computacional pues suele ser cierto que consumo y tamaño son **inversamente proporcionales a sensibilidad y precisión**.

Existen algunos factores más como pueden ser:

- ▶ El precio del sensor.
- ▶ La usabilidad.
- ▶ La resistencia a agentes externos: si un sensor de temperatura es resistente a la lluvia o una cámara lo es a temperaturas extremas.
- ▶ Amplitud en el rango de la medida.
- ▶ La reparabilidad: si es fácilmente reparable...

Tipos de sensores para la captura de información

Existe una gran cantidad de tipología de sensores. Esto es fruto de que el mercado está dando respuesta a la gran demanda de **especificidad** que se ha visto anteriormente (es decir, cada sensor se está haciendo cada vez más específico) e intenta resolver una casuística concreta.

A continuación presentamos una clasificación genérica para que tengas una visión panorámica de los sensores que existen a día de hoy:



Figura 12. Tipos de sensores que puedes encontrar en el mercado.

Conversión analógico-digital (A/D)

La conversión analógico-digital es uno de los pasos esenciales para poder procesar cualquier información. De por sí, dicha conversión daría para una asignatura en sí misma (incluso para un curso académico), no obstante, en este apartado resumiremos dos puntos:

- ▶ En qué consiste la conversión analógico-digital y cómo funciona.
- ▶ Qué tipos de ventajas aporta dicha conversión.

Conversión

En primer lugar, lo que hace es **tomar fotografías de la realidad que se está midiendo**. Es decir, si estamos midiendo temperaturas, la conversión analógico-digital no estaría midiendo constantemente la temperatura en sí, sino que la mediría a intervalos de tiempos. El valor tomado en cada intervalo, además, se discretizaría, entendiendo por «discretizar» el reducirlo a un formato concreto de datos.

Pongamos un ejemplo más concreto y aplicable a un caso real de nuestro día a día. Estamos en un centro comercial y escuchamos la música que nos gusta por los

altavoces. Queremos guardar esa canción en nuestro móvil y utilizamos la aplicación micrófono para guardar el sonido del exterior.

- ▶ Nuestro móvil tiene un sensor de sonido: un micrófono.
- ▶ Posteriormente un conversor analógico-digital. Dicho conversor A/D muestrea la señal de audio: toma fotos de ella a periodos concretos.
- ▶ Convierte cada medida (que es una intensidad) en un dígito binario (por ejemplo, 2.1 voltios).

Esa conversión de intensidades físicas a dígitos binarios es lo que se conoce como conversor A/D.

Tras dicha conversión, la información almacenada (las fotografías de la señal de audio) ocupa una cantidad mucho menor que el sonido original y permite la realización de un **pre-procesado** (se verá más en detalle en qué consiste esta fase) que, entre otras tareas, filtra la señal del ruido existente y la protege frente a posibles errores; todo esto mediante software sin la necesidad de dispositivos hardware adicionales.

En este gráfico se aprecia cómo la intensidad de una determinada señal a lo largo del tiempo (valores enteros como 3, 5, 7, 7, 7, etc.) se convierte en formato digital/binario.

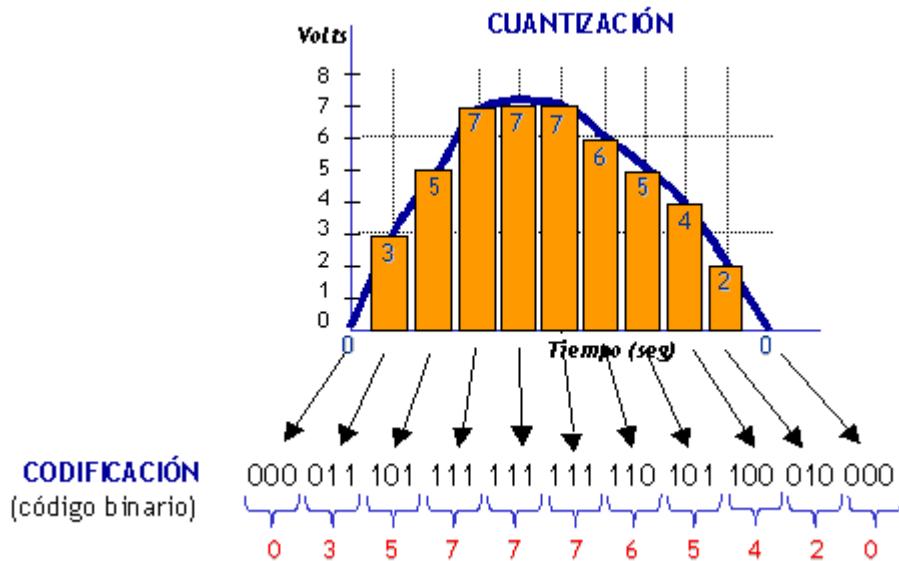


Figura 13. Ejemplo de conversión analógico-digital.

Fuente: <http://panamahitek.com/escalando-unidades-de-conversion-analogica-digitales/>.

Ventajas

Las ventajas de un conversor A/D son numerosas:

- ▶ Reduce enormemente la lógica de procesamiento lo que le permite ser configurable por medio de software.
- ▶ Permite la integración de multitud de sensores en dispositivos corrientes sin necesidad de una integración compleja.
- ▶ La información almacenada es infinitamente menor, lo que ahorra también espacio de almacenamiento.
- ▶ Se puede realizar una transmisión de una información determinada por radio o infrarrojo tras usar un conversor A/D sin necesitar gran ancho de banda ni perder información.
- ▶ Permite y facilita la compresión de información, puesto que la información ha sido convertida a formato digital.

¿Existen desventajas? Por supuesto. La principal es la **pérdida de información**, ya que la señal se está muestreando.

Para solventar esta desventaja, el **teorema de Nyquist-Shannon** propone la frecuencia mínima para muestrear una determinada señal. Este teorema queda fuera del alcance de este temario, sin embargo, hacemos referencia a la existencia de dicha frecuencia mínima dada su relevancia.

Preprocesamiento de la información

Antes de empezar con el apartado dedicado al procesamiento de la información capturada, es necesario añadir una sección dedicada al preprocesamiento. Esta fase podría tener una entidad propia *per se*. De hecho, algunos de los temas de esta asignatura están centrados en el preprocesamiento.

En muchos libros, además, preprocesamiento y procesamiento son sinónimos, siendo en algunos casos uno sustitutivo de otro.

En esta asignatura entenderemos el preprocesamiento como el tratamiento realizado inmediatamente después de la captura de la información y que será común para cualquier procesado posterior que se haga.

Por ejemplo, imaginemos un sistema de percepción computacional que se encarga de seguir un objeto en pantalla y detectar su forma. En este caso, son dos tareas las que se realizan para una misma captura de información: vídeo e imagen.

El preprocesamiento del vídeo y de la imagen consistirá en eliminar ruido en la imagen; asegurar que la iluminación es adecuada; eliminar componentes borrosos del vídeo o incluso en equilibrar las componentes de color.

Por otro lado, el **procesamiento** de dicha captura será doble:

- ▶ Un procesamiento encargado de facilitar la detección de objetos (eliminando detalles del fondo, seleccionando los objetos principales, etc.).
- ▶ Otro procesamiento encargado de analizar la forma y contorno de los objetos.

¿Podrían incluirse las fases de preprocesamiento dentro del procesamiento? Por supuesto, pero entonces habría que realizarla dos veces, una para cada tipo de procesamiento.

**Por tanto, el criterio de esta asignatura es definir el preprocesamiento como
el conjunto de tareas que acomodan la captura de información al
procesamiento de la misma.**

Adicionalmente, el preprocesamiento puede realizarse, bien sobre la información en analógico, o bien después de la conversión a digital.

- ▶ En el caso de realizarlo sobre la información en **analógico**, suele tratarse de un preprocesamiento que involucra hardware como filtros basados en circuitos, mejores sensores, etc.
- ▶ En el caso de realizar el preprocesamiento sobre la información ya **digitalizada** (discretizada), este será eminentemente realizado por software o código.

En general, el preprocesamiento realizado de forma digital suele ser más potente, pero trabaja sobre información ya deteriorada no solo por la captura de la señal, sino también por la propia conversión A/D que ha propagado el error de medida.

Por el contrario, el preprocesamiento realizado de forma analógica suele ser muy eficaz, pero menos flexible y poco configurable. Es decir, al ser muy específico, no puede adaptarse a variaciones en los errores observados.

En general, puede decirse que una combinación de ambos es lo más adecuado, aunque actualmente el avance de la tecnología en sensorización y en computación han desplazado la balanza hacia el preprocesamiento realizado de forma digital, ya que es el elegido en numerosos casos por coste, eficiencia y flexibilidad, dado que los sensores proporcionan una medida de la realidad muy precisa.

Las tareas más comunes de preprocesamiento suelen cubrir los siguientes aspectos:

Eliminación de ruido

El concepto de ruido es muy amplio y genérico, pero puede entenderse como **pérdidas** en la calidad de la información o bien **alteraciones** de la información. El ruido más común es el que se produce en una televisión, donde la imagen no es nítida y se aprecia como algunos píxeles de la pantalla no concuerdan con el contexto de la imagen.

Actualmente, la mayoría de los sensores ya poseen integrados mecanismos de reducción de ruido, pero en muchos casos, es necesario una eliminación de ruido adecuada para asegurar que la señal preprocesada cumple con los requisitos correctos para una posterior extracción de características y toma de decisiones.



Figura 14. Ejemplos de ruido en una determinada imagen.

Fuente: <http://www.clickblog.it/post/194357/cose-il-rumore-dellimmagine-e-come-avertirlo>.

Detección de anomalías

La detección de anomalías se produce cuando una de las medidas realizadas carece de sentido o está fuera de un rango determinado.

Aunque principalmente es algo a evitar, la detección de anomalías puede conducir al estudio de casos concretos que deriven en un conocimiento más amplio de la problemática. Por ejemplo, en banca, la detección de anomalías suele estar asociada

con la detección de comportamientos fuera de lo habitual como pueden ser el fraude o el blanqueo de capitales.

En la siguiente imagen se puede apreciar una anomalía en uno de los latidos de un determinado momento de la señal.

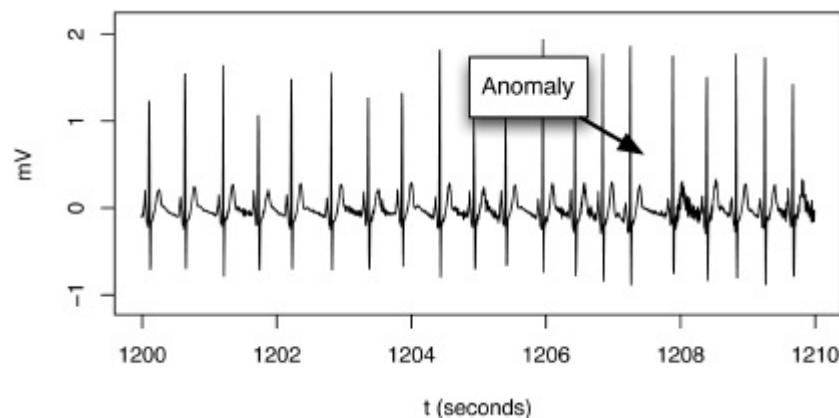


Figura 15. Detección de anomalías en una señal de Electrocardiograma (ECG).

Fuente: Dunning y Friedman (2014).

Corrección de errores

Aunque puede parecer propio de la codificación o transmisión de señales, la corrección de errores permite subsanar rápidamente pequeñas perturbaciones en la captura de información que son estadísticamente corregibles.

Por ejemplo, en una imagen: si en una región todos los píxeles son oscuros y un píxel es completamente blanco, se puede corregir mediante el análisis de los píxeles vecinos.

La corrección de errores no suele ser una de las tareas que se realice por defecto en un preprocesamiento. De hecho, suele venir de una etapa del procesado de información que puede automatizarse y es común al resto del procesamiento.

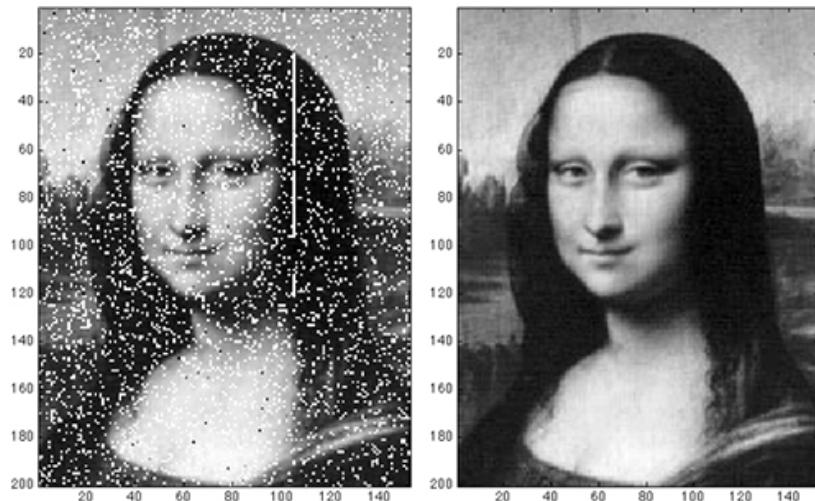


Figura 16. Ejemplo de corrección de errores: (Izq.) Imagen donde se han aplicado errores forzando que algunos píxeles sean blancos. (Dcha.) Imagen reconstruida.

Fuente: <http://cnnespanol.cnn.com/2013/01/18/la-mona-lisa-va-a-la-luna-en-la-primer-transmision-laser-interplanetaria/>.

En definitiva, la etapa de preprocessamiento tiene como fin dotar al sistema de percepción computacional de las herramientas necesarias para:

- ▶ Saber si un determinado sensor está funcionando correctamente: un número elevado de anomalías, errores o ruido puede conducir a la conclusión de que el sensor no está funcionando correctamente.
- ▶ Corregir las posibles desviaciones de los sensores a la hora de realizar una determinada medida.
- ▶ Evitar que los errores en la medida se propaguen hasta la toma de la decisión o, en otras palabras, asegurar en todo momento que la decisión se está tomando con la información más limpia posible.

2.5. Procesamiento de la información



Accede al vídeo «Procesamiento de la información» a través del aula virtual

Llegados a este punto, la información ya está depurada, limpia y disponible para realizar el procesamiento de la información propiamente dicho. Este depende completamente de la finalidad que se persiga (detectar caras en un aeropuerto, leer matrículas de forma automática, detectar enfermedades en la voz, etc.) y es el núcleo de esta asignatura.

Por ello, en este apartado vamos a dar una visión global de los diferentes **métodos de procesamiento** de información existentes y las familias en las que se divide. Todos estos pueden aplicarse de forma secuencial y no suelen ser excluyentes. De hecho, lo habitual en problemas reales es aplicar un conjunto de ellos para encontrar la solución adecuada.

Filtros y suavizado de la información

En general, cuando se habla de filtros se hace referencia a la operación matemática, bien sobre la señal unidimensional, bien sobre la imagen, que permite eliminar o potenciar los detalles de la información a procesar. Un filtro puede eliminar los detalles de una imagen (contrastos, contornos, bordes, etc.) para permitir, por ejemplo, detectar qué objetos existen en dicha imagen.

El **filtrado** suele ser una operación computacionalmente costosa pues requiere recorrer toda la señal unidimensional o toda la imagen, que es una señal bidimensional.

Las operaciones de filtrado normalmente poseen una **operación inversa teórica**, pero puesto que el filtrado y el suavizado de la información conllevan de por sí pérdida de información, volver a la información original es, aparte de costoso, no muy satisfactorio. Además, y no menos importante, se puede decir que la mayoría de filtros son no lineales.

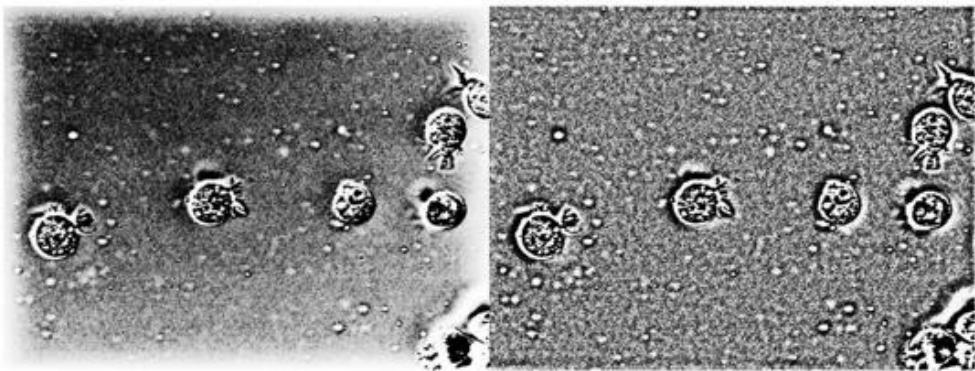


Figura 17. Ejemplo de filtrado de imagen para bajas frecuencias (respetando las altas frecuencias que se corresponden con los detalles de la imagen).

Fuente: <https://blogs.mathworks.com/steve/2013/07/10/homomorphic-filtering-part-2/>.

Segmentación y detección de regiones

Por segmentación se entiende la división de la información en conjuntos de datos con propiedades similares entre ellos.

Un ejemplo claro se aprecia en la segmentación de imágenes. Si consideramos la foto de un elefante en la sabana, la segmentación deberá distinguir las zonas de la imagen más generales como son el suelo, el elefante, el cielo y la vegetación. Se podría haber resumido también en una segmentación de dos elementos: elefante y resto del paisaje. Esta elección es dependiente de la aplicación que se persiga.

La segmentación y detección de regiones es también una de las **operaciones computacionalmente más costosas** que existe en el procesamiento de información.

La dificultad de la segmentación (y que aún se encuentra bajo investigación) es la **detección automática del número de segmentos** dentro de una imagen. Es decir, ¿cuántas subimágenes o grupo de píxeles existen dentro de una imagen?

Para realizar una buena segmentación es necesario que la comparación se realice considerando **características extraídas** de la imagen como pueda ser el color, la textura o incluso la coherencia de las regiones.

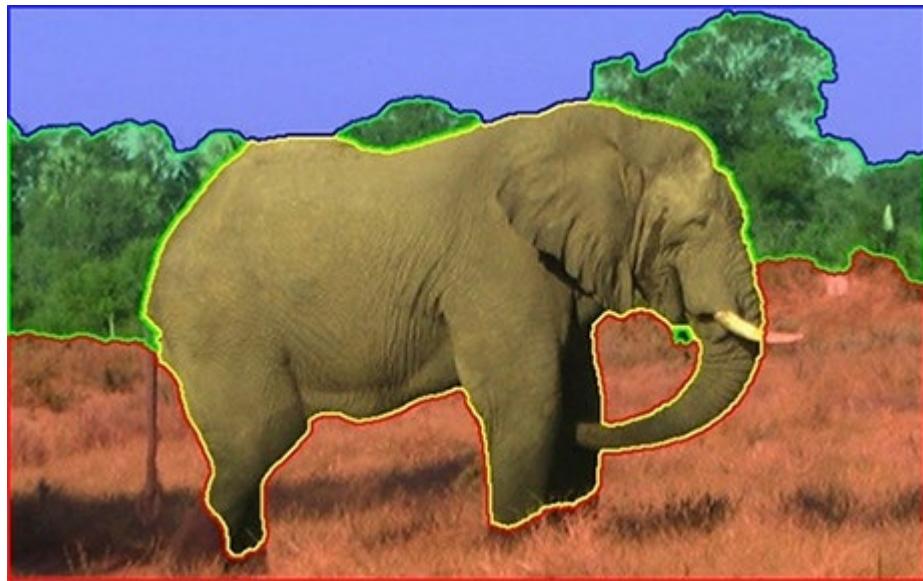


Figura 18. Ejemplo de filtrado de imagen para bajas frecuencias (respetando las altas frecuencias que se corresponden con los detalles de la imagen).

Fuente: <http://ieeexplore.ieee.org/document/7789625/>.

Extracción de características

El siguiente paso lógico a la segmentación es la extracción de características, entendida como un resumen desde el punto de vista analítico de cada región obtenida.

En otras palabras, la extracción de características hace referencia a la capacidad para **distinguir regiones** donde hay diferentes texturas, intensidades o, en el caso de señales unidimensionales, diferentes **patrones de comportamiento**.

La extracción de características puede aplicarse inicialmente sobre la señal unidimensional o sobre la imagen directamente, pero se obtendrá ruido debido a que los extractores de características considerarán el **ruido como parte de la información**. Por eso es recomendable, aunque dependerá del problema en concreto a resolver, aplicar la extracción de características cuando la información esté completamente limpia y preprocesada lo máximo posible.

Ejemplos de características que se pueden extraer:

- ▶ Componentes en frecuencia de diferentes secciones de una imagen.

- ▶ Coeficientes de transformaciones no lineales como puede ser la transformada de Fourier o de Laplace.
- ▶ Suavidad del contorno de un objeto detectado.
- ▶ Área del segmento encontrado después de la segmentación.
- ▶ Descriptores de texturas de una determinada región.

De todas las fases del procesado de la información, es la que más se está desarrollando últimamente y donde hay más foco de investigación. Muchos de los resultados han ayudado a avanzar en la encriptación y la compresión de datos, entre otros.

En definitiva, la extracción de características permite **facilitar la comparación** entre dos regiones dentro de una imagen, entre dos objetos previamente segmentados o incluso entre dos señales unidimensionales.

En la siguiente imagen vemos como la extracción de características acaba con un vector de características que facilita su posterior comparación con otro vector.

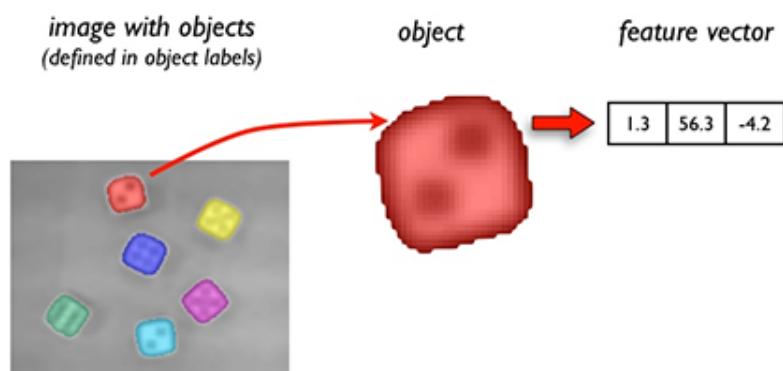


Figura 19. Ejemplo de filtrado de imagen para bajas frecuencias (respetando las altas frecuencias que se corresponden con los detalles de la imagen).

Fuente: http://perclass.com/doc/guide/feature_extraction/intro.html

2.6. Toma de decisión



Accede al vídeo «Toma de decisión» a través del aula virtual

Para recapitular, cuando se llega a la toma de decisiones se han realizado los siguientes pasos que vemos con un ejemplo sobre la lectura de matrículas de forma automática:

Se ha definido un conjunto de fuentes de información que pueden resultar de interés para resolver un determinado problema.

Ejemplo: leer de forma automática las matrículas de un coche.

Una vez fijadas las fuentes de información, se han decidido qué sensores emplear para capturar dicha información.

Ejemplo: cámaras estándar que capturan fotografías cada segundo.

El preprocesamiento consiste en corregir bien si existe desenfoque en la imagen capturada.

Ejemplo: algo normal, debido a la velocidad a la que se captura la imagen.

El procesamiento consiste en un filtrado para eliminar los detalles de menor importancia, una segmentación en color y una detección de bordes.

Posteriormente se ha hecho una extracción de características.

Ejemplo: detectar dónde está ubicado cada número y letra de la matrícula.

Figura 20. Resumen y ejemplo de las fases anteriores de la de toma de decisiones.

Llegados a este punto tenemos, en el caso de España, cuatro números y tres letras. De todos los métodos presentados en este apartado, ninguno de ellos nos ayuda en:

- ▶ Asegurar que en efecto hay cuatro números y tres letras.
- ▶ Asegurar que estoy leyendo realmente una matrícula y no la marca del coche.
- ▶ Asociar la región correspondiente a un número con un número concreto. Es decir, ¿cómo asociar un determinado conjunto de píxeles a un número?
- ▶ Detectar que no es una matrícula española.

Todas estas tareas deben realizarse en el módulo de toma de decisiones. Dicho módulo es el encargado de aplicar la lógica final para o bien tomar una decisión: ¿Es una matrícula registrada?; o bien dar soporte en la toma de decisiones: marcar en una determinada imagen qué zonas corresponden con paisajes, por ejemplo.

Esta fase debe realizarse principalmente tras la extracción de características. Considerarla basándonos únicamente en filtrados, segmentaciones o incluso en el preprocesado no suele ser común y, en muchos casos complica, enormemente la toma de decisiones. Sin embargo, existen casos donde los módulos de toma de decisiones pudieran no existir.

Por ejemplo, el coche que es capaz de conducir de forma automática (sin supervisión); estos coches poseen un módulo capaz de detectar las líneas de las carreteras. Dicho módulo tiene todos los ingredientes de un sistema de percepción computacional: captura de la información, preprocesamiento y extracción de características.

No obstante, dicho módulo no toma ninguna decisión, sino que proporciona la propia extracción de características a otro módulo que aglutina la información de otras fuentes y toma la decisión. Con lo cual, puede decirse que dicho módulo de percepción computacional no toma una decisión, sino que ayuda a otros módulos tanto a aprender como a dar soporte en la toma de decisiones.

2.7. Referencias bibliográficas

Dunning, T. y Friedman, E. (2014). *Time Series Databases: New Ways to Store and Access Data*. Sebastopol: O'Reilly Media.

González, R. C. y Woods R. E. (2007). *Digital image processing*. Upper Saddle River: Prentice-Hall, Inc.



Accede a los ejercicios de autoevaluación a través del aula virtual

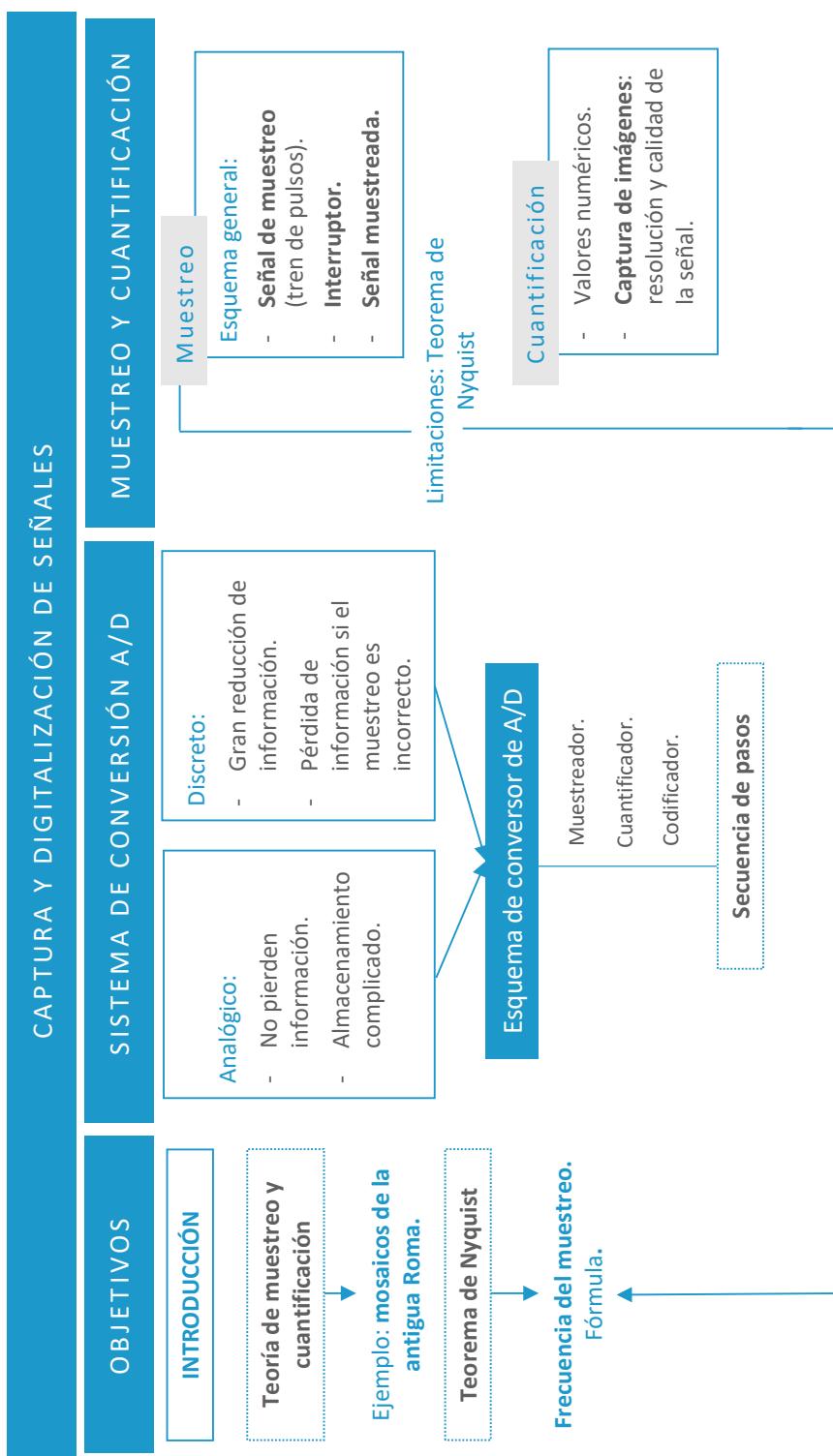


Accede al vídeo «Resumen del tema» a través del aula virtual

Percepción Computacional

Captura y digitalización de señales

Esquema. Tema 3



Ideas clave. Tema 3

3.1. ¿Cómo estudiar este tema?

Para estudiar este tema deberás leer con atención las ideas clave que se desarrollan a continuación.

3.2. Objetivos

El muestreo de señales es una parte esencial en la captura de información. En él se basa el resto de procesamiento y la información que se pierda en esta fase no podrá estar disponible.

Es por eso que este tema pretende proporcionar las bases para entender correctamente el funcionamiento del muestreo y la posterior cuantificación de señales.

3.3. Introducción

 Accede al vídeo «Introducción. Captura y digitalización de señales» a través del aula virtual

La teoría de muestreo y cuantificación es uno de los temas más áridos en toda asignatura de procesamiento de señales. Reflejar sobre el papel qué significa muestrear una señal requiere de una complejidad matemática muy alta.

Sin embargo, la mejor manera de entenderlo es verlo con un ejemplo, y para ello recurrimos a la cuantificación más antigua que se conoce: los mosaicos de la antigua Roma.



Figura 21. Ejemplo de discretización o cuantificación: detalle del mosaico de Medusa .y las estaciones.
Fuente: <http://www.man.es/man/dms/man/estudio/publicaciones/boletin-man/MAN-Bol-2015/2015-33-Jabaloyas.pdf>.

El ejercicio de crear un mosaico es claro: con un conjunto «limitado» de colores (como mucho entre cien o doscientos colores) que se corresponden con diferentes tipos de piedra, elaborar un dibujo que refleje la complejidad de la realidad como las sombras, claroscuros, tonalidad, movimiento e incluso la expresión.

Intuitivamente podemos llegar a comprender que cuanto mayor sea el número de **colores** disponibles, mayor la gama cromática que podemos alcanzar y, por lo tanto, mayor el número de matices y expresiones o detalles que se pueden mostrar. Lo mismo sucede con el **tamaño** de la tesela. Parece intuitivo pensar que cuanto más pequeña sea la tesela, mejor nos aproximaremos a la realidad.

Pero si tuviésemos que hacerlo ahora, ¿qué cantidad de colores elegirías?, ¿qué tamaño de tesela?, ¿serían todas del mismo tamaño?, ¿cuál sería el mínimo número de colores y cuál el tamaño de la tesela para que el mosaico fuera lo más barato y

bello a la vez? Estas preguntas no son tan obvias de responder. Y los romanos seguro que tampoco tenían las respuestas.

La teoría de muestreo y cuantificación nos ayuda a decidir esos mínimos valores.

Nos da los umbrales a partir de los cuales podemos tener una calidad suficiente de señal o de imagen, sin necesidad de emplear mucho almacenamiento (teselas en el ejemplo del mosaico romano).

Dicha teoría de muestreo funciona de la siguiente manera: imaginemos que queremos entender el funcionamiento de la noche y el día. Vamos a asumir que tenemos siempre los ojos cerrados y que los abrimos en un momento concreto. En este ejercicio, abrir los ojos es una metáfora para **muestrear una realidad**; en definitiva, no es más que observar.

Pongamos que la **primera muestra**, la primera vez que abrimos los ojos, es de día. Si abrimos los ojos una vez al mes, siempre a la misma hora, solo veremos que es de día pudiendo llegar a la conclusión de que en nuestro planeta solo está el día y no hay noche.

Como sabemos teoría de muestreo, decidimos acelerar la **frecuencia de muestreo** a una vez al día, cada 24h. Lamentablemente observamos el mismo efecto: es de día. Decidimos hacer el experimento cada hora y vemos cada día doce muestras de luz y doce muestras de oscuridad (asumiendo que el día y la noche tienen igual número de horas). Evidentemente, con este muestreo se ha observado el fenómeno de la noche. Lo mismo hubiera pasado si miro cada segundo, evidentemente.

Luego, hasta este punto tenemos que si se observa:

- ▶ Cada 24h, siempre hay luz.
- ▶ Cada hora, hay doce observaciones de luz y doce de oscuridad.

Si ahora realizamos el experimento midiendo cada doce horas, realizaremos dos medidas al día: una de noche y una de día. Hemos dado con la **frecuencia mínima**, también llamada frecuencia de muestreo, para observar el fenómeno de la noche y el día. Parece casualidad que:

La **frecuencia del fenómeno** sea una vez cada 24 horas, es decir, la secuencia luz-noche se repite a las 24h, mientras que la **frecuencia mínima** para observar el fenómeno es justo el doble, una vez cada 12h.

Sería aventurarse mucho decir que:

$$f_{muestreo} \geq 2 f_{observación}$$

Esta igualdad se conoce como **teorema de Nyquist**, a dicha frecuencia de muestreo se la denomina también **Frecuencia de Nyquist** y es la idea principal sobre la que trata este tema. En otras palabras, para procesar información digital hay que perder calidad de la señal e información. La cuestión es, ¿cuál es la mínima información que puedo perder?

3.4. Sistemas de conversión analógico-discreto



Accede al vídeo «Sistema de conversión analógico-discreto» a través del aula virtual

En esta sección contaremos en qué consiste un conversor analógico-discreto más en detalle, así como su utilidad y funcionamiento. Antes de nada, es importante entender dos conceptos: analógico y discreto.

Analógico

Por analógico se entiende toda señal o imagen que es continua y que procede de una fuente o fenómeno físicos como puede ser un sonido, un objeto en movimiento, etc. Si continuamos con el ejemplo anterior del mosaico, lo analógico sería el dibujo original del que partió el artista que diseño el mosaico.

- ▶ Este tipo de señales se caracterizan por tener todo el detalle de la realidad, **no pierden nada de información** cuando se transmiten de un punto a otro.
- ▶ El almacenamiento de una señal analógica es complicado y muy poco eficiente (sirva como ejemplo los discos de vinilo o los *cassettes* antiguos).

Discreto

Por discreto (también digital, son sinónimos en este contexto) se entiende una señal o imagen que **solo puede alcanzar unos determinados valores**, por lo que está discretizada. Normalmente, una señal discreta/digital viene de muestrear y discretizar la señal analógica.

- ▶ La mayor ventaja es la reducción de información que supone.
- ▶ La desventaja, si el muestreo no se realiza correctamente, puede ser que la señal pierda la información más importante.

Esquema de un conversor A/D

A continuación, se presenta un esquema general de un conversor A/D. Este conversor consta de las siguientes partes:

- ▶ En primer lugar, el **muestreador**, que es el encargado de convertir la señal analógica a discreto tomando fotos de la señal a una frecuencia determinada.

- ▶ En segundo lugar, un **cuantificador**, encargado de transformar la foto tomada en el muestreo y convertirla en un conjunto limitado de valores.
- ▶ Finalmente, un **codificador** que convierte el valor del cuantificador en binario o en un sistema numérico que pueda transmitirse.

En algunos ejemplos de la literatura, cuantificador y codificador van de la mano y suelen ser una misma pieza. No obstante, pueden separarse con vistas a que se apliquen diferentes tipos de codificación (junto con su redundancia para evitar errores).

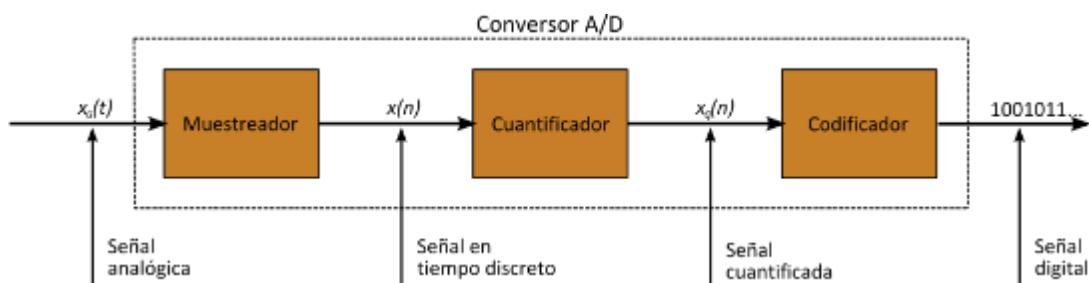


Figura 22. Esquema general de un conversor analógico-discreto.

Fuente: <http://elastixtech.com/fundamentos-de-telefonia/transmision-de-la-voz/>.

Las ventajas principales de hacer este proceso son las siguientes:

- ▶ Permite, en primer lugar, reducir la complejidad de la señal de entrada y reducirla únicamente a un determinado número de posibilidades (**cuantificador**).
- ▶ El filtrado de ruidos, anomalías y otros efectos producidos por una baja calidad son mucho más fácilmente procesables con un filtrado digital que con un filtrado analógico.
- ▶ Los filtros digitales pueden modificarse mediante software, a diferencia de los analógicos, que deben modificarse por hardware, lo que los encarece y hace difícil su mantenimiento.
- ▶ La señal analógica, por el mero hecho de ser convertida a señal digital, ya reduce la cantidad de información a procesar posteriormente, pero otra características es que dicha señal digital puede comprimirse y encriptarse.

- ▶ El almacenamiento de una señal digital es más sencillo y barato que el almacenamiento analógico, así como más perdurable en el tiempo.

Sin embargo, las señales digitales (un archivo MP3 por ejemplo) no pueden oírse directamente, lo único que pueden oírse son las señales analógicas. El proceso de conversión de analógico a digital o discreto debe permitir el proceso inverso, asumiendo que por la propia definición de este proceso se perderá información, pero la mínima como para que la señal analógica reconstruida suene muy similar a la original.

Para ello, deben seguirse los siguientes pasos que se presentan a continuación:

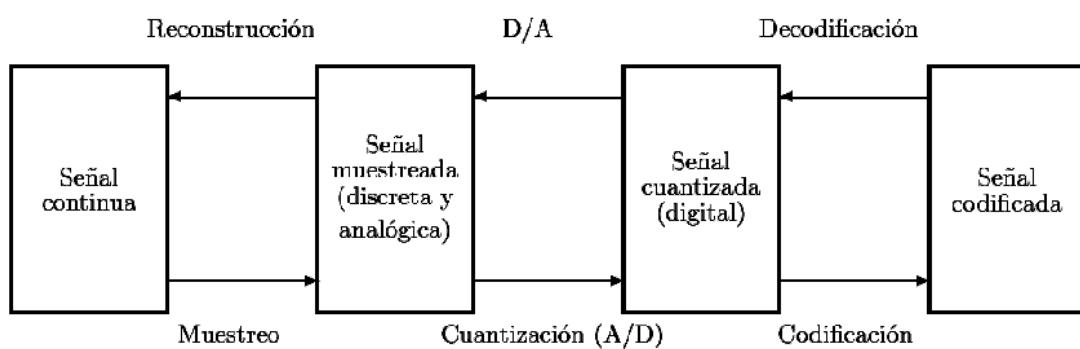


Figura 23. Reconstrucción de una señal analógica.

Fuente: <http://www.lip.uns.edu.ar/pdi/index31.htm>.

En este tema solo veremos el proceso de conversión A/D. El proceso inverso, (conversión D/A), aunque similar en estructura y pasos a dar, contiene otras peculiaridades que se desvían del objetivo de esta asignatura.

Para entender el diagrama anterior, vamos a explicarlo con un ejemplo: vamos a un concierto de nuestro cantante favorito y nos llevamos nuestro móvil. En el momento en que suena nuestra canción favorita, sacamos el móvil y grabamos, ¿qué sucede en ese instante?

- ▶ Primero, el micrófono del móvil captura toda la información analógica proveniente del exterior (no solo la voz del cantante, sino también el ruido de ambiente).

- ▶ Dicho micrófono, aunque puede capturar toda la información, lleva asociado un sistema de muestreo que «observa» la señal de forma periódica, es decir, mide la intensidad de voz cada cierto periodo de tiempo.
- ▶ La señal, una vez muestreada y cuantificada, ya puede almacenarse en el dispositivo móvil. Es la única manera de almacenar información en un dispositivo de este tipo.
- ▶ Para que la información ocupe lo menos posible, la información se comprime y se filtra por defecto para que quede lo más limpia posible.

A continuación, puede verse un esquema de los pasos seguidos para guardar un sonido de voz en un dispositivo móvil:

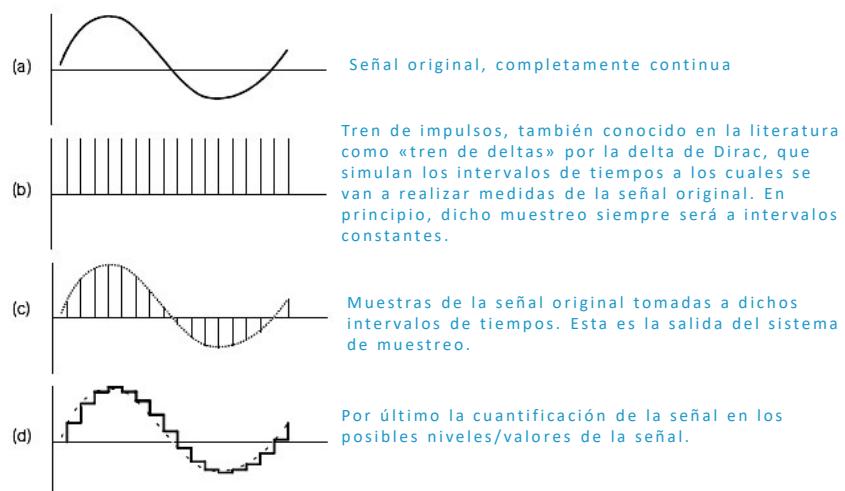
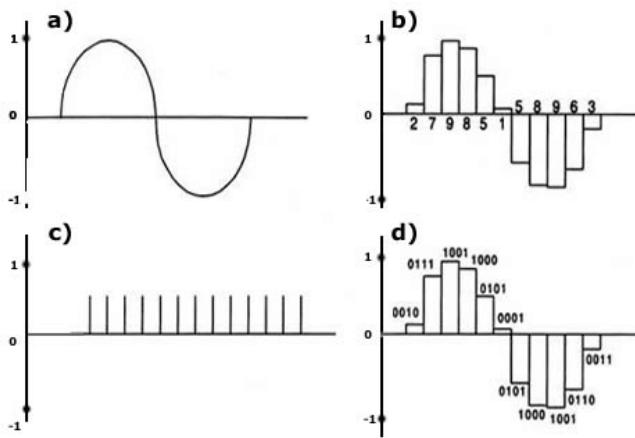


Figura 24. Secuencia de pasos para la conversión analógico-discreto.
Fuente: Adaptado de <http://www.sapiensman.com/tecnoficio/docs/doc39.php>.

Señal original.

Tren de impulsos que simulan los intervalos de tiempos a los que la señal se muestrea.



Señal discretizada incluyendo los valores discretos asignados a cada muestra.

Conversión de dichos valores a binario (o cualquier otro formato digital).

Figura 25. Secuencia para la conversión analógico-discreto incluyendo también la digitalización de la señal.

Fuente: Adaptado de <http://acusticaysonido.com/?p=315>.

3.4. Muestreo y cuantificación



Accede al vídeo «Muestreo y cuantificación I» a través del aula virtual



Accede al vídeo «Muestreo y cuantificación II» a través del aula virtual

En el apartado anterior, se ha presentado una visión general de los sistemas de conversión analógico-discreto. Ahora, en este apartado, se describen los componentes individuales y la manera en la que se implementan.

Muestreo

Como hemos dicho antes, el muestreo consiste en tomar fotos de la realidad (ya sea una señal o una imagen) a una determinada frecuencia (posteriormente veremos qué criterio) de tal manera que la información manejada se reduce, pero no la información necesaria para comprender el mensaje.

A continuación, se presenta un esquema general de muestreo. Dicho sistema consta de:

- ▶ Una señal de muestreo, conocida como **tren de pulsos**, que define la frecuencia y el tiempo de muestreo, es decir, durante cuánto tiempo se va a observar la señal.
- ▶ Un **interruptor** que, en función de la señal de muestreo, se abrirá o cerrará permitiendo tomar las fotos de la señal original.
- ▶ Por último, la **señal muestreada**.

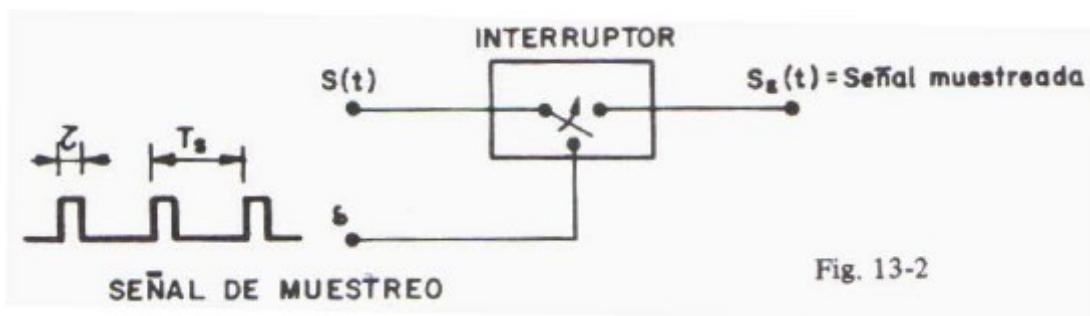


Fig. 13-2

Figura 26. Sistema de muestreo basado en tren de pulsos.

Fuente: http://www.ifent.org/lecciones/digitales/secuenciales/teorema_muestreo.htm.

El mecanismo es sencillo en su visión reducida, pero presenta ciertas **limitaciones**:

- ▶ El muestreo dependerá de la señal de muestreo, tanto de la anchura de los pulsos como de la separación entre ellos.
- ▶ El interruptor en sí mismo presenta discontinuidades muy pronunciadas cuando conmuta tan rápidamente. Esto puede ocasionar (y de hecho los produce) picos de señal en la señal muestreada.
- ▶ El comutador puede estropearse debido al propio mecanismo y su uso continuado, cualquier solución que no involucre un sistema mecánico ofrecerá ventajas considerables. Es por ello que muchos muestreadores multiplican la señal por dicha señal de muestreo.

Ahora bien, de todos estos parámetros anteriores, solo hay uno que sí que
puede modificarse: la frecuencia de muestreo.

Dicha frecuencia viene definida por el **teorema de Nyquist-Shannon** (profundizaremos más sobre esto en la lección magistral de este tema) que indica que si una señal tiene un ancho de banda f , la frecuencia de muestreo deberá de ser:

$$f_{muestreo} \geq 2f$$

Sin embargo, existen dos aspectos importantes a tener en cuenta si no se cumple dicho teorema.

¿Qué sucede si muestreo por debajo de la frecuencia de Nyquist?

Lo que sucederá es lo que se ve en la siguiente imagen. Partiendo de la señal sinusoidal original, si se muestrea a menor frecuencia se verá por tanto una variación de señal que no corresponde con la verdadera. Es decir, veremos que la señal va más despacio.

Este efecto sucede cuando vemos un coche en marcha y sus ruedas parecen girar en el sentido contrario de la marcha.

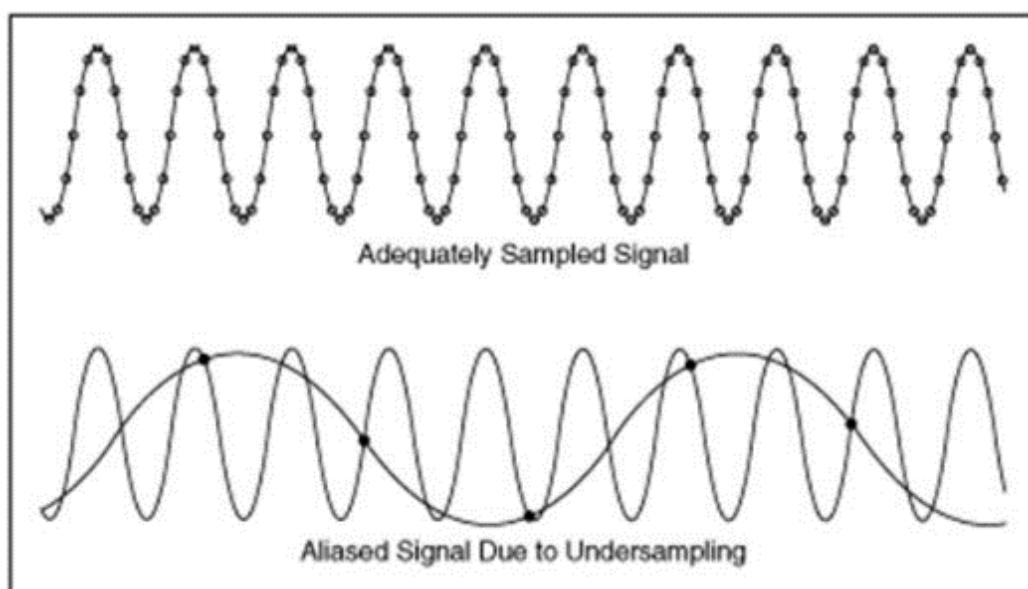


Figura 27. Ejemplo visual de submuestreo o *undersampling*.

Fuente: <https://www.masterhacks.net/blog-ingenero/que-es-el-aliasing/>

¿Qué sucede si muestreamos a mucha más frecuencia?

Entonces, estaremos capturando información de más, lo que produce una reconstrucción errónea de la señal, tal y como se ve en la imagen siguiente. Es decir, consideramos puntos consecutivos, entre los cuales no hay apenas variación, como puntos que poseen una gran variabilidad.

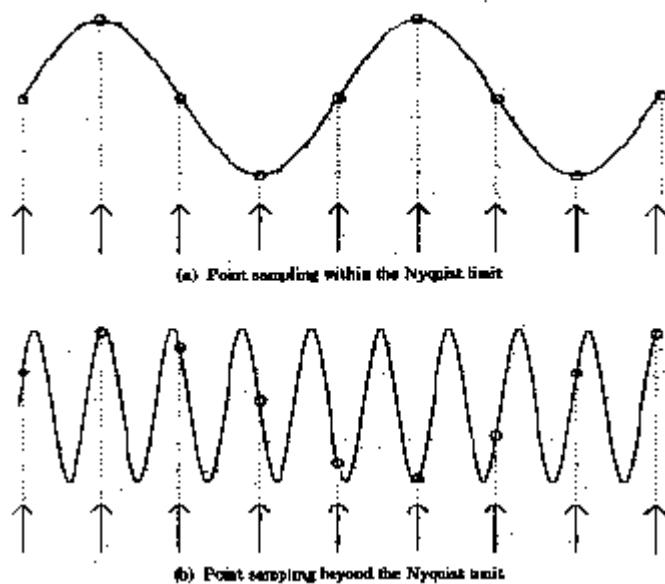


Figura 28. Ejemplo de sobremuestreo de señales u *oversampling*.

Fuente: <https://www.siggraph.org/education/materials/HyperGraph/aliasing/alias1.htm>.

Cuantificación

La cuantificación es el proceso de convertir las muestras tomadas de la señal original en **valores numéricos**.

Dichos valores numéricos poseen una resolución que dependerá del número de bits con los que cuente el procesador. Como se aprecia en la imagen siguiente, a mayor número de bits, mayor parecido con la señal original.

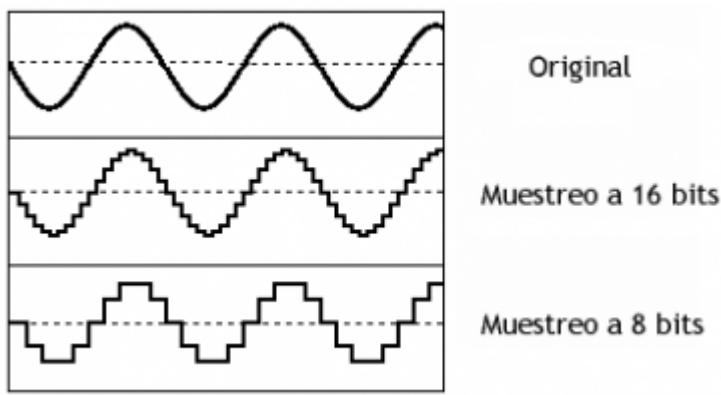


Figura 29. Ejemplo de cuantificación con diferentes niveles de calidad (bits).

Fuente: <http://www.fotoseimagenes.net/frecuencia-de-muestreo>.

En general, los sistemas de cuantificación proporcionan una **cuantificación uniforme**, pero en determinados casos, donde la riqueza de valores es más amplia, suele aumentarse el número de valores en una determinada región de la señal para que así se puede alcanzar mayor resolución donde más se necesita.

La elección del número de bits, así como de la uniformidad del rango es algo que depende de la aplicación y, al contrario del teorema de Nyquist, no existe un teorema concreto que definan ambos parámetros simultáneamente. Es cierto que el teorema de Shannon sí que da una idea de cuántos números de bits harían falta para codificar una determinada señal. Esto lo veremos en detalle en la sesión magistral.

Esta cuantificación realmente está implícita en la propia captura de imágenes. Por ejemplo, las cámaras CCD (ver imagen a continuación) poseen un cuantificador por fotodiodo, con lo cual una imagen nunca es continua, ya es discreta de por sí, y lo mismo sucede con las señales digitales.

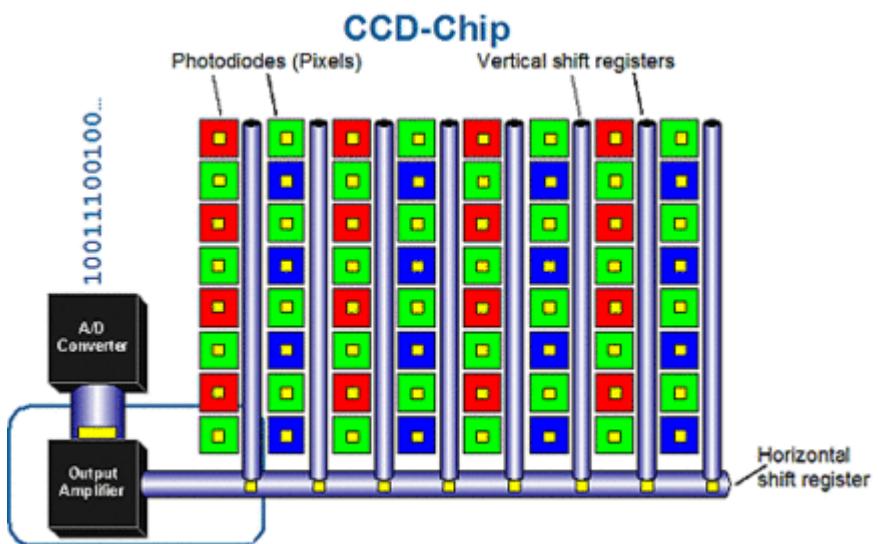


Figura 30. Esquema de funcionamiento de captura de imágenes basada en CCD.
Fuente: <http://distanciafocal.com.ar/%C2%BFcomo-elegir-el-tipo-de-sensor/>.

La cuantificación puede verse como un tema de **resolución o calidad de la señal**. De hecho, en función del número de bits empleados para cuantificar una señal, así será su propia calidad. Normalmente, en imagen, se trabaja con 256 niveles de color, siendo 0 el color negro y 255 el color blanco en escala de grises.

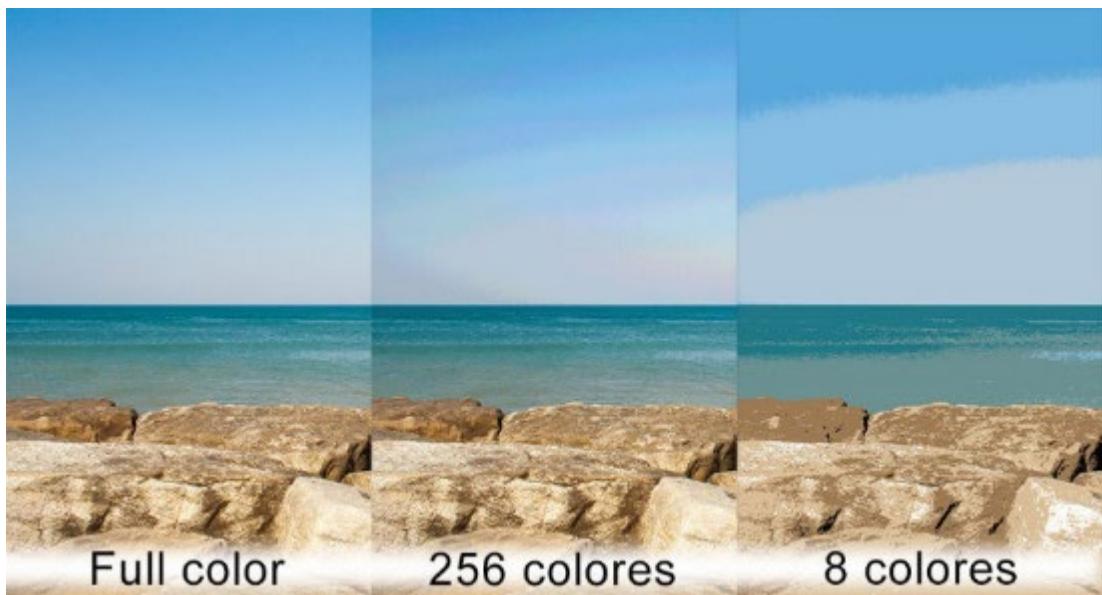


Figura 31. Cuantificación con diferentes niveles de resolución: Full color, 256 colores, 8 colores.
Fuente: <http://www.comercialgumara.es/blog/pixel-vs-imagenes-vectoriales/>.

Así como el muestreo puede volver a recomponerse, siempre y cuando se cumpla el teorema de Nyquist (como veremos en la clase magistral), la información difícilmente

puede recuperarse en el caso de la resolución baja. Existen algunos algoritmos, basados en la transformada del Coseno, similar a la transformada de Fourier, pero que su funcionalidad se basa en inferir qué rango de valores posibles podrían haber tomado antes de la cuantificación.

Por último, podemos indicar que hoy día muestreo y cuantificación están tan relacionados que se vinculan incluso a nivel *hardware* y de dispositivo.



Accede a los ejercicios de autoevaluación a través del aula virtual

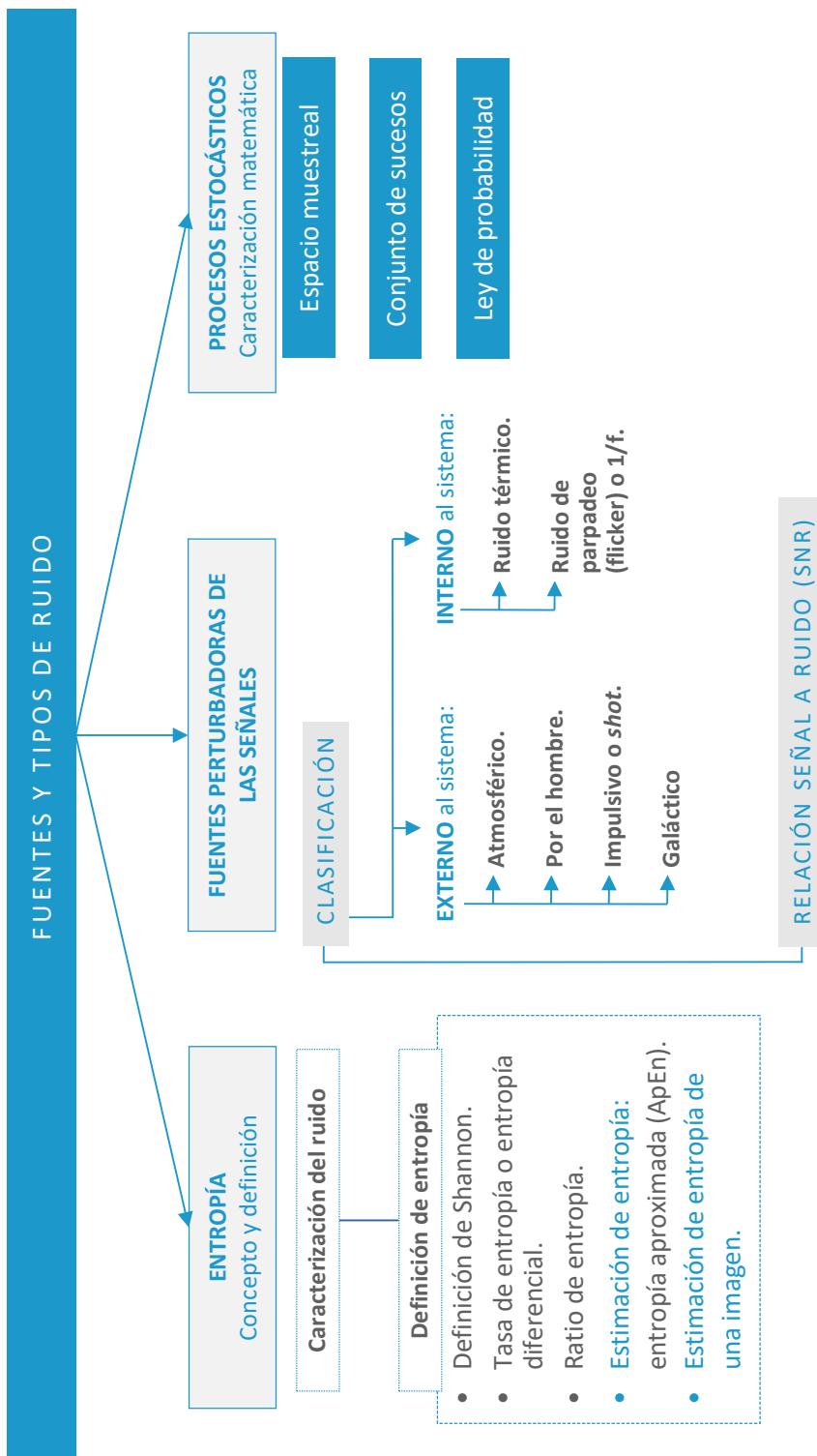


Accede al vídeo «Resumen del tema» a través del aula virtual

Percepción Computacional

Fuentes y tipos de ruido

Esquema. Tema 4



Ideas clave. Tema 4

4.1. ¿Cómo estudiar este tema?



Accede al vídeo «Introducción. Fuentes y tipos de ruido» a través del aula virtual

Para estudiar este tema deberás leer con atención las ideas clave que se desarrollan a continuación.

En el siguiente tema se analiza el **componente aleatorio de las señales**. Como veremos, toda señal estará afectada por algún tipo de perturbación o ruido en un escenario real. El conocimiento de los diferentes tipos de ruido nos permitirá poder diseñar técnicas adecuadas para su eliminación completa o parcial, reduciendo su efecto en la fuente de información.

Con este objetivo, en el tema se presenta, en primer lugar, el concepto de **entropía**. Este hace referencia al grado de incertidumbre e irregularidad en la señal, de forma que nos permite cuantificar el impacto del ruido. Posteriormente, se describen los diferentes **tipos** de ruido. Y finalmente, se presentan las herramientas matemáticas que permiten su modelado: los **procesos estocásticos**.

4.2. Entropía: concepto y estimación



Accede al vídeo «Entropía: concepto y estimación» a través del aula virtual

El ruido es toda señal no deseada y de naturaleza aleatoria que modifica la intensidad de la señal original a percibir.

En los sistemas de captación de estímulos, como luz o sonido, en primer lugar se lleva a cabo la transformación de estos a señales eléctricas mediante sensores. Por ejemplo, considérese el micrófono o la cámara fotográfica, que captan el sonido o la luz y los convierten en una señal eléctrica de naturaleza analógica o digital. Posteriormente, estas señales pueden almacenarse en memorias o ser transmitidas por medio de un sistema de comunicación.

Caracterización del ruido

En el mundo real, cualquier señal estará afectada por elementos incontrolables que generen ruido. Generalmente, este se superpone a la señal de interés en forma de **ruido aditivo**. Es decir, nuestra imagen o señal acústica recibida tendrá la siguiente forma:

$$S(t) = f(t) + r(t)$$

Donde:

- ▶ S denota la señal recibida.
- ▶ f denota la señal original.
- ▶ r denota el componente de ruido.

Por tanto, la primera etapa en el tratamiento de una señal debe centrarse en la **identificación y eliminación** del mayor número de artefactos ruidosos posible; en la mayoría de los casos no será viable eliminarlos por completo de la señal capturada.

La **caracterización matemática** del ruido permite abordar el problema de forma analítica. Como se verá más adelante en este tema, las señales que manejamos en la vida real incluyen cierto nivel de ruido, por lo que pueden verse como el resultado de

un proceso aleatorio, tal y como sucede cuando lanzamos un dado. La naturaleza aleatoria del ruido provoca que la medición de una misma señal no produzca siempre el mismo resultado.

No estamos ante señales deterministas sino aleatorias: procesos estocásticos.

Como ejemplo, puede considerarse el escenario en el que se capta una señal sonora. En este caso, el resultado de la medición variará dependiendo del nivel de intensidad de otras fuentes sonoras sobre las que no tengamos control (ruido de ambiente).

Definición de entropía

Como se ha comentado anteriormente, las señales contienen información y, además, están afectadas por diferentes fuentes de ruido. En este contexto, surge el concepto de entropía. Al igual que en física, el término hace referencia a la **complejidad de la señal**. La adición de ruido contribuye a incrementar el grado de complejidad de una señal, resultando en una mayor entropía.

En teoría de la información, la entropía se define como la cantidad de información de una fuente aleatoria (en promedio). Por tanto, el concepto de entropía sirve para caracterizar una **variable aleatoria**. En el caso de las señales, estas pueden modelarse como una sucesión de realizaciones de una variable aleatoria en el tiempo (proceso estocástico), por lo que veremos cómo extender la definición de entropía a elementos aleatorios de esta naturaleza.

Dada una variable aleatoria discreta X , que toma valores del conjunto $\{X_1, X_2 \dots, X_M\}$ y de acuerdo a una probabilidad $P(X_i)$, **Shannon** definió la entropía de la siguiente forma:

$$H(X) = E\{-\log[P(X)]\} = \sum_{i=1}^M -\log[P(X_i)]P(X_i)$$

Donde $-\log[P(X_i)]$ se interpreta como la cantidad de información.

Así, valores poco probables de la variable X están asociados con una mayor cantidad de información (efecto sorpresa) frente a valores de mayor probabilidad.

En el caso de que la variable X posea únicamente dos posibles resultados, $\{X_1, X_2\}$, es decir, cuando se trata de una variable caracterizada por una distribución de Bernoulli, puede observarse como la entropía de X , en función de la probabilidad $P(X = X_1)$, alcanza su máximo cuando esta es 0.5. En este caso, ambos sucesos poseen la misma probabilidad y en promedio obtendremos siempre la misma cantidad de información de X .

El siguiente gráfico muestra la variación de la entropía de X como función de la probabilidad asociada al suceso X_1 .

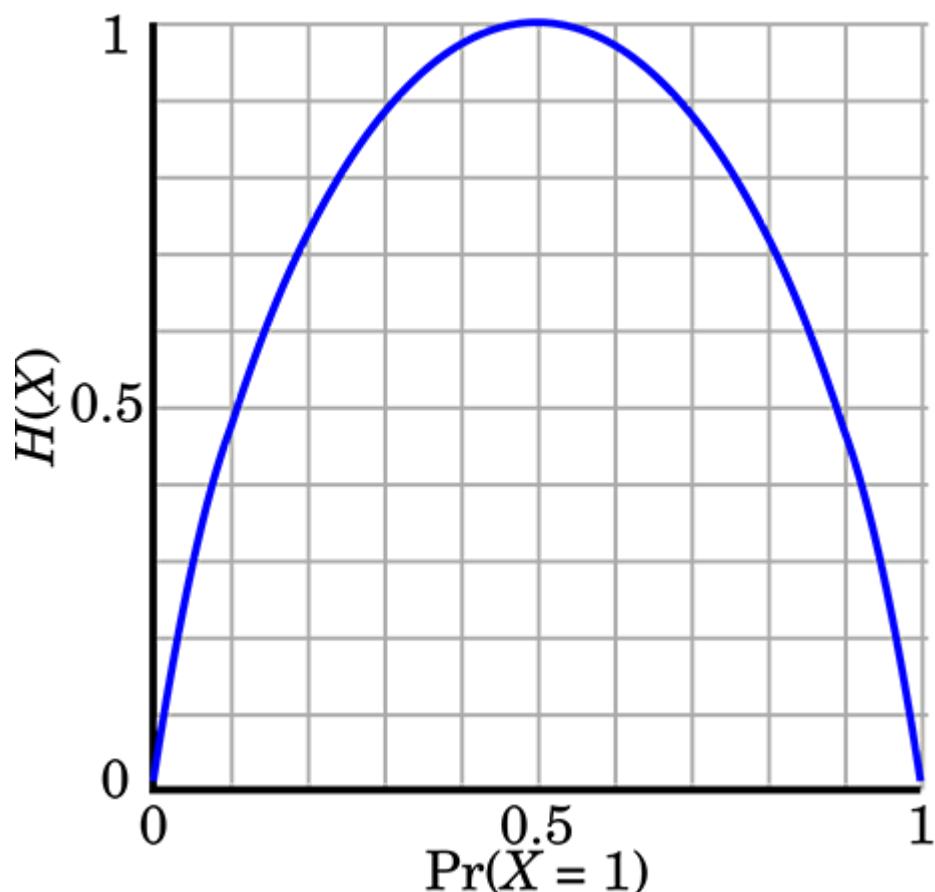


Figura 32. Dependencia de la entropía con la probabilidad de observación del suceso X_1 para una variable aleatoria X con una distribución de Bernoulli.

Fuente: [https://es.wikipedia.org/wiki/Entrop%C3%ADa_\(informaci%C3%B3n\)](https://es.wikipedia.org/wiki/Entrop%C3%ADa_(informaci%C3%B3n)).

Las señales pueden modelarse, desde un punto de vista matemático, como un conjunto de variables aleatorias (proceso estocástico). Por ejemplo: una señal de voz de una duración determinada puede verse como una serie temporal finita, de forma que cada una de sus muestras representa la realización de una variable aleatoria.

La inclusión de nuevas muestras en la serie contribuirá a aumentar el contenido de información en esta, lo que muestra la dependencia de la entropía del proceso con la longitud de este. Por tanto, tiene sentido medir la variación de la entropía de la señal debida a la inclusión de una nueva muestra, hablamos de **la tasa de entropía o entropía diferencial**.

Considérese una señal de longitud N dada por $x = [x_1, x_2 \dots, x_N]$ y compuesta por la secuencia de la realización de N variables aleatorias. La entropía del proceso estocástico viene dada por:

$$\begin{aligned} H_N &= E\{-\log[p(x_1, x_2 \dots, x_N)]\} \\ &= - \int_{-\infty}^{\infty} \log[p(x_1, x_2 \dots, x_N)] p(x_1, x_2 \dots, x_N) dx_1 \dots dx_N \end{aligned}$$

Donde $p(x_1, x_2 \dots, x_N)$ es la función de densidad de probabilidad conjunta de las variables que componen el proceso estocástico.

A partir de esta expresión, el **ratio de entropía**, o simplemente entropía, de la señal se define de la siguiente forma:

$$E_N = \lim_{N \rightarrow \infty} H_{N+1} - H_N$$

Existen diferentes métodos para la estimación de la entropía de una señal. En este caso, propondremos la **entropía aproximada (ApEn)** como procedimiento de estimación.

Este algoritmo consiste en estimar la entropía de subseries de longitud m y $m+1$, de forma que la diferencia de ambas estimaciones proporciona el valor final de entropía. A continuación, se describe cómo obtener la estimación de la entropía para subseries de **longitud m** . El mismo procedimiento se aplicaría para longitud $m + 1$.

Para nuestra serie original $x = [x_1, x_2 \dots, x_N]$:

- ▶ Se extraen todas las subseries de longitud m representadas por $x_i^{(m)}$.
- ▶ Posteriormente, se define una tolerancia r que permite identificar el número $N^{(m)}(i)$ de las subseries $x_j^{(m)}$ que se asemejan a $x_i^{(m)}$, de acuerdo a una métrica de la distancia entre ambas $d[x_i^{(m)}, x_j^{(m)}] \leq r$.
- ▶ Por tanto, la probabilidad de encontrar en la serie original una subserie similar a $x_i^{(m)}$ viene dada por la siguiente expresión:

$$C^{(m)}(i) = N^{(m)}(i)/(N - m + 1)$$

Donde $N - m + 1$ es el número total de subseries de longitud m que podemos encontrar en la serie original.

El término $C^{(m)}(i)$ proporciona una estimación discreta de la función de densidad de probabilidad $f(x^{(m)})$. Esta puede emplearse para aproximar la entropía de la variable aleatoria $x^{(m)}$, que vendrá dada por el promedio de la cantidad de información de esta fuente aleatoria, empleando la definición de entropía de Shannon. Así, la entropía de los procesos representados por $x^{(m)}$ se calcula como:

$$H_N^{(m)} = -1/(N - m + 1) \sum_{i=1}^{N-m+1} C^{(m)}(i)$$

El **ratio de entropía de la serie** (proceso estocástico) inicial, es decir, el incremento de entropía por la inclusión de una nueva muestra se obtiene como:

$$E_N = H_N^{(m+1)} - H_N^{(m)}$$

El valor de la entropía de una señal puede interpretarse como el grado de incertidumbre de la misma. De forma equivalente, refleja la capacidad para predecir un estado o valor posterior a partir del conocimiento u observación de los valores que previamente ha tomado la señal. Un mayor valor de entropía reflejará, por tanto, mayor complejidad y caos en la señal bajo estudio.

Como resultado, la entropía nos da una idea del nivel de impacto del ruido en una señal.

Si tomamos una muestra de una misma señal en las mismas condiciones pero en instantes de tiempo diferentes, la señal con mayor entropía será aquella con un mayor nivel de ruido.

Hasta ahora, se ha presentado el concepto de entropía tomando como contexto la influencia del ruido en una señal, y se ha descrito cómo estimar su valor en **series temporales**, señales dependientes de la variable tiempo de una única dimensión.

La naturaleza y el modelado matemático de las **imágenes** son diferentes al de este tipo de señales. Una imagen no lleva implícita la variable tiempo, como sí ocurre en una señal de voz o un electrocardiograma, sino que representa la **luz capturada** en cada posición del espacio. Además, la información de una imagen se representa en dos dimensiones.

Por tanto, en imágenes, de la misma forma que en series se caracterizó el ratio de incremento de la entropía respecto a nuevas muestras, podría pensarse en un ratio de entropía respecto a la unidad de área representada. Para la **estimación de la entropía en una imagen** se toma el histograma de los niveles de intensidad de esta.

La estimación final se obtiene como la entropía de la variable aleatoria caracterizada por dicho histograma.

Al igual que sucedía en señales unidimensionales, la entropía tenderá a incrementarse, o al menos permanecerá igual, si el área de la imagen considerada para la estimación se amplía. Así, menores valores de entropía se asociarán con patrones repetitivos en la imagen que conduzcan a un histograma con picos marcados (textura). En cambio, la entropía aumenta si existe una mayor variabilidad en los valores de intensidad observados en la imagen, no hay patrones marcados que produzcan un histograma más plano. En este sentido, se deduce que el ruido contribuye a aumentar la entropía de la imagen, ya que provoca que la variabilidad de los niveles de intensidad observados aumente.

4.3. Fuentes perturbadoras de las señales



Accede al vídeo «Fuentes perturbadoras de las señales» a través del aula virtual

Clasificación de los tipos de ruido

Las señales eléctricas se ven afectadas por diferentes tipos de ruido:

RUIDO EXTERNO
AL SISTEMA

Se identifica como ruido externo a cualquier señal no deseada procedente de fuentes ajenas al sistema de tratamiento de la información. Dentro de este grupo, se identifican diferentes tipos que se citan a continuación.

Figura 33. Definición de ruido externo al sistema.

Ruido atmosférico

El ruido atmosférico viene dado por las señales eléctricas derivadas de las descargas naturales que tienen lugar bajo la ionosfera. Las tormentas o las cargas eléctricas de las nubes son fuentes de este tipo de ruido que, generalmente, afecta en mayor medida a sistemas de comunicaciones que utilizan el espectro radioeléctrico como canal.

Aproximadamente, **la potencia del ruido atmosférico es inversamente proporcional a la frecuencia**. Así, el ruido atmosférico tiene mayor impacto en bandas de baja y media frecuencia, mientras que el ruido de menor potencia afecta a las bandas VHF y UHF. Como resultado, el ruido atmosférico afecta a la banda de comunicaciones AM y decrece significativamente en frecuencias de TV y FM. Más allá de los 30 MHz, el ruido atmosférico tiene menor impacto negativo que el propio ruido del receptor.

Producido por el hombre

Se refiere a los artefactos eléctricos generados por fuentes como automóviles, motores eléctricos, interruptores, líneas de alto voltaje... Se conoce también como **ruido industrial**. La intensidad de estas señales ruidosas cobra mayor magnitud en grandes núcleos urbanos y zonas industriales. En estas áreas, el ruido de esta naturaleza prevalece sobre otras fuentes de ruido en el rango de frecuencias entre 1 MHz y 600 MHz.

Ruido impulsivo o *shot*

Este tipo de ruido origina la aparición de valores anómalos (*outliers*) en la señal. Se caracteriza por un aumento brusco de la intensidad durante un corto periodo de tiempo. Por lo general, su origen es un agente externo al sistema de información: un rayo o una interferencia derivada de una chispa de un motor. Sin embargo, no debe confundirse con el ruido atmosférico o producido por el hombre, ya que la duración de estos es más prolongada en el tiempo.

Galáctico

Tiene su origen en las perturbaciones producidas más allá de la atmósfera terrestre. Como principales fuentes de ruido galáctico aparecen el sol y otras estrellas.

- ▶ Solar: el sol es una gran fuente de emisión de energía en forma de radiación electromagnética. Por tanto, estas señales afectan a los sistemas de telecomunicaciones. El **rango frecuencial** de estas emisiones es muy amplio, incluyendo las bandas comúnmente empleadas para los sistemas de comunicación por radio. La intensidad de la emisión producida por el sol varía de forma cíclica, con un período de unos once años aproximadamente. En los niveles más altos, esta radiación puede llegar a hacer inutilizables algunas bandas frecuenciales.
- ▶ Cósmico: de la misma forma que el sol, otras estrellas próximas a nuestro planeta emiten energía en forma de radiación electromagnética que puede llegar a afectar a nuestras señales y sistemas de comunicaciones.

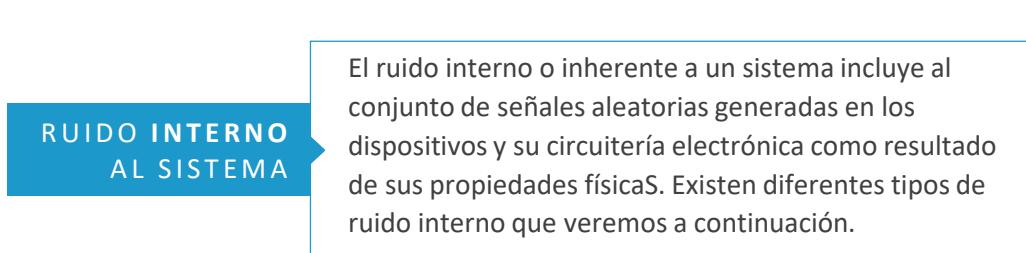


Figura 34. Definición de ruido externo al sistema.

Ruido térmico

Esta fuente de ruido se debe a la agitación aleatoria de los electrones en los elementos de un circuito electrónico. Este movimiento solo podría cancelarse en condiciones de cero absoluto de temperatura. Por tanto, se trata de una fuente de **ruido inevitable** que siempre va estar presente en un sistema de captación y procesado de señales. El movimiento de los electrones es mayor conforme la temperatura del conductor aumenta, dando lugar a pequeñas corrientes eléctricas. Esta señal ruidosa se distribuye en una amplio rango de frecuencias, de forma que

siempre afectará en cierto grado al sistema, a pesar de llevar a cabo diferentes etapas de filtrado.

Ruido de parpadeo (*flicker*) o 1/f

Se denomina 1/f debido a que su potencia decae por debajo de 1 KHz cuando la frecuencia aumenta. Por lo que tiene mayor impacto sobre las bajas frecuencias. Las causas físicas de este tipo de ruido no están del todo claras. Se origina en elementos como transistores o resistencias, y se hipotetiza que se debe a procesos de intermodulación en estos elementos.

Relación señal a ruido (SNR)

Ante una fuente de información afectada por artefactos ruidosos, el valor de la relación señal a ruido (SNR, *signal to noise ratio*) indica cuantitativamente **la calidad de la señal** de interés. Esta relación viene dada por el cociente entre la potencia de la señal recibida y la potencia estimada del ruido. Así, un valor mayor que la unidad indica una mayor presencia de señal frente al ruido. Generalmente, la relación entre ambos términos de potencia viene expresada en decibelios (dB), por lo que la SNR se calcularía como:

$$SNR = 10 \log(P_S/P_N)$$

Donde:

- ▶ P_S se corresponde con la potencia de la señal.
- ▶ P_N se corresponde con la potencia del ruido.

4.4. Caracterización matemática del ruido: procesos estocásticos



Accede al vídeo «Caracterización matemática del ruido: procesos estocásticos» a través del aula virtual

El término proceso estocástico ha sido previamente empleado en este tema para referirnos a una **señal aleatoria**. En nuestro caso, cualquier señal será el resultado de la combinación de la señal de interés y una señal no deseada, de naturaleza aleatoria y caótica, que contribuye a aumentar la entropía. Esta señal no deseada es el ruido.

Por lo que la señal resultante es, en sí misma, una señal aleatoria. Al igual que sucede con una variable aleatoria, de la que tomamos una muestra y obtenemos valores de acuerdo a una función de densidad de probabilidad, las señales que manejamos son realizaciones de un proceso estocástico. Cada vez que extraemos una muestra de la fuente de información, obtenemos una señal diferente.

En esta sección, se proporciona una definición formal de proceso estocástico que permita comprender el modelado y caracterización del ruido en el tratamiento de señales.

Una variable aleatoria está caracterizada por los siguientes tres elementos:

Espacio muestral. Es el conjunto de todos los resultados que pueden observarse en la realización de un experimento.

Conjunto de sucesos. Subconjunto del espacio muestral.

Ley de probabilidad. Asignación de probabilidad a cada uno de los sucesos observables.

Un proceso estocástico puede verse como una variable aleatoria para la que el resultado de un experimento viene dado en forma de señal. De la misma forma que una variable aleatoria, está caracterizado por la terna mencionada: espacio muestral, conjunto de sucesos y ley de asignación de probabilidades.

En la siguiente imagen vemos la interpretación del concepto de proceso estocástico X :

- ▶ Para diferentes realizaciones del proceso (a_1 , a_2 y a_3), se obtienen diferentes señales como resultado: $x(t, a_1)$, $x(t, a_2)$ y $x(t, a_3)$.
- ▶ Si nos fijamos en un único instante de tiempo t_0 , obtenemos muestras de la variable aleatoria $X(t_0)$.

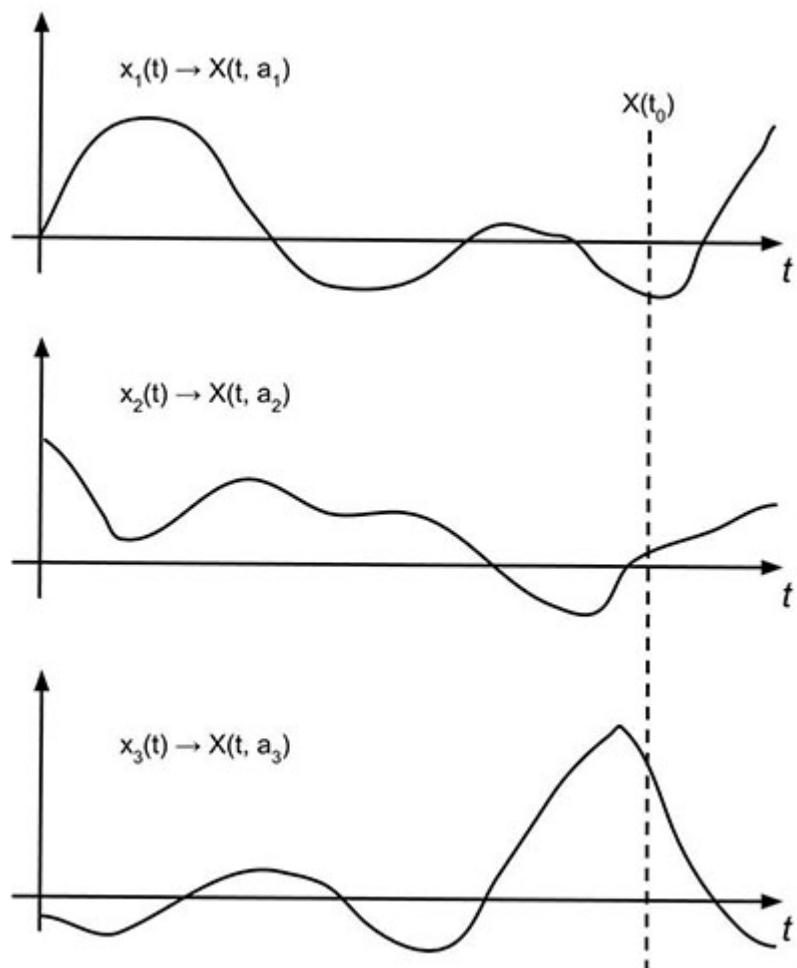


Figura 35. Interpretación del concepto de proceso estocástico.

En la práctica, como se ha mencionado previamente en este tema, tendremos señales ruidosas que, desde un punto de vista matemático, serán modeladas como un proceso estocástico. La componente de ruido se supondrá aditiva, por lo que la señal capturada $y(t)$ tendrá la siguiente forma:

$$y(t) = x(t) + \varepsilon(t)$$

Donde:

- ▶ $x(t)$ refleja la señal de interés.
- ▶ Y el término $\varepsilon(t)$ corresponde al ruido.

Considérese, por ejemplo, que la señal de interés corresponde a un tono de frecuencia f y que la componente ruidosa obedece a una distribución gaussiana de media nula y varianza s .

En el siguiente gráfico, podemos ver esta señal objetivo (parte superior) y una realización del proceso estocástico que corresponde a la señal observada (parte inferior).

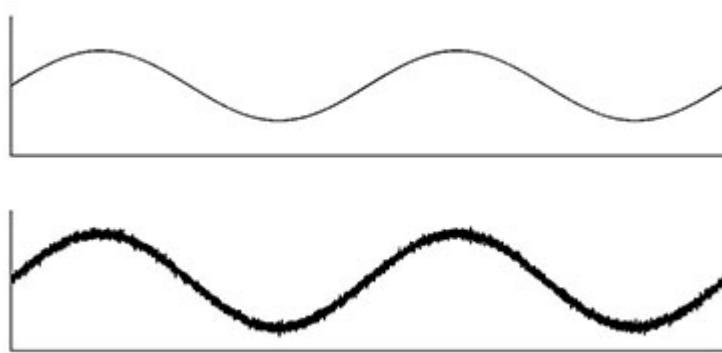


Figura 36. Ejemplo de una señal ruidosa modelada como un proceso estocástico.

Como puede apreciarse, el componente de ruido dota a la señal de naturaleza aleatoria que nos impide conocer con exactitud el valor de la misma en un instante t . A fin de caracterizar el proceso estocástico, el objetivo será conocer las propiedades

estadísticas del mismo. Las funciones de distribución y densidad de probabilidad permiten modelar estadísticamente el proceso. Estas funciones vendrían dadas de la siguiente forma:

- ▶ Función de distribución: $F_X(x, t) = P(X(t) \leq x)$
- ▶ Función de densidad de probabilidad: $f_X(x, t) = dF_X(x, t)/dx$

A partir de estas funciones, puede definirse la estacionariedad del proceso:

- ▶ Un proceso es estacionario en **sentido estricto** si la función de densidad de probabilidad que caracteriza al proceso no varía con el tiempo. Es decir, para c constante tal que $c > 0$, se tendrá lo siguiente: $F_X(x, t) = f_X(x, t + c)$
- ▶ Un proceso es estacionario en **sentido amplio** si los momentos estadísticos que lo caracterizan (media, varianza, etc.) no varían respecto al tiempo.

Retomemos el ejemplo anterior. En esta ocasión, la señal capturada muestra, además del ruido gaussiano, otro componente que provoca una clara tendencia a lo largo del tiempo. El siguiente gráfico recoge este nuevo ejemplo. Como resultado de esta tendencia, las propiedades estadísticas de la señal no se mantienen a lo largo del eje temporal, por lo que no puede considerarse una señal estacionaria. Será necesario eliminar el componente de ruido que provoca esta tendencia para eliminar la no estacionariedad presente en nuestra información.

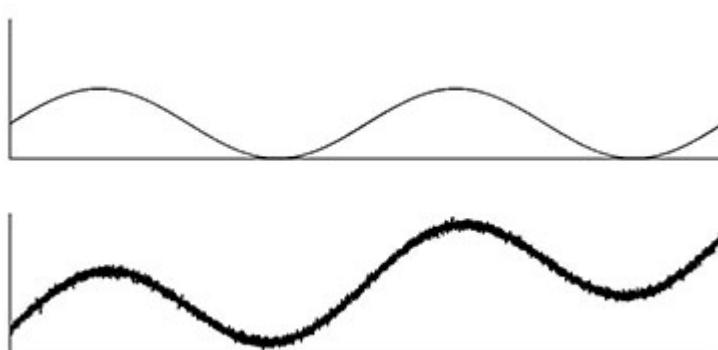


Figura 37. Proceso estocástico no estacionario.



Accede a los ejercicios de autoevaluación a través del aula virtual

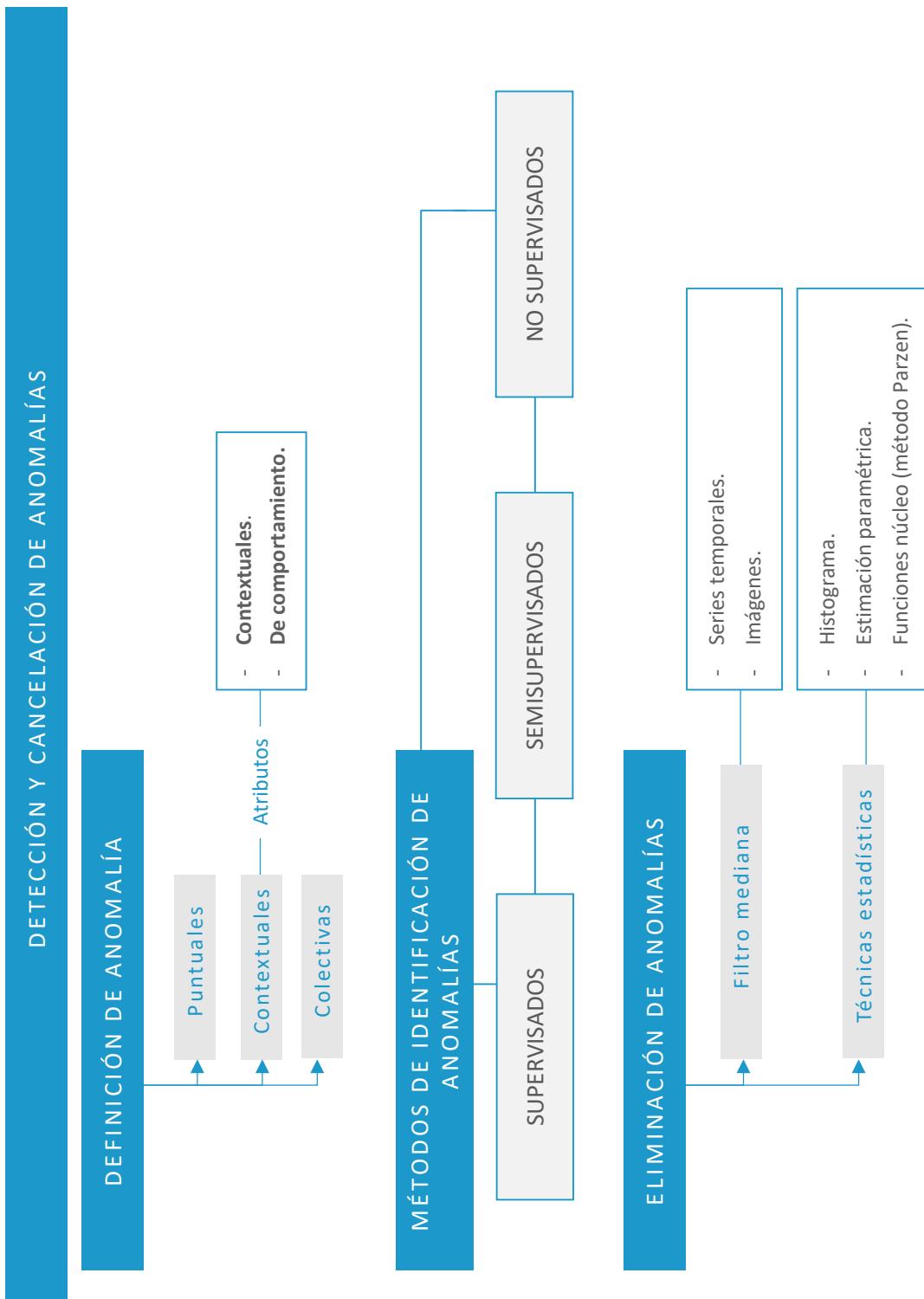


Accede al vídeo «Resumen del tema» a través del aula virtual

Percepción Computacional

Detección y cancelación de anomalías

Esquema. Tema 5



Ideas clave. Tema 5

5.1. ¿Cómo estudiar este tema?



Accede al vídeo «Introducción. Detección y cancelación de anomalías» a través del aula virtual

Para estudiar este tema deberás leer con atención las ideas clave que se desarrollan a continuación.

En el siguiente tema se aborda, como una etapa más en la preparación de las señales capturadas, la detección y eliminación de muestras de datos atípicas conocidas como anomalías (*outliers*).

El tema se organiza de la siguiente forma:

- ▶ Definición de anomalía. En primer lugar, se proporciona una definición de anomalía en el contexto del procesado de datos y señales. Así, se indican los **diferentes tipos de anomalías** que podremos observar en un conjunto de datos y las características de cada uno de ellos.
- ▶ Métodos de identificación. Se indica la taxonomía de los métodos que pueden emplearse para la **identificación** de anomalías.
- ▶ Eliminación de anomalías. Por último, se profundiza en aquellos métodos que son comúnmente empleados para la **detección y eliminación** de anomalías en señales, dado que estas son la fuente de información con la que se trabaja en la asignatura.

El siguiente esquema permite memorizar de forma sencilla los elementos de este tema.

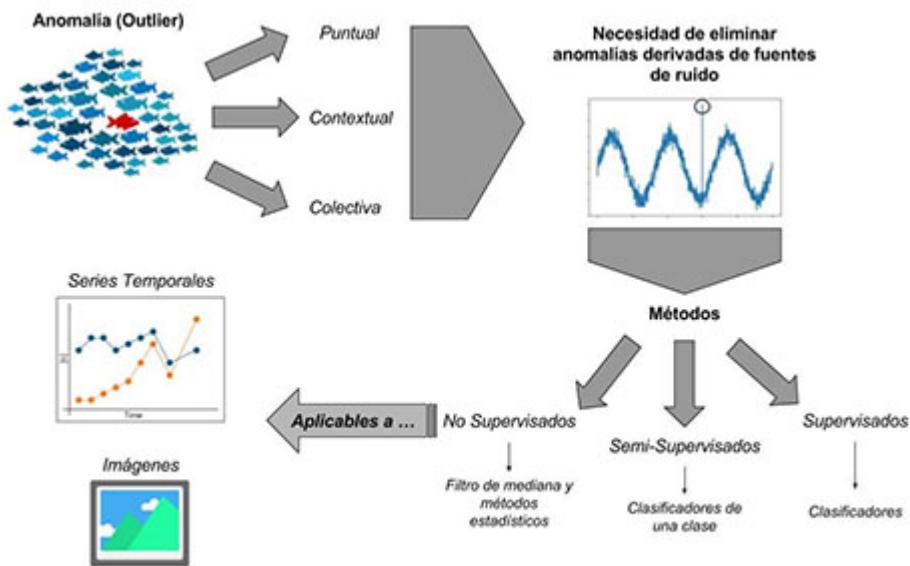


Figura 38. Conceptos relacionados con la presencia e identificación de anomalías en señales como series temporales e imágenes.

5.2. Definición de anomalía



Accede al vídeo «Definición de anomalía» a través del aula virtual

La detección de anomalías tiene como objetivo la **identificación de valores atípicos en la fuente de información**, comúnmente conocidos por su vocablo en inglés, *outliers*. Estos se definen como patrones inusuales que no se ajustan al comportamiento esperado. La aparición de *outliers* en una señal o imagen refleja la existencia de ruido, generalmente de tipo impulsivo motivado, por ejemplo: por un valor de pico en un campo eléctrico cercano, o de inestabilidades en el procedimiento de captura, por ejemplo: el movimiento brusco de una cámara.

La detección de anomalías tiene aplicación directa en diferentes escenarios prácticos. A continuación se citan algunas:

- ▶ Detección de intrusos en una red. Identificación de patrones atípicos en el tráfico de red que pueden indicar un ataque.
- ▶ Diagnóstico médico. Reconocimiento de lesiones con una bajo índice en la población que pueden indicar la existencia de alguna patología.
- ▶ Detección de transacciones fraudulentas. La gran mayoría de las transacciones que se realizan son lícitas y solo una pequeña proporción se corresponden con actividades fraudulentas.
- ▶ Predicción de fuga de clientes en grandes compañías. En los sectores de banca, seguros y telecomunicaciones, una pequeña parte de los clientes abandona la compañía, por lo que la identificación de estos comportamientos puede realizarse mediante técnicas de detección de anomalías.

Existen diferentes tipos de anomalías. Vamos a definir cada uno de ellos:

Anomalías puntuales

En el caso de que una muestra individual pueda considerarse notablemente diferente respecto al resto de los datos, esta puede ser tomada como un *outlier*. Este tipo de anomalía es el más simple y el foco de la mayoría de los trabajos de investigación sobre este tema.

Un claro ejemplo correspondiente a un escenario real sería el fraude cometido con una tarjeta de crédito. Si nos fijamos en una variable como la cuantía de la transacción, aquellas transacciones para las que la cantidad sea muy alta en comparación con el promedio del resto de transacciones previas son susceptibles de ser anomalías puntuales y, por tanto, sospechosas de fraude. Así, una anomalía puntual se expresa mediante la aparición de **valores pico** que se alejan excesivamente del conjunto de valores que encontramos.

En la siguiente figura podemos ver una señal en la que una de las muestras toma un valor que no observamos en ninguna otra. Se trata, claramente, de una muestra candidata a ser una anomalía.

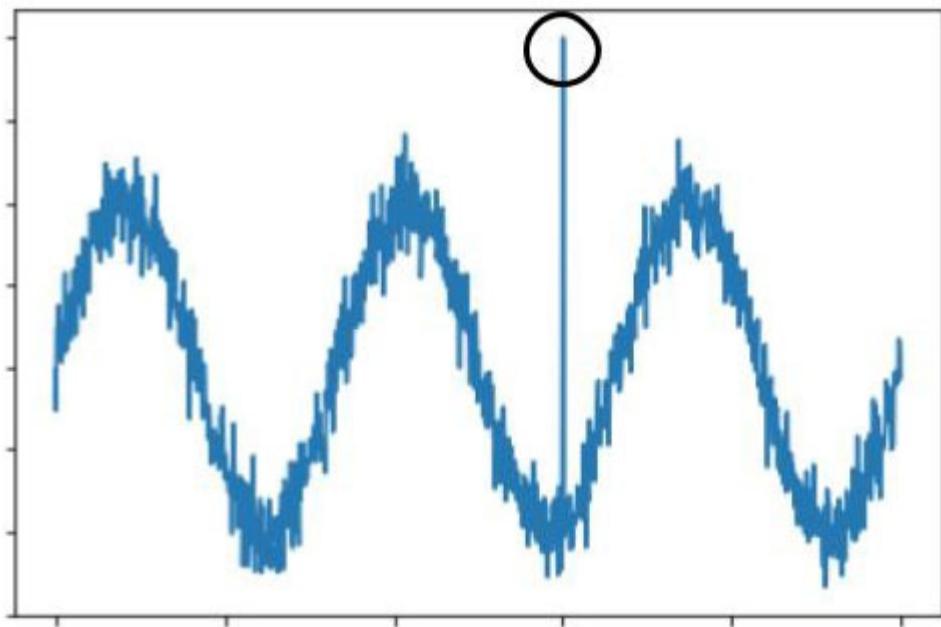


Figura 39. Ejemplo de anomalía puntual en una señal temporal.

Anomalías contextuales

Si una muestra de datos es anómala en un contexto específico (pero no de otro modo), se denomina anomalía contextual. La noción de contexto viene dada por la naturaleza de los datos. Cada muestra de datos se define teniendo en cuenta los siguientes atributos.

Atributos contextuales: estos se usan para determinar el contexto (o vecindad) para esa muestra. Vienen dados por la **naturaleza de la fuente de datos**. Por ejemplo, en conjuntos de datos espaciales, la longitud y la latitud de una ubicación son los atributos contextuales. En una serie temporal, el tiempo es un atributo contextual que determina la posición de una muestra en toda la secuencia.

Atributos de comportamiento: estos definen el carácter no contextual de una instancia. Es decir, representa el **valor de la muestra**. Siguiendo con el ejemplo de los datos espaciales, si se trata de cuantificar la precipitación promedio en cualquier punto de la superficie mundial, la cantidad de lluvia en cualquier lugar es un atributo de comportamiento.

El comportamiento anómalo se determina usando los valores de los atributos de comportamiento dentro de un contexto específico.

Una instancia de datos podría ser una anomalía contextual en un determinado contexto, pero una instancia de datos idéntica (en términos de atributos de comportamiento, es decir, de su valor) podría ser considerada normal en un contexto diferente. Esta propiedad es clave para identificar atributos contextuales y de comportamiento para una técnica de detección de anomalía contextual.

A diferencia de las anomalías puntuales definidas previamente, en las que únicamente se lleva a cabo una comparación de las muestras de datos disponibles para identificar un valor atípico, en señales temporales (series temporales) e imágenes se tiene en cuenta el contexto para definir un valor anormal.

Por ejemplo, en una imagen, es posible identificar un píxel anómalo si su intensidad es muy diferente a la de los píxeles vecinos. De la misma forma, en una serie temporal también será el entorno de un punto el que nos dé la información contextual necesaria para identificar un valor anómalo, tal y como se ejemplifica en la siguiente imagen.

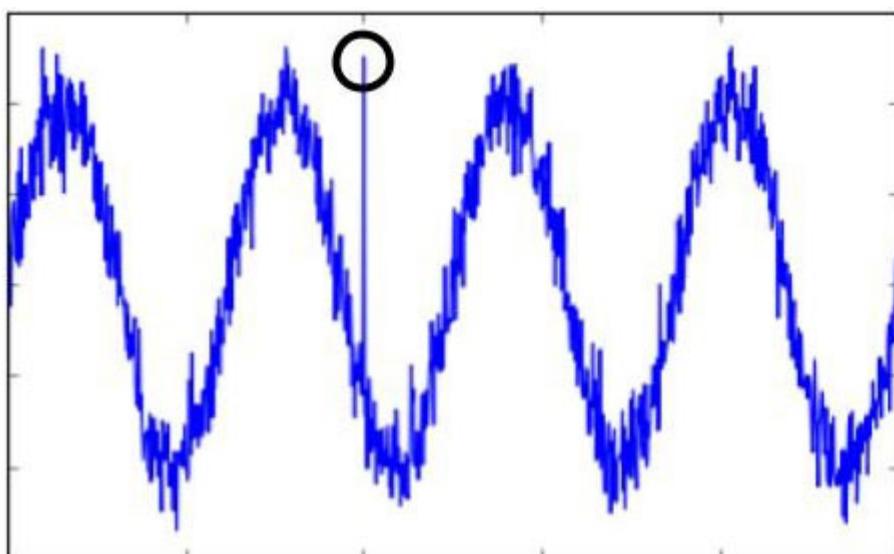


Figura 40. Ejemplo de anomalía contextual identificada en una serie temporal.

En ella vemos que la serie toma en algún momento valores similares al de la anomalía, pero el contexto nos indica que en este caso es una muestra atípica.

Anomalías colectivas

Si una **colección de instancias de datos** relacionadas es anómala con respecto a todo el conjunto de datos, se denomina anomalía colectiva. Las instancias de datos individuales en una anomalía colectiva pueden no ser anomalías por sí mismas, pero su ocurrencia conjunta como colección es anómala.

El siguiente gráfico ilustra un ejemplo de anomalía colectiva en una señal electrocardiográfica. La región resaltada denota una anomalía porque la señal toma aproximadamente el mismo valor durante un tiempo inusualmente largo. Sin embargo, ese valor no es una anomalía por sí mismo.

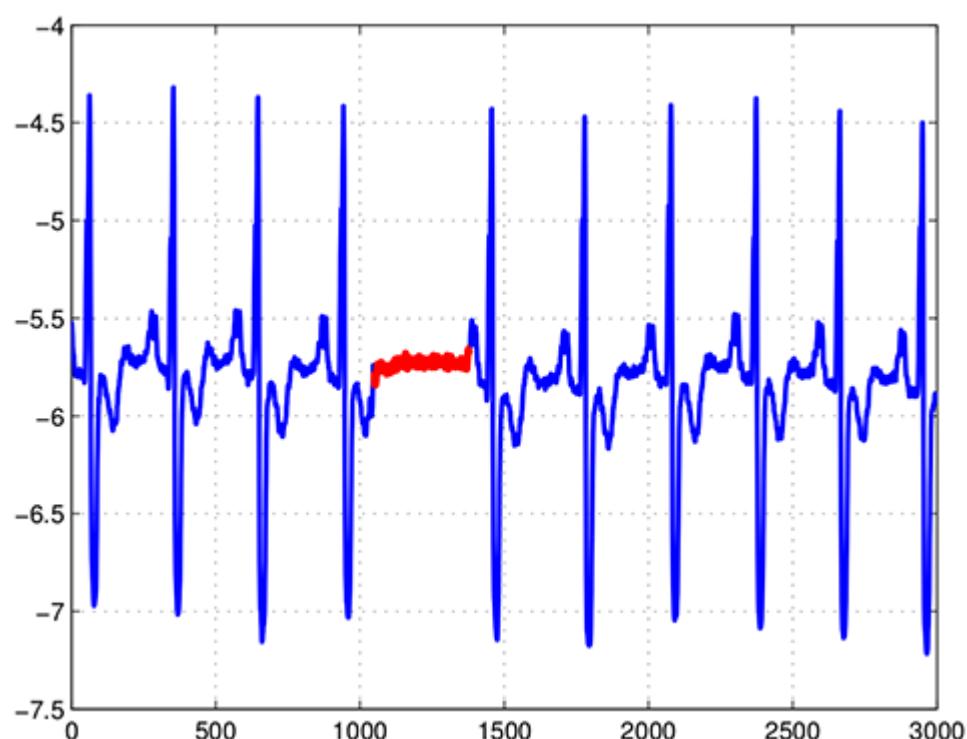


Figura 41. Ejemplo de anomalía colectiva en una señal de ECG.

Fuente: <https://www.datasciencecentral.com/profiles/blogs/anomaly-detection-for-the-oxford-data-science-for-iot-course>.

En este tema se tratará de identificar la presencia de anomalías en nuestras fuentes de información como una etapa más en la limpieza y preparación de la señal. Nos ceñiremos al escenario en el que la anomalía se corresponde con un artefacto ruidoso que debería ser eliminado a fin de preservar la calidad de la información.

En este sentido, los tres tipos de anomalías descritos previamente pueden tener como origen un elemento perturbador de la señal. Generalmente, las anomalías o artefactos a los que nos enfrentaremos vendrán dados por valores de pico inusuales derivados de la presencia de **fuentes ruidosas de tipo impulsivo**.

5.3. Métodos de identificación de anomalías



Accede al vídeo «Métodos de identificación de anomalías» a través del aula virtual

Adiferencia de los problemas convencionales de clasificación, donde se cuenta con un conjunto de datos de entrenamiento, previamente etiquetados con su correspondiente clase, y un conjunto de test que permite la estimación objetiva del rendimiento del modelo, hay múltiples configuraciones posibles cuando se habla de detección de anomalías.

Básicamente, la configuración de detección de anomalías que se utilizará depende de las etiquetas disponibles en el conjunto de datos. Así, podemos distinguir entre tres tipos principales:

Métodos supervisados

En este caso, se dispone de un conjunto de datos en los que cada muestra está asociada a una etiqueta que indica si es o no una anomalía. A partir de estos datos, se puede **entrenar un clasificador que capture el patrón característico de la**

anomalía. Posteriormente, este clasificador se emplearía como identificador automático de estos valores atípicos.

Este escenario es muy similar al reconocimiento tradicional de patrones en el que las dos clases (no anomalías vs. anomalía) están fuertemente desequilibradas. Como consecuencia, debe tenerse en cuenta que no todos los algoritmos de clasificación se adaptan perfectamente a esta tarea, pues depende de la función de error que trate de optimizarse durante su entrenamiento.

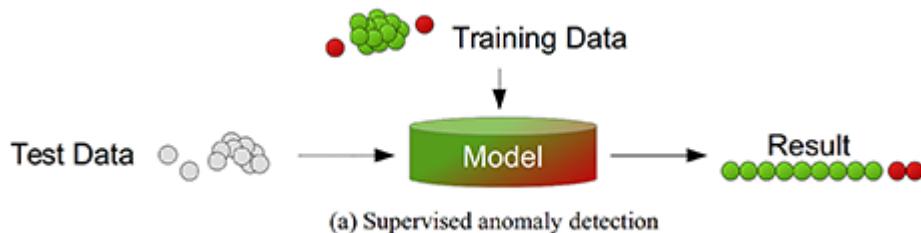


Figura 42. Métodos y escenarios para la implementación de técnicas de detección de anomalías: supervisado.
Fuente: Goldstein y Uchida, 2016.

Por ejemplo: los árboles de decisión, que buscan divisiones de los datos en las que se maximice la pureza de las categorías, no responden bien a conjuntos desequilibrados, mientras que las máquinas de vector de soporte (SVM, *Support Vector Machines*) o las redes neuronales artificiales (ANN, *Artificial Neuronal Network*) proporcionarán un resultado más plausible.

Sin embargo, la identificación de anomalías basada en técnicas supervisadas **no está muy extendida**, pues el punto de partida es la suposición de que las anomalías son conocidas y etiquetadas correctamente. Para muchas aplicaciones, las anomalías no se conocen de antemano o pueden ocurrir espontáneamente como novedades durante la fase de test, por lo que estas técnicas no tendrían cabida.

La utilización de modelos derivados del aprendizaje supervisado como herramientas de identificación de anomalías es común en aplicaciones prácticas como la detección de fraude o el diagnóstico médico, donde el número de muestras positivas es notablemente menor en proporción y, por tanto, se consideran una anomalía.

Métodos semisupervisados

Se trata de un procedimiento similar al anterior, pues también se emplea un conjunto de datos para entrenar un modelo. Sin embargo, en este caso, los datos de entrenamiento **contienen únicamente muestras no anómalas**. La idea fundamental es que el modelo aprenda la clase normal, de forma que se detectaría la anomalía al identificarse una desviación del patrón aprendido.

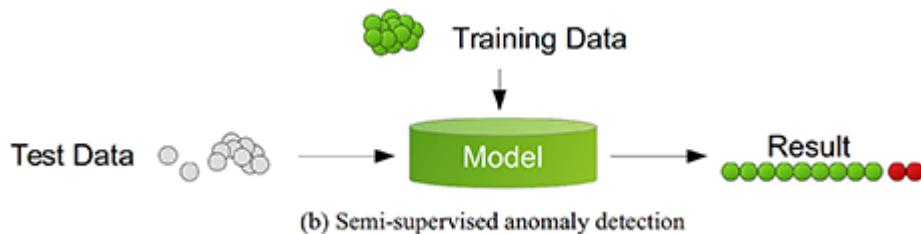


Figura 43. Métodos y escenarios para la implementación de técnicas de detección de anomalías: semisupervisado.

Fuente: Goldstein y Uchida, 2016.

Esta aproximación se conoce como clasificadores de una sola clase (*one-class classifiers*). Usualmente, suelen emplearse para tal fin los algoritmos SVM de una clase y los *autoencoders*. Además, cualquier método de estimación de densidad puede usarse para modelar la función de densidad de probabilidad de las muestras normales. Por ejemplo, los modelos de mezcla gaussiana o la estimación basada en funciones núcleo.

Métodos no supervisados

Representa el procedimiento **más flexible** para la implementación de un método de identificación de anomalías pues no requiere de ningún tipo de conocimiento previo sobre los datos (etiquetas). La idea es que un algoritmo de detección de anomalías no supervisado califique los datos únicamente en función de las propiedades intrínsecas de estos. Normalmente, las **distancias o densidades** se utilizan para dar una estimación de lo que es normal y lo que es un valor atípico.

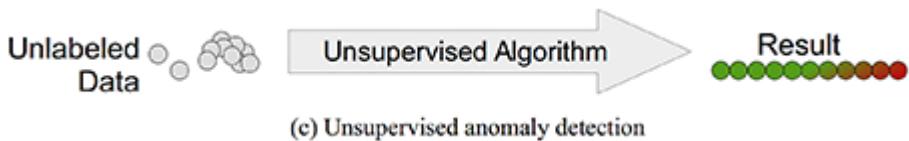


Figura 44. Métodos y escenarios para la implementación de técnicas de detección de anomalías: no supervisado.

Fuente: Goldstein y Uchida, 2016.

La salida devuelta por un algoritmo de detección de anomalías puede ser de dos tipos diferentes. Por un lado, puede emplearse **una etiqueta para cada muestra**. Esta indica de forma categórica si una instancia es anómala o no. Este es el caso, por lo general, de los métodos basados en aprendizaje supervisado.

En segundo lugar, puede obtenerse un **valor continuo** que refleje el grado de anormalidad de la muestra. Los algoritmos de detección de anomalías semisupervisados y no supervisados producen una salida de este tipo. Esto se debe a razones prácticas, pues el procedimiento de identificación solo informa de los casos más sospechosos, por lo que es necesario establecer un orden en las instancias a partir del grado de anomalía asignado.

El objetivo de la asignatura es el manejo de señales que, generalmente, se corresponden con:

- ▶ Series temporales: señales unidimensionales con el tiempo como variable independiente.
- ▶ Imágenes: señales bidimensionales con las coordenadas espaciales como variables independientes.

En este contexto, como se indicó previamente, las anomalías son el resultado de una fuente ruidosa que conlleva la aparición de elementos espurios en la señal. Por tanto, el propósito de la etapa de detección de anomalías es eliminarlas para mejorar la calidad de la señal en etapas posteriores. Es decir, no formaría parte de esta etapa la identificación de patrones anómalos de naturaleza no ruidosa en la señal.

Por ejemplo: el resultado de una arritmia reflejada en una señal de ECG o la presencia de determinados objetos en una imagen. De acuerdo a este contexto, el procedimiento habitual para la detección y eliminación de anomalías será la utilización de **métodos no supervisados** dado que, inicialmente, no disponemos de ninguna descripción de las posibles anomalías que pudiera haber en la señal.

5.4. Eliminación de anomalías



Accede al vídeo «Eliminación de anomalías» a través del aula virtual

A

continuación se explican los procedimientos no supervisados más habituales para la eliminación de anomalías en señales.

Filtro de mediana

El filtro de mediana ha sido comúnmente empleado sobre señales 1D y 2D para la eliminación de ruido impulsivo. Estos artefactos se reconocen fácilmente mediante la inspección visual de la señal, pues están asociados a valores de pico que destacan notablemente sobre el resto de la señal (anomalía puntual) o sobre el entorno más cercano (anomalía de tipo contextual, pues el valor atípico iría en discordancia con aquellos de su entorno).

En imágenes, este tipo de anomalías se conoce como ruido «sal y pimienta» (*salt & pepper*), ya que el efecto que genera es el de píxeles colocados de forma aleatoria que toman valores extremos de intensidad (1 o 0). Aquí vemos una imagen afectada por este tipo de ruido.



Figura 45. Imagen con ruido de tipo «sal y pimienta» reflejado en píxeles con una valor extremo de intensidad.

Fuente: http://in.mathworks.com/matlabcentral/mlc-downloads/downloads/submissions/16201/versions/3/previews/toolbox_image/html/content.html.

El filtro de mediana es una operación que se aplica punto a punto mediante una ventana deslizante. El tamaño de esta viene determinado por el usuario.

En el caso de señales unidimensionales como series temporales, se trata de una ventana de longitud N , mientras que en imágenes la ventana se define en ambas coordenadas y es de tamaño $N \times N$. El valor de N es impar, ya que la ventana se centra en el punto de la señal que se pretende filtrar. Así, el valor resultante en este punto viene dado por la mediana de los puntos considerados por la ventana.

El filtro, tal y como se puede apreciar en su definición, no crea nuevos valores de la señal, sino que **selecciona como salida uno de los valores entrantes**. A continuación veremos el resultado de aplicar el filtro de mediana a una serie temporal y a una imagen.

En la mitad izquierda del siguiente gráfico tenemos una serie temporal donde pueden observarse dos puntos que representan sendas anomalías frente al resto de valores de la señal. Para eliminar estos *outliers*, se aplica un filtro de mediana con una ventana de longitud 3. El resultado del filtro aparece en la parte derecha del gráfico. Tal y como se aprecia, el filtro permite suprimir los valores atípicos, estimando su valor como la mediana de los puntos de su vecindad.

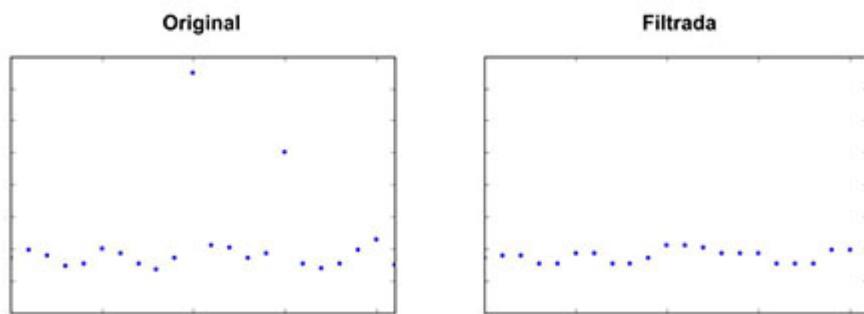


Figura 46. Aplicación de un filtro de mediana sobre una serie temporal afectada por la presencia de *outliers*.

Y a continuación vemos el resultado de utilizar el filtro de mediana para la eliminación de ruido de tipo «sal y pimienta» en la imagen anterior (figura 8). El tamaño de ventana empleado ha sido 3×3 . Como puede verse en la imagen filtrada, la aplicación del filtro conlleva cierta distorsión de la imagen, pues se modifica el valor de intensidad de los píxeles que no están afectados por este tipo de ruido.

El filtro de mediana es muy similar a un filtro de promedio que obtendría, para cada ventana, el valor medio de los píxeles o puntos considerados. Esta operación es equivalente a la utilización de un filtro paso-bajo en frecuencia, por lo que las variaciones rápidas de la señal, reflejadas como contrastes significativos en una imagen, quedan suavizados por el filtro.

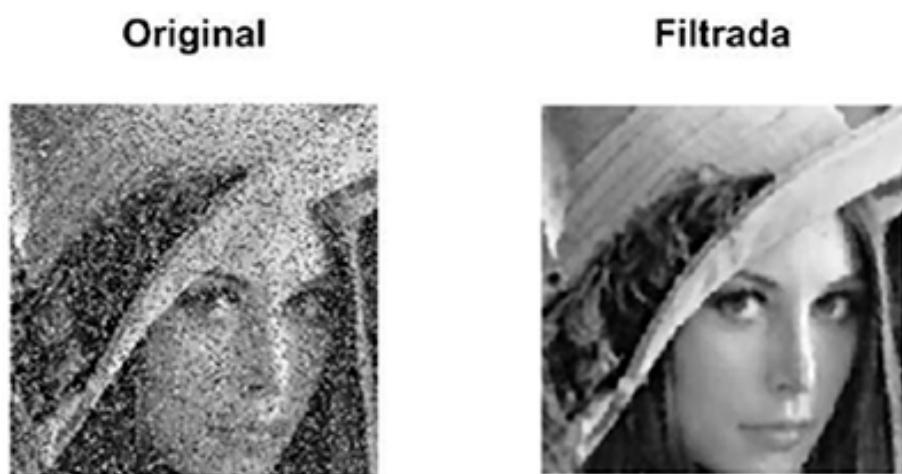


Figura 47. Efecto del filtro de mediana sobre una imagen con ruido impulsivo «sal y pimienta».

Fuente: Adaptado de http://in.mathworks.com/matlabcentral/mlc-downloads/downloads/submissions/16201/versions/3/previews/toolbox_image/html/content.html#17 .

Técnicas estadísticas

Otra técnica de detección y corrección de anomalías empleada habitualmente se basa en la utilización de la **función de densidad de probabilidad de los datos**. Dada la función de densidad $f(x)$, donde x es uno de los valores que puede tomar la variable aleatoria correspondiente, puede obtenerse una medida que cuantifica el grado de anomalía para la muestra x_1 como la inversa de $f(x_1)$.

Aquellos valores muy poco probables tenderán a ser identificados como atípicos y, por tanto, debe emplearse una estrategia adecuada para su tratamiento, por ejemplo: eliminarlos o estimar su valor como la media de los puntos vecinos.

Otra estrategia para la identificación de valores atípicos a partir de la función $f(x)$ sería considerar como tales aquellos **que estén en los extremos** del dominio de $f(x)$. Por ejemplo, a partir de $f(x)$ puede identificarse los valores x_a y x_b , tales que:

$$P(x \leq x_a) = P(x \geq x_b) = P_{min}$$

Si la muestra x_1 es menor que x_a o mayor que x_b , sería considerada una anomalía.

La utilización de estos procedimientos puede llevarse a cabo de forma global o local:

- ▶ En el primer caso, la función $f(x)$ corresponde al conjunto total de muestras disponibles. Es decir, todos los puntos de la serie temporal o todos los píxeles de la imagen. En este caso, podríamos detectar **anomalías de tipo puntual**.
- ▶ A fin de identificar **anomalías contextuales**, sería necesario aplicar este método en un entorno local del punto a estudiar. Para ello, la función $f(x)$ se correspondería únicamente con la vecindad del punto que se pretende evaluar como anomalía. Tal y como sucede con el filtro de mediana explicado previamente, sería necesario emplear una ventana deslizante centrada en el punto objetivo para definir dicha vecindad.

Además, ambos métodos basados en $f(x)$ requieren fijar inicialmente un umbral de decisión: para comparar el *score* obtenido con el primero de los métodos y para definir el valor P_{min} que identifica los valores extremos de una distribución en el caso del segundo procedimiento.

Este umbral determina la definición de anomalía en nuestro conjunto de datos y que ha de ser establecido por el usuario.

En la definición de los métodos basados en el empleo de la función de densidad de probabilidad $f(x)$ se ha asumido el conocimiento de esta. Sin embargo, esta función es desconocida, por lo que será necesario aplicar técnicas de estimación para obtener una aproximación a la misma.

Estas son algunas técnicas que pueden ser utilizadas para la estimación de esta función.

Histograma

Representa la técnica más sencilla. A partir de las muestras de una variable, esta se discretiza mediante la división de su dominio en un número limitado de intervalos de igual tamaño e identificados por su punto medio. Estos puntos representan los valores discretos que la variable puede tomar. Así, se obtiene la frecuencia (probabilidad) asociada a cada posible valor a partir del conjunto total de datos inicial contando el número de muestras de la variable que caen en cada intervalo.

La **elección del número de intervalos** empleado para la discretización de la variable tiene una influencia muy significativa en la aproximación obtenida. Un número demasiado pequeño de intervalos resultará en una aproximación excesivamente simple que no captura las particularidades de la distribución objetivo. Sin embargo, un número excesivo de intervalos conlleva que la estimación resultante presente discontinuidades (valores nulos) y cambios bruscos en su perfil.

Existen diferentes reglas para obtener el número óptimo de intervalos a considerar. Entre ellas, una de las más comunes es la **regla de Freedman-Diaconis**, que viene dada por la siguiente expresión:

$$T = 2IQR(x)/\sqrt[3]{n}$$

Donde:

- ▶ x hace referencia a la muestra disponible de la variable.
- ▶ n es el número de muestras.
- ▶ IQR es el rango intercuartil. Este se obtiene de la diferencia entre los percentiles 75 % y 25 %, y representa el grado de dispersión de la muestra, es decir, si la distribución está extendida en el dominio de x o si, por el contrario, está concentrada en un rango de valores.

Estimación paramétrica

Se asume que la función de densidad de probabilidad que caracteriza estadísticamente a la variable es de tipo normal. Por tanto, la media y varianza de esta distribución son los parámetros a obtener. Para ello, se emplean las estimaciones derivadas de la muestra disponible:

$$\mu_x = 1/n \sum_{i=1}^n x_i$$

$$\sigma_x = \sqrt{1/n \sum (x_i - \mu_x)^2}$$

Obviamente, la principal limitación de este método viene dada por la suposición inicial sobre la forma de la distribución. El error en la estimación será más representativo, por tanto, cuanto más difiera la distribución real de la variable del perfil normal inicialmente supuesto.

Funciones núcleo (método de Parzen)

Se trata de un procedimiento híbrido entre la estimación basada en histogramas y la estimación paramétrica. En este caso, la estimación de la función de densidad de probabilidad viene dada por la superposición de funciones núcleo centradas en cada uno de las muestras x_i observadas inicialmente. La expresión de la función estimada se obtiene de la siguiente forma:

$$\hat{f}_x = 1/n \sum_{i=1}^n g(x - x_i, \theta),$$

Donde:

- ▶ $g(x, \theta)$ es la función núcleo.
- ▶ θ representa el conjunto de parámetros de esta función.

Comúnmente, se emplea una normal Gaussiana como función núcleo, de forma que el conjunto de parámetros θ viene dado únicamente por la varianza de la normal, dado que cada función núcleo se centra en la muestra x_i correspondiente.

Es habitual emplear el mismo valor de varianza para el conjunto de funciones núcleo, por lo que la función de densidad de probabilidad estimada se obtendría como:

$$\hat{f}_x = 1/n \sum_{i=1}^n 1/\sqrt{2\pi\sigma^2} \exp[-(x_i - x_i)^2/(2\sigma^2)]$$

Como se observa, el efecto de la varianza de las funciones núcleo normales es similar al del tamaño del intervalo para el cálculo del histograma. De hecho, el histograma puede verse como un caso particular de estimación basada en funciones núcleo, en el que estas funciones vendrían dadas por pulsos uniformes de altura igual a la unidad y centrados en el punto medio del intervalo.

Una regla extendida para la obtención de un valor adecuado de la varianza de las funciones núcleo es fijar esta al siguiente valor:

$$\sigma = 1.06\sigma_x n^{-1/5}$$

5.5. Referencias bibliográficas

Goldstein, M. y Uchida, S. (2016). A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data. *PLOS ONE*, 11(4): e0152173.

Recuperado de <https://doi.org/10.1371/journal.pone.0152173>



Accede a los ejercicios de autoevaluación a través del aula virtual

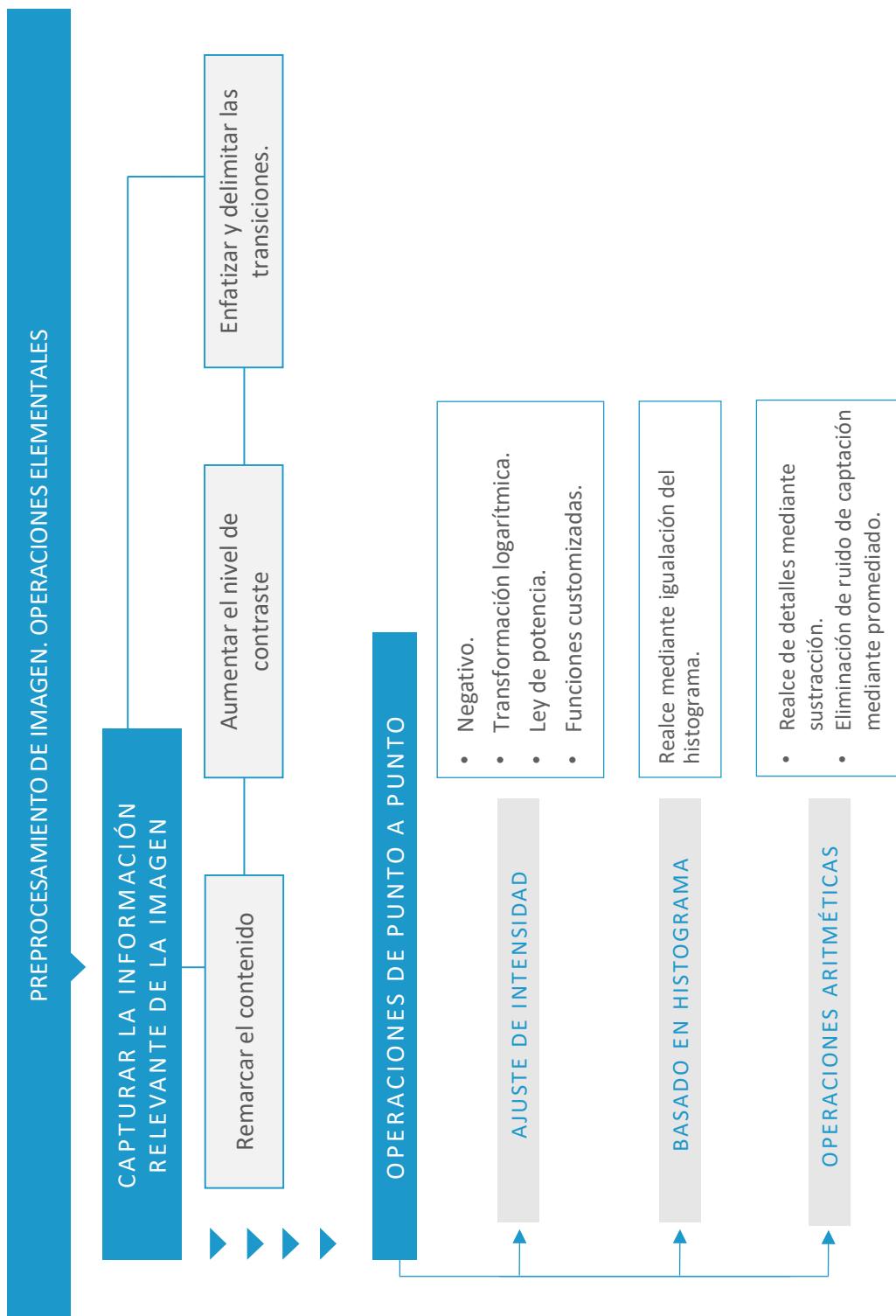


Accede al vídeo «Resumen del tema» a través del aula virtual

Percepción Computacional

Procesamiento de imagen. Operaciones elementales

Esquema. Tema 6



Ideas clave. Tema 6

6.1. ¿Cómo estudiar este tema?



Accede al vídeo «Introducción. Operaciones elementales» a través del aula virtual

Para estudiar este tema deberás leer con atención las ideas clave que se desarrollan a continuación.

Dentro del marco de la asignatura, este tema se centra en el procesado de imagen y, concretamente, se ubica en las primeras etapas de tratamiento de una imagen. En él se describen técnicas básicas para comenzar el procesado y que persiguen realizar el contenido de la imagen bajo estudio. Es decir, en un contexto o aplicación determinados, se busca **capturar la información relevante de la imagen** frente a la más prescindible.

Formalmente, las operaciones de realce de imagen consisten en la utilización de una o varias técnicas para resaltar en ella determinadas propiedades. Estas características a magnificar dependen del propósito final del escenario práctico en el que nos encontramos. Por ejemplo, lesiones concretas en imágenes sobre tejidos corporales para la identificación de una patología o los bordes de un objeto en un sistema de visión automática.

A modo ilustrativo, resumimos las siguientes operaciones que pertenecen al ámbito del realce de imagen:

- ▶ La supresión del ruido de fondo derivado del proceso de captura de una imagen. En este caso, se pretende **remarcar el contenido** de la imagen en detrimento de la textura en el fondo de esta. El objetivo es que el fondo de la imagen presente

una intensidad lo más constante posible, ya que el detalle de interés no está en esa parte de la señal.

- ▶ El ajuste de intensidad que persigue **aumentar el nivel de contraste** de la imagen, de forma que puedan distinguirse con mayor nitidez los elementos presentes en ella.
- ▶ El realce de bordes, cuyo objetivo es **enfatizar y delimitar las transiciones** entre objetos diferentes a fin de poder ser distinguidos posteriormente con mayor facilidad.

El propósito final de las operaciones de realce de imagen es facilitar las etapas posteriores destinadas a la extracción de información. Así, estas etapas pueden ser llevadas a cabo por:

- ▶ Un observador humano. El resultado de las operaciones de realce debe hacer que la imagen sea más fácilmente «digerible», es decir, que puedan distinguirse con mayor nitidez las estructuras y elementos implicados en el caso de uso a tratar.
- ▶ Una máquina. Las operaciones de realce representan, en este caso, una etapa de preprocessado, cuyo objetivo es generar inputs adecuados para etapas posteriores destinadas a extraer información de la imagen. Así, el indicar cuáles son los bordes existentes entre diferentes estructuras puede no generar una imagen fácilmente interpretable para un humano, pero sí para una etapa posterior del sistema automático de procesado.

Existen diferentes operaciones para el realce de imagen. En este tema, se pone el foco en las **operaciones elementales de tipo punto a punto**. Estas operaciones se caracterizan por que el valor de un píxel en la imagen procesada es función únicamente del valor de ese mismo píxel en la imagen original. Matemáticamente, se expresa de la siguiente forma:

$$B(x, y) = T(A(x, y))$$

Donde:

- ▶ B es la imagen procesada.
- ▶ A es la imagen original.
- ▶ La coordenada (x, y) identifica al píxel sobre el que se aplica la operación.
- ▶ T es la función u operador de transformación.

La siguiente figura esquematiza la aplicación de este tipo de operadores.

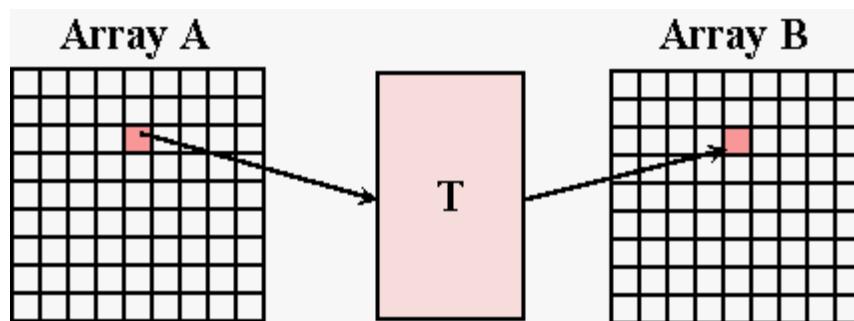


Figura 48. Aplicación de un operador punto a punto T sobre una imagen A y dando como resultado una imagen B.

Fuente: www.cis.rit.edu

Con el fin de sintetizar la información proporcionada en este tema, en el siguiente esquema se reflejan los principales conceptos.

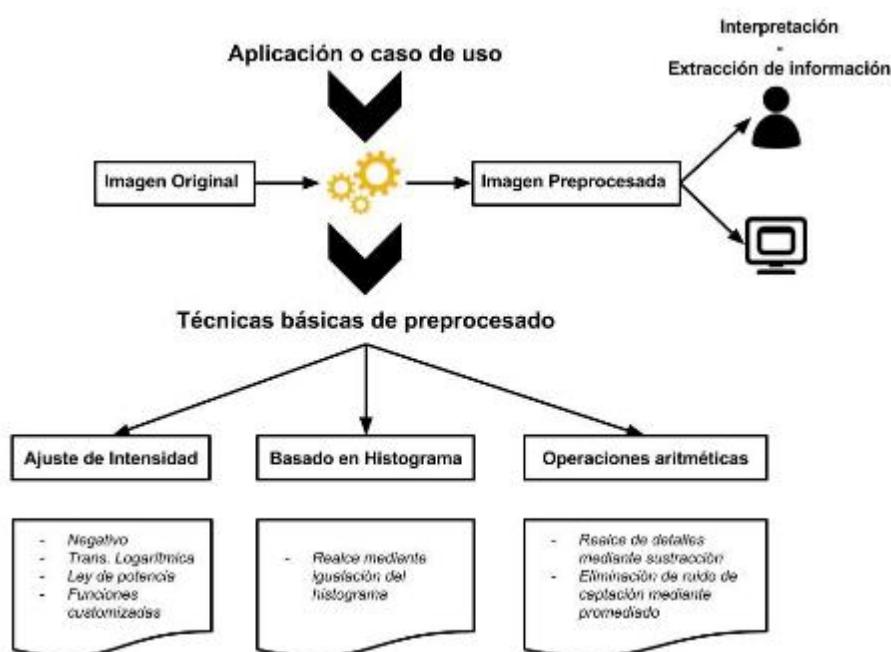


Figura 49. Síntesis de los conceptos elementales descritos en el tema: preprocesado de imagen mediante operaciones elementales.

6.2. Ajuste de intensidad



Accede al vídeo «Ajuste de intensidad» a través del aula virtual

Las operaciones de ajuste de intensidad consisten en diferentes expresiones del operador T de la ecuación anterior. La figura 3 recoge algunas de las operaciones de transformación más comunes para una imagen con L niveles de intensidad diferentes.

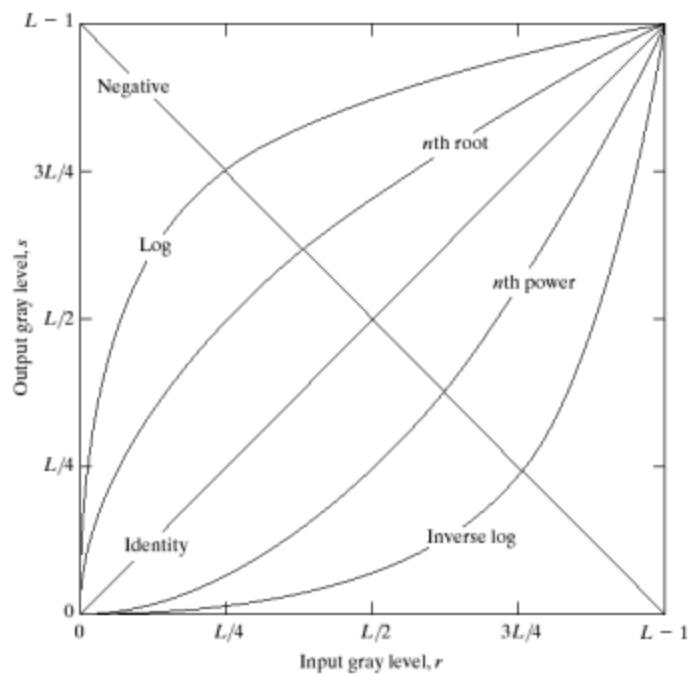


Figura 50. Perfil de las funciones de transformación punto a punto más comúnmente empleadas.

Fuente: González y Woods, 2008.

Negativo de una imagen

El negativo de una imagen representa la imagen invertida respecto a la original. La expresión matemática de la transformación viene dada por la siguiente ecuación:

$$T(u) = L - u$$

Donde L es el mayor nivel de intensidad que puede tomar un píxel.

Como puede apreciarse, se trata de una **transformación lineal**. La figura 4 muestra un ejemplo de la aplicación de esta transformación a una imagen médica con el objetivo de resaltar un detalle de interés que, en este caso, viene dado por una lesión. Tal y como se aprecia, la lesión destaca con mayor nitidez en la imagen transformada, a pesar de que la transformación no varía el contenido visual de la señal al ser de naturaleza lineal.

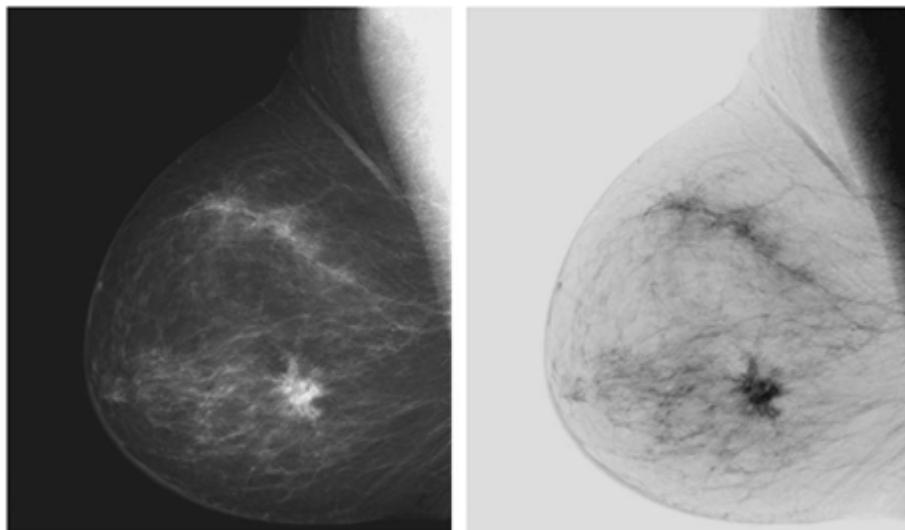


Figura 51. Ejemplo de aplicación del operador de transformación para la obtención del negativo en una imagen médica.

Fuente: González y Woods, 2008.

Esta transformación tiene especial utilidad para realzar estructuras de mayor intensidad, es decir, de tonos más próximos al color blanco (que representa el valor máximo de intensidad) encerradas en áreas oscuras de la imagen.

Transformaciones logarítmicas

La expresión de los operadores logarítmicos es la siguiente:

$$T(u) = C \log(1 + u)$$

Donde C es una constante.

El **perfil del operador** puede apreciarse en la figura 3. Como puede observarse, esta transformación mapea un rango pequeño de valores menores de intensidad en la imagen de entrada en un rango amplio de salida. Sin embargo, sucede lo contrario para los valores de intensidad mayores, que tienden a ser concentrados en un rango estrecho de valores de salida.

Por tanto, esta transformación se emplea cuando se pretende expandir el rango de intensidad de los píxeles oscuros al mismo tiempo que se equiparan entre sí los píxeles más claros.

La utilidad práctica de este operador puede apreciarse en aquellas situaciones en las que la imagen de entrada posee un rango dinámico muy amplio y su representación, por tanto, nos llevará a distinguir únicamente píxeles negros y blancos.

Esta situación suele observarse con la representación de imágenes en el dominio de la frecuencia. El espectro de la imagen típicamente tiene valores significativamente mayores que el resto en el entorno de la frecuencia origen: gran parte de la potencia de la señal se concentra en las frecuencias más bajas. Por tanto, a la hora de representarlo gráficamente, no es posible apreciar la evolución de la señal hacia frecuencias mayores.

La figura 5 muestra un ejemplo de **cómo emplear la transformación logarítmica** para obtener una representación más informativa del espectro de una imagen. En este caso, se han realzado los píxeles en niveles menores de intensidad, manteniéndose prácticamente intactos aquellos píxeles cercanos al máximo valor de intensidad.

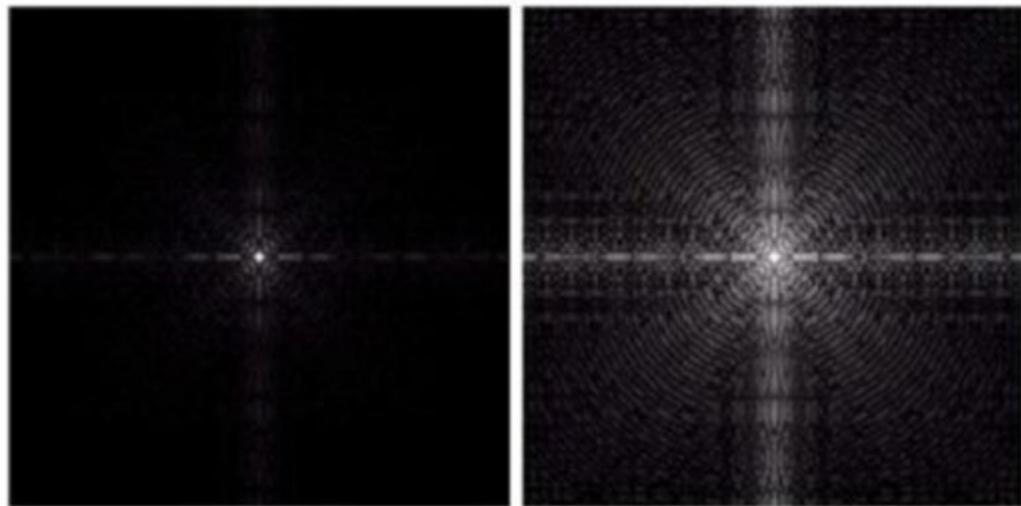


Figura 52. Ejemplo de utilización de una transformación logarítmica para el realce de las componentes frecuenciales más altas en el espectro de una imagen.

Fuente: González y Woods, 2008.

Ley de potencia

La función de transformación de acuerdo a la ley de potencia viene dada por la siguiente expresión:

$$T(u) = Cu^\gamma$$

Donde C y γ son constantes positivas. El perfil de las funciones obtenidas de acuerdo a esta expresión se muestra en la siguiente imagen.

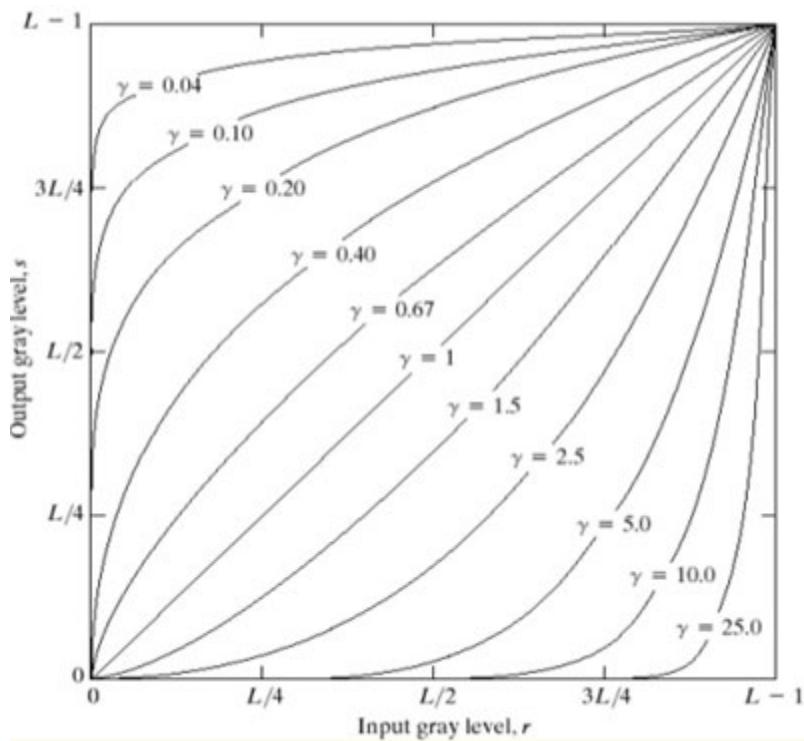


Figura 53. Funciones de transformación punto a punto obtenidas de acuerdo a la ley de potencias.

Fuente: González y Woods, 2008.

Como en el caso de las transformaciones logarítmicas, valores de γ menores que la unidad tienden a expandir el rango de intensidad de los píxeles más oscuros, mientras que este se comprime para los píxeles más claros.

La ventaja de estas funciones respecto a la transformación logarítmica es la posibilidad de generar una amplia familia de transformaciones simplemente mediante la variación del parámetro γ .

Así, las curvas generadas con $\gamma > 1$ tienen exactamente un efecto opuesto a las correspondientes a $\gamma < 1$.

El origen de este tipo de transformación se encuentra en la respuesta observada en algunos dispositivos antiguos destinados a la captura de imagen o la impresión. Esta respuesta obedece a una función de este tipo, por lo que es necesario aplicar una transformación opuesta para restaurar la imagen. Este procedimiento se conoce como **corrección gamma**, haciendo referencia al exponente de la función de transformación.

Funciones definidas a trozos

Las funciones anteriores tienen una expresión matemática para todo el dominio de aplicación, es decir, para cualquier valor de intensidad de entrada. Sin embargo, puede ser interesante aplicar diferentes transformaciones dependiendo del rango de intensidad de los píxeles sobre los que se opera. En esta situación, se emplean operadores definidos a trozos o tramos.

La **principal ventaja** de estas funciones es que su forma o expresión puede ser arbitrariamente compleja. Por ejemplo, puede dividirse el rango de intensidades de entrada en tantos tramos como se desee, definiéndose una transformación específica para cada uno de ellos.

Sin embargo, el **inconveniente más relevante** es que requieren un alto grado de participación del usuario para su definición. Generalmente, un humano define la función mediante la inspección visual del efecto producido por cada una de las transformaciones que se aplican sobre los diferentes tramos.

Una de las aplicaciones más comunes de este tipo de funciones es el **realce del contraste en una imagen**. En algunas situaciones, la imagen puede tener un rango dinámico pequeño debido a una pobre iluminación de la escena, el sensor empleado o la configuración de la lente empleada en la captura. Por tanto, la idea del realce del contraste es aumentar el rango dinámico de la imagen de entrada. Para ello, se trata de remarcar las diferencias entre píxeles cuyos valores de intensidad están concentrados en un rango intermedio. El perfil de la transformación será, en consecuencia, similar al mostrado en la figura 7.

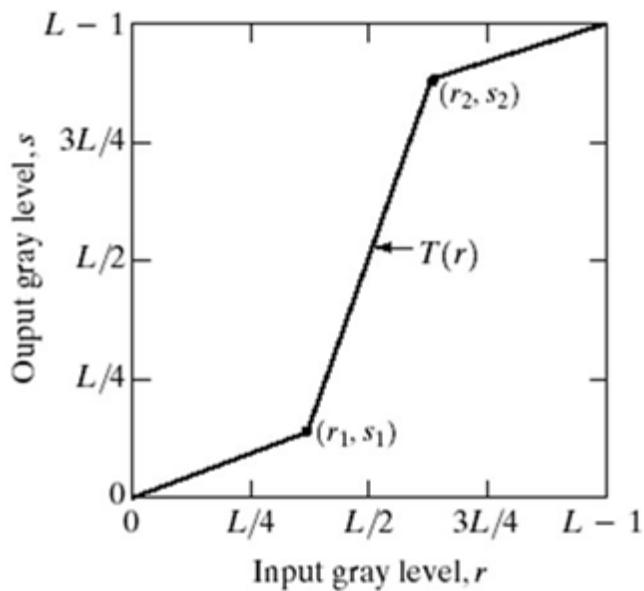


Figura 54. Perfil de la función de transformación definida a tramos para conseguir el realce de contraste.

Fuente: González y Woods, 2008.

Por otro lado, la figura 8 muestra el efecto de la aplicación de una transformación de este tipo sobre una imagen con un rango dinámico reducido. Como puede observarse, el resultado obtenido es un realce significativo respecto al contraste inicial entre los objetos y el fondo de la imagen.

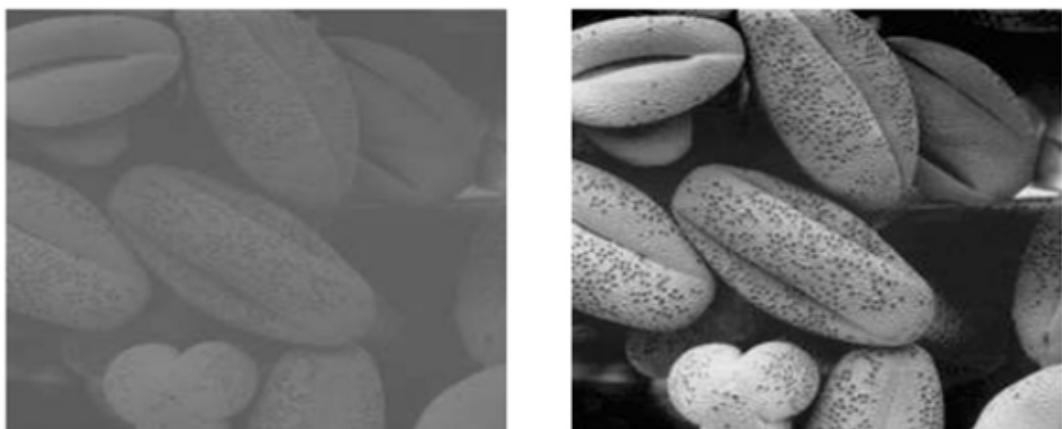


Figura 55. Ejemplo de realce de una imagen mediante la utilización de una función de transformación definida a tramos.

Fuente: <https://www.mathworks.com/>.

6.3. Procesado sistemático del histograma



Accede al vídeo «Procesado del histograma» a través del aula virtual

Como se indicaba previamente, la expresión de la función de transformación definida a trozos para el realce de contraste se obtiene de forma manual, tras diferentes ciclos de prueba. Este modo de proceder conlleva demasiado y puede llegar a convertirse en una tarea tediosa. Una opción alternativa a este método para el realce de contraste es el procesado de la imagen basado en las propiedades de su histograma.

El histograma de una imagen proporciona una estimación de la función de densidad de probabilidad de la variable, la cual representa el valor de intensidad de los píxeles.

Para una imagen cuyos píxeles toman un nivel de intensidad de entre L niveles diferentes, el histograma se obtiene mediante el conteo del número de píxeles en cada nivel. Para obtener una versión normalizada en la que los conteos puedan interpretarse como probabilidades, el número de píxeles asociados a cada nivel se divide entre el número total de puntos en la imagen.

La **operación de igualación del histograma de una imagen** permite realizar el contraste de esta, ensalzando las estructuras presentes en la imagen de forma automática sin intervención externa del usuario.

Esta operación parte de la modelización matemática de la imagen de tamaño $N = n_r \times n_c$ como la realización de N variables aleatorias, una por cada píxel Y_i , que toma valores en el rango $L' - 1$. Se denota por f_{Y_i} a la función de densidad de probabilidad de cualquiera de estas variables aleatorias.

El objetivo de la transformación es convertir cada una de estas variables aleatorias en variables uniformes X_i en el rango $[0, \dots, L - 1]$.

Para ello, como es sabido, una variable aleatoria puede convertirse en uniforme transformándola a través de su **función de distribución**. Por tanto, basta con obtener la variable X_i como la transformación de Y_i a partir de su función de distribución:

$$X_i = F_Y(Y_i) = \int_{-\infty}^{Y_i} f_Y(\tau) d\tau$$

Sin embargo, la función de distribución de Y no es conocida, por lo que debe emplearse una **estimación** de la misma. Dicha estimación se obtiene a partir del histograma.

El valor de este para el nivel de intensidad l puede interpretarse como la probabilidad de que la variable Y tome dicho valor. Es decir:

$$P(Y = l) = p_l \cong \frac{n_l}{N}$$

Donde n_l es el número de píxeles con un nivel de intensidad l en la imagen. Por tanto, la función de distribución puede aproximarse mediante el histograma acumulado:

$$F_Y(l) = \sum_{k=0}^l p_k, \quad l = 0, \dots, L - 1$$

La figura 9 muestra un ejemplo en el que se emplea la igualación de histograma para realzar las estructuras de una imagen. Además del efecto logrado en la propia imagen, puede apreciarse el impacto de la transformación en el histograma. Este se caracteriza por un perfil más uniforme tras la transformación. En otras palabras, hay un mayor grado de dispersión en los valores de intensidad que toman los píxeles de la imagen transformada. En el caso de la original, estos tienden a estar concentrados en un rango menor de valores de intensidad.

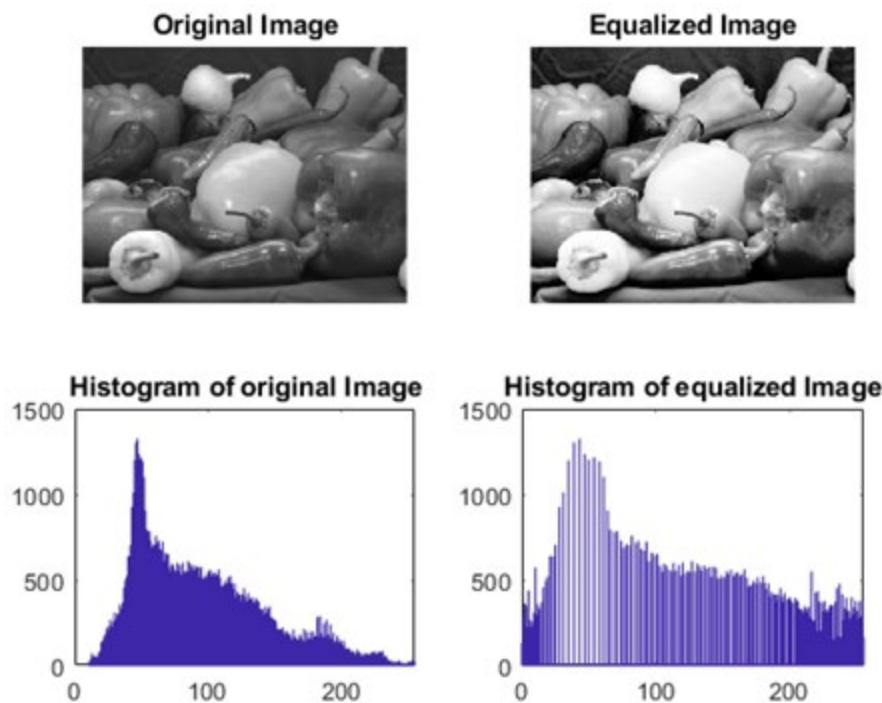


Figura 56. Ejemplo del realce de contraste mediante igualación del histograma.

Fuente: <https://www.mathworks.com/>.

6.4. Suavizado y realce mediante operadores aritméticos



Accede al vídeo «Suavizado y realizdo mediante operadores aritméticos» a través del aula virtual

De las cuatro operaciones aritméticas posibles (suma, resta, multiplicación y división), la **resta** y la **suma** son los dos operadores más útiles en las etapas de preprocesado para conseguir el realce de una imagen.

Realce de imagen mediante el operador resta

La diferencia $C(x, y)$ entre dos imágenes $A(x, y)$ y $B(x, y)$ se obtiene mediante la resta entre cada par de píxeles en ambas:

$$C(x, y) = B(x, y) - A(x, y)$$

Esta operación permite **realizar las diferencias entre dos imágenes**. Sirva como ejemplo la figura 10.

- ▶ En la esquina superior izquierda se tiene la imagen original en la que cada píxel se codifica mediante 8 bits: hay 256 niveles diferentes de intensidad.
- ▶ La imagen de la esquina superior derecha se obtiene asignando un valor 0 al cuarto bit de cada píxel. Aparentemente, no se aprecia ninguna diferencia entre ambas imágenes.
- ▶ Sin embargo, esta hipótesis puede ser contrastada mediante la resta de las dos señales píxel a píxel. Este resultado se muestra en la imagen de la esquina inferior izquierda, que refleja cierta diferencia entre ambas imágenes.
- ▶ Para apreciarse mejor, en la esquina inferior derecha se ensalzan estas diferencias mediante una operación de igualación de histograma sobre la imagen anterior.

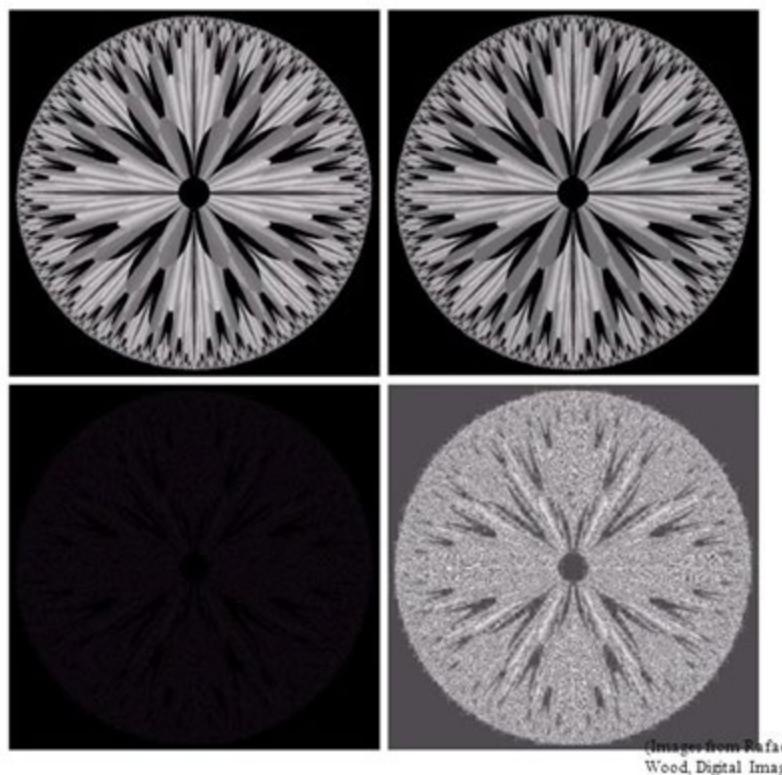


Figura 57. Utilización del operador resta entre dos imágenes para el realce de diferencias entre ambas.
Fuente: González y Woods, 2008.

Una de las aplicaciones prácticas más comunes de la sustracción de imágenes puede encontrarse en el **ámbito médico**. Partiendo de una imagen de referencia de un tejido vascular, en el que puede apreciarse una determinada sustancia, se toma a continuación una secuencia de imágenes de este tejido. Calculando la diferencia entre cada una de estas imágenes y la imagen de referencia, puede apreciarse el movimiento de la sustancia de interés, obteniéndose una especie de película a partir de las sucesivas diferencias.

Suavizado de imagen mediante el operador suma

De forma similar a la resta, el operador suma aplicado a dos o más imágenes consiste en la adición de los valores de intensidad de los píxeles correspondientes en cada una de ellas. El interés de este operador radica en su utilización para la **obtención de la imagen promedio** a partir de un conjunto, lo que permite reducir el ruido de captación.

Así, para un conjunto de M imágenes $A_i(x, y)$ ($i = 1, \dots, M$), el promedio de estas se obtiene de la siguiente forma:

$$C(x, y) = 1/M \sum_{i=1}^M A_i(x, y)$$

Cada una de las instantáneas $A_i(x, y)$ puede verse como la realización de un **proceso estocástico espacial**, una variable aleatoria cuyas observaciones suponen imágenes. Por tanto, estas instantáneas serían expresadas matemáticamente de la siguiente forma:

$$A_i(x, y) = F(x, y) + \eta(x, y)$$

Donde:

- $F(x, y)$ representa la imagen ideal libre de ruido.

- ▶ El proceso estocástico $\eta(x, y)$ se asume estacionario y gaussiano, con media nula y varianza σ^2 .

Si el ruido aditivo $\eta(x, y)$ se asume incorrelado entre dos instantáneas cualesquiera de nuestro conjunto, puede demostrarse que el promedio de todas ellas tiende a la imagen $F(x, y)$:

$$\lim_{M \rightarrow \infty} C(x, y) = F(x, y)$$

La potencia del ruido de captación se atenúa de acuerdo a un factor M en la imagen promedio resultante:

$$\sigma_C^2 = \sigma^2/M$$

Por último, reseñar que esta estrategia de realce basada en la atenuación del ruido de captación asume que las instantáneas de partida se encuentran perfectamente alineadas. De no ser así, el resultado final mostraría una clara difuminación de los bordes de las estructuras en la imagen.

En el siguiente ejemplo (figura 11) se muestra en la parte izquierda la imagen libre de ruido, en el centro una de las instantáneas capturadas afectada por ruido aditivo gaussiano y, en la derecha, el resultado de aplicar el promediado sobre un conjunto de instantáneas ruidosas.

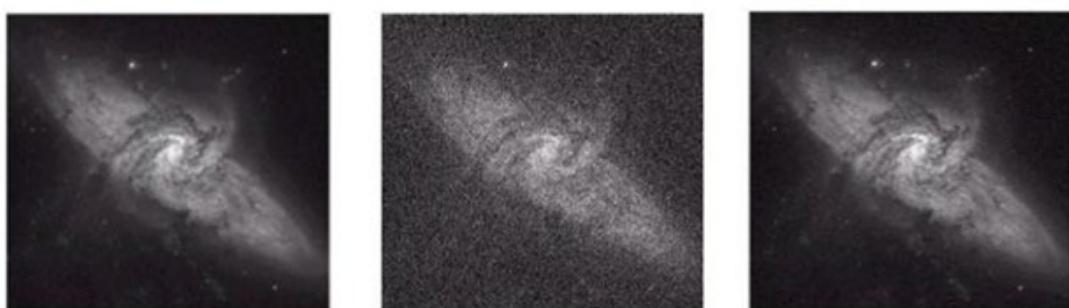


Figura 58. Ejemplo de utilización del promediado de imágenes para la atenuación del ruido de captación.
Fuente: González y Woods, 2008.

6.5. Referencias bibliográficas

González, R. C. y Woods, R. E. (2008). *Digital image processing*. New Jersey: Pearson Education.



Accede a los ejercicios de autoevaluación a través del aula virtual

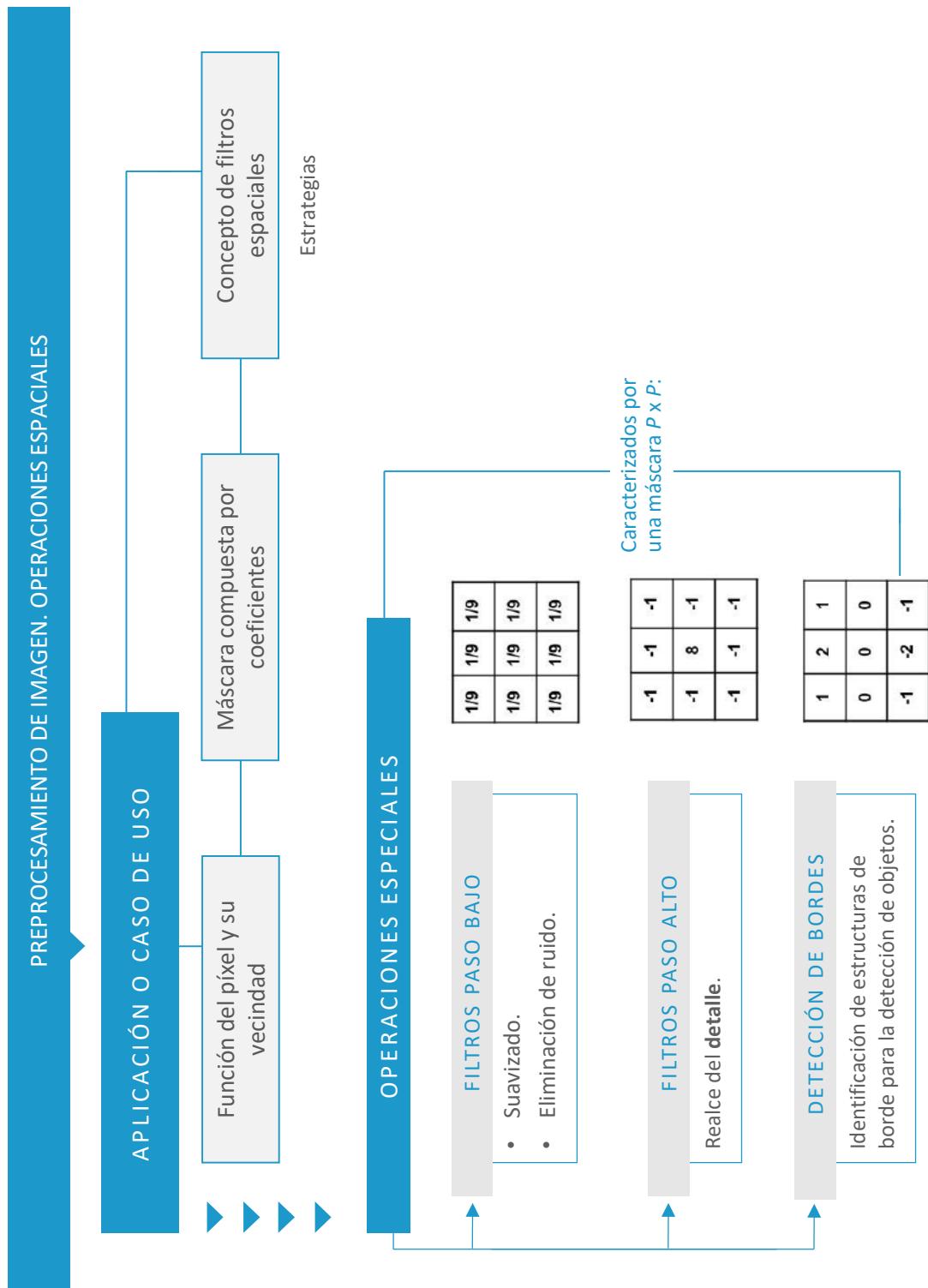


Accede al vídeo «Resumen del tema» a través del aula virtual

Percepción Computacional

Procesamiento de imagen. Operaciones espaciales

Esquema. Tema 7



Ideas clave. Tema 7

7.1. ¿Cómo estudiar este tema?



Accede al vídeo «Introducción. Operaciones espaciales» a través del aula virtual

Para estudiar este tema deberás leer con atención las ideas clave que se desarrollan a continuación.

En el tema anterior se describieron un conjunto de técnicas de realce de imagen. Estas persiguen ensalzar las estructuras presentes en la imagen, de forma que, en etapas posteriores, pueda obtenerse información de interés. Los operadores analizados fueron calificados como elementales, dada la naturaleza de las operaciones que, en ese caso, correspondía a funciones de transformación punto a punto.

Operadores espaciales

En el presente tema se aborda la utilización de operadores más complejos para el realce de imagen. Concretamente, se estudiarán diferentes operaciones espaciales. Para este tipo de operadores, el valor del píxel resultante es **función del píxel** correspondiente en la imagen original **y de la vecindad de este**. Además, se emplea una subimagen de tamaño similar a esta vecindad y compuesta por elementos denominados coeficientes que se conoce como **máscara**. Así, el operador puede expresarse de la siguiente forma:

$$v = T(u, u_1, u_2, \dots, u_{P^2-1}; \lambda, \lambda_1, \lambda_2, \dots, \lambda_{P^2-1})$$

Donde:

- ▶ v denota al valor de intensidad del píxel resultante.
- ▶ u es el píxel correspondiente en la imagen original.
- ▶ $u_i(i = 1, \dots, P^2 - 1)$ son los píxeles vecinos ($P^2 - 1$) para una vecindad de tamaño $P \times P$ centrada en u .
- ▶ $\lambda, \lambda_i(i = 1, \dots, P^2 - 1)$ son los coeficientes de la máscara que define el operador espacial.

Si el operador es de **tipo convolutivo**, el valor del píxel resultante será una función lineal de los valores de intensidad de los píxeles de entrada, expresándose el operador T de la siguiente forma:

$$v = \lambda u + \sum_{i=1}^{P^2-1} \lambda_i u_i$$

Como se ha indicado, el operador espacial queda definido por los coeficientes λ_i de la máscara. Por tanto, es común proporcionar una descripción del operador mediante la representación de esta máscara.

Por ejemplo, para un operador espacial 3×3 , la máscara tendría el aspecto mostrado en la siguiente figura:

λ_1	λ_2	λ_3
λ_4	λ	λ_5
λ_6	λ_7	λ_8

Figura 59. Máscara de un operador espacial de dimensión 3×3 .

Típicamente, se toma P impar, de forma que la máscara se centra en el píxel bajo estudio. Mediante esta representación, puede obtenerse una interpretación del operador a través de una inspección visual de su máscara.

En los siguientes epígrafes de este tema se presentan algunas implementaciones de filtros espaciales comúnmente empleados en las etapas de preprocesado de una imagen. La figura 2 recoge los principales conceptos expuestos en el tema:

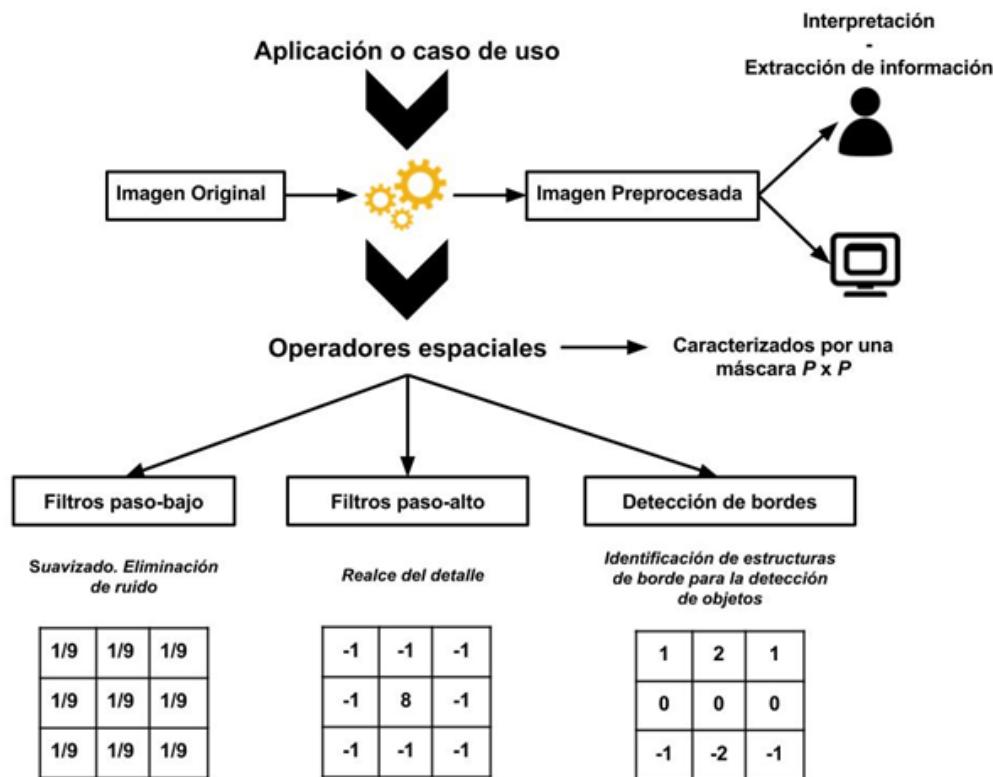


Figura 60. Esquema de los conceptos y descripción de los principales operadores espaciales sobre imágenes.

El **concepto de filtro** tiene su origen en las funciones de transferencia de los sistemas diseñados íntegramente en el dominio de la frecuencia. Sin embargo, en este tema, se emplea el término filtro para referirnos a las operaciones llevadas a cabo directamente sobre los píxeles de la imagen. Así, el término filtrado espacial se emplea para denominar a este tipo de operaciones frente a los filtros diseñados y aplicados en el dominio frecuencial.

En este punto, cabe recordar el filtro de mediana descrito en anteriormente. Se trata, al igual que los operadores descritos en este tema, de un filtro espacial, pues su respuesta se calcula directamente sobre el píxel de interés y su vecindad. Sin embargo, el filtro de mediana supone una dependencia no lineal de la respuesta y el conjunto de píxeles de entrada.

Una cuestión a considerar cuando se implementan filtros espaciales es la estrategia a seguir para aquellos píxeles cercanos al borde de la imagen. Cuando la máscara se centra en uno de estos píxeles, parte de ella queda fuera de la imagen. Concretamente, para aquellos **píxeles alejados del borde** una distancia inferior a $(P - 1)/2$ píxeles, parte de la máscara quedará fuera de la imagen. En este caso, existen diferentes estrategias para manejar esta situación:

- ▶ La más simple de todas ellas es aplicar la máscara solo a aquellos píxeles que estén situados a una **distancia mayor o igual a $(P - 1)/2$ píxeles** del borde.
 - La principal limitación de esta estrategia radica en que la imagen de salida será de tamaño menor que la imagen original.
- ▶ Otra opción es **calcular la respuesta del filtro** para aquellos puntos cercanos al borde, teniendo en cuenta únicamente los píxeles de la imagen que caen dentro de la máscara. Esta opción es tomada comúnmente cuando se requiere que la imagen de salida conserve el tamaño de la imagen original.
- ▶ Por último, cabe la posibilidad de **añadir filas/columnas a la imagen** con un valor determinado. Habitualmente, estas filas/columnas añadidas contienen píxeles con valor 0, 1 o el promedio de la intensidad de la imagen. También es posible replicar la última fila/columna de la imagen a fin de añadir estos nuevos píxeles.
 - La limitación fundamental de esta estrategia es que la respuesta del filtro en los píxeles cercanos al borde está influenciada por el valor de intensidad de los píxeles añadidos de forma artificial.

7.2. Filtros paso bajo y paso alto



Accede al vídeo «Técnicas de filtrado» a través del aula virtual

A continuación se exponen las características de los filtros espaciales empleados habitualmente para la selección de frecuencias bajas y altas.

Tal y como se ha mencionado, estos filtros se definen y aplican sobre los píxeles originales de la imagen y, a pesar de su nombre, no hacen referencia a filtros diseñados y aplicados en el dominio de la frecuencia.

Filtros espaciales paso bajo (suavizado)

El efecto del filtro paso bajo se obtiene mediante el **cálculo del promedio** a partir del conjunto de píxeles definidos por la máscara, por tanto, los coeficientes de esta serán mayores que cero. Además, a fin de preservar la componente continua de la señal original, estos coeficientes son normalizados de forma que la suma de todos ellos sea la unidad.

La idea detrás de estos filtros de promediado es simple. Si se reemplaza el valor de intensidad de un píxel por el promedio de los píxeles vecinos, **se atenúan las variaciones bruscas de intensidad**. Por definición, un cambio significativo de intensidad en píxeles cercanos está asociado a una componente frecuencial alta que, de esta manera, sería eliminada por el filtro. Este suele ser el caso del ruido de captación, que representa una componente aleatoria de la señal de entrada caracterizada por transiciones bruscas entre píxeles contiguos.

En el caso de no contar con un conjunto de instantáneas de la misma escena para, tal y como se describió en el tema anterior, obtener el promedio de todas ellas y reducir el ruido, la aplicación de un **filtro espacial de promediado** es una solución plausible a fin de mejorar la calidad de la imagen y ensalzar sus estructuras. La principal desventaja de este filtro es el efecto de suavizado o desenfoque sobre algunas de las estructuras caracterizadas por componentes frecuenciales elevadas como, por ejemplo, los bordes entre objetos. La figura 3 muestra las máscaras correspondientes a sendos filtros espaciales de promediado (paso bajo).

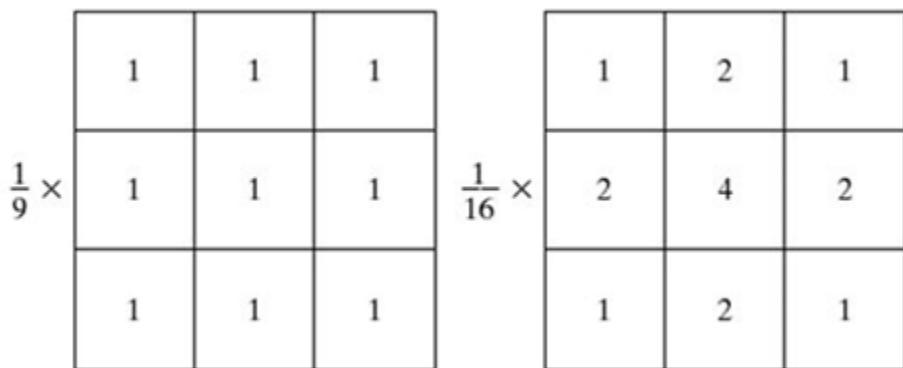


Figura 61. Máscaras correspondientes a dos filtros espaciales de promediado.

La máscara de la izquierda posee el mismo valor para todos sus coeficientes, mientras que la máscara de la derecha, al tener coeficientes de diferente valor, lleva a cabo el **cálculo del promedio ponderado**.

La máscara de la izquierda refleja un filtro en el que se lleva a cabo el promediado simple de los píxeles de la vecindad, pues todos los coeficientes tienen el mismo valor.

Un filtro de promediado en el que todos sus coeficientes son iguales se denomina *box filter*.

En el caso de la máscara de la derecha, como puede observarse, hay coeficientes que tienen más peso que otros. En este caso, el promediado está ponderado, dando mayor relevancia a unos píxeles frente a otros. Concretamente, en el ejemplo de la figura, el píxel central es el de mayor peso, seguido de los píxeles contiguos.

Los filtros mostrados en el ejemplo tienen un tamaño 3×3 . Sin embargo, puede considerarse cualquier **tamaño del filtro**. Este factor influye, a su vez, en el resultado del filtrado y debe escogerse de acuerdo al nivel de ruido presente en la imagen original y al nivel de calidad exigido por el escenario práctico en el que nos encontramos. Por lo general, un filtro de mayor dimensión tenderá a eliminar el ruido de forma más eficiente, pero a costa de producir una imagen más desenfocada, ya que el promediado se computa a partir de un mayor número de píxeles.

Filtros espaciales paso-alto (afilado)

El principal objetivo de los filtros espaciales para el afilado de la imagen es realzar los detalles más finos de la imagen o, también, corregir el desenfoque producido como consecuencia del proceso de adquisición.

En el epígrafe anterior, se describían los filtros espaciales enfocados en el suavizado de la imagen para la supresión de ruido. Las máscaras que caracterizan estos filtros implementan una operación de promediado que, matemáticamente, se corresponde con la operación de integración o suma de elementos. Los filtros de afilado o ensalzado del detalle persiguen un objetivo opuesto. Es decir, tratan de **magnificar las variaciones o contrastes** entre estructuras de la imagen. Por tanto, las máscaras correspondientes a estos filtros incluirán coeficientes tales que lleven a cabo cálculos de diferenciación. El **cálculo de diferencias** es necesario para la obtención de la derivada de una función. A su vez, la derivada representa la operación inversa a la integración desde un punto de vista matemático.

La derivada de una función se hace más pronunciada en aquellos puntos en los que hay cambios más significativos.

Así, los operadores espaciales paso alto que aproximan la derivada de la función imagen en las coordenadas (x, y) ensalzan los bordes y otras discontinuidades (por ejemplo, el ruido), atenuando aquellas áreas de la imagen en las que la variación es poco relevante o nula. En definitiva, actúan como un filtro de característica paso alto que retiene las componentes frecuenciales más elevadas de la imagen.

Los filtros espaciales paso alto se basan en la aproximación de las derivadas de primer y segundo orden. Nótese que estas operaciones matemáticas se llevan a cabo sobre funciones discretas, como son las imágenes. Los valores de intensidad que pueden observarse en estas son finitos y, por tanto, las variaciones entre píxeles así lo serán. Además, la menor distancia entre dos puntos de la función entre los que cuantificar una variación vendrá dada por dos píxeles adyacentes.

Para la definición de la aproximación de las derivadas de primer y segundo orden en un contexto digital como el descrito, emplearemos por simplicidad **señales unidimensionales**; el procedimiento es extensible a señales bidimensionales sin pérdida alguna de generalidad.

La **derivada de primer orden** de una señal discreta $f[n]$ puede estimarse como:

$$\frac{\partial f}{\partial n} = f[n+1] - f[n]$$

De forma similar, la **derivada de segundo orden** puede computarse de la siguiente forma:

$$\frac{\partial^2 f}{\partial^2 n} = f[n+1] + f[n-1] - 2f[n]$$

Para definir las propiedades de ambas operaciones, podemos observar la siguiente imagen donde se muestra el cálculo de ambas derivadas sobre una secuencia numérica que representa diferentes tipos de discontinuidades.

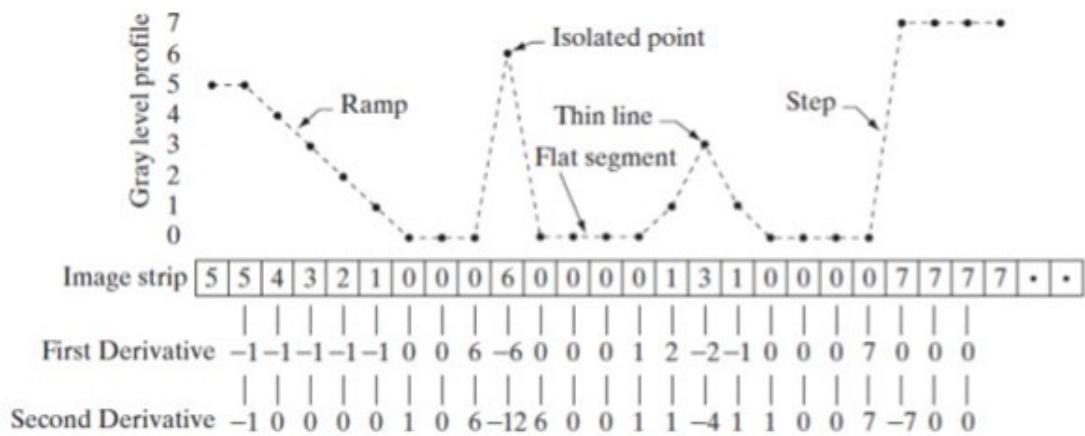


Figura 62. Estimación de derivadas de primer y segundo orden sobre una secuencia numérica. Análisis de la capacidad de estos operadores para la identificación de diferentes tipos de discontinuidades.

Fuente: González y Woods, 2008.

En primer lugar, se aprecia como la primera derivada es distinta de cero durante toda la rampa inicial de la secuencia, mientras que la segunda derivada solo toma valores distintos de cero durante el comienzo y final de la rampa. Dado que los **bordes** en una imagen vienen dados por este tipo de perfil en forma de rampa, se puede extraer que la primera derivada produce bordes gruesos, mientras que los identificados por la segunda derivada son mucho más finos.

Posteriormente, se aprecia un **punto aislado en la secuencia**. Para este, la respuesta es mucho más significativa en el caso de la segunda derivada. Esto hecho no es sorprendente, ya que la derivada de segundo orden magnifica los cambios de forma más pronunciada que la derivada de primer orden. Por tanto, este será el resultado con este tipo de puntos aislados y, por consiguiente, con elementos espurios como el ruido.

La línea delgada que encontramos a continuación en nuestra secuencia produce un resultado similar. La respuesta de la segunda derivada sería aún más pronunciada si esta línea fina alcanzara el nivel de intensidad del punto aislado que se ha visto previamente. Finalmente, la secuencia muestra una discontinuidad brusca que es capturada de forma muy parecida por ambos operadores de derivada.

A continuación, exponemos un resumen de las **conclusiones** extraídas de la comparación del resultado producido por las derivadas de primer y segundo orden.

COMPARACIÓN ENTRE DERIVADAS DE PRIMER Y SEGUNDO ORDEN	
DERIVADA DE PRIMER ORDEN	DERIVADA DE SEGUNDO ORDEN
Produce bordes más gruesos en la imagen resultante.	Tiene una respuesta más pronunciada a detalles más finos como líneas delgadas o puntos aislados.
Presenta una respuesta más acusada a cambios con forma de escalón en el nivel de intensidad de la imagen.	Tiende a producir una doble respuesta en cambios de tipo escalón. Además, se aprecia que su respuesta es más pronunciada en puntos que en líneas delgadas, y en estas que en un escalón.

Figura 63. Conclusiones de la comparación entre derivadas de primer y segundo orden.

De acuerdo a las propiedades descritas, la siguiente máscara (figura 6) corresponde a un **filtro espacial paso alto** basado en el cálculo de la derivada de primer orden. En este caso, el filtro está diseñado para la detección de bordes en posición vertical, tal y como reflejan los coeficientes de la máscara. Este filtro es equivalente al cómputo del **operador gradiente** en la dirección del eje horizontal de la imagen.

1	0	-1
1	0	-1
1	0	-1

Figura 64. Ejemplo de un filtro espacial paso alto basado en el cómputo de la derivada de primer orden (gradiente) de la señal.

En el caso de un filtro de segundo orden, este vendría dado por una máscara similar a la mostrada en la figura 7. El operador equivalente a este filtro es el **laplaciano** de la señal. En este caso, el filtro es de carácter isotrópico, pues es capaz de identificar discontinuidades en cualquiera de las direcciones de la imagen.

-1	-1	-1
-1	8	-1
-1	-1	-1

Figura 65. Ejemplo de un filtro espacial paso alto basado en el cómputo de la derivada de segundo orden (laplaciano) de la señal.

Finalmente, la siguiente imagen (figura 8) ilustra el efecto de aplicar el operador laplaciano sobre la imagen de la izquierda. Como se puede ver, la imagen resultante (derecha) contiene los bordes de las estructuras identificadas en la imagen original.

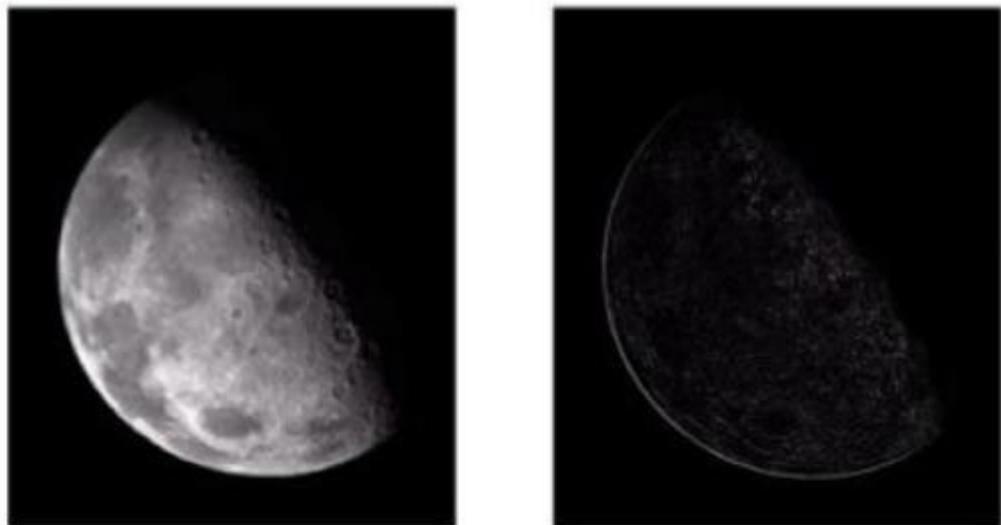


Figura 66. Aplicación del operador laplaciano para el realce de detalles en una imagen.

Fuente: González y Woods, 2008.

7.3. Detección de bordes



Accede al vídeo «Detección de bordes» a través del aula virtual

Ya se ha visto cómo pueden ser identificados los bordes de las estructuras presentes en una imagen mediante la aplicación de filtros espaciales paso alto. En este punto, se profundiza en el conocimiento de operadores destinados a la detección de bordes. Esta denominación se emplea para identificar a los operadores basados en el **cálculo del gradiente**. Como se ha visto, se corresponden con los filtros espaciales que aproximan el cómputo de la derivada de primer orden en direcciones determinadas de la imagen.

A continuación se detallan los filtros espaciales más comunes para la detección de estructuras de borde en una imagen.

Operadores de Roberts

Estos operadores se caracterizan por tener un tamaño de máscara 2×2 , que supone una novedad con las máscaras vistas hasta el momento. Los coeficientes de las máscaras se muestran en la siguiente figura, indicándose en negrita el píxel sobre el que se calcula la operación.

1	0
0	-1

0	1
-1	0

Figura 67. Máscaras que definen los operadores de Roberts para la detección de bordes.

La máscara de la izquierda aproxima el resultado del operador gradiente en la dirección diagonal dada por el ángulo 135° . En el caso de la máscara de la derecha,

como puede deducirse por sus coeficientes, lleva a cabo el cálculo del gradiente en la diagonal de 45°.

Operadores de Prewitt

Estos operadores vienen definidos por máscaras de tamaño 3 x 3 que definen el cálculo del operador gradiente tanto en la dirección vertical como en la dirección horizontal de la imagen. Estas serían las máscaras correspondientes a este operador.

1	1	1
0	0	0
-1	-1	-1

-1	0	1
-1	0	1
-1	0	1

Figura 68. Máscaras correspondientes a los operadores de Prewitt para la identificación de bordes con orientación horizontal y vertical.

Operadores de Sobel

Los operadores de Sobel son similares a los de Prewitt. La diferencia estriba en que los primeros dan más peso a los píxeles cercanos al píxel de interés (centro de la máscara) para el cálculo de diferencias.

1	2	1
0	0	0
-1	-2	-1

-1	0	1
-2	0	2
-1	0	1

Figura 69. Máscaras correspondientes a los operadores de Sobel para la identificación de bordes con orientación horizontal y vertical.

Algoritmo de Canny

Por último, se incluye en este epígrafe del tema uno de las transformaciones más ampliamente utilizadas y efectivas para la detección de bordes como es el filtro de Canny.

Este operador se basa en los siguientes **criterios**:

- ▶ El filtro debe identificar el mayor número de bordes de entre todos los que hay en la imagen. Es decir, el operador, a fin de funcionar correctamente, debe minimizar el número de falsos negativos y falsos positivos en la identificación de los bordes.
- ▶ El punto identificado como borde por parte del operador debe corresponderse con el centro de la estructura de borde identificada en la imagen.
- ▶ Un borde de la imagen debe ser identificado una única vez e, idealmente, el ruido presente en la imagen no debe dar lugar a falso bordes.

Así, los **pasos** que sigue el algoritmo de detección de bordes de Canny son los siguientes:

- ▶ Se aplica, en primer lugar, un filtro paso bajo de tipo gaussiano que suaviza la imagen y reduce el posible ruido en la misma.
- ▶ Se calcula el gradiente de la imagen en las diferentes direcciones.
- ▶ Se aplica una transformación sobre los píxeles no máximos que consiste en poner a 0 el valor de intensidad de los píxeles que no forman parte de un máximo local.
- ▶ Se aplica un umbral sobre la intensidad de los píxeles para binarizar la imagen y quedarnos con las estructuras que potencialmente pueden corresponderse con bordes.
- ▶ Se eliminan aquellas estructuras más débiles que no están conectadas con bordes mayores.

A continuación mostramos un ejemplo en el que se aplica el operador de Canny para la detección de los bordes en una imagen.

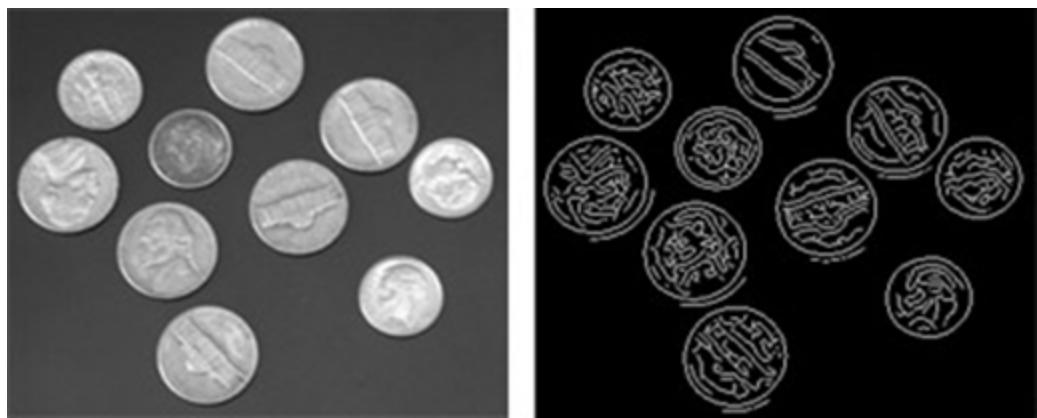


Figura 70. Resultado de la aplicación del operador de Canny sobre una imagen.

Fuente: <https://www.mathworks.com/>.

7.4. Referencias bibliográficas

González, R. C. y Woods, R. E. (2008). *Digital image processing*. New Jersey: Pearson Education.



Accede a los ejercicios de autoevaluación a través del aula virtual

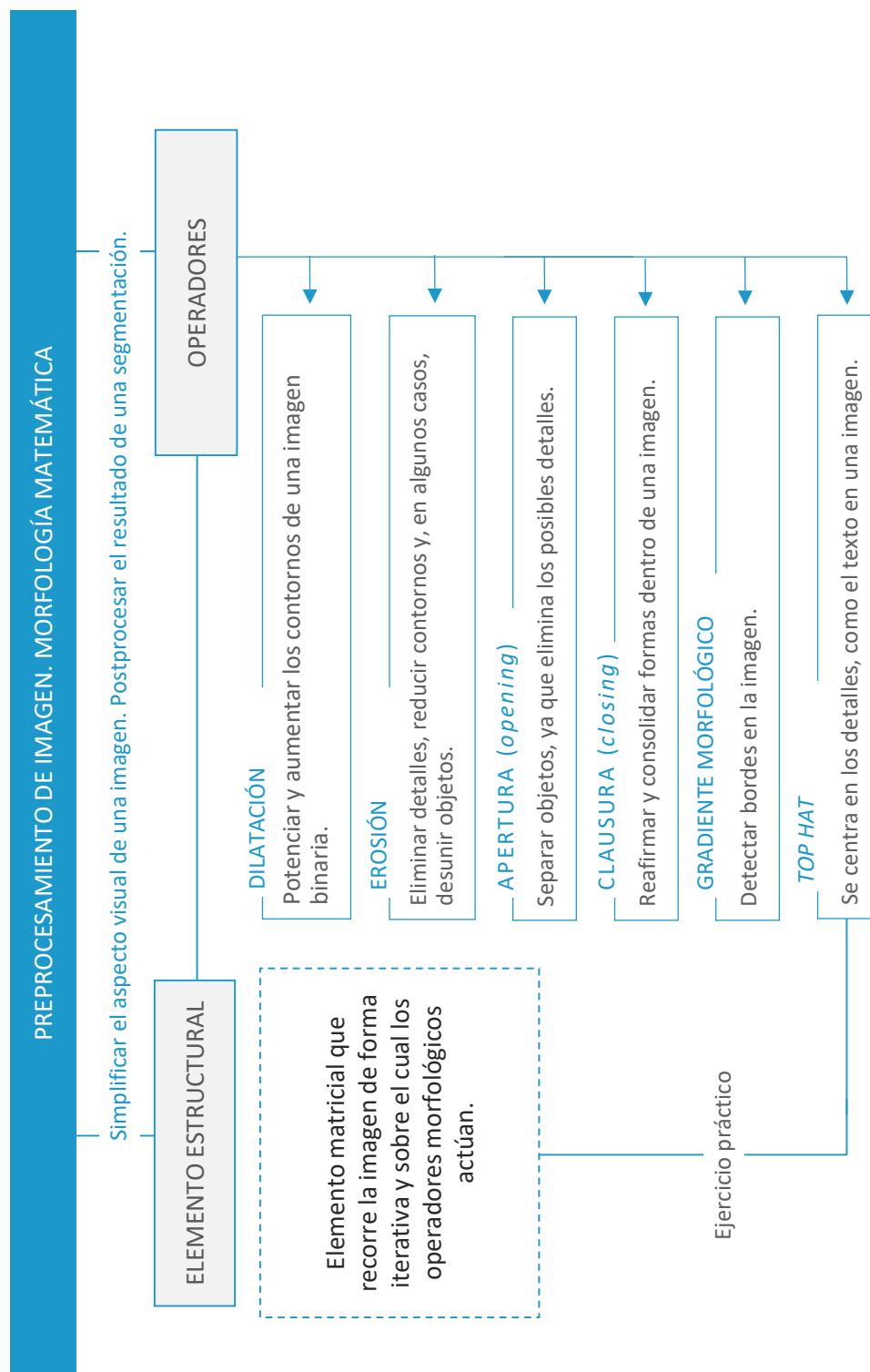


Accede al vídeo «Resumen del tema» a través del aula virtual

Percepción Computacional

Procesamiento de imagen. Morfología matemática

Esquema. Tema 8



Ideas clave. Tema 8

8.1. ¿Cómo estudiar este tema?

Para estudiar este tema deberás leer con atención las ideas clave que se desarrollan a continuación.

8.2. Introducción a la morfología matemática



Accede al vídeo «Introducción. Morfología matemática» a través del aula virtual

La morfología matemática es una de las ramas de las matemáticas que estudia el análisis y tratamiento de estructuras geométricas basándose en topología y geometría. Nació en la Escuela de Minas de París como fruto de la tesis doctoral de Jean Serra, supervisada por Georges Matheron.

La principal utilidad de la morfología matemática es la de **simplificar el aspecto visual de una imagen**, pero conservando sus características o composición principales. Es decir, una de sus utilidades es la de eliminar detalles, ruido o incluso componentes que pudieran distraer a otros algoritmos de segmentación.

A continuación presentamos un ejemplo sencillo donde se puede apreciar cómo los operadores **dilatación (Dilation)** y **cierre (Closing)** potencian más la eliminación de los detalles de la superficie quedándose con la forma de la hoja, incluso podemos ver cómo han eliminado la rama de la hoja.

Por otro lado, los operadores **erosión (Erosion)** y **apertura (Opening)**, potencian los detalles de la hoja como son las hebras que existen en ella.

Además, se observa que los operadores de apertura y cierre conservan el tamaño y las proporciones de la imagen original. No sucede así con los operadores de erosión y dilatación. No obstante, todos estos detalles los veremos en este tema.

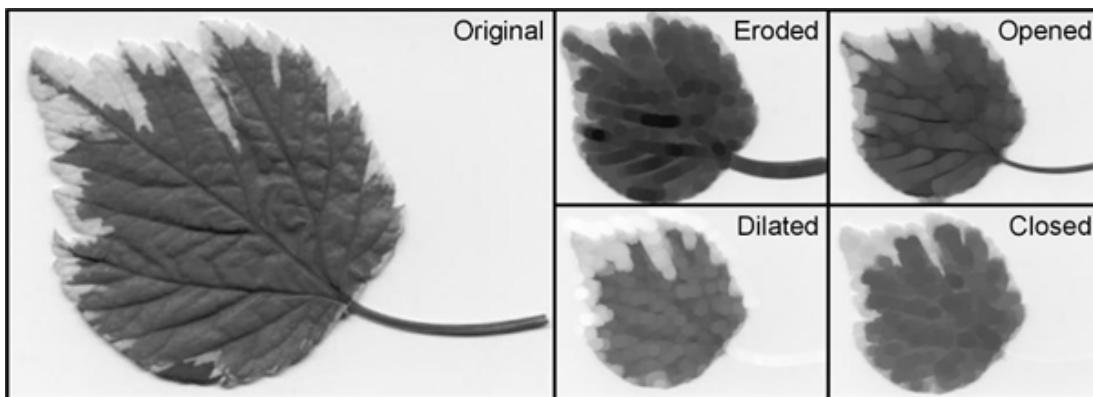


Figura 71. Resultado de aplicar los operadores más comunes de morfología matemática.

Fuente: <https://imagej.nih.gov/ij/plugins/gray-morphology.html>

Por otro lado, la morfología matemática ayuda mucho a postprocesar el resultado de una segmentación.

A continuación se presenta una ilustración con los efectos de dilatación y erosión sobre dos objetos ya segmentados: dichos objetos vienen representados por el conjunto X. Al aplicar los operadores de dilatación y erosión con el operador B, más adelante se explicará cómo funciona este operador denominado **elemento estructural**, se puede potenciar o disminuir los tamaños de los objetos en X e incluso hacer desaparecer algunos de sus elementos.

Con estos dos operadores, podríamos depurar los resultados de la segmentación eliminando aquellos residuos procedentes de una sobresegmentación o suavizar los contornos de una imagen.

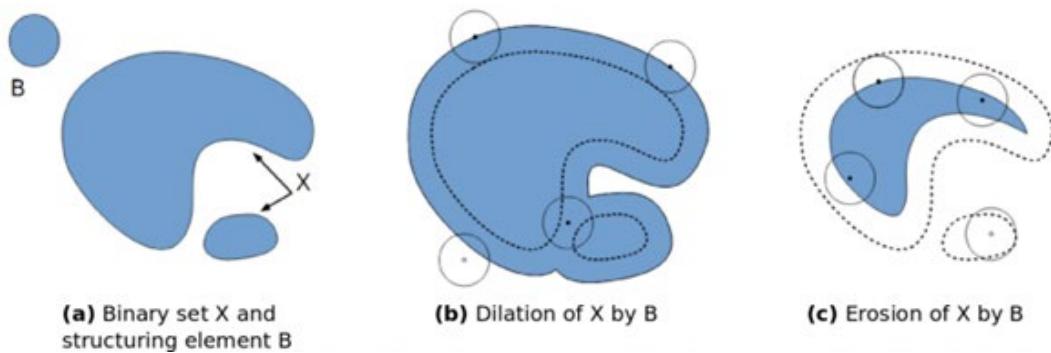


Figura 72. Efecto de dos operadores morfológicos: dilatación y erosión.

Fuente: <https://imagej.net/MorphoLib>

Así, en esta imagen vemos:

- ▶ La imagen original junto con \$B\$, el elemento estructural (a).
- ▶ Dilatación de la imagen original usando \$B\$, en este caso se aprecia cómo los dos grupos se han unido (b).
- ▶ Erosión de la imagen original usando \$B\$. El elemento más pequeño ha desaparecido y el más grande, aunque ha disminuido, ha conservado su forma (c).

Puede aplicarse también a señales unidimensionales, pero su mayor uso está concentrado en las imágenes binarias y en las imágenes de escala de grises.

Los operadores morfológicos, *per se*, no aportan información sobre la forma de los objetos ni son capaces de extraer características. Su utilidad se centra en el **limpiado de imágenes ya segmentadas** o la eliminación de detalles tanto en la imagen original como en la imagen segmentada para que los algoritmos de extracción de características puedan obtener resultados más precisos.

8.3. Definición de elemento estructural



Accede al vídeo «Definición de elemento estructural» a través del aula virtual

El elemento estructural es un ingrediente esencial de la morfología matemática. En algunos contextos se nombra también como elemento estructurante.

La función del elemento estructural es similar a la función de máscara en un filtro gaussiano de imágenes: es un **elemento matricial** (mínimo de 3x3 píxeles) que recorre la imagen de forma iterativa y sobre el cual los operadores morfológicos actúan. Es decir, un operador morfológico actúa sobre una imagen como suma de todos los efectos producidos sobre el elemento estructural cuando este recorre la imagen.

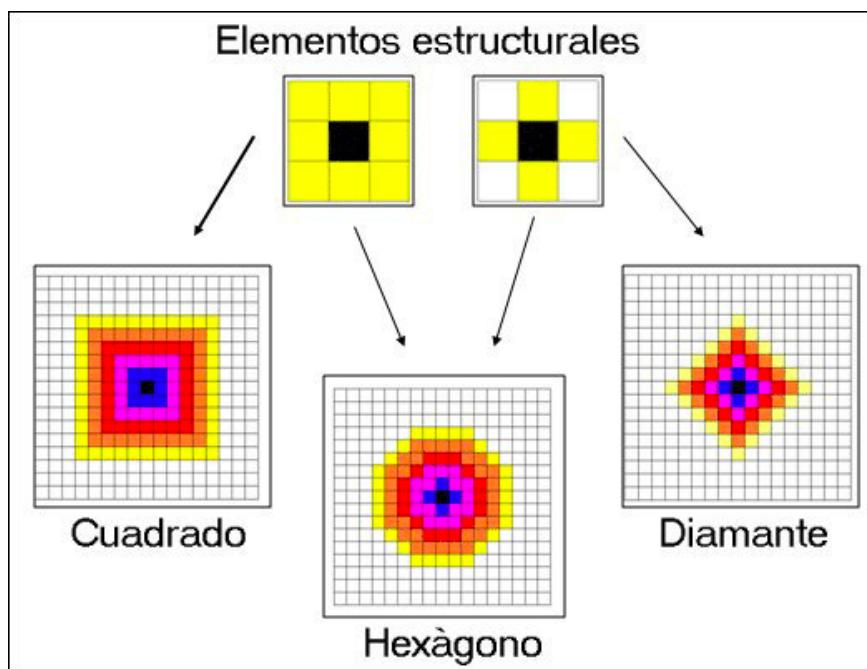


Figura 73. Ejemplos de elementos estructurales con diferentes formas.

Fuente: https://www.researchgate.net/figure/Figura-47-Elementos-estructurales-de-la-Morfologia-matematica_fig9_313905159

La definición de elemento estructural es esencial para el operador morfológico.

Supongamos que queremos contar el número de círculos existentes en la siguiente imagen:

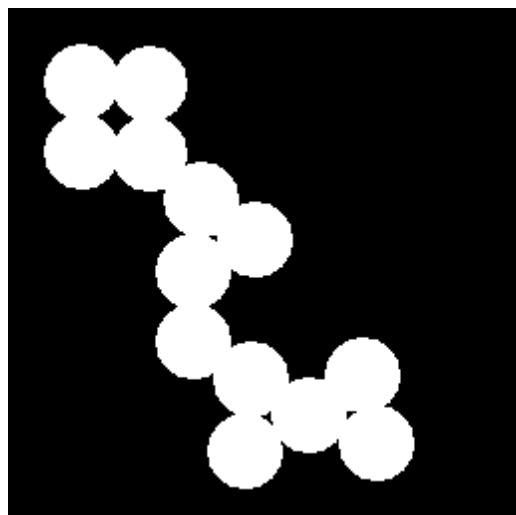


Figura 74. Ejemplo de imagen con círculos solapadas.

Fuente: <https://es.mathworks.com/help/images/ref/bwulterode.html>

La dificultad radica en contar el número de círculos. Independientemente del operador morfológico que empleemos (erosión, dilatación, etc.), el elemento estructural ha de ser lo más parecido a lo que se quiere encontrar y procesar. En este caso, el elemento estructural que mejor funciona es el circular o también llamado **de hexágono** en algunos contextos.

La razón por la cual puede llamarse hexágono viene dada por el hecho de que un círculo no puede aproximarse perfectamente en una matriz pixelada. De esta manera se puede alcanzar el resultado que buscamos.



Figura 75. Resultado de aplicar elementos estructurales circulares a la imagen.

Fuente: <http://matlab.izmiran.ru/help/toolbox/images/imerode.html>

A continuación, se presenta el resultado de haber utilizado otro elemento estructural como es el caso de una línea, de longitud 20 píxeles y orientación 90° (vertical) y 180° (horizontal) respectivamente.

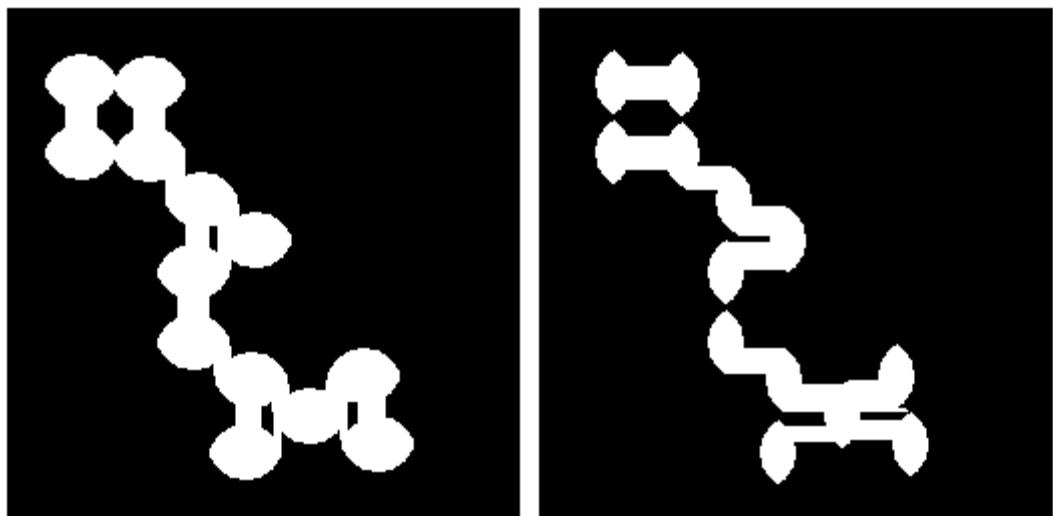


Figura 76. Ejemplos de usar un elemento estructural en forma de línea horizontal y vertical.

Fuente: <http://matlab.izmiran.ru/help/toolbox/images/imerode.html>

No obstante, el elemento estructural por sí mismo carece de sentido y ha de entenderse más en detalle con los operadores morfológicos que se verán a continuación.

8.4. Erosión y dilatación



Accede al vídeo «Erosión y dilatación» a través del aula virtual

La erosión y la dilatación son los operadores morfológicos esenciales. A partir de ellos, se crean el resto de operadores. Aunque lo parezca (y antes de dar definición alguna sobre ellos), es importante hacer constar que la erosión no es la inversa de la dilatación ni viceversa.

La **dilatación** se define matemáticamente como:

$$A \oplus B = \bigcap_{b \in B} A_b$$

Donde:

- ▶ A es la imagen original.
- ▶ B es el elemento estructural.

La función de la operación dilatación consiste en potenciar y aumentar los contornos de una imagen binaria, haciendo que los detalles se magnifiquen e incluso que algunos objetos dentro de una imagen, que originalmente estén separados, se fusionen.

Este tipo de funcionalidad tiene mucha utilidad cuando, tras una segmentación, hay objetos que se han quedado separados, pero cercanos en la imagen. La manera de implementar computacionalmente el operador dilatación (luego lo veremos también con el operador erosión) es de forma iterativa, comprobando píxel a píxel dentro de la imagen qué relación tienen el elemento estructural (B) y la imagen original (A).

En este caso, el operador mirará si el elemento estructural B coincide al menos en un píxel con la imagen original A. En ese caso, añadirá a dicha imagen el elemento estructural B, haciendo que la imagen aumente en la zona de los contornos y manteniéndose constante en el interior de la misma. Posteriormente veremos con un ejemplo cómo funciona este procedimiento.

La **erosión**, matemáticamente, se define como:

$$A \ominus B = \{z \in E | Bz \subseteq A\}$$

Donde:

- ▶ A es la imagen original.
- ▶ B es el elemento estructural.
- ▶ E es una generalización del espacio euclídeo, es decir, engloba todos los posibles puntos existentes incluyendo los de A.

La principal utilidad de la erosión como operador morfológico es el de eliminar detalles, reducir contornos y, en algunos casos, desunir objetos.

La erosión no conserva la forma original: tras usar este operador, la forma resultante puede haberse erosionado tanto que no conserve la morfología original. La manera de implementar computacionalmente el operador erosión es también de forma iterativa, comprobando píxel a píxel dentro de la imagen qué relación tienen el elemento estructural (B) y la imagen original (A).

En este caso, el operador mirará píxel a píxel si el elemento estructural B está contenido íntegramente en la imagen original A. En ese caso, se quedará únicamente con la posición/píxel donde ambos elementos (A y B) coincidan. De esta manera, el contorno de la imagen se reduce considerablemente.

Sin embargo, la mejor manera de ver ambos operadores es mediante un ejemplo visual. Primero empezaremos con el de dilatación y posteriormente aplicaremos el operador erosión sobre la imagen resultante de la dilatación.

Vamos a empezar con este ejemplo, donde a la izquierda está el elemento estructural, aunque tenga forma de cruz, puede ser un elemento estructural de disco o diamante de radio 3 píxeles. Y a la derecha, la imagen sobre la que se quiere aplicar el operador morfológico.

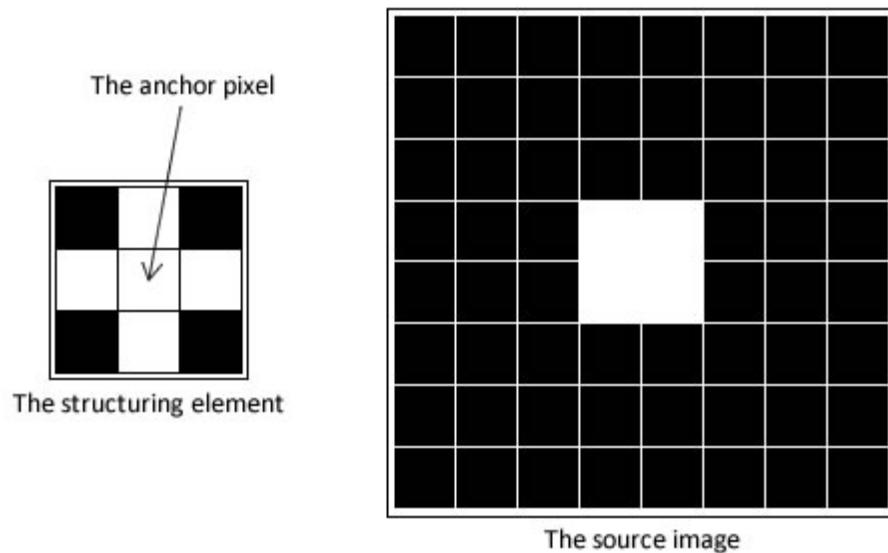


Figura 77. Elemento estructural en forma de cruz y la imagen sobre el que se va a realizar la operación morfológica.

Fuente: <http://aishack.in/tutorials/mathematical-morphology/>

Independientemente del elemento estructural empleado, la imagen resultante después de aplicar la dilatación ha de ser mayor (en el sentido de número de píxeles en blanco) que en la imagen original.

Como se ha descrito anteriormente, el operador dilatación irá moviendo el elemento estructural a lo largo de la imagen A, comprobando si A y B coinciden en al menos un píxel. Inicialmente, no existe tal solape y, por lo tanto, el operador morfológico no devuelve ningún valor en blanco.

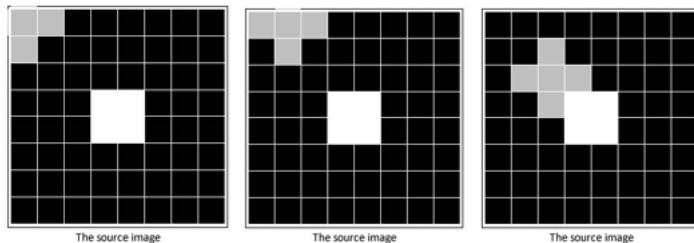


Figura 78. Elemento estructural en forma de cruz y la imagen sobre el que se va a realizar la operación morfológica.

Fuente: Adaptado de <http://aishack.in/tutorials/mathematical-morphology/>

Sin embargo, cuando el elemento estructural, toca en **al menos un píxel**, la zona de blancos dentro de la imagen original, es entonces donde el operador dilatación aplica su definición. Es decir, en aquellas posiciones donde ambos coincidan se hará la unión de ambos operadores, tal y como puede verse en la siguiente imagen.

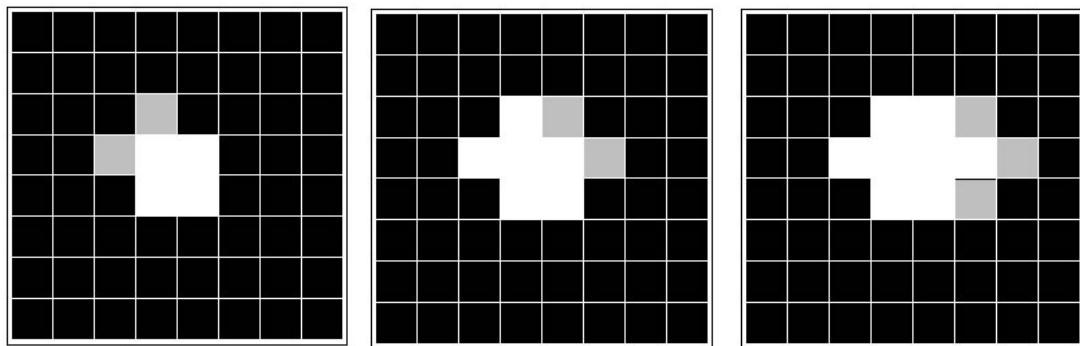


Figura 79. Paso a paso de cómo el elemento estructural entra en contacto con una zona dentro de la imagen con píxeles diferentes a cero (negro).

Fuente: Adaptado de <http://aishack.in/tutorials/mathematical-morphology/>

En este caso, y puesto que se trata de una dilatación, se conserva la unión de ambos elementos. Finalmente, tras recorrer toda la imagen, el resultado es que el cuadrado se ha dilatado, ha aumentado de tamaño con la forma del elemento estructural. En ningún momento se aprecia que se haya conservado la forma inicial (se trataba de un cuadrado) y tiene forma de cruz.

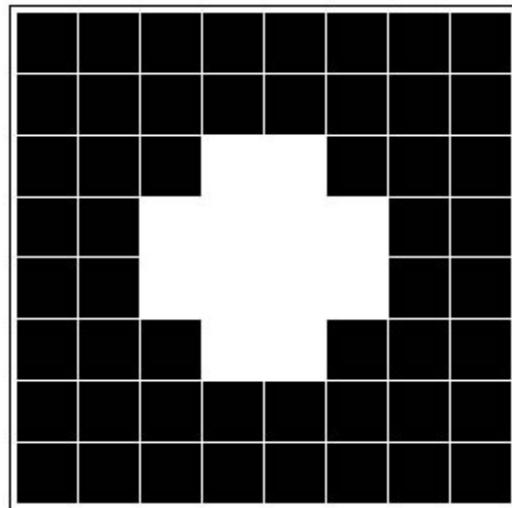


Figura 80. Resultado final del operador morfológico dilatación sobre un cuadrado de 2x2 píxeles y con un elemento estructural en forma de cruz.

Fuente: <http://aishack.in/tutorials/mathematical-morphology/>

El siguiente proceso va a consistir en aplicar la erosión a la imagen dilatada anterior. Es importante tener en mente que al aplicar el operador erosión, el número de **píxeles en blanco se verá disminuido** en el resultado.

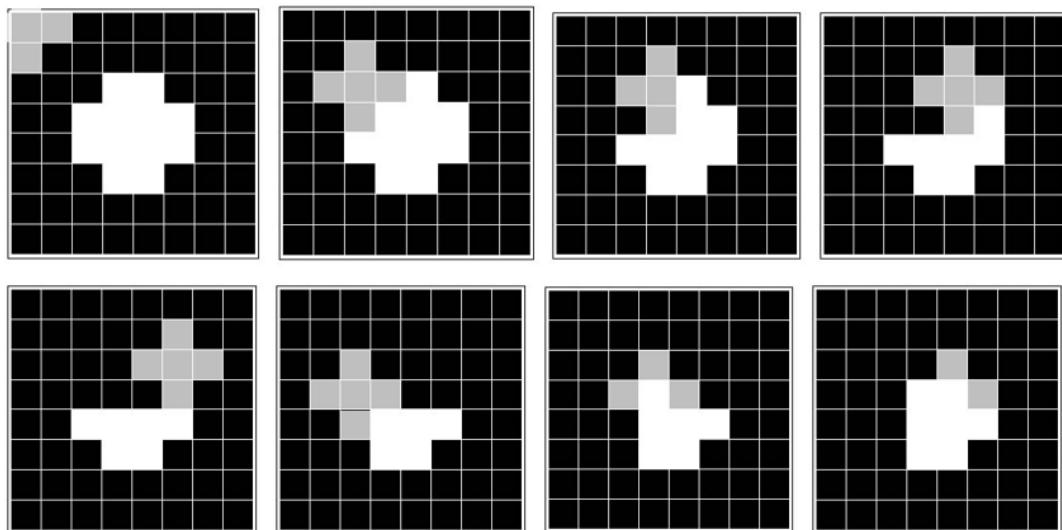


Figura 81. Proceso de erosión con un operador en forma de cruz.
Fuente: Adaptado de <http://aishack.in/tutorials/mathematical-morphology/>

En la imagen resultante se han eliminado detalles en el contorno, quedándose con la forma inicial de cuadrado de 2x2 píxeles, aunque la imagen no conserva la forma original. Tras la erosión, la imagen tiene menos píxeles en blanco que procesar y para ciertas funciones, como el conteo de objetos, puede llegar a ser más fácil que con la imagen original. Por otro lado, si lo que se quiere es detectar el centro de un objeto, el resultado de la erosión también ayudaría, pues facilitaría el cálculo del centro del objeto al haber reducido los contornos y detalles del mismo.

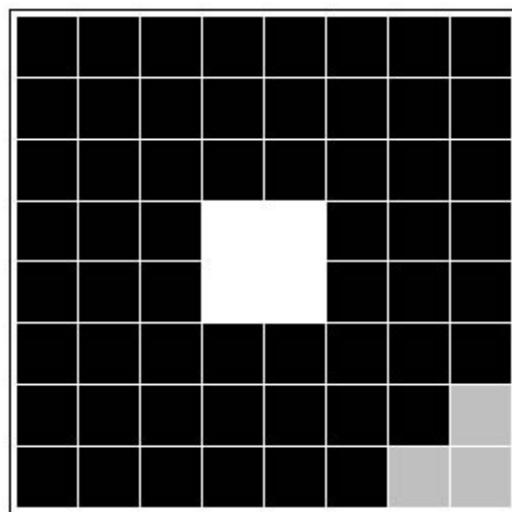


Figura 82. Resultado final del operador morfológico erosión sobre una cruz y con un elemento estructural en forma de cruz. El resultado es una imagen que no conserva la forma original.

Fuente: <http://aishack.in/tutorials/mathematical-morphology/>

Por último, y aunque en este caso haya sucedido así, repetimos que la erosión no es la inversa de la dilatación ni viceversa.

8.5. Apertura y clausura



Accede al vídeo «Apertura y clausura» a través del aula virtual

A diferencia de los operadores anteriores, la apertura y clausura son los primeros operadores combinación de varias operaciones morfológicas. Su mayor propiedad es que son capaces de respetar, en la medida de lo posible, la morfología adicional. Con lo cual, tienen las propiedades de la erosión y la dilatación, pero **manteniendo la forma original**. En otras palabras, son capaces de eliminar o potenciar los detalles y aun así, mantienen la proporción original.

La **apertura**, conocida en inglés como *opening*, es un operador morfológico que es la sucesión de una erosión seguida por una dilatación. Matemáticamente, la apertura se define como:

$$A \circ B = (A \ominus B) \oplus B$$

Donde:

- ▶ A es la imagen original.
- ▶ B es el elemento estructural.

Principalmente, este operador sirve para separar objetos, ya que elimina los posibles detalles que haya en la imagen en función del elemento estructural.

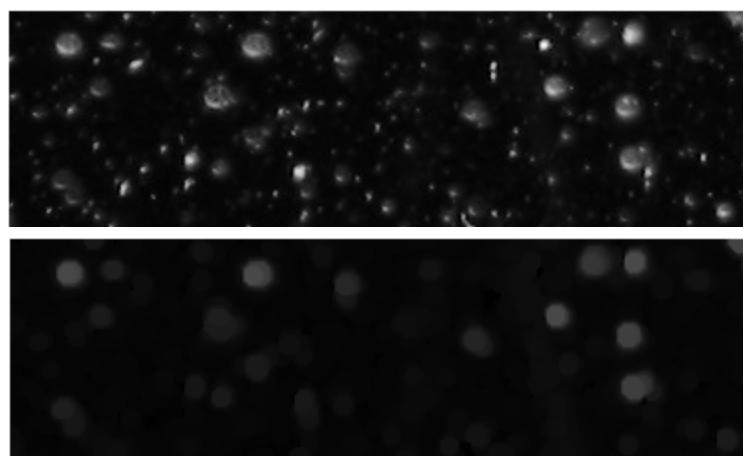


Figura 83. Resultado de aplicar el operador morfológico apertura.

Fuente: <https://es.mathworks.com/help/images/ref/imreconstruct.html>

En este ejemplo, vemos el resultado de aplicar el operador morfológico apertura con un elemento estructural disco de 5 píxeles de radio. El efecto es el de haber eliminado los detalles (puntos) más pequeños, manteniendo los objetos más grandes.

La **clausura**, conocida en inglés como *closing*, es un operador morfológico que realiza primeramente una dilatación y posteriormente una erosión.

Matemáticamente, la clausura se define como:

$$A \cdot B = (A \oplus B) \ominus B$$

Donde:

- ▶ A es la imagen original.
- ▶ B es el elemento estructural.

La clausura suele emplearse para reafirmar y consolidar formas dentro de una imagen.

Como podemos apreciar en la siguiente imagen, el operador ha transformado la figura inicial en un solo bloque.



Figura 84. Clausura con un disco de radio 20 píxeles.

Fuente: https://users.cs.cf.ac.uk/Dave.Marshall/Vision_lecture/node34.html

8.6. Gradiente morfológico



Accede al vídeo «Gradiente morfológico» a través del aula virtual

Otro operador muy empleado en morfología matemática es el gradiente morfológico. Se define como la diferencia entre la dilatación y la erosión. La idea intuitiva es la de **detectar bordes en la imagen**; ya que la dilatación aumenta una imagen y la erosión la disminuye, la diferencia ha de ser similar a los bordes de una imagen.

Aquí tenemos un ejemplo de gradiente morfológico aplicado a una imagen con granos de arroz. Se ha utilizado un elemento estructural de disco con radio de 3 píxeles

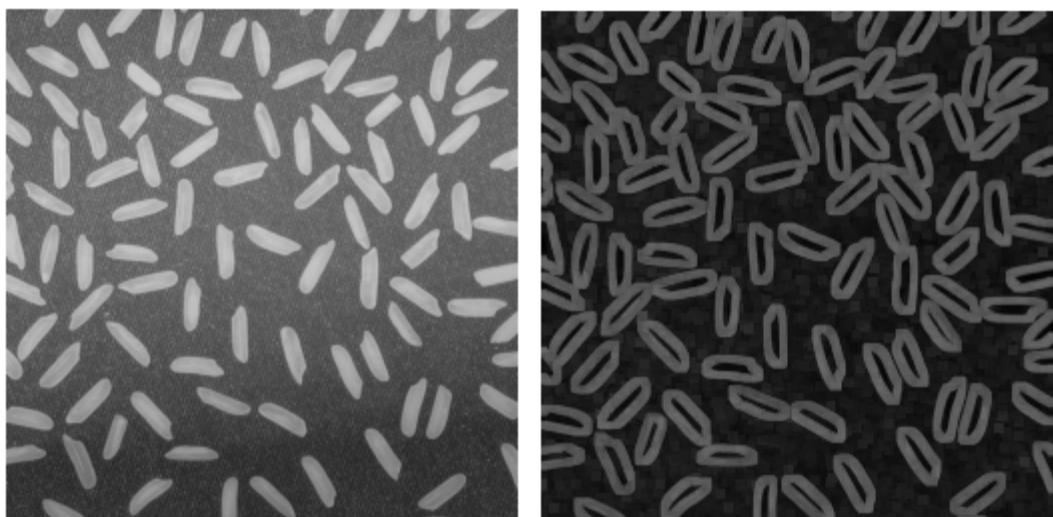


Figura 85. Ejemplo de gradiente morfológico.

Fuente: Adaptado de <https://imagej.net/MorphoLibJ>

8.7. Top Hat



Accede al vídeo «Top Hat» a través del aula virtual

El operador Top Hat se define como la diferencia entre la imagen original y el operador morfológico apertura. Devuelve una imagen con únicamente los detalles que la apertura ha eliminado.

Esta transformada es muy útil en el caso de la detección de texto dentro de una imagen, como leer las marcas de una moneda, ya que **se centra en los detalles**. Pero si quisiésemos contar el número de monedas, esta transformada no aportaría ningún valor.

En esta imagen, vemos el resultado de aplicar la transformada *Top Hat* mediante un disco de radio de 5 píxeles.

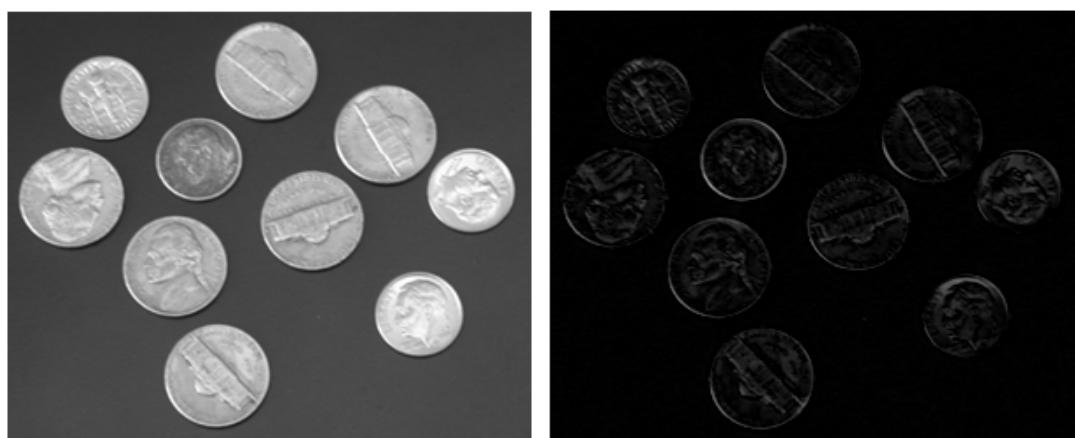


Figura 86. Detección de los detalles de una moneda mediante la transformada *Top Hat*.

Fuente: Adaptado de <https://imagej.net/MorphoLibJ>

8.8. Ejercicio práctico

Con todo lo anterior explicado, vamos a realizar un ejercicio práctico. Partiendo de esta conocida imagen, el objetivo es quedarse únicamente con el fotógrafo y eliminar el resto de componentes, incluyendo la cámara y los edificios de atrás.



Figura 87. Detección de los detalles de una moneda mediante la transformada *Top Hat*.

Fuente: <https://blogs.mathworks.com/steve/2012/11/20/image-effects-part-2/>

Existen múltiples soluciones para hacer esto, pero los pasos que vamos a describir son una aproximación adecuada.

Lo primero de todo es eliminar las patas de la cámara. Esto podemos hacerlo con morfología matemática mediante un **operador de clausura** haciendo uso de un elemento estructural que sea perpendicular a la dirección de las patas de la cámara. La razón de esto es eliminar dichas patas lo máximo posible, puesto que al hacer la clausura difuminaría estos objetos. Así, en esta imagen vemos la aplicación de clausura con un elemento estructural de línea de 45 px y 45° de inclinación.



Figura 88. Resultado de aplicar un operador de clausura.

Fuente: Adaptado de <https://blogs.mathworks.com/steve/2012/11/20/image-effects-part-2/>

La forma del fotógrafo se ha conservado en gran medida, luego, para eliminar las posibles líneas provocadas por el anterior operador, es conveniente realizar un operador de apertura (para consolidar las formas y eliminar detalles). De esta manera, y aunque parezca muy borroso, la figura del cámara se ha conservado habiéndose eliminado la cámara y el fondo.

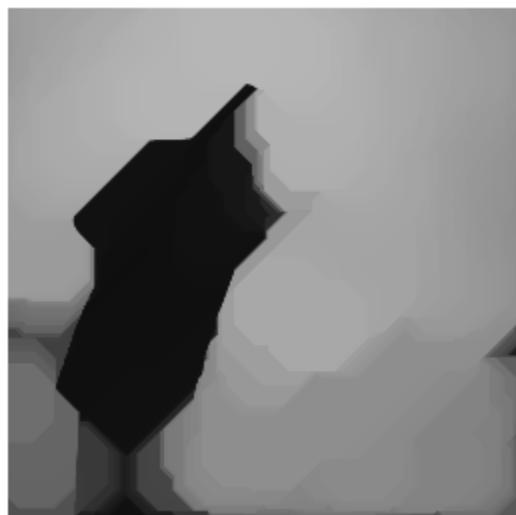


Figura 89. Resultado de aplicar un operador de apertura.

Fuente: Adaptado de <https://blogs.mathworks.com/steve/2012/11/20/image-effects-part-2/>

Únicamente queda aplicar este filtro a la imagen original. En este caso lo haremos multiplicando ambas imágenes, componente a componente, una manera muy sencilla de aplicar el filtro.

El resultado puede apreciarse en la imagen siguiente, donde hemos eliminado todo el fondo y la cámara, quedándonos únicamente con la parte correspondiente al fotógrafo. Además, como resultado del filtrado de multiplicación de las dos imágenes se ha obtenido un reescalado que permite ver más detalles del abrigo que inicialmente no eran perceptibles.



Figura 90. Resultado final del ejercicio práctico.

Fuente: Adaptado de <https://blogs.mathworks.com/steve/2012/11/20/image-effects-part-2/>

8.9. Referencias bibliográficas

González, R. C. y Woods, R. E. (2008). *Digital image processing*. New Jersey: Pearson Education.



Accede a los ejercicios de autoevaluación a través del aula virtual

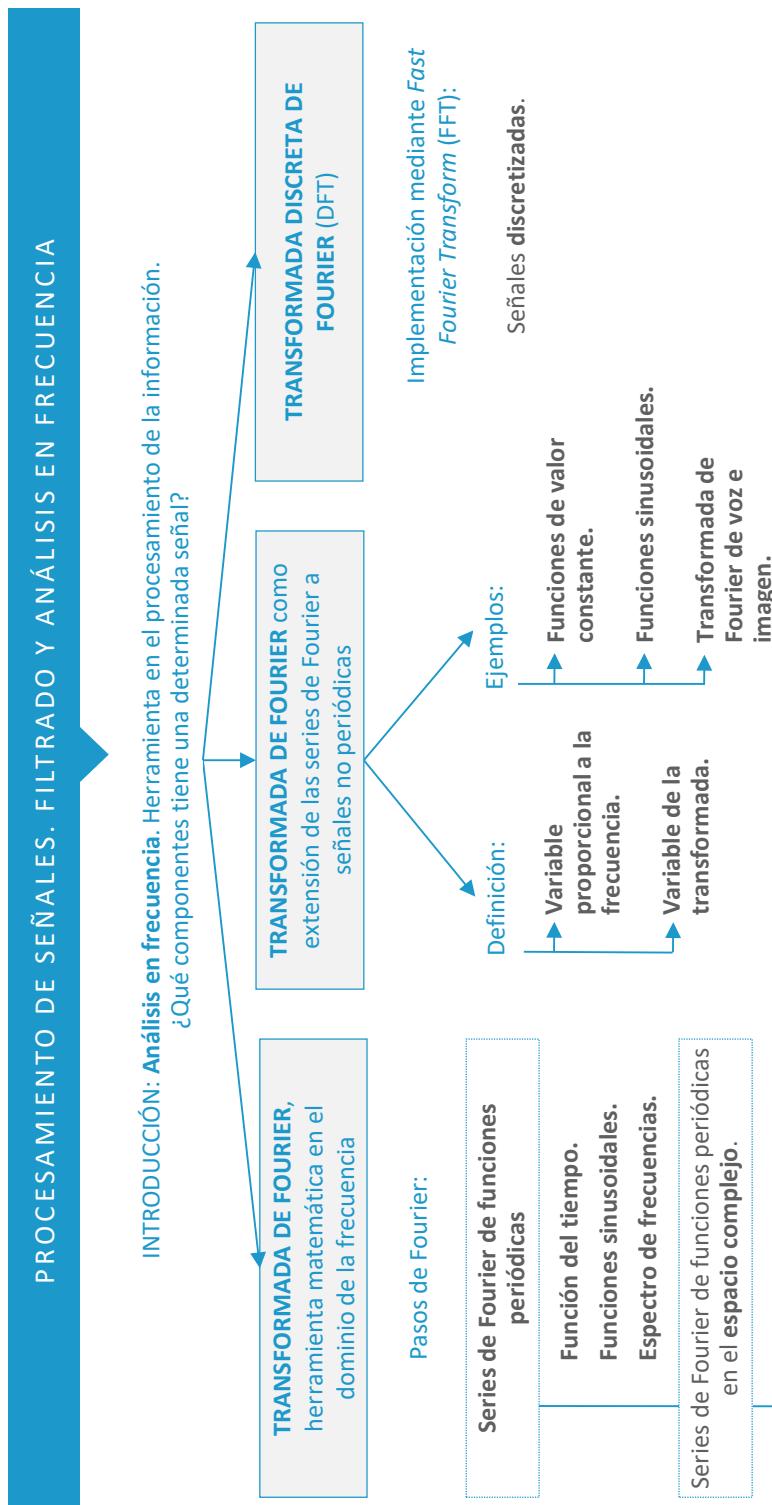


Accede al vídeo «Resumen del tema» a través del aula virtual

Percepción Computacional

Procesamiento de señales. Filtrado y análisis en frecuencia

Esquema. Tema 9



Ideas clave. Tema 9

9.1. ¿Cómo estudiar este tema?

Para estudiar este tema deberás leer con atención las ideas clave que se desarrollan a continuación.

9.2. Objetivos

Esta asignatura persigue presentar el concepto de análisis en frecuencia desde su origen (series de Fourier) hasta su implementación actual (FFT, *Fast Fourier Transform*). Para ello, se revisan los conceptos de análisis en frecuencia y transformada de Fourier, así como las ventajas y desventajas del uso de estas herramientas en el procesamiento computacional.

Se sentarán las bases del análisis en frecuencia, lo que te permitirá abordar otros algoritmos y herramientas como transformada del coseno (DCT), Wavelets o filtrados en frecuencia para imágenes.

9.3. Introducción al análisis en frecuencia



Accede al vídeo «Introducción. Filtrado y análisis en frecuencia» a través del aula virtual



Il análisis en frecuencia es una de las herramientas más potentes e innovadoras que existen en el procesamiento de la información. En este apartado se va a describir, con un ejemplo muy cotidiano: la señal de fibra

óptica y teléfono que llega a nuestros hogares, en qué consiste el análisis en frecuencia y por qué es tan importante.

Si el cable de red que va al router lo conectásemos a un osciloscopio o a cualquier otro dispositivo que nos permitiera medir voltajes e intensidades, veríamos algo parecido a esta imagen.

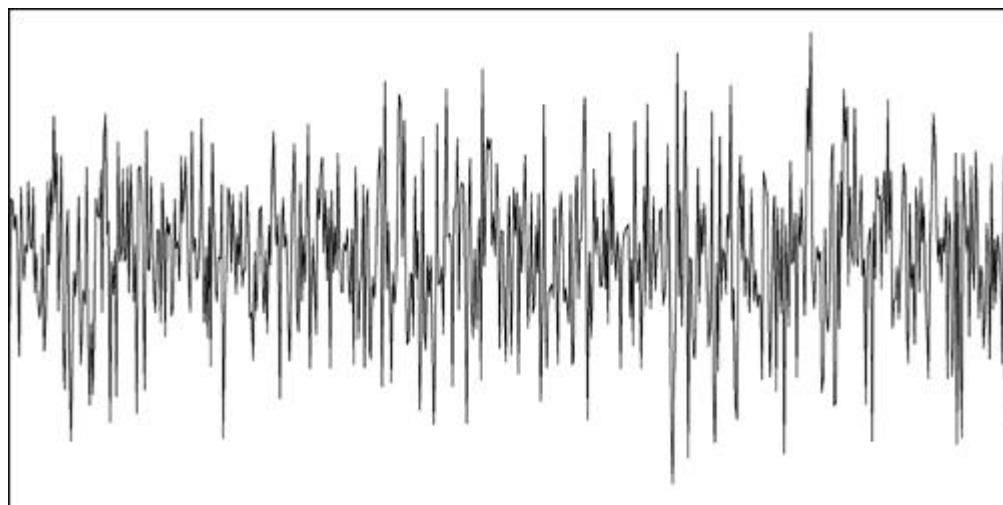


Figura 91. Ejemplo de señal compleja aperiódica.

Es el claro ejemplo de una señal compleja aperiódica o señal aleatoria, en algunos contextos se llama también caótica. En esta señal unidimensional podemos ver variaciones de intensidad y de voltaje correspondientes a la superposición de nuestra voz (llamadas telefónicas), Internet (datos que estamos recibiendo) y en algunos casos, incluso canales de televisión.

Todo ello sucede simultáneamente. Sin embargo, al esquematizar qué es lo que está pasando, nos encontramos con el siguiente ejemplo:

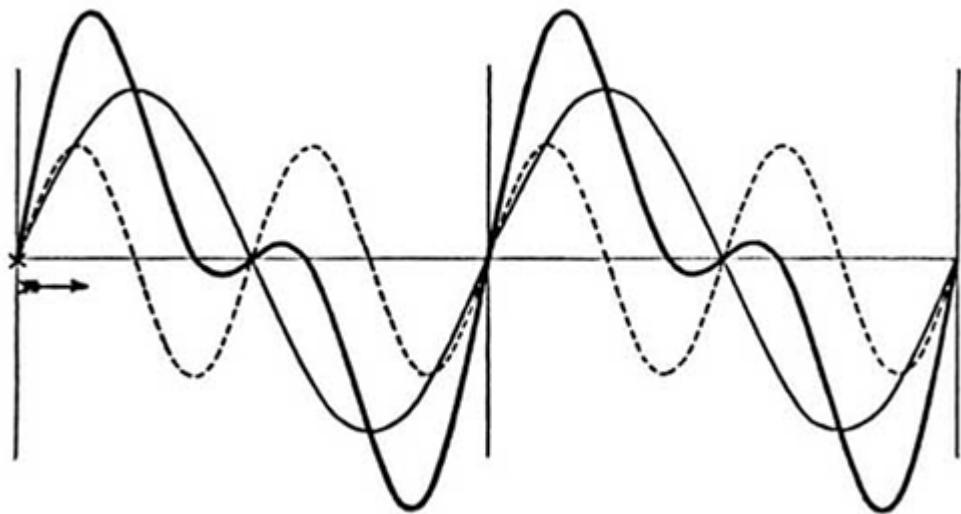


Figura 92. Ejemplo de señales superpuestas en el tiempo.

En este ejemplo, encontramos que existen tres tipos de señales que se superponen. De estas señales, podemos distinguir:

- ▶ La línea continua fina, correspondiente a un tipo de señal.
- ▶ La línea discontinua, correspondiente a un tipo de señal.
- ▶ La línea continua gruesa, suma de ambas señales.

Si nos centramos únicamente en la línea continua gruesa, no podemos saber *a priori* si es el resultado de la suma de dos señales, de un número determinado de señales o si es una única señal. De hecho, si se realizara un análisis de estas señales en el tiempo, podríamos obtener características como la media, el valor máximo, el valor mínimo, etc. Pero en cualquier caso, no podríamos saber el origen.

El **análisis en frecuencia** es la herramienta encargada de, dada la señal continua gruesa, analizar si esta proviene de la suma de dos o más señales y cómo de intensas (potencia) son dichas señales.

Volviendo al ejemplo de la fibra óptica y la voz en nuestras casas, el gráfico anterior sería una simplificación de la realidad. El análisis en frecuencia, en definitiva, es lo

que permitiría separar la fibra de la voz y de la televisión. Permitiría realizar un filtrado, que a nuestros ojos, parece inviable.

¿Y qué hace exactamente el análisis en frecuencia? Principalmente, lo que hace es barrer el espectro de frecuencias y detectar, mediante un producto escalar, cuánta señal está sucediendo a esa determinada frecuencia. Posteriormente lo veremos más en detalle.

Tomemos el siguiente gráfico. Si miramos el dominio del tiempo, vemos dos señales superpuestas. Si analizamos las frecuencias, vemos que ambas señales tienen comportamientos en frecuencias diferentes, lo que nos permitiría separar las señales en frecuencia, aunque estén completamente solapadas en el tiempo.

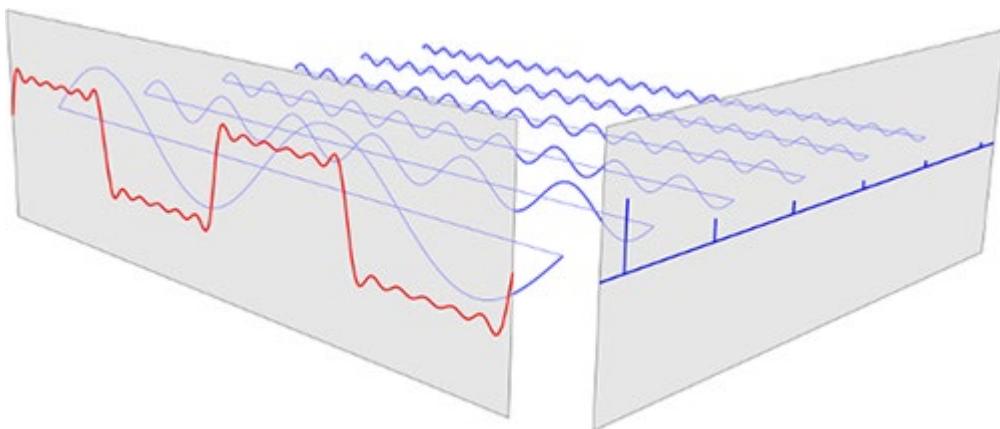


Figura 93. Ejemplo de cómo mediante el uso del dominio de la frecuencia podemos separar dos señales que en el tiempo son coincidentes.

Fuente: <https://tex.stackexchange.com/questions/127375/replicate-the-fourier-transform-time-frequency-domains-correspondence-illustrati>.

Un ejemplo más técnico es el de la siguiente figura. Podemos ver una señal en el tiempo claramente caótica. Es casi imposible entender qué está sucediendo en esa señal o a qué hace referencia. No obstante, el análisis en frecuencia ya nos indica que existen diferentes componentes, con lo cual, si se quisiera escuchar cada uno de forma clara, habría que realizar filtrados de frecuencia y no filtrados temporales.

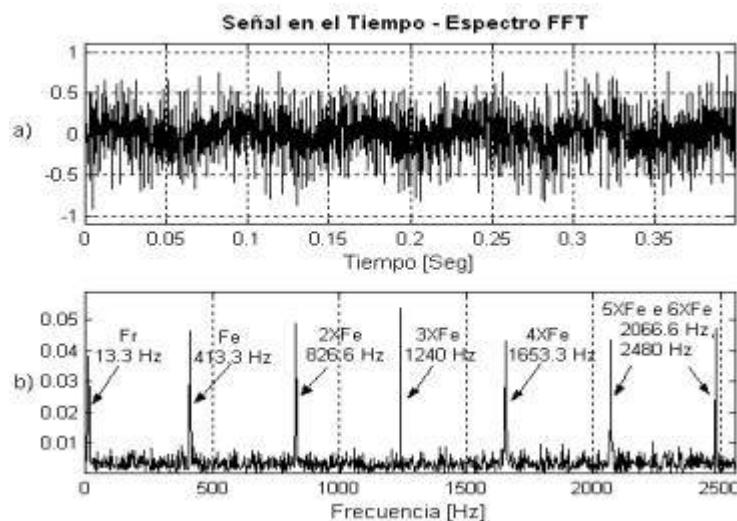


Figura 94. Comparación de representaciones Tiempo-Frecuencia.

Fuente: https://scielo.conicyt.cl/scielo.php?script=sci_arttext&pid=S0718-07642004000500003.

La herramienta principal de análisis en el dominio de la frecuencia es la **transformada de Fourier**, que permite:

- ▶ Entender qué componentes de frecuencia posee una determinada señal temporal y cuál es la intensidad de dichas componentes.
- ▶ Separar comportamientos temporales de comportamientos de frecuencia.
- ▶ Realizar filtrados en frecuencia y separar un determinado número de señales de forma sencilla y con razonamiento matemático, cuya labor en el análisis temporal habría sido imposible.

Por último, el uso de la transformada de Fourier está muy presente en nuestros días, tan presente que sin esta herramienta matemática hoy no podríamos hablar por el móvil, comunicarnos con los satélites, enviar sondas a Marte, navegar por Internet o incluso detectar si, cuando introducimos una moneda en una máquina de *parking*, es de 1 € o de 2 €.

La transformada de Fourier, junto con otras fórmulas matemáticas, está considerada por la comunidad científica como uno de los avances más considerables de la humanidad.

9.4. La transformada de Fourier



Accede al vídeo «La transformada de Fourier» a través del aula virtual

A

ntes de comenzar a definirla, es importante entender qué camino siguió Fourier para llegar a formularla.

- Entender el origen de la transformada de Fourier: series de Fourier de funciones periódicas.
- Cómo se extienden dichas funciones al campo complejo.
- Definición de la transformada de Fourier como una extensión de las series de Fourier a señales no periódicas.

Figura 95. Pasos seguidos por Fourier.

Series de Fourier de funciones periódicas

En primer lugar, Fourier entendía que **cualquier función del tiempo** $f(t)$:

- ▶ Real.
- ▶ Periódica de período T , si para todo t se cumple que $f(t) = f(t + T)$
- ▶ Frecuencia continua:

$$\omega_1 = \frac{2\pi}{T}$$

Puede ser expresada mediante una serie infinita de funciones sinusoidales (senos y cosenos) de frecuencias ω_n , múltiplos de ω_1 , es decir:

$$\omega_n = n \cdot \omega_1$$

Donde $n = 1, 2, \dots, \infty$.

Por lo que en forma de ecuación tenemos que:

$$f(t) = A_0 + \sum_{n=1}^{\infty} A_n \cos(n \cdot \omega_1 t) + \sum_{n=1}^{\infty} B_n \sin(n \cdot \omega_1 t)$$

En resumen, la restricción inicial de que $f(t)$ fuera una función periódica puede parecer realmente muy limitativa, pero lo cierto es que muchos fenómenos físicos poseen comportamientos periódicos, luego entender funciones periódicas complejas en términos de senos y cosenos era algo ciertamente avanzado, ya que las **funciones sinusoidales** tienen propiedades matemáticas muy positivas: continuidad, derivabilidad, perpendicularidad/ortogonalidad desde el punto de vista del producto escalar, etc.

Únicamente queda calcular los términos A_0 , A_n con $n = 1, 2, \dots, \infty$ y B_n con $n = 1, 2, \dots, \infty$. Dichos términos tienen las siguientes relaciones (que no vamos a demostrar en este apartado):

$$\bar{f}(t) = \frac{1}{T} \int_{t_0}^{t_0+T} f(t) dt = A_0$$
$$B_n = \frac{2}{T} \int_0^T A_n = \frac{2}{T} \int_0^T f(t) \cos(n \cdot \omega_1 t) dt$$

La interpretación de estos tres términos es sencilla y, si has estudiado matemáticas, será más sencilla aún de entender:

- ▶ Fourier imaginó que toda función periódica se podría expresar como una combinación lineal (sumas, restas y coeficientes) de funciones sinusoidales.

- ▶ Dichas funciones sinusoidales son ortogonales entre sí, lo que quiere decir que las funciones están definidas de tal manera que no pueden expresarse como combinación lineales entre sí.
- ▶ Los coeficientes A_0 , A_n y B_n son el producto escalar de la función $f(t)$ y la función sinusoidal correspondiente, en otras palabras, puede entenderse como la proyección de $f(t)$ sobre las funciones sinusoidales.
- ▶ En otras palabras, cuanto más se parezca $f(t)$ a una sinusoidal, mayor serán los coeficientes A y B .

Este último punto es de vital importancia. El producto escalar nos da una medida de cómo de parecidas son dos funciones, entre otras muchas propiedades. Con lo cual, aquellas componentes A_0 , A_n y B_n que sean más altos indicarán un mayor parecido con una sinusoidal.

Esto ya empieza a estar relacionado con lo que hemos hablado en la introducción, es decir, el poder encontrar una herramienta que nos diga qué componentes de frecuencia tiene una determinada señal, en este caso llamada $f(t)$.

Por último, y antes de continuar, únicamente decir que el gráfico de A_n y B_n en función de n o de ω_n se conoce como el **espectro de frecuencias** de la función periódica $f(t)$.

Series de Fourier de funciones periódicas en el espacio complejo

Cuando hacemos referencia al espacio complejo, Fourier quiere incluir funciones $f(t)$ que tengan valores en el campo complejo, es decir, con el término $j = \sqrt{-1}$.

Evidentemente, no existe ninguna función que pueda medirse que dé un valor complejo, pero muchos fenómenos físicos se modelan con estos, principalmente los campos electromagnéticos, por lo que si se quiere extender las series de Fourier a esos fenómenos físicos, es necesario llevar las series de Fourier a este espacio.

Para entender dicha extensión, el alumno debe de estar familiarizado con la **fórmula de Euler**:

$$e^{j\theta} = \cos \theta + j \sin \theta$$

Para ello escribimos la serie de la siguiente manera:

$$f(t) = A_0 + \sum_{n=1}^{\infty} C_n \cos(n \cdot \omega_1 t + \phi_n)$$

Donde se cumplen las relaciones:

- ▶ $A_n = C_n \cos \varphi$.
- ▶ $B_n = C_n \sin \varphi$.

Finalmente, $f(t)$ puede expresarse como suma de exponenciales complejas al obtener G_n a partir de $f(t)$ usando la integral:

$$f(t) = A_0 + \sum_{n=1}^{\infty} G_n e^{jn \cdot \omega_1 t}$$
$$G_n = \frac{1}{T} \int_{-T/2}^{T/2} f(t) e^{-jn \cdot \omega_1 t} dt$$

En este tema no se va a justificar matemáticamente los pasos que conducen a esta demostración puesto que está fuera del enfoque de la asignatura. No obstante, recomendamos que intentes resolver esta demostración.

La interpretación de $f(t)$ en función de las exponenciales complejas es de nuevo similar a las series de Fourier que vimos inicialmente: es la proyección de $f(t)$ sobre dichas exponenciales complejas.

Definición de la transformada de Fourier

Llegados a este punto, podemos preguntarnos para qué sirve la transformada de Fourier si ya tenemos los coeficientes A_n y B_n , que ya nos indican las componentes espectrales de $f(t)$. Muy sencillo.

La transformada de Fourier sirve para extender la serie de Fourier a funciones no periódicas (periódicas con período infinito).

En realidad es un proceso límite si juzgamos que la función no periódica la aproximamos a una que se repite a una distancia (tiempo) muy lejana de nosotros.

En definitiva, las expresiones son las siguientes:

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-j\omega t} dt$$
$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{j\omega t} d\omega$$

Donde:

- ▶ $F(\omega)$ representa la transformada de Fourier.
- ▶ $f(t)$ se representa como la inversa de la transformada de Fourier.

La reflexión que aplica ahora es: ¿por qué $F(\omega)$ es función de ω ? ¿Qué es ω ? Desde ahora en adelante (y en toda la literatura que existe sobre este tema), se utiliza ω como **variable proporcional a la frecuencia**:

$$\omega = 2\pi f$$

Donde f es la frecuencia.

El uso de ω queda justificado debido a la nomenclatura y no confundir la F de Fourier con la f de frecuencia. En otras palabras, ω es la **variable de la transformada**, que ya no tiene información sobre el tiempo (t). Nótese que la única variable t es la variable de integración que desaparece cuando la integral se resuelve.

Y además, de todo esto se deduce que no se pierde ningún tipo de información cuando se hace la transformada.

Una de las preguntas más comunes cuando se estudia la transformada de Fourier es si existe algún dispositivo que permita verla. La respuesta rápida es no. Debes recordar que la transformada de Fourier, $F(\omega)$, está en el espacio complejo y es solo una entelequia matemática. Lo único que puede hacerse es una aproximación del módulo de la transformada, $|F(\omega)|$, mediante computación (se verán algunos ejemplos).

De hecho, gracias a la computación podemos obtener representaciones de la transformada de Fourier muy interesantes, como esta que vemos a continuación. En ella puede apreciarse cómo las **componentes temporales se transforman en componentes frecuenciales**.

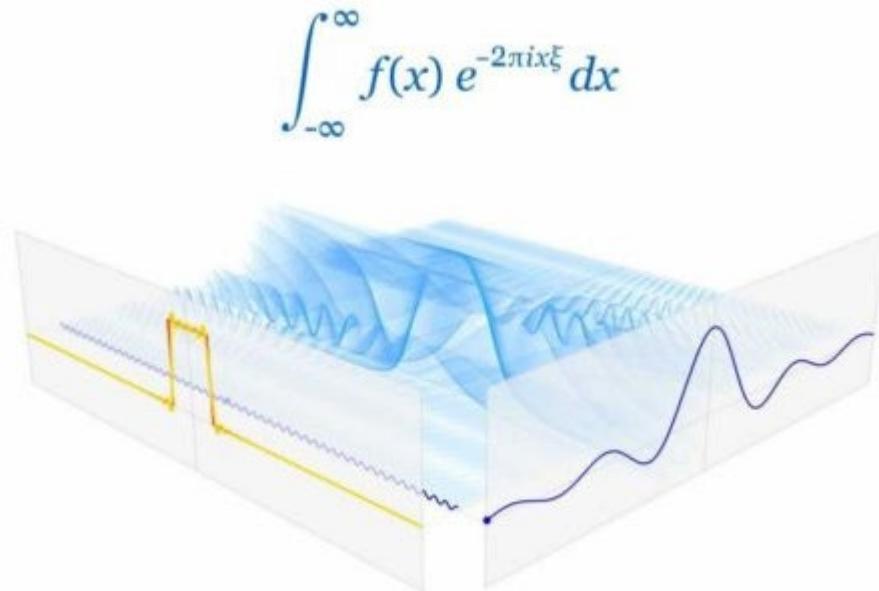


Figura 96. Vista simultánea de la función pulso (en amarillo) y su transformada de Fourier (azul).

Fuente: <http://puyaa.ir/fourier-analysis-nutshell/>

Ejemplos de transformada de Fourier

A continuación exponemos algunos ejemplos de transformada de Fourier:

PARES DE TRANSFORMADAS DE FOURIER

Función (t)		Función (w)	
$\delta(t)$		1	
1		$2\pi\delta(w)$	
$w(t)$		$\pi\delta(w) + \frac{1}{jw}$	
$\Pi(t/\tau)$		$\frac{\sin(\omega\tau/2)}{\omega\tau/2}$	
$\cos(\omega_0 t)$		$\pi[\delta(w - \omega_0) + \delta(w + \omega_0)]$	
$\sin(\omega_0 t)$		$j\pi[\delta(w + \omega_0) - \delta(w - \omega_0)]$	

Figura 97. Tabla con los pares de transformadas más comunes.

Fuente: <http://chismesycircuitos.blogspot.com.es/2011/09/tabla-transformadas-de-fourier-ut-1-del.html>

De esta tabla, solo vamos a reflexionar sobre dos funciones: la función constante y las señales sinusoidales.

Funciones de valor constante. Desde el punto de vista de percepción computacional, un valor constante puede ser ruido o la componente continua de una señal unidimensional. Con respecto a estas funciones de valor constante, se aprecia como la energía se concentra en el dominio de la frecuencia entorno a la frecuencia cero.

- ▶ Hasta cierto punto parece lógico, ya que una señal continua no posee frecuencia alguna.
- ▶ Esto quiere decir que, en el dominio de la frecuencia, las componentes constantes en el tiempo están muy cercanas a cero y, por lo tanto, serán fácilmente eliminables por filtros.

Funciones sinusoidales. Tonos puros de sonido, como puede ser el tono producido por una nota en un instrumento musical. En estas se observa que se transforman en otras dos funciones denominadas **delta de Dirac**, las cuales indican que la función solo está en un determinado punto.

- ▶ Lógico si pensamos que la transformada de Fourier mide las componentes frecuenciales de una señal y, si esta es un tono puro, solo podrá tener un determinado valor en frecuencia.
- ▶ En la tabla se aprecian dos deltas a ambos lados de los ejes horizontales. Por ahora, no debes plantearte a qué se debe este fenómeno, pero es el reflejo de que se está trabajando en el dominio complejo de la frecuencia y, aunque físicamente no existen frecuencias negativas, matemáticamente sí son posibles.

Veamos ahora dos ejemplos con datos reales, uno para voz y otro para imagen.

La transformada de Fourier de voz suele ocupar unos 4 KHz (ancho de banda de la voz). Es decir, cuando se hace la transformada de Fourier de cualquier fichero de sonido donde hable una persona, nunca (o muy difícilmente) se superarán los 4 KHz.

Esto puede verse en el siguiente ejemplo donde en rojo vemos la señal de voz (un segundo de duración) y en azul, la transformada de Fourier calculada con uno de los algoritmos más importantes (*Fast Fourier Transform*, FFT). La transformada de Fourier puede variar si, en vez de tomar un segundo de duración de voz, usamos una hora, pero en ningún caso variará significativamente y casi nunca superará esos 4 KHz de ancho de banda.

De hecho (se verá mejor en la parte de filtrado basado en Fourier), si quisiésemos eliminar ruidos que no fueran voz, tendríamos que eliminar las componentes superiores a 4 KHz y oiríamos la voz nítidamente. Esto es lo que hacen nuestros dispositivos móviles cuando hablamos: filtran todas las componentes por encima de un valor y solo transmiten frecuencias menores que 4KHz.

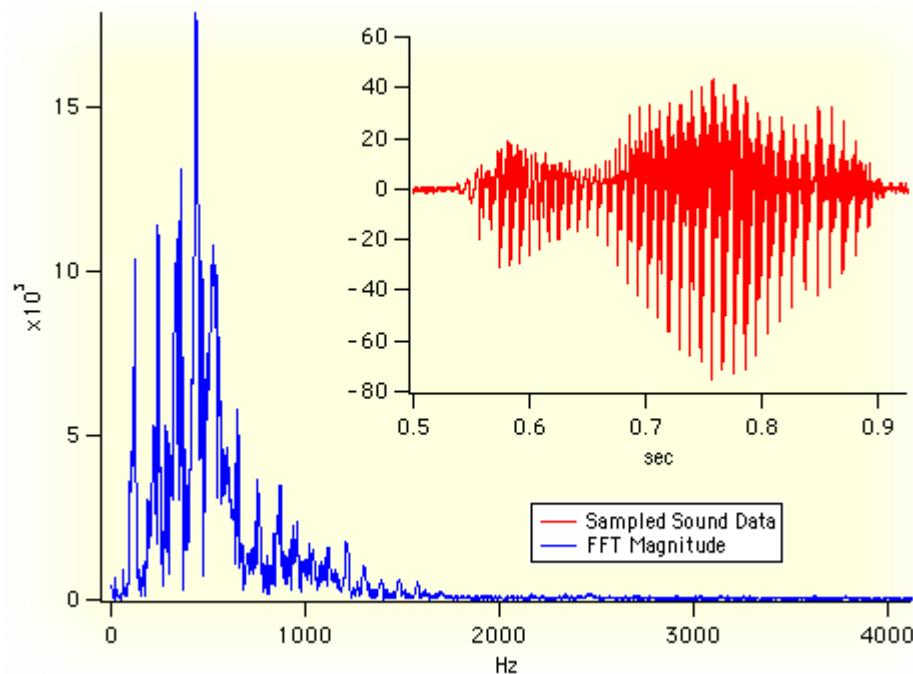


Figura 98. Ejemplo de transformación de Fourier.

Fuente: <http://www.wavemetrics.com/index.html>

La transformada de Fourier de imagen. Con respecto a las imágenes, la transformación es más difícil de interpretar, pero aún sigue siendo válida.

Primero, se presentan ejemplos con frecuencias puras para que se aprecie cómo se aplicaría Fourier:

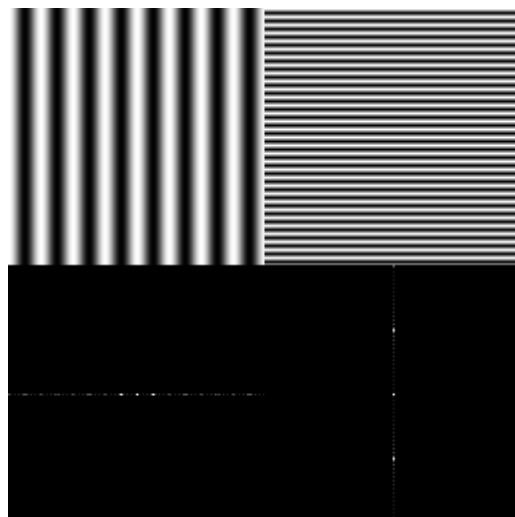


Figura 99. Transformada de Fourier de patrones de imágenes puros (rayas verticales y horizontales).

Fuente: <https://www.cs.unm.edu/~brayer/vision/fourier.html>

En sí misma, la transformada de Fourier puede ofrecer también la posibilidad de proporcionar características que identifiquen de forma unívoca qué está visualizándose en la imagen. Un ejemplo claro es el de los caracteres escritos a mano. En la siguiente figura, las componentes en frecuencia varían en función de los trazos. Las más circulares, como la «B» y la «Q», ofrecen componentes en frecuencia completamente diferentes de otras letras como la «T» o la «K».

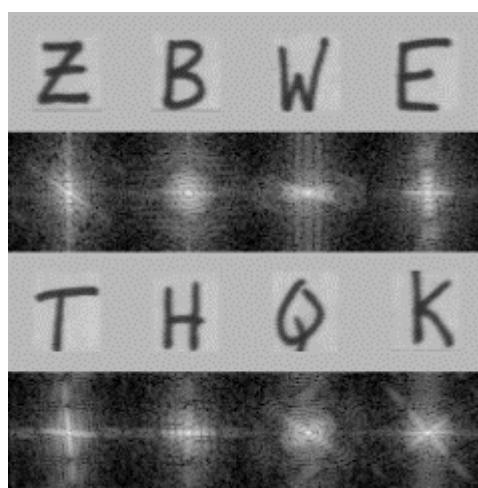


Figura 100. Transformada de Fourier de caracteres escritos a mano.

Fuente: <https://www.cs.unm.edu/~brayer/vision/fourier.html>

Como hemos explicado, se aprecia cómo las componentes en frecuencia son diferentes y permitirían una extracción de características diferente para poder distinguir las letras.

Por último, un uso de la transformada de Fourier es eliminar o detectar el ruido que existe en la imagen. La transformada de Fourier se ve afectada al desenfocarse la imagen: las componentes espectrales varían, ya que se pierde nitidez, y el ruido aparece muy marcado en una de las componentes verticales, donde aparece una raya más blanca que pasa por el origen.

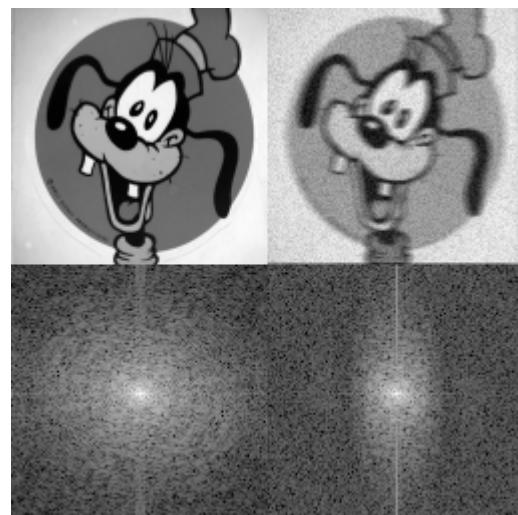


Figura 101. Efecto en la transformada de Fourier de desenfocar la imagen y aplicar ruido en ella.

Fuente: <https://www.cs.unm.edu/~brayer/vision/fourier.html>

9.5. Transformada discreta de Fourier (DFT) y su implementación mediante *Fast Fourier Transform* (FFT)



Accede al vídeo «La transformada discreta de Fourier y su implementación mediante la FFT» a través del aula virtual

Tal como hemos visto antes, la transformada de Fourier de $f(t)$ está pensada para señales continuas, es decir, para las que no han sido discretizadas o no han sufrido un proceso de conversión analógico-discreto.

Sin embargo, si se quiere aplicar la transformada de Fourier a señales discretizadas (digitales), hay que realizar algunas modificaciones en la definición y es por eso que haremos especial hincapié.

Además, la transformada discreta de Fourier (DFT, *Discrete Fourier Transform*) ha recibido gran atención por parte de matemáticos e ingenieros para realizar su implementación; la FFT (Fast Fourier Transform) es la mejor aproximación que existe actualmente, desde luego la más eficiente y que mejores resultados tiene. Cabe decir que la mayoría de lenguajes de programación poseen una implementación de la FFT.

En este apartado no vamos a abordar una comprensión matemática de la DFT, ya que conlleva un conocimiento de complejos y teoría matemática más avanzada que con la transformada continua de Fourier.

En este caso, para que tengas una visión global de la DFT únicamente diremos que:

- ▶ Ya no se habla de $f(t)$, sino de **secuencias**.
- ▶ Se representa mediante las variables x, y o z .

- ▶ Las variables n o m (si se quiere hacer distinción entre ambas) entre **corchetes**, no paréntesis.
- ▶ Por lo que dada una secuencia discreta $x[n]$ de longitud N, su **transformada discreta de Fourier** viene dada por la siguiente expresión:

$$X_K = \sum_{n=0}^{N-1} x[n]e^{-2\pi jk\frac{n}{N}}$$

Intuitivamente, se parece a la transformada de Fourier, pero posee algunas distinciones:

- ▶ El sumatorio está acotado entre 0 y N-1.
- ▶ La secuencia $x[n]$ puede no ser periódica, pero será de longitud N.
- ▶ No existe el concepto de frecuencia, sino que existe la variable k . Dicha variable provoca que la DFT sea también discreta y tenga los mismos puntos que la secuencia original N.

Aquí vemos dos ejemplos que facilitan la comprensión de la teoría. En primer lugar, se ve una esquematización de la DFT y su inversa. Hay que tener en cuenta que, en algunos libros, el factor $\frac{1}{N}$ se incluye en la propia DFT. Nosotros preferimos dejar ese factor para la **transformada inversa**, de ahí que no haya salido en la formulación anterior de la DFT.

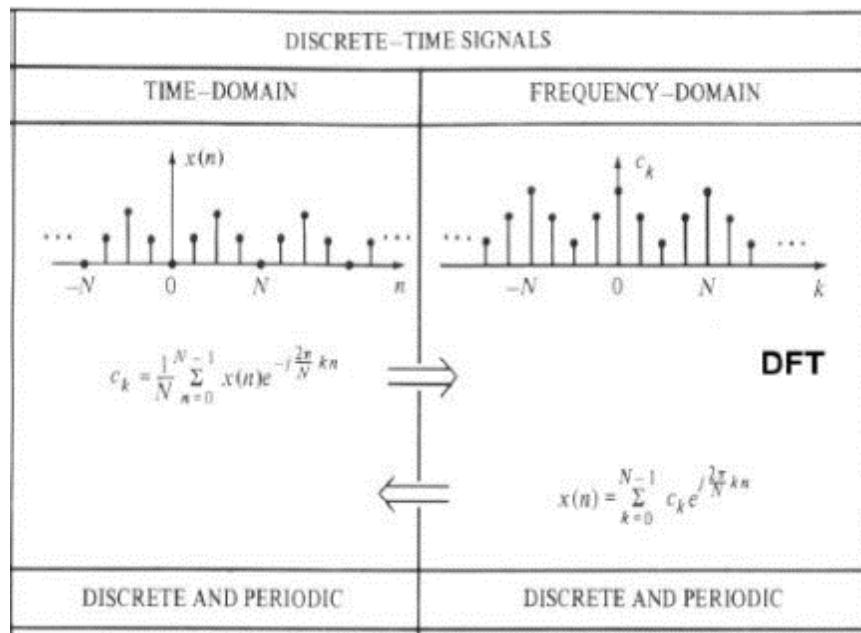


Figura 102. Comparación de la transformada discreta de Fourier entre la secuencia original y su propia transformada. En este caso se trata de una señal periódica.

Fuente: <http://kcchao.wikidot.com/dft>

En segundo lugar, un ejemplo de resolución de la DFT. Es evidente que el número de muestras de una secuencia, así como el número de muestras elegido para dar resolución a la DFT, afecta a la calidad de la transformada. En este caso, vemos un ejemplo con la función $\cos \frac{\pi n}{2}$, representando el módulo de la DFT. Se puede apreciar cómo la variación de los números de puntos empleados influye en la resolución de la transformada. De hecho, existe un número óptimo para evitar que aparezca ruido en la transformada.

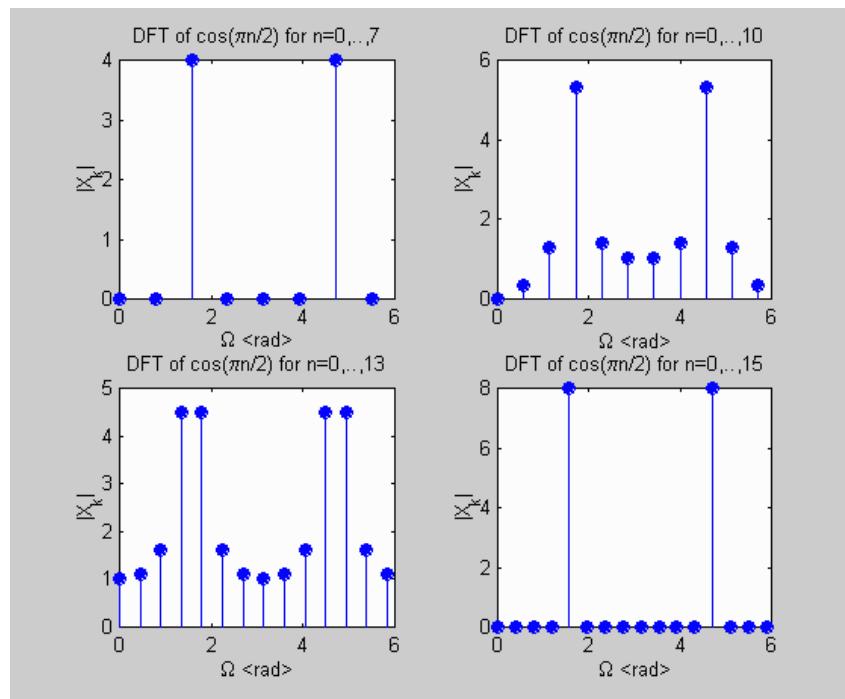


Figura 103. Comparación de la transformada discreta de Fourier entre la secuencia original y su propia transformada. En este caso se trata de una señal periódica.

Fuente: <http://www.ee.nmt.edu/~wedeward/EE342/SP99/example16.html>

Finalmente, no queríamos dejar de hablar sobre el esquema general de la transformada FFT. No deja de ser una aproximación que impone ciertas limitaciones, entre ellas que las duraciones de las secuencias deben de ser múltiplos de 2 para facilitar el cálculo. Sin embargo, la **FFT reduce la complejidad** de $O(n^2)$ a $O(n \log n)$, lo cual es un avance notable.

No existe una implementación única de la FFT. La original, creada por Cooley y Tukey de IBM en 1960 y revisada por C. S. Burrus de la Universidad Rice University, permitió procesar voz en tarjetas inteligentes para dispositivos con capacidades muy limitadas, pero actualmente existe una gran cantidad de implementaciones, todas ellas en función del problema que se pretenda resolver: más rapidez, más precisión, menos consumo, etc.

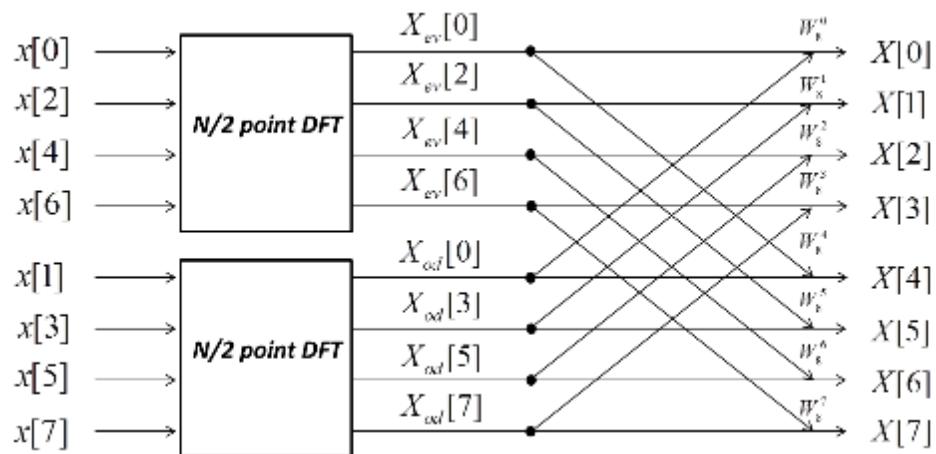


Figura 104. Funcionamiento de la FFT y ahorro de costes al calcular de forma paralela diferentes coeficientes, siempre y cuando las señales tengan una duración potencia de 2.

Fuente: <http://www.vocal.com/noise-reduction/fft-algorithms/>



Accede a los ejercicios de autoevaluación a través del aula virtual

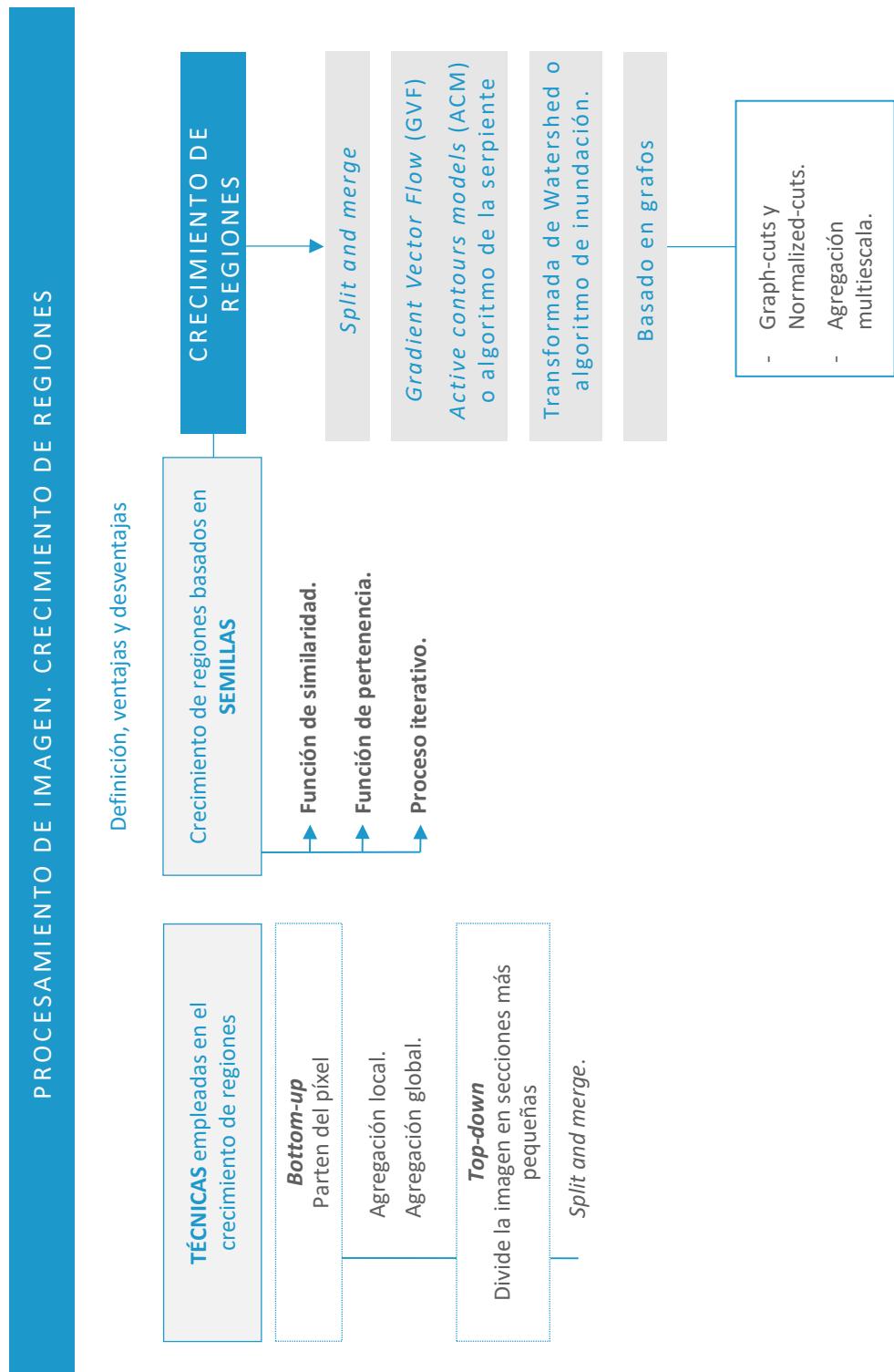


Accede al vídeo «Resumen del tema» a través del aula virtual

Percepción Computacional

Procesamiento de imagen. Crecimiento de regiones

Esquema. Tema 10



Ideas clave. Tema 10

10.1. ¿Cómo estudiar este tema?

Para estudiar este tema deberás leer con atención las ideas clave que se desarrollan a continuación.

10.2. Segmentación y crecimiento de regiones

Como se ha visto en temas anteriores, la segmentación consiste en dividir una determinada imagen en regiones/segmentos de propiedades similares. Esas propiedades pueden venir definidas bien por el color, la textura o incluso por el hecho de estar acotados por contornos muy bien definidos.

El crecimiento de regiones es un algoritmo generalmente **no supervisado** (existe posibilidad de aplicar supervisión, pero no es lo habitual) que, partiendo de unas regiones concretas en la imagen, normalmente elegidas por una persona, encuentra los límites de dicha región donde el punto se encuentra.

En otras palabras, dada una región inicial, el algoritmo explora en la vecindad de dicha región y considera que aquellos píxeles más parecidos son de la misma región.

A continuación, un ejemplo de crecimiento de regiones. Inicialmente se parte de un punto y se expande alrededor de dicho punto hasta que la región encontrada empieza a no ser uniforme y el algoritmo se detiene.



Figura 105. Ejemplo de crecimiento de regiones.

Por supuesto, este enfoque posee ventajas y desventajas. Dentro de las **desventajas** encontramos el hecho de que necesita una buena inicialización, es decir, requiere de intervención humana que indique dónde comenzar a crecer.

De hecho, este tipo de algoritmos se utiliza en biomedicina: imágenes biomédicas como ecografías, radiografías, etc., donde el médico indica qué región le interesa y el algoritmo encuentra las mejores fronteras de cada región.

Adicionalmente, como desventaja, nos encontramos con que este tipo de algoritmos son computacionalmente muy costosos por dos motivos:

Son algoritmos iterativos y no paralelizables. Es decir, hasta que no acaba una iteración no puede comenzar la siguiente.

Las **funciones** que exploran si un píxel debe pertenecer o no a una región son **complejas de evaluar y costosas de implementar**, lo que hace que estos algoritmos sean dependientes de estas funciones de evaluación.

Como **ventaja** se encuentra el hecho de que el algoritmo, hasta cierto punto, es autónomo para encontrar las fronteras de las regiones de interés. Esta **autonomía** está basada en funciones de comparación y en funciones de maximización y minimización. Si esas funciones no están correctamente parametrizadas o definidas, el algoritmo nunca será capaz de encontrar una segmentación adecuada.

10.3. Técnicas empleadas en el crecimiento de regiones



Accede al vídeo «Segmentación y crecimiento de regiones» a través del aula virtual

Existen dos familias de técnicas para el crecimiento de regiones:

- ▶ Aquellas técnicas que se encargan de resolver el problema **partiendo de píxeles singulares** y encontrando regiones más globales. Llamadas *bottom-up*: parten del mayor grado de detalle (el píxel) y agrupan en función de él. Podemos decir que existe una división también dependiendo de cómo se realice la agregación, lo que da lugar a dos tipos de sub-familias:
 - Técnicas basadas en **agregación local**, es decir, para agregar únicamente nos fijamos en los vecinos más cercanos del píxel en cuestión (o de la región que se está agregando).
 - Técnicas basadas en agregación **global**, donde se tiene una visión global de toda la imagen y dos píxeles solo se unen bajo una misma región siempre y cuando a nivel global tenga sentido dicha agregación.
- ▶ Aquellas que se encargan de dividir la imagen en **secciones más pequeñas** de forma iterativa, viendo hasta qué punto dos regiones pueden unirse. Este algoritmo es conocido como *split and merge* y a estas técnicas también se denominan *top-down*.

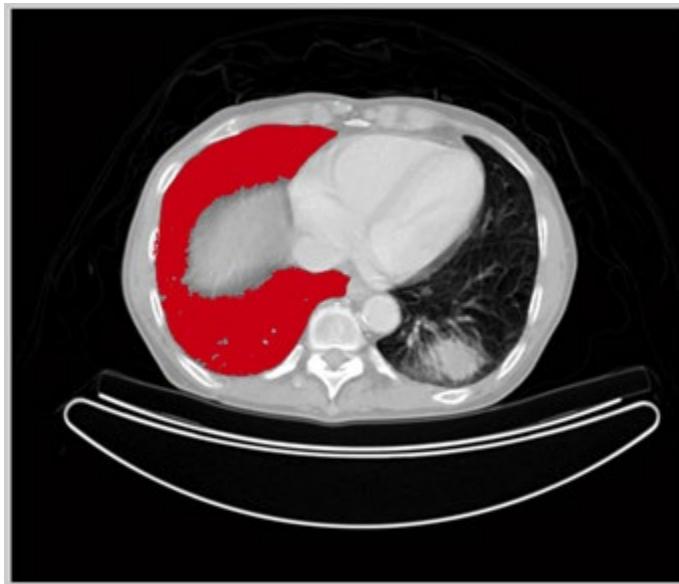


Figura 106. Resultado de un algoritmo de crecimiento de regiones (zona roja).

Fuente: <https://es.mathworks.com/matlabcentral/fileexchange/19084-region-growing>

Como hemos comentado, el uso de este tipo de algoritmos está muy extendido en biomedicina y segmentación de imágenes médicas, donde no existe la posibilidad de obtener colores.

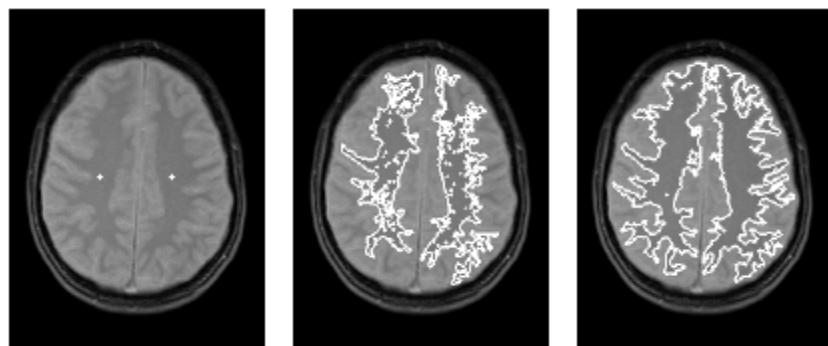


Figura 107. Evolución de un algoritmo *bottom-up* de crecimiento de regiones.

Fuente: <https://www.creatis.insa-lyon.fr/~grenier/?cat=8>

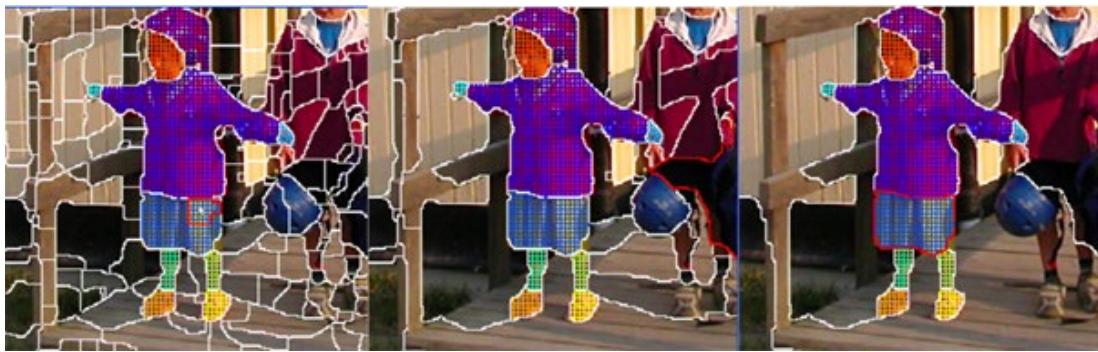


Figura 108. Ejemplo de algoritmo *split and merge*.

Fuente: <https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/shape/hat/>

En esta última imagen vemos que, partiendo de pequeñas divisiones de la imagen (*split*) se decide unir (*merge*) aquellas regiones con mayor parecido.

10.4. Crecimiento de regiones basado en semillas



Accede al vídeo «Crecimiento de regiones basado en semillas» a través del aula virtual

El crecimiento basado en semillas es el más intuitivo y para ello hacen falta los siguientes ingredientes, que serán comunes al resto de métodos de crecimiento de regiones.

Inicialmente asumiremos que una semilla vendrá dada por un píxel. Otras alternativas contemplan el uso de una forma bien definida por el usuario de forma interactiva sobre la imagen, bien ya predefinida: un círculo, cuadrado, etc.

Por lo tanto, para trabajar con este método basado en semillas necesitamos:

- ▶ Un conjunto de píxeles: p_1, p_2, \dots, p_n que servirán como semillas.
- ▶ La **posición** de esos píxeles vendrá dada por la propia posición que ocupe dentro de la imagen, y las **propiedades** de cada píxel, por la función $I(p_i)$ donde se elige el nombre I , ya que comúnmente será la intensidad de color de dicho píxel. Otros

valores de la función pueden ser la saturación u otras codificaciones de color que no sean las dadas por el criterio RGB.

- ▶ Una **función de similaridad** entre dos píxeles p_i y p_j . Esta similaridad vendrá dada por $S(p_i, p_j)$ y podrá aplicarse bien entre píxeles o bien entre regiones como posteriormente veremos.
- ▶ Una **función de pertenencia** a una región; dada $\pi_r(p_i)$ donde r hace referencia a la región en concreto y p_i a un píxel cualquiera. Esta función es 1 si $S(r, p_i) > T$, es decir, si la función de similaridad supera un cierto umbral. En cualquier otro caso, será 0. Dicho umbral se fijará de forma automática por el algoritmo o bien de forma manual, si así lo decide el usuario final.

Con lo cual, un algoritmo de crecimiento de regiones basado en semillas lo que intenta es encontrar el número mínimo de regiones que maximiza todas y cada una de las funciones de similaridad y de pertenencia dentro de cada región.

Este **proceso iterativo** posee las siguientes características:

- ▶ Al ser iterativo, es un proceso que posee gran coste computacional y cuya optimización debe realizarse, en muchos casos, de forma secuencial y no paralela.
- ▶ Solo considera regiones adyacentes, por lo que si dos píxeles no adyacentes, por función de similaridad S y de pertenencia pudieran pertenecer a una misma región, no serán unidos dentro de una misma región.
 - Esto conduce en la gran mayoría de casos a una **sobresegmentación**: hay un número elevado de segmentos/grupos dentro de la imagen, y por ende, otro postprocesado para eliminar dicha sobresegmentación.
- ▶ Cada píxel tiene que tener una región, así que en muchos casos, algunos píxeles son forzados a pertenecer a una determinada región simplemente por estar en su adyacencia.
- ▶ Si la región a detectar no es homogénea y posee geometrías complicadas, este método no proporcionará un resultado preciso debido a que el algoritmo tenderá a crecer en todas direcciones, respetando la vecindad del píxel.

- ▶ Sin embargo, por su sencillez e interactividad permite encontrar una primera aproximación de la segmentación muy certera en aquellas imágenes donde el contraste sea muy alto.
- ▶ Además, suele servir como origen para otros algoritmos de crecimientos de regiones más específicos y complejos.

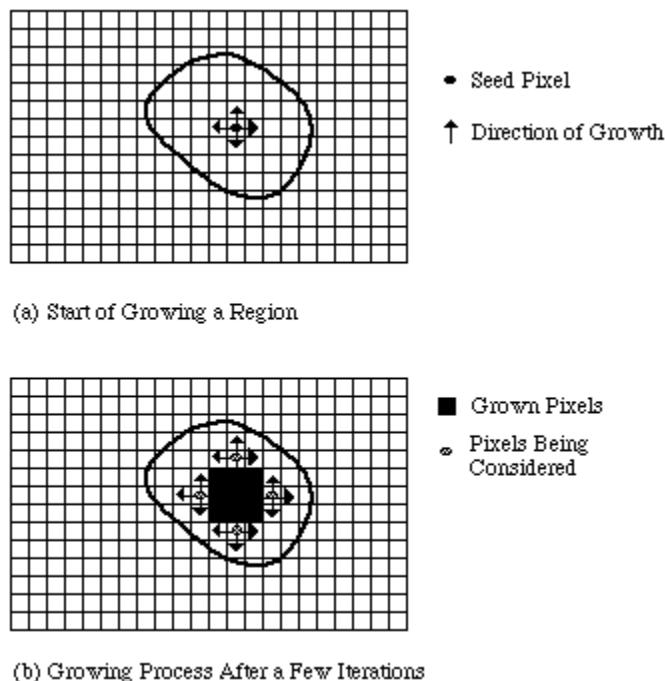


Figura 109. Comienzo del algoritmo de crecimiento de regiones basado en semillas. La dirección de crecimiento suele ser otro parámetro del algoritmo.

Fuente: https://users.cs.cf.ac.uk/Dave.Marshall/Vision_lecture/node35.html

10.5. Crecimiento de regiones basado en *Split and Merge*



Accede al vídeo «Crecimiento de regiones basado en *Split and Merge*» a través del aula virtual

Para explicar el algoritmo basado en *split and merge* voy a usar el siguiente esquema y cuatro subimágenes.

1	IMAGEN ORIGINAL i	Esta imagen es considerada en este método (y en este paso) como un segmento en sí mismo. La idea es ver si se pueden encontrar segmentos con funciones de similaridad y pertenencia más precisas.
2	DIVISIÓN EN CUATRO PARTES	Se divide la imagen en cuatro partes, evaluándose las funciones de similaridad y pertenencia en cada una. Se eligen cuatro en función del diseñador del algoritmo, aunque pueden escogerse el número de particiones que se desee. Para ser más óptimos computacionalmente, se suele dividir en cuatro y cada una a su vez en otras cuatro subpartes.

Figura 110. Pasos a seguir con el algoritmo *Split and Merge*.

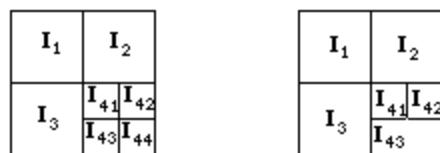
En este ejemplo, únicamente el subconjunto I_4 no verifica las condiciones de similaridad y pertenencia. Esto quiere decir que la entropía dentro de dicho segmento o la variabilidad de intensidades hace que esté por debajo de un determinado umbral y por lo tanto haya que realizar de nuevo el proceso de dividir la imagen en cuatro partes. En este caso, únicamente en el segmento I_4 .

De las cuatro subimágenes que se han creado, los segmentos I_{43} e I_{44} tienen propiedades similares y por lo tanto se fusionan (*merge*). ¿Por qué se dividieron si después hay que fusionarlos? Simplemente por el funcionamiento del algoritmo. Primero realiza divisiones de forma automática, en este caso en cuatro, y posteriormente evalúa si dos segmentos deberían de seguir juntos o no.



(a) Whole Image

(b) First Split



(c) Second Split

(d) Merge

Figura 111. Esquema de pasos principales en el algoritmo *split and merge*.Fuente: https://users.cs.cf.ac.uk/Dave.Marshall/Vision_lecture/node34.html

En la figura 7 vemos entonces que:

- ▶ La imagen original, todos los píxeles poseen la misma importancia (a).
- ▶ Se divide la imagen en partes iguales (b).
- ▶ Únicamente el segmento I_4 se subdivide en más subelementos debido a que dicho segmento es el menos uniforme (c).
- ▶ Los segmentos I_{43} e I_{44} se fusionan (*merge*) en uno debido a que tienen propiedades similares (d).

Este proceso de *split and merge* es «paralelizable», ya que el análisis se puede ejecutar de forma simultánea para los segmentos I_1 , I_2 , I_3 e I_4 .

Además, el coste computacional de este algoritmo va decreciendo a medida que va avanzando, ya que los subconjuntos/segmentos cada vez son más pequeños en tamaño y hay menor número. Teóricamente se puede asegurar que el coste computacional de este algoritmo está acotado.

Para verlo más fácilmente, pondré un ejemplo con una imagen de 16x16 píxeles. En ella puede apreciarse que ya de por sí existen regiones con valores muy similares (valores iguales a 7).

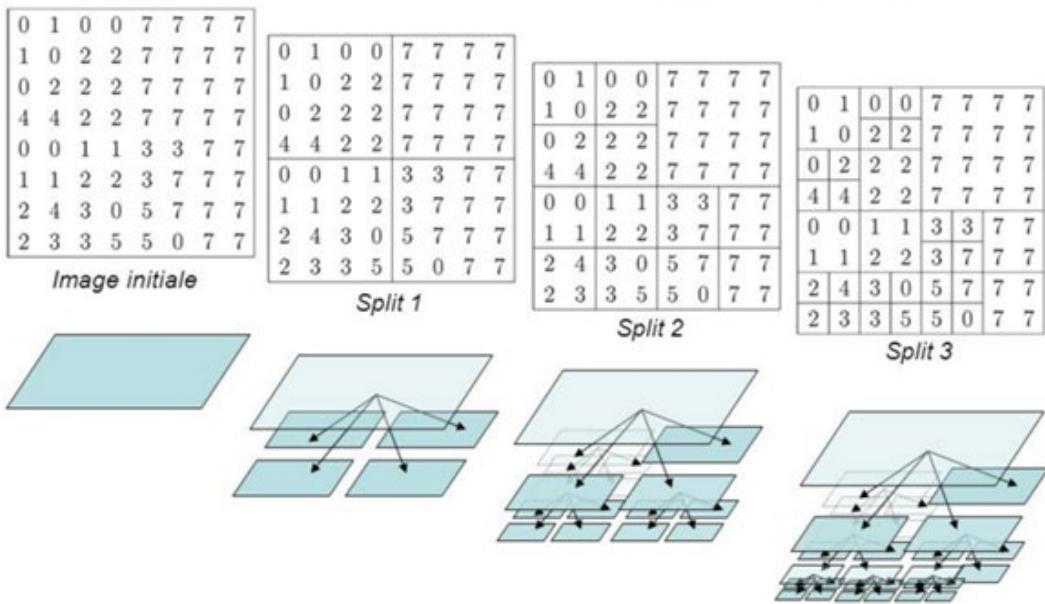


Figura 112. Máscaras correspondientes a los operadores de Prewitt para la identificación de bordes con orientación horizontal y vertical.

Fuente: <http://slideplayer.fr/slide/518235/>

- ▶ En la primera iteración (*Split 1*), se divide la imagen también en cuatro partes. De todas ellas, hay una que ya no se dividirá más puesto que todos los píxeles son iguales entre sí.
- ▶ Del resto de regiones, se dividen de nuevo por la mitad y en una tercera división se aprecia cómo algunos subconjuntos con valores similares no se han dividido de nuevo. Es aquí donde entra en juego el umbral que veíamos en la segmentación basada en crecimiento de regiones con semillas. Si ese umbral se disminuye, el algoritmo es más estricto y por lo tanto producirá sobresegmentación.
- ▶ En paralelo, se puede ver cómo se va construyendo un grafo jerárquico que explica cómo la imagen se ha ido dividiendo y cómo cada segmento contiene varios subsegmentos. Ese grafo se utiliza en otras técnicas (como graph-cuts) para poder segmentar de una manera más precisa e incluso para poder unir segmentos parecidos.

Por último, es importante indicar que aunque el algoritmo *split and merge* es uno de los más empleados, tiene una **desventaja** muy importante y es que, una vez que dos segmentos han sido separados, jerárquicamente están separados en ramas diferentes, el algoritmo no evaluará si pueden ser fusionados en un mismo segmento.

Este proceso debe hacerse de forma posterior en un postprocesado que ha de recorrer todos los segmentos y evaluar la función de similaridad y pertenencia. Sin embargo, aunque esto puede parecer costoso, como el número de segmentos es pequeño, puede realizarse sin problema.

10.6. Crecimiento de regiones basado en *Gradient Vector Flow* (GVF)

 Accede al vídeo «Crecimiento de regiones basado en *Gradient Vector Flow*» a través del aula virtual

Los métodos conocidos como *Gradient Vector Flow* (GVF en adelante) o *Active Contour Models* (ACM) suelen estar clasificados como métodos de segmentación basados en modelos u orientados a modelo y contorno. Sin embargo, pueden considerarse como métodos de crecimiento de regiones en un ámbito más amplio.

Esta familia de algoritmos se conoce popularmente como el **algoritmo de la serpiente**, puesto que simula el comportamiento de este animal a la hora de ir encontrando el contorno deseado.

Este tipo de métodos se basa en ir modificando el contorno de una determinada figura base (initialmente suele ser un círculo) e irla deformando de forma iterativa hasta que se adapta a la forma del objeto o región que se quiere segmentar.

En este sentido, son muy dependientes de la inicialización y de las funciones de coste o de optimización, es decir, el criterio que siguen estas curvas para readaptarse al contorno buscado.

A continuación se presentan tres ejemplos de cómo funciona visualmente este algoritmo.

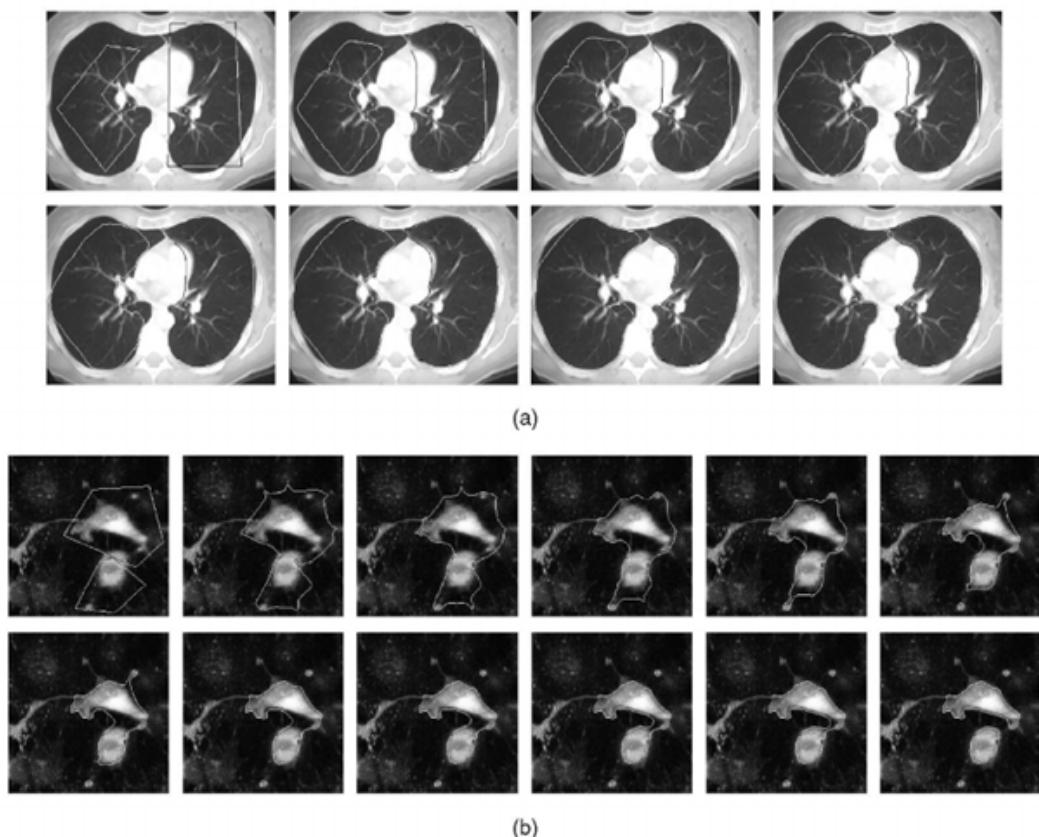


Figura 113. Funcionamiento del algoritmo GVF.

Fuente: Paragios y Ramesh, 2004.

En la figura 9, (a) y (b) son imágenes biomédicas donde se aprecia como el algoritmo comienza con un contorno inicial y dicho contorno se va adaptando a la forma a segmentar.

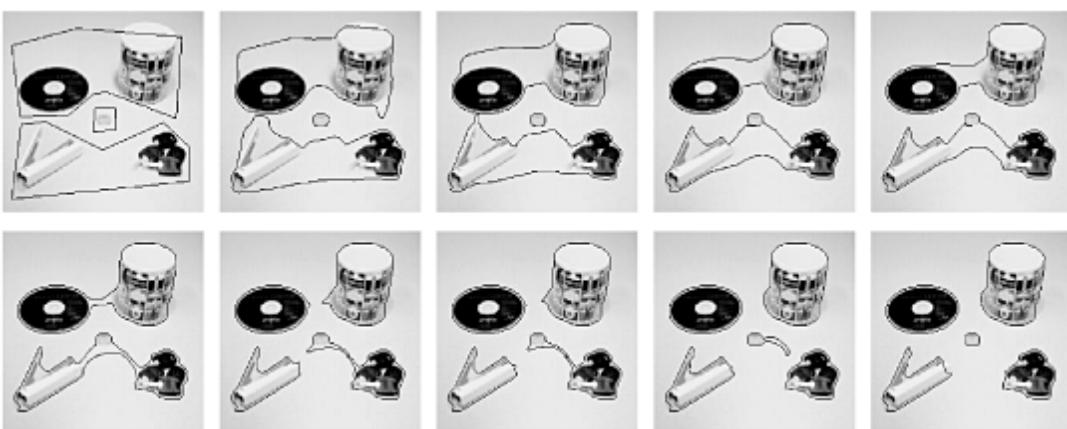


Figura 114. Funcionamiento del algoritmo GVF para la segmentación de varios objetos.

Fuente: Paragios y Ramesh, 2004.

En este último ejemplo, la segmentación es relativamente sencilla debido a que el fondo de la imagen está muy diferenciado.

No obstante, aunque GVF y ACM suelen ser sinónimos en la literatura, se puede decir que GVF es una extensión de ACM o algoritmos de serpiente. La **diferencia** más significativa radica en que GVF convergen en las concavidades del contorno y no necesitan una inicialización tan cercana al contorno como en el caso de los algoritmos ACM.

De hecho, veamos la diferencia de forma matemática. El algoritmo de serpiente original (v) es un contorno bidimensional dinámico definido de forma paramétrica como $v(s) = [x(s), y(s)]$ donde $s \in [0,1]$ y que minimiza la siguiente función de energía:

$$E = \int_0^1 E_{int}(v(s)) + E_{Imagen}(v(s)) + E_{con}(v(s)) ds$$

Donde:

- ▶ E_{int} denota la energía del contorno interior.
- ▶ E_{Imagen} representa la energía debida a la intensidad propia de la imagen.
- ▶ Y E_{con} es una función de contorno que modela la parte exterior del contorno.

La definición de estas energías en modo de función queda fuera del ámbito de este tema, pero en literatura pueden encontrarse numerosos ejemplos de funciones para estas energías.

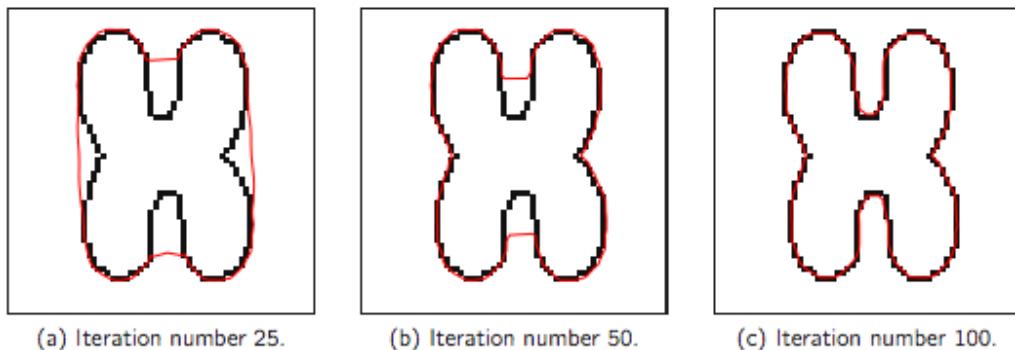


Figura 115. Esquema de funcionamiento del algoritmo GVF.

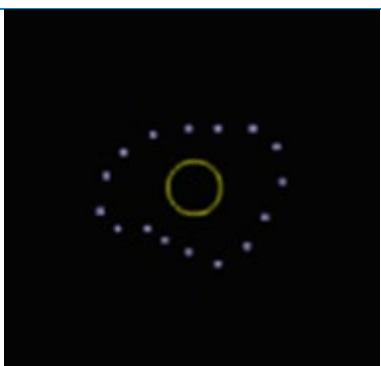
Fuente: Cartas-Ayala, s. f.

En el caso de los métodos GVF, y aquí radica la diferencia con los algoritmos de serpiente, la función de energía E_{con} utiliza una superficie paramétrica en vez de una curva, es decir, es función de $[u(x, y), v(x, y)]$ y no solo de $v(s)$.

Recomendamos leer lo siguiente:

- ▶ Kass, M., Witkin, A. y Terzopoulos, D. (1987). Snakes: Active contour models. *International Journal of Computer Vision*, 1(4), 321–331. Recuperado de <https://link.springer.com/article/10.1007/BF00133570>
- ▶ Leroy, B., Herlin, I. y Cohen, L. (1996). Multi-resolution algorithms for active contour models. En M. O. Berger et al. (Eds.). *ICAOS '96. Lecture Notes in Control and Information Sciences*, 219. Heidelberg: Springer Books. Recuperado de https://link.springer.com/chapter/10.1007/3-540-76076-8_117

Por último, presentamos una tabla comparativa entre GVF y ACM donde se presentan las ventajas y desventajas de ambos.

GVF vs ACM	Curva inicial	Curva final
El algoritmo de serpiente tradicional debe de inicializarse con un contorno muy similar al objeto a segmentar y aun así, no converge en las concavidades.		
El algoritmo GVF puede empezar muy alejado del contorno a segmentar, pero será capaz de converger en las concavidades.		
Incluso, si la inicialización de GVF atraviesa el contorno, será capaz de converger, a diferencia de ACM.		
GVF converge a contornos definidos de forma subjetiva, es decir, se aprecian a nivel visual, pero no están claramente definidos.		

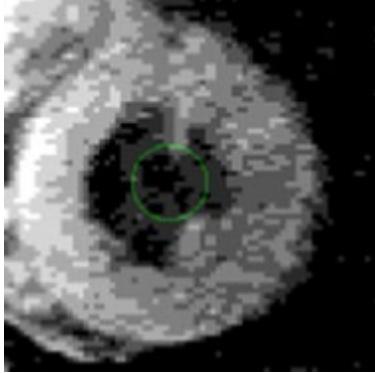
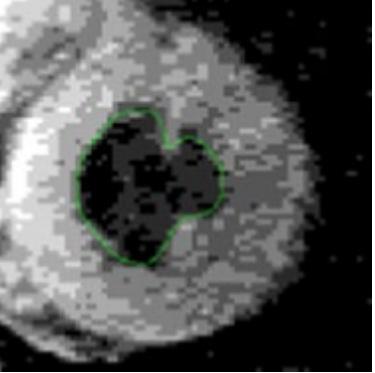
<p>GVF puede trabajar también con escalas de grises (aquí se aprecia cómo el algoritmo se adapta para encontrar las paredes del corazón en una imagen de resonancia magnética)</p>		
--	---	--

Tabla 1. Comparación entre GVF y ACM.

Fuente: Adaptado de <http://www.iacl.ece.jhu.edu/static/gvf/>

10.7. Crecimiento de regiones basado en Watershed



Accede al vídeo «Crecimiento de regiones basado en Watershed» a través del aula virtual

La transformada Watershed o segmentación basada en Watershed suele estar englobada dentro de la morfología matemática. Sin embargo, la motivación para ubicarla dentro de este tema se debe a que su funcionamiento está más cercano al crecimiento de regiones que a la propia morfología matemática.

La segmentación basada en Watershed se conoce popularmente como **algoritmos de inundación**, ya que utilizan el símil de la inundación de pantanos (o superficies) para explicar muy visualmente el algoritmo.

Con respecto a otras técnicas de segmentación, el objetivo de esta técnica es dividir en regiones la imagen de nivel de grises.

Generalmente una de las regiones se corresponde con el fondo de la imagen, que puede contener otros objetos fuera de interés, y el resto con los objetos o regiones que se pretende extraer. El objetivo último de esta técnica es **determinar los contornos** que definen dichos objetos.

El punto de partida es considerar que los contornos de una imagen se corresponden con las líneas donde el nivel de gris varía más rápidamente que en un determinado entorno vecino. En este punto, hay que hacer uso del **gradiente de la imagen**, que proporciona la posición de los píxeles donde la variación de intensidad es más pronunciada, muy similar al concepto de derivada matemática.

El uso de dicho gradiente nos permite asumir que los contornos de la imagen original se corresponden con las líneas de cresta de la imagen gradiente.

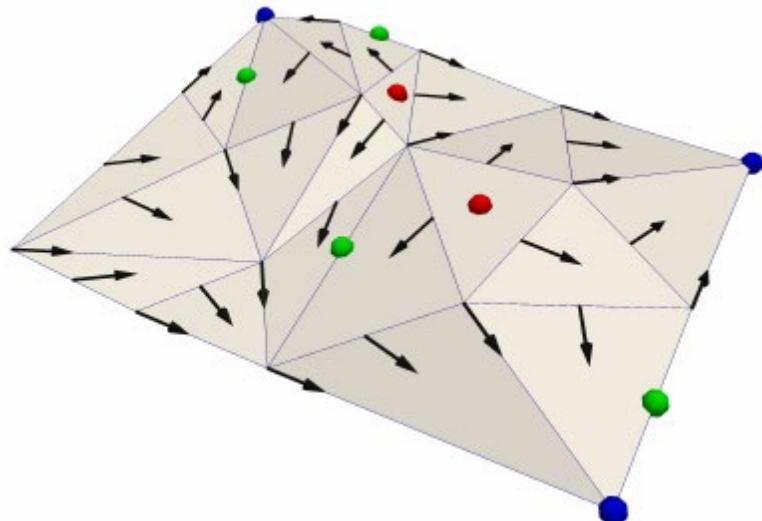


Figura 116. Ejemplo visual de gradiente de una imagen.

Fuente: https://www.researchgate.net/figure/Example-of-a-discrete-gradient-vector-field-on-a-triangulated-terrain-Paired-simplices_fig1_261674491.

El concepto de Watershed se basa en visualizar una imagen en tres dimensiones (3D): dos coordenadas espaciales frente a niveles de gris. En esta interpretación topográfica, de ahí que estos algoritmos estén dentro del campo de la morfología matemática (nacieron dentro del mundo de la topografía), se considerarán tres tipos de puntos:

1. Puntos que corresponden a mínimos locales.
2. Puntos en los que, si se coloca una gota de agua, esta cae con certeza en un único mínimo: los puntos que verifican esta condición se denominan **puntos Watershed o catchment basis**.
3. Puntos en los que el agua caería con igual probabilidad en más de uno de estos mínimos: los puntos que verifican esta condición forman las **líneas Watershed o líneas de cresta**.

El objetivo principal de los algoritmos de segmentación es alcanzar estas líneas divisorias o de cresta.

La idea básica es la siguiente: supongamos que se hace un pequeño agujero en cada mínimo local y que todo el relieve topográfico es inundado desde abajo. El agua va subiendo e inundando las cuencas. Cuando el agua de dos cuencas está a punto de juntarse, se construye un dique (*dam*) para evitar la fusión. La inundación continúa y llega a un punto en que solo se ve la parte de arriba de los diques por encima de la línea de agua. Las líneas de Watershed forman un camino conexo, dando por lo tanto bordes continuos entre las regiones. Este ejemplo puede verse en la imagen a continuación.

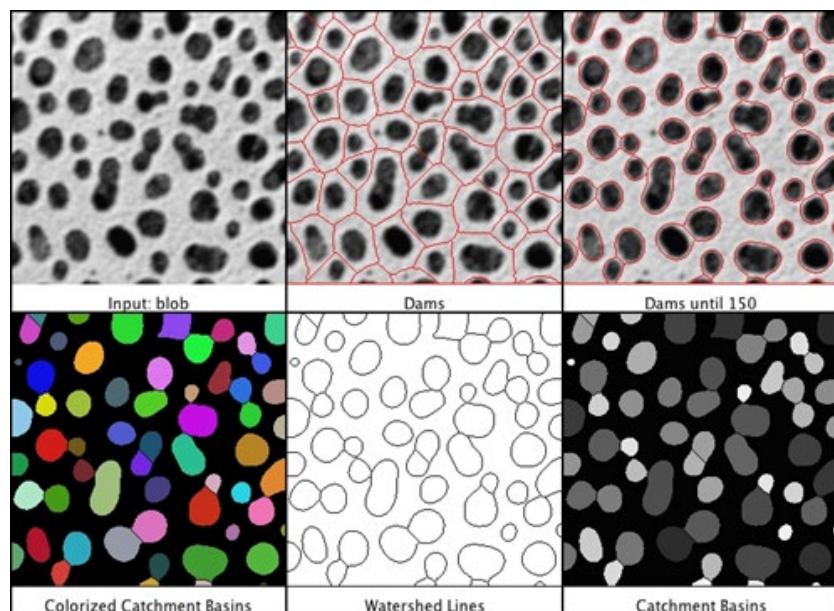


Figura 117. Pasos más comunes en la ejecución de la segmentación basada en Watershed.

Fuente: <http://bigwww.epfl.ch/sage/soft/watershed/>.

La segmentación basada en Watershed produce normalmente **sobresegmentación** y puede deberse a diferentes aspectos como la diferencia de texturas o patrones, ruido o incluso cambios en las tonalidades del color. Para ello, lo más habitual suele ser:

- ▶ Eliminar los contornos irrelevantes una vez realizado el Watershed.
- ▶ Modificar la imagen gradiente de tal forma que las regiones de depresión o valles se correspondan únicamente con los objetos deseados.
- ▶ Imponer marcadores como mínimos de la imagen gradiente.
- ▶ Suprimir todos los demás mínimos del gradiente (los irrelevantes) rellenando los correspondientes valles.
- ▶ Preservar las líneas de cresta más importantes de la imagen gradiente, localizadas entre los marcadores.

10.8. Crecimiento de regiones basado en grafos



Accede al vídeo «Crecimiento de regiones basado en grafos» a través del aula virtual

Por último, presentamos un apartado dedicada a los algoritmos de crecimientos de regiones basadas en **grafos**. Estos algoritmos entienden la imagen como un grafo con cierta conectividad entre los píxeles y con unas **propiedades de conectividad**: funciones de similaridad, peso, distancia, etc.

En función de esas propiedades, el algoritmo itera el grafo eliminando las conexiones más débiles (menos parecidas) y potenciando las más fuertes.

Dentro de esta familia se encuentran los siguientes algoritmos principales:

Graph-cuts y **Normalized-cuts**: dado un grafo, eliminan aquellas conexiones, también conocidas como *edges*, entre los píxeles (llamados en la literatura como *nodes*), que son más débiles.

Agregación multiescala: partiendo de un grafo, crean diferentes niveles de segmentación donde el algoritmo va encontrando qué subsegmentos (subgrafos) pertenecen a un mismo segmento. Muy similar al concepto de *split and merge*, pero utilizando métodos como graph-cuts para realizar el *split*.

A continuación, se presenta un esquema resumido de ambos algoritmos.

Graph-cuts

Para Graph-cuts se presenta el esquema típico de funcionamiento:

- ▶ En primer lugar, se obtiene la imagen original.
- ▶ En segundo lugar, se crea el grafo obteniendo para cada conexión entre píxeles (nodos dentro del grafo) una función de similaridad o parecido. La vecindad puede ser sencilla, como en este caso, o compleja si consideramos conectividades con nodos/píxeles más lejanos.
- ▶ Se buscan aquellas conectividades más débiles o que poseen una similaridad menor. El corte del grafo por esos puntos más débiles es lo que otorga el nombre de Graph-cuts al algoritmo.
- ▶ Por último, se convierte el grafo en imagen de nuevo, esta vez con una frontera que refleja la segmentación realizada.

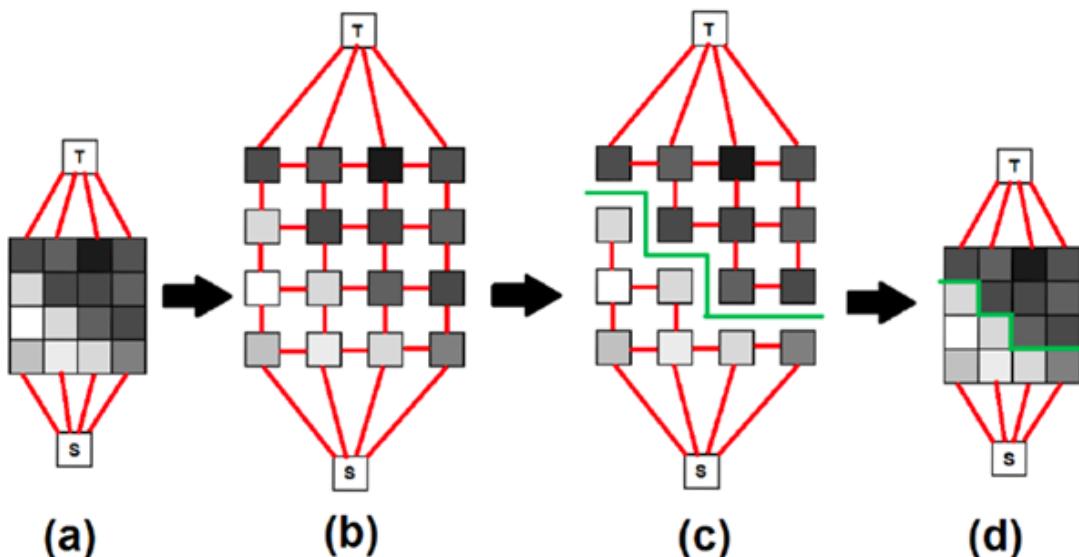


Figura 118. Funcionamiento simplificado de graph-cuts.

Agregación multiescala

El funcionamiento es el siguiente:

- ▶ Se parte de la imagen original, donde cada píxel representará un nodo.
- ▶ Se realiza, en función de un criterio de similaridad, una agrupación basada en los nodos más próximos. Esta agregación de por sí ya es una segmentación, aunque proporciona resultados muy pocos precisos.
- ▶ Una vez agrupados los nodos en segmentos, se obtiene un **representante** (normalmente el valor medio) de dicho segmento, lo que conduce a una nueva escala (nueva capa) con menos nodos, pero cada nodo representando a un subconjunto de nodos iniciales.
- ▶ Se vuelve a repetir la agregación, esta vez para los nodos representantes y se itera este proceso hasta que, o bien se alcanza un número determinado de segmentos, o bien se verifica una condición de calidad, es decir, que los segmentos creados, independientemente del número que sean, tienen unas propiedades bien definidas.
- ▶ Finalmente, la imagen se segmenta deshaciendo las agrupaciones realizadas por cada uno de los niveles.

De esta manera, si existen regiones parecidas entre ellas, pero separadas de forma espacial en la imagen, el algoritmo encontrará la manera de asociarlas a un mismo segmento.

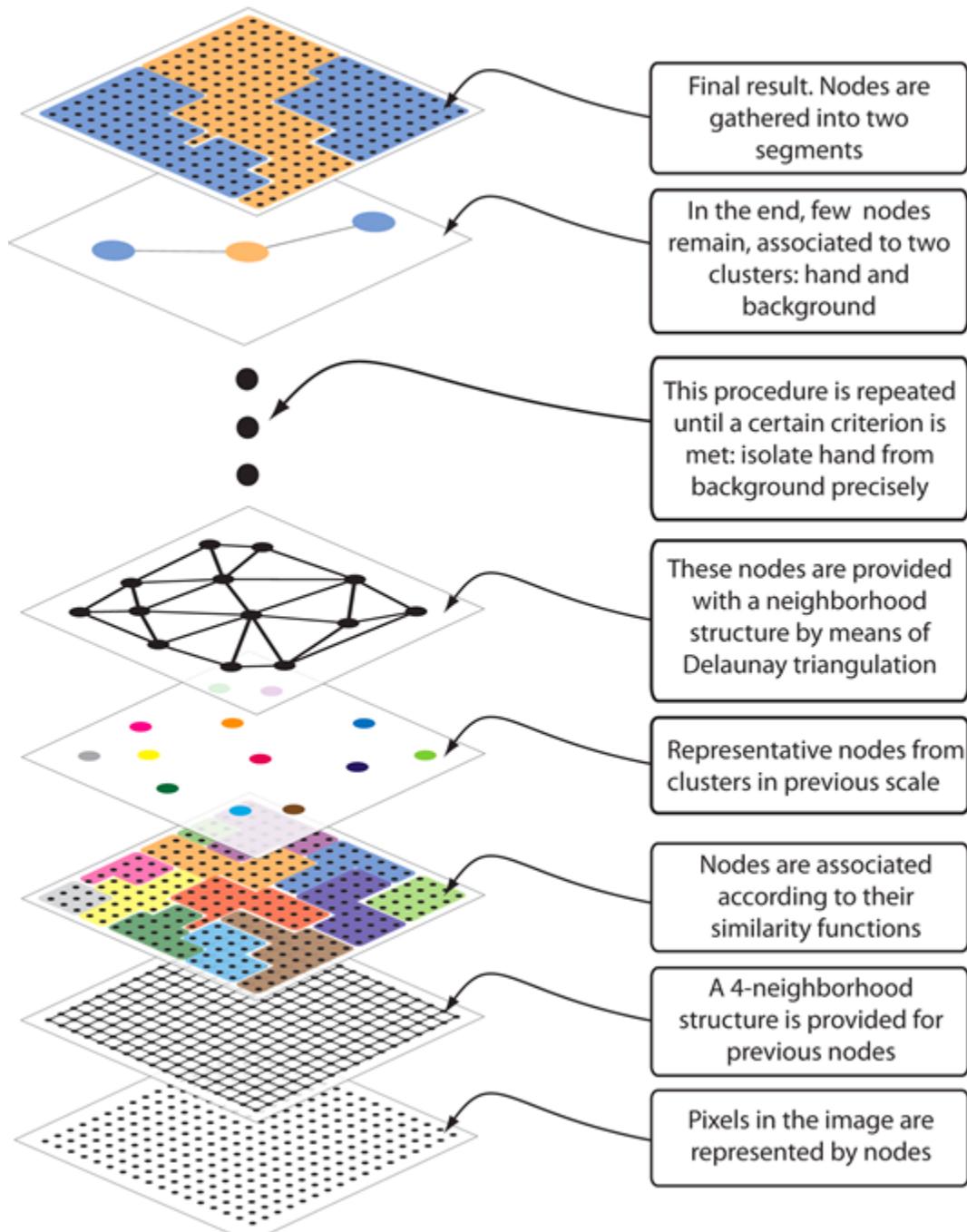


Figura 119 Procedimiento de agregación multiescala comenzando por la imagen original (parte inferior) y terminando con la imagen segmentada (parte superior).

10.9. Referencias bibliográficas

Cartas-Ayala, A. (Sin fecha). Gradient Vector Flow Snakes. Manuscrito presentado para su publicación. Recuperado de

http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/AV1011/cartas.pdf

González, R. C. y Woods, R. E. (2008). *Digital image processing*. New Jersey: Pearson Education.

Paragios, N. y Ramesh, V. (2004). Gradient Vector Flow Fast Geometric Active Contours. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(3), 402-407. Recuperado de

https://www.researchgate.net/publication/8337757_Gradient_Vector_Flow_Fast_Geometric_Active_Contours



Accede a los ejercicios de autoevaluación a través del aula virtual

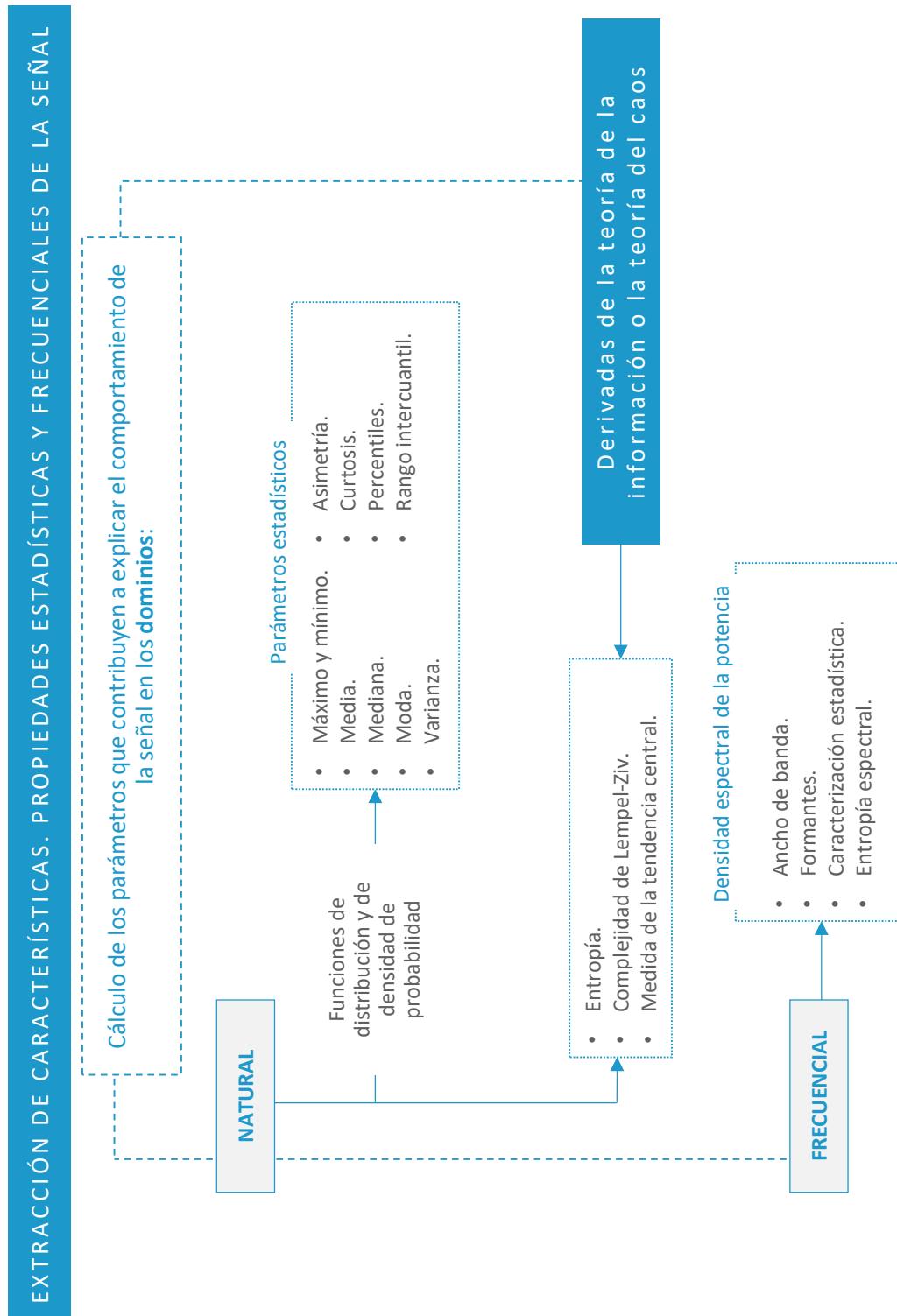


Accede al vídeo «Resumen del tema» a través del aula virtual

Percepción Computacional

Extracción de características. Propiedades estadísticas y frecuenciales de la señal

Esquema. Tema 11



Ideas clave. Tema 11

11.1. ¿Cómo estudiar este tema?



Accede al vídeo «Introducción. Extracción de características. Propiedades estadísticas y frecuenciales» a través del aula virtual

Para estudiar este tema deberás leer con atención las ideas clave que se desarrollan a continuación.

Este tema tiene como objetivo proporcionar herramientas con las que caracterizar el comportamiento de nuestras señales. Se van a explicar las principales variables que describen las propiedades de la señal y, como resultado, tendremos un conjunto de atributos de diversa naturaleza que cuantifican diferentes cualidades de la fuente de información que manejamos.

Estos atributos contribuyen a sintetizar la información contenida en la señal, de forma que pueda ser manejada más eficientemente en una etapa final basada en técnicas de reconocimiento de patrones.

En temas anteriores, se ha descrito cómo las señales, generalmente series temporales de naturaleza unidimensional, e imágenes, definidas en dos dimensiones, que manejamos pueden ser modeladas matemáticamente como procesos estocásticos. Asumiremos en este tema que nuestras señales o procesos estocásticos son estacionarios en sentido amplio. Por tanto, las propiedades estadísticas de la señal no varían en función de las coordenadas del proceso.

De acuerdo a este modelo, las muestras observadas en una señal $f(\cdot)$ pueden ser consideradas como **realizaciones de una variable aleatoria**. Por ejemplo, si nuestra señal se trata de un electrocardiograma, cuya amplitud refleja el potencial del campo

eléctrico generado por la actividad muscular del corazón, las muestras de la señal pueden tomarse como observaciones de una variable aleatoria que toma el valor de dicho potencial. Por tanto, puede caracterizarse el comportamiento estadístico de esta variable.

Las funciones de distribución y de densidad de probabilidad definen por completo el comportamiento estadístico de una variable.

Sin embargo, en algunas aplicaciones prácticas es deseable disponer de elementos cuantitativos más fácilmente manejables que una función para describir sus propiedades. Por ejemplo: el valor medio de la variable sería uno de estos atributos. Estos elementos permiten obtener una caracterización estadística parcial de la variable frente a la descripción completa que nos ofrecen las funciones mencionadas.

Además de la caracterización puramente estadística de la variable aleatoria que representa el valor de la señal, pueden calcularse otros parámetros que contribuyen a explicar el comportamiento de la señal en el dominio en el que queda definida (por ejemplo: en el tiempo, en el caso de series temporales o en las dos coordenadas espaciales para imágenes). En este sentido, es habitual emplear **medidas cuantitativas derivadas de la teoría de la información o la teoría del caos**.

Por último, cabe destacar que hemos mencionado la extracción de atributos de la señal de partida mediante su análisis en el dominio natural. Sin embargo, el análisis en el dominio frecuencial proporciona atributos de gran valor para la definición de su comportamiento. Estos atributos se derivan de la densidad espectral de potencia de la señal de partida, que se obtiene como la transformada de Fourier de su función de autocorrelación.

La siguiente figura muestra un esquema de los principales conceptos expuestos en este tema.

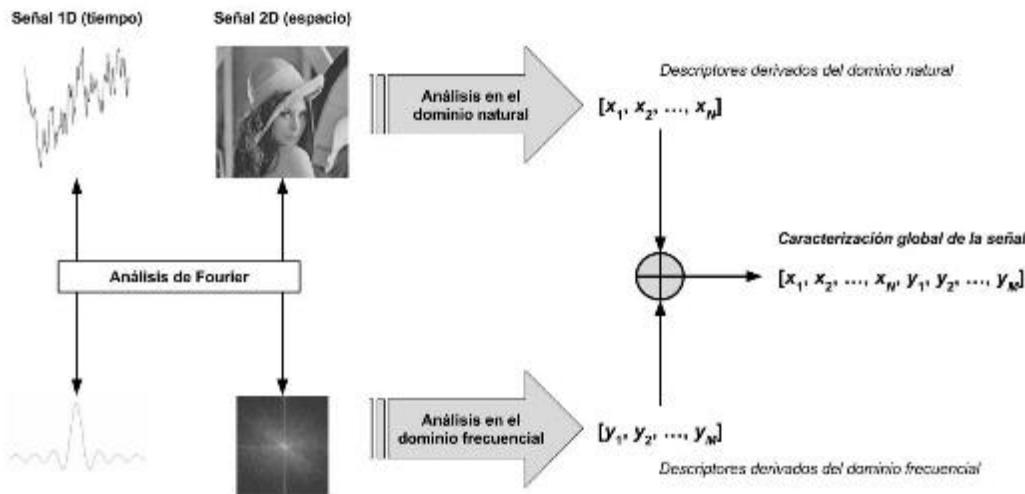


Figura 120. Proceso de caracterización de señales en su dominio natural y frecuencial para la generación de un vector global de características.

11.2. Caracterización de señales en el dominio natural



Accede al vídeo «Caracterización de señales en el dominio natural» a través del aula virtual

En primer lugar, se indicarán los **parámetros más comunes** a partir del análisis de la señal en el dominio en el que se define. Entre estos, se tendrán parámetros estadísticos y otros derivados de la teoría de la información.

Caracterización estadística parcial

Máximo y mínimo

La diferencia entre ambos valores define el **rango dinámico** de la señal.

Media

Asumamos que nuestra señal $f(\cdot)$ está compuesta por M muestras. En el caso de una serie unidimensional, esta tendría longitud M . Para una imagen, estaría formada por un total de M píxeles. Utilizamos f_j para denotar los diferentes valores que toma nuestra señal, asumiendo que puede ser uno de N diferentes, es decir, $j = 1, \dots, N$.

Finalmente, supongamos que k_j muestras de las M observadas tienen un valor f_j . Puede deducirse de forma directa que el número total de muestras es la suma de los cardinales asociados a cada uno de los posibles valores f_j :

$$\sum_j k_j = M$$

A partir de este escenario, podemos calcular el **valor medio de la muestra** de la siguiente forma:

$$\bar{f} = \sum_{j=1}^N \frac{k_j}{M} f_j$$

Es decir, el valor medio se obtiene como la media ponderada de todos los valores f_j posibles. El peso en la ponderación viene dado por la relación $\frac{k_j}{M}$, que aproxima la probabilidad de observar un valor f_j en la señal. Esta aproximación se obtiene, por tanto, como la fracción entre el número de observaciones del valor f_j y el número total de muestras en nuestra señal.

Cabe destacar que en un ejercicio aparentemente tan simple como el cálculo de la media se llevan cabo diferentes pasos:

- ▶ En primer lugar, se discretiza la variable continua que refleja la amplitud de la señal. Teóricamente, nuestra señal $f(\cdot)$ toma valores en un rango continuo. Sin

embargo, se ha asumido para la estimación de la media que $f(\cdot)$ es discreta y toma únicamente uno de los N valores posibles f_j ($j = 1, \dots, N$).

- ▶ Posteriormente, se ha aproximado la función de densidad de probabilidad de esta variable discreta. Para ello, se ha aproximado la probabilidad asociada a cada posible valor de la variable como la frecuencia de observación de dicho valor.

Este ejercicio de discretización y estimación de la función de densidad de probabilidad de la variable discreta se repetirá para el cálculo de otros parámetros estadísticos que permiten una caracterización parcial.

El valor medio obtenido para la señal $f(\cdot)$ es una estimación de su media real η_F , que viene definida de la siguiente forma:

$$\eta_F = E\{F\} = \int_{-\infty}^{\infty} f p_F(f) df$$

Donde:

- ▶ F es la variable aleatoria que representa el valor de la señal.
- ▶ $p_F(f)$ es la función de densidad de probabilidad de la variable.
- ▶ F y $E\{\cdot\}$ denota el operador esperanza matemática.

Mediana

El valor mediano o mediana de una variable F es el valor f_{med} de esta para el que la probabilidad de observar una muestra de F en el intervalo $(-\infty, f_{med}]$ es 0.5. Es decir, el área bajo la función de densidad de probabilidad a ambos lados de este valor es la misma y es 0.5. Por tanto, puede expresarse de la siguiente forma:

$$P(F \leq f_{med}) = P(F > f_{med}) = \int_{-\infty}^{f_{med}} p_F(f) df = \int_{f_{med}}^{\infty} p_F(f) df = 0,5$$

La definición se aplica de la misma forma a variables discretas, ya que su función de densidad de probabilidad puede escribirse en términos de funciones $\delta(x)$. Sin embargo, la estimación de la mediana suele obtenerse como el valor de la observación ubicada en la posición central tras la ordenación de las muestras disponibles para la variable. Si el número de muestras disponibles es par, se toma como estimación de la mediana el valor de la **media aritmética** entre las dos observaciones centrales.

Moda

La moda de la variable F (f_{mod}) viene dada por el valor de la misma que se toma con mayor probabilidad. Es decir, vendría dada por el valor en el cual la función de densidad de probabilidad $p_F()$ alcanza su máximo:

$$f_{mod} = \operatorname{argmax}\{p_F(f)\}$$

Varianza

La varianza de una variable aleatoria es el promedio de las desviaciones, en términos cuadráticos, de los valores que toma la variable respecto a su media. A partir de un conjunto de observaciones, la varianza puede estimarse de la siguiente forma:

$$\bar{\sigma}^2 = \sum_{j=1}^N \frac{k_j}{M} (f_j - \bar{f})^2$$

Como puede apreciarse, para la **estimación de la varianza** se lleva a cabo la misma aproximación empleada para el cálculo de la media basada en la discretización de la variable y la estimación de su función de densidad de probabilidad.

De forma precisa y de acuerdo a su definición, la varianza viene dada por la esperanza matemática de las desviaciones cuadráticas de F respecto a su media:

$$\sigma_F^2 = E\{(F - \eta_F)^2\}$$

Asimetría

La asimetría (en inglés, *skewness*) de una variable aleatoria cuantifica el grado de asimetría de su función de densidad de probabilidad respecto al eje vertical que pasa por su media.

Este **estadístico**, de forma coloquial, refleja hacia qué lado de la media es más pronunciada la cola de la distribución.

- ▶ Un valor positivo de la asimetría refleja que la cola de la función de densidad de probabilidad a la derecha de la media es más larga que a la izquierda.
- ▶ En caso contrario, el valor de la asimetría será menor que 0.
- ▶ En el caso de una variable caracterizada por una **función simétrica**, como es el caso de las variables gaussianas, su asimetría es 0.

La figura 2 muestra gráficamente la interpretación de la asimetría de una variable aleatoria.

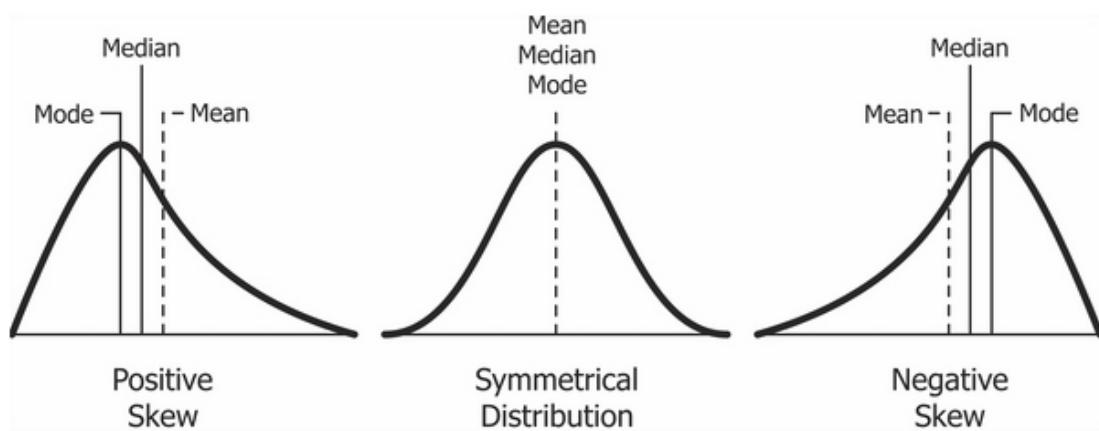


Figura 121. Interpretación de la asimetría estadística de una variable.

Fuente: <https://medium.com/@cybersiftIO/anomaly-detection-vs-ransomware-b83510a3a860>

La asimetría γ viene dada por la siguiente expresión:

$$\gamma = E \left[\left(\frac{f - \eta_F}{\sigma} \right)^3 \right]$$

Donde:

- σ es la desviación típica de la variable F , que viene dada por la raíz cuadrada positiva de la varianza.

Curtosis

La curtosis de una variable caracteriza las **colas** de su función de densidad de probabilidad. Como referencia, se suele emplear la curtosis de la distribución normal, que es 3. Un valor inferior (curtosis negativa) refleja que los *outliers* que genera la distribución son menos extremos que los derivados de una distribución normal. La interpretación contraria es aplicable a un valor positivo de curtosis.

La siguiente imagen refleja la propiedad evaluada por este parámetro.

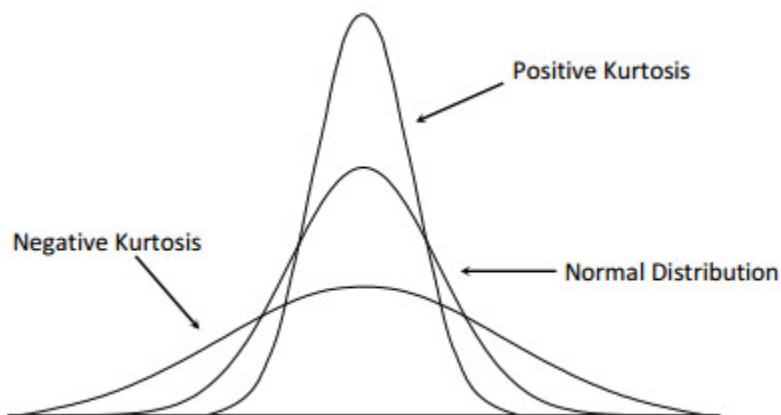


Figura 122. Interpretación de la curtosis de una variable.

Fuente: <https://medium.com/@cybersiftIO/anomaly-detection-vs-ransomware-b83510a3a860>

La expresión para el cálculo de la curtosis (δ) es la siguiente:

$$\delta = E \left[\left(\frac{f - \eta_F}{\sigma} \right)^4 \right]$$

Percentiles

Diferentes valores de percentil contribuyen a dar una idea de la distribución estadística de la variable a caracterizar. El percentil f_u de la variable F representa el umbral para el cual la probabilidad acumulada es u . Matemáticamente, viene dado por la siguiente expresión:

$$P(F \leq f_u) = u$$

Generalmente, se calcula el valor del percentil para los umbrales 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8 y 0.9. A fin de reducir el número de parámetros, en los umbrales 0.25, 0.5 y 0.75.

Rango intercuartil

Los percentiles para los umbrales de probabilidad 0.25 y 0.75 se denominan primer cuartil y tercer cuartil, respectivamente. El rango intercuartil (IQ) es la diferencia entre ambos percentiles:

$$IQ = f_{0.75} - f_{0.25}$$

Otros parámetros

Se indican a continuación otros parámetros que sintetizan información de interés sobre el comportamiento de la señal en su dominio original.

Entropía

Se incluye en este apartado, a pesar de que se corresponde con un parámetro estadístico más derivado de la función de densidad de probabilidad de la variable aleatoria a caracterizar. La entropía ha sido descrita en detalle en temas anteriores

de la asignatura. Como se indicó, cuantifica el promedio de la cantidad de información generada por una variable aleatoria.

De forma intuitiva, una variable tendrá mayor entropía si su función de densidad de probabilidad tiende a ser más uniforme en el rango de definición de esta. Es decir, mayor entropía está asociada a mayor incertidumbre sobre el valor que *a priori* puede tomar la variable.

Por tanto, la **distribución uniforme** será la que tenga una mayor entropía asociada, pues ninguno de los valores que la variable puede tomar posee, *a priori*, mayor probabilidad de ser observado que otros.

Complejidad de Lempel-Ziv

La complejidad de Lempel-Ziv es una medida no paramétrica de complejidad de una fuente de información. Generalmente, su definición aplica a señales unidimensionales en forma de serie, sin embargo, es extensible a imágenes. Su cálculo se basa en la obtención del número de subsecuencias diferentes que hay contenidas en la señal original. Para cada una de ellas, se calcula la **frecuencia de repetición**.

Una mayor tasa de repetición reflejará una menor complejidad de la fuente de datos, ya que el contenido de la señal indica que su origen es menos caótico y, por tanto, su comportamiento es más predecible y hay **menor incertidumbre** asociada.

Medida de la tendencia central

Las ecuaciones derivadas de la teoría del caos pueden emplearse para generar distintos tipos de gráficos. Entre estos, tenemos el que se conoce como el **gráfico de diferencias de primer orden**: considerando una serie temporal $f[n]$, la representación de $f[n + 2] - f[n + 1]$ frente a $f[n + 1] - f[n]$.

Centrado en el origen, ofrece una representación gráfica del **grado de variabilidad** en la serie temporal $f[n]$. Con este enfoque, en lugar de definir categóricamente una serie temporal como caótica o no caótica, se cuantifica el grado de variabilidad o caos.

A partir de la representación de las diferencias de primer orden se obtiene la métrica de variabilidad conocida como medida de la tendencia central.

Un mayor valor de esta métrica refleja mayor variabilidad y, por tanto, un mayor grado de caos y de dificultad para predecir el comportamiento de la fuente de datos.

11.3. Características derivadas del análisis en frecuencia



Accede al vídeo «Caracterización de señales en el dominio frecuencial» a través del aula virtual

La representación de una señal en sus componentes frecuenciales permite apreciar atributos de esta que no son observables en su dominio natural. Dicha representación viene descrita por la teoría de Fourier, que define las operaciones de transformación directa e inversa a partir de una base ortogonal de funciones de la frecuencia.

La **densidad espectral de potencia** $S(\omega)$, abreviada como DEP, de una señal refleja la potencia contenida por unidad de frecuencia. Así, aquellas regiones de ω en las que esta función tome una amplitud mayor reflejarán las componentes frecuenciales más significativas de la señal. Por ejemplo: en el caso de una señal de voz, su densidad espectral de potencia se concentra en el rango entre 300 Hz y 4 KHz, aproximadamente.

A continuación, se indican diferentes **descriptores** de la función de densidad espectral de potencia que caracterizan su perfil y comportamiento.

Ancho de banda

El ancho de banda de una señal se define como el rango de frecuencias en el que su **densidad espectral de potencia es mayor que cero**. Se calcula como la diferencia entre las componentes frecuenciales máxima y mínima de la señal.

Generalmente no se suele emplear la definición anterior, que refleja el ancho de banda absoluto de la señal. Así, es habitual considerar el ancho de banda de acuerdo a criterios como la diferencia entre componentes frecuenciales para las que $S(\omega)$ refleja una atenuación de 3 dB respecto a su máximo o el rango de frecuencias donde se concentra el 90 % de la potencia de la señal.

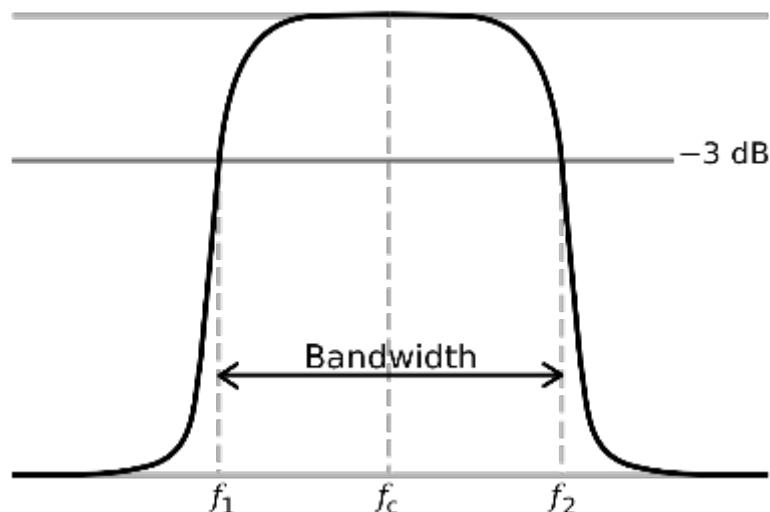


Figura 123. Definición del ancho de banda a 3 dB de una señal.

Fuente: Adaptado de [https://en.wikipedia.org/wiki/Bandwidth_\(signal_processing\)](https://en.wikipedia.org/wiki/Bandwidth_(signal_processing))

Cualitativamente, puede interpretarse que una señal caracterizada por un mayor ancho de banda presenta diferencias más significativas en la velocidad de variación de su perfil (variaciones lentas y rápidas). Por tanto, la señal contiene cambios más diversos y está asociada a una mayor cantidad de información.

Formantes

Las frecuencias formantes se identifican como **máximos locales** en la función de densidad espectral de potencia (como se ve en la figura 5). Estas componentes se asocian con las principales frecuencias contenidas en la señal bajo estudio.

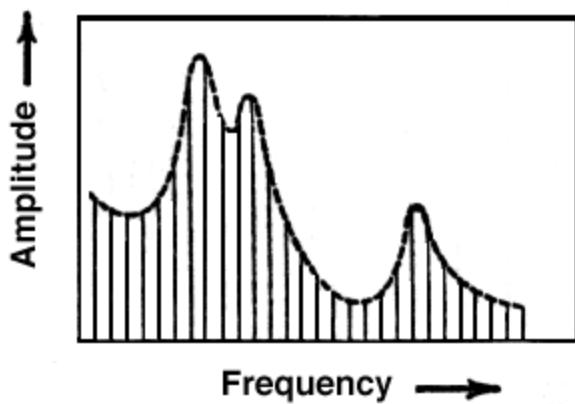


Figura 124. Ejemplo de una función de densidad espectral de potencia en la que se aprecian tres frecuencias fromantes.

Fuente: <https://www.sfu.ca/sonic-studio/handbook/Formant.html>

Caracterización estadística

La **normalización** de la función de densidad espectral de potencia, de tal forma que el área contenida bajo la misma sea igual a la unidad, hace que pueda ser interpretada como una función de densidad de probabilidad. Esta caracterizaría el comportamiento estadístico de la variable aleatoria que viene dada por la frecuencia. Así, el área bajo dicha función de probabilidad hasta la frecuencia ω_0 representaría la probabilidad de observar componentes frecuenciales menores o iguales que ω_0 en nuestra señal.

Así, a partir de esta **función de densidad de probabilidad**, pueden obtenerse parámetros de naturaleza estadística como la media, la mediana o la varianza. Concretamente, la frecuencia mediana se ha empleado comúnmente para la caracterización del espectro de una señal.

Entropía espectral

Por último, a partir de la función de densidad espectral normalizada se obtiene la entropía espectral. Este parámetro no es más que el cálculo de entropía para la variable aleatoria que representa la componente frecuencial.



Accede a los ejercicios de autoevaluación a través del aula virtual

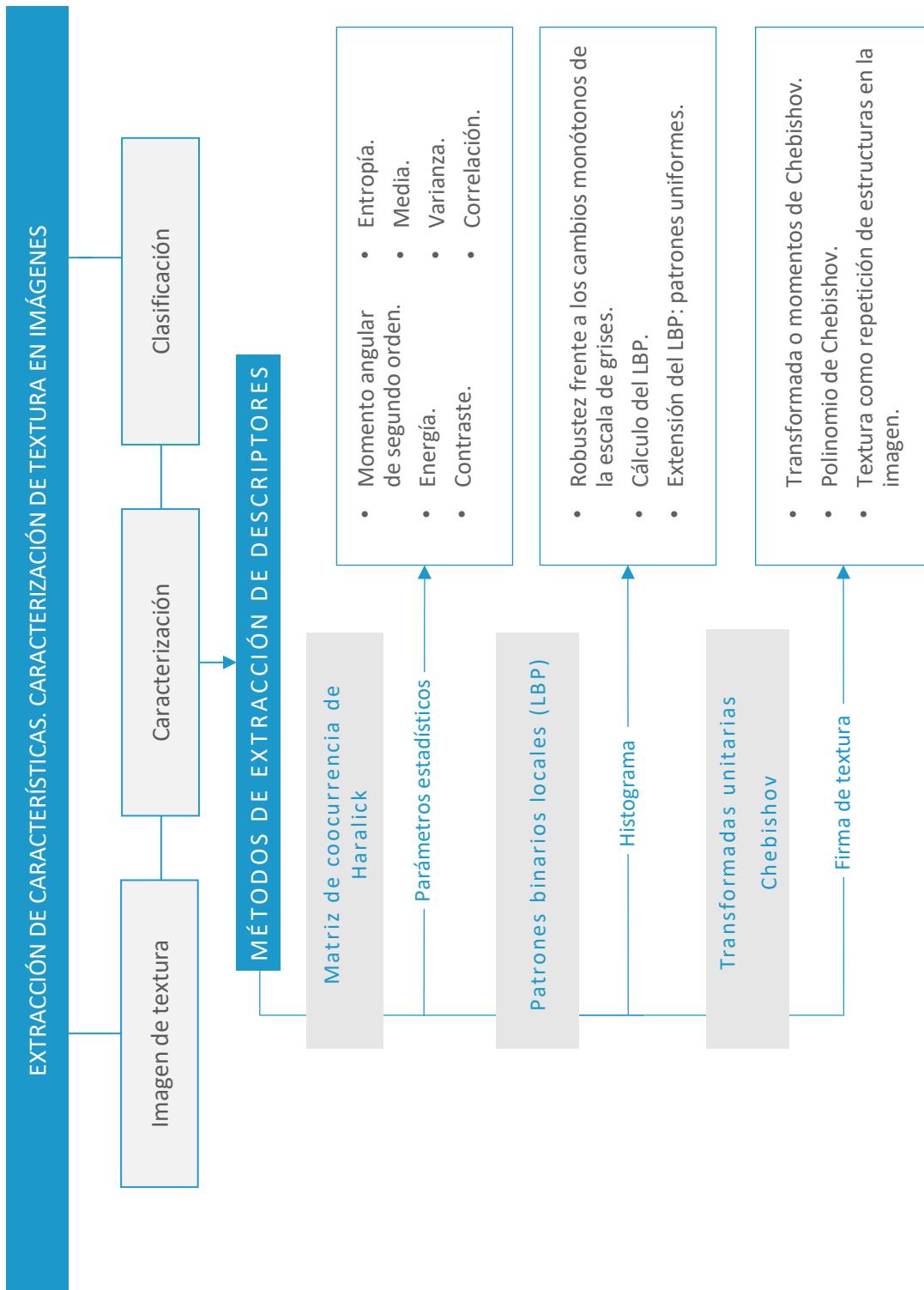


Accede al vídeo «Resumen del tema» a través del aula virtual

Percepción Computacional

Extracción de características. Caracterización de textura en imágenes

Esquema. Tema 12



Ideas clave. Tema 12

12.1. ¿Cómo estudiar este tema?



Accede al vídeo «Introducción. Extracción de características. Caracterización de texturas» a través del aula virtual

Para estudiar este tema deberás leer con atención las ideas clave que se desarrollan a continuación.

La textura ha sido objeto de estudio durante las últimas décadas para la comunidad científica inmersa en el procesamiento de imágenes. Sin embargo, no hay consenso sobre su definición. A partir de las definiciones proporcionadas por diferentes investigadores podemos decir que:

La textura es una propiedad relacionada con la aspereza de una superficie y se caracteriza por la variación de la intensidad de píxeles en el dominio espacial.

La textura puede encontrarse en imágenes de diferente naturaleza, por ejemplo: un tejido biológico específico (tejido muscular o nervioso), un área de terreno, la superficie de un objeto o la piel de una persona o animal. Por tanto, la caracterización de la textura en una imagen es necesaria en diferentes escenarios reales. Específicamente, la textura juega un papel relevante en tareas como el diagnóstico médico, la teledetección o la identificación biométrica.

En este tema, se aborda la caracterización de la textura para su **identificación automática**. En algunos problemas, el desafío consiste en identificar automáticamente a qué tipo de textura pertenece la imagen de entrada. Por ejemplo, considérese un escenario de diagnóstico médico en el que ha de indicarse si la

muestra de tejido corresponde a una zona dañada o, por el contrario, a una zona sana.

Así, tomemos la figura 1, donde se recogen diferentes ejemplos de textura. En la parte superior se aprecian tres texturas diferentes y en la parte inferior reúne también diferentes tipos de textura. Esta imagen refleja aquellos problemas de **segmentación** en los que se trata de delimitar diferentes zonas de la imagen de forma automática. En este caso, puede pensarse en una instantánea de un paisaje en la que se pretende distinguir áreas como el cielo, el terreno y la pared de una casa.

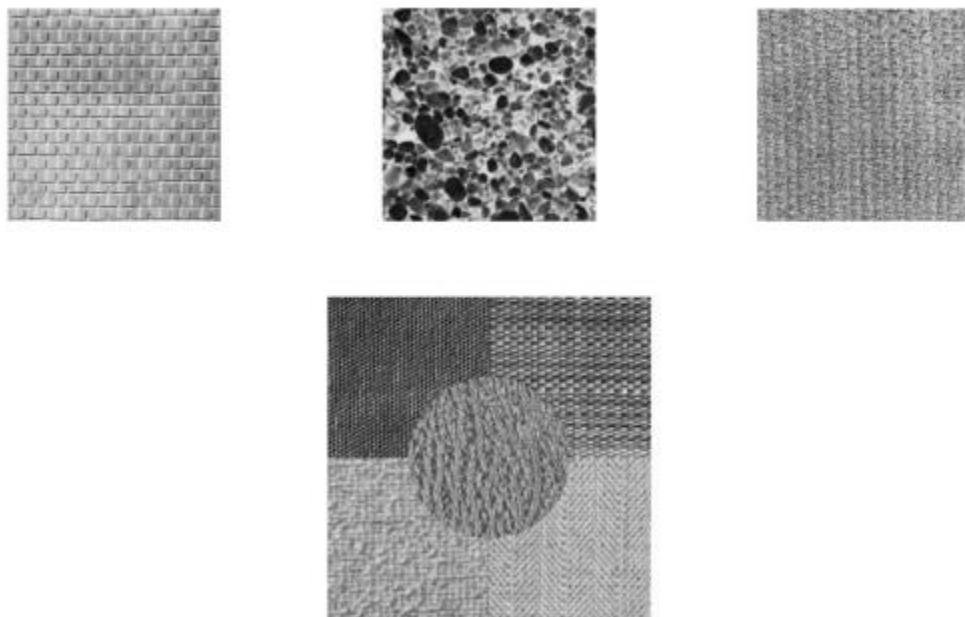


Figura 125. Ejemplo de tres texturas diferentes correspondientes a un muro de ladrillo, un suelo empedrado y un tejido textil.

Fuente: Adaptado de <http://www.ux.uis.no/~tranden/brodatz.html>

A fin de identificar una textura, el primer paso es la extracción de descriptores que permitan diferenciarla de otras. Existen diferentes métodos que pueden ser empleados para la caracterización. En este tema se describirán tres de ellos:

- ▶ La matriz de coocurrencia de Haralick.
- ▶ Los patrones binarios locales.
- ▶ La caracterización basada en el análisis de la imagen en un dominio transformado.

El siguiente esquema recoge, de forma breve, los conceptos desarrollados a lo largo de este tema.

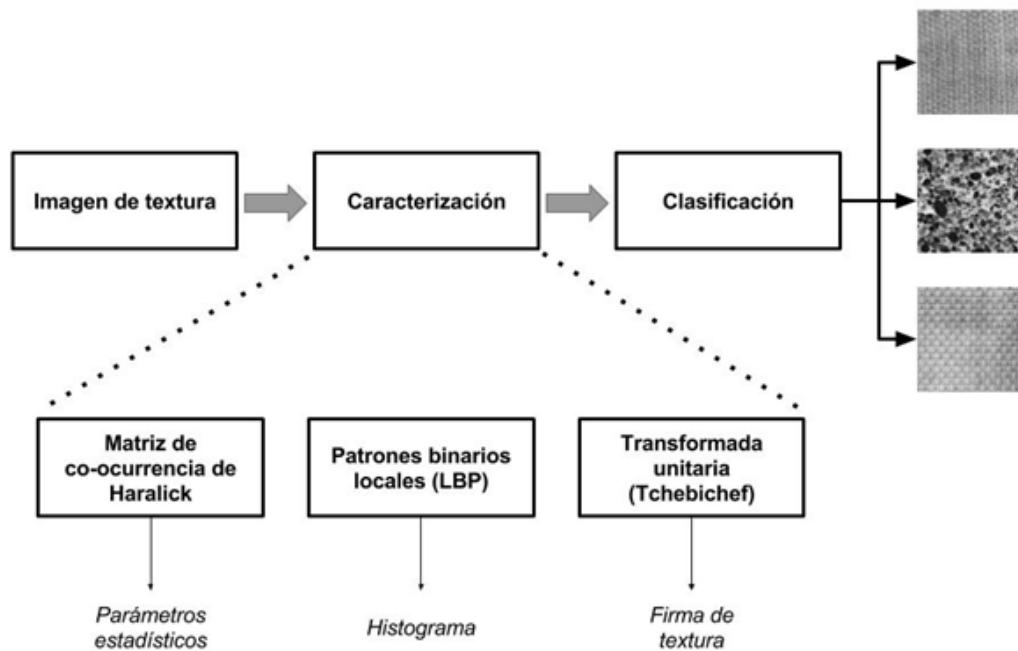


Figura 126. Principales herramientas para la caracterización de la textura en una imagen.

12.2. Matriz de coocurrencia de Haralick



Accede al vídeo «Matriz de co-ocurrencia de Haralick» a través del aula virtual

La textura está caracterizada por la relación espacial entre las intensidades correspondientes a sus píxeles. La matriz de coocurrencia refleja estas relaciones, obteniéndose para una dirección determinada entre píxeles vecinos: 0° , 45° , 90° y 135° . A partir de la matriz coocurrencia es posible obtener un gran número de descriptores estadísticos que facilitan una descripción precisa de la textura.

Esta matriz es cuadrada y de dimensión N_g , donde N_g es el número de niveles utilizados para la cuantificación de los valores de intensidad (gris) en la imagen. El elemento $[i, j]$ de la matriz se genera contando el número de veces que un píxel con

valor de intensidad i es adyacente a un píxel con valor j . Finalmente, se divide la matriz completa por el número total de las comparaciones realizadas. En otras palabras, cada elemento de la matriz puede interpretarse como la probabilidad de que un píxel con valor i se encuentre adyacente a un píxel de valor j .

La identificación de píxeles adyacentes depende de la dirección considerada. Así, dado que los píxeles de una imagen son cuadrados, pueden considerarse los cuatro ángulos indicados anteriormente para el cálculo de la matriz.

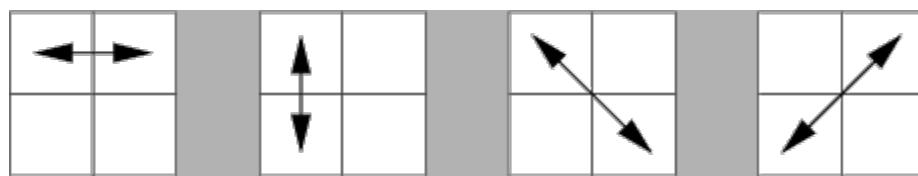


Figura 127. Cuatro direcciones posibles para la identificación de píxeles adyacentes en una imagen y el cómputo de la matriz de coocurrencia de Haralick.

Fuente: http://murphylab.web.cmu.edu/publications/boland/boland_node26.html

Comúnmente, se emplea $N_g = 8$ y se calculan las cuatro matrices posibles (una por cada ángulo diferente). La figura 4 muestra la estructura de la matriz resultante.

$$\mathbf{G} = \begin{bmatrix} p(1,1) & p(1,2) & \cdots & p(1,N_g) \\ p(2,1) & p(2,2) & \cdots & p(2,N_g) \\ \vdots & \vdots & \ddots & \vdots \\ p(N_g,1) & p(N_g,2) & \cdots & p(N_g,N_g) \end{bmatrix}$$

Figura 128. Matriz de coocurrencia de Haralick con N_g niveles y para una dirección determinada.

Fuente: http://murphylab.web.cmu.edu/publications/boland/boland_node26.html

A partir de estas matrices, se calculan características como las que se describen a continuación:

Momento angular de segundo orden

$$\sum_i \sum_j p^2(i,j)$$

Energía

$$\sqrt{\sum_i \sum_j p^2(i,j)}$$

Contraste

$$\sum_i \sum_j (i,j)^2 p(i,j)$$

Entropía

$$-\sum_i \sum_j p(i,j) \log(p(i,j))$$

Media

$$\mu_x = \sum_i \sum_j i p(i,j), \mu_y = \sum_i \sum_j j p(i,j)$$

Varianza

$$\sigma_x = \sqrt{\sum_i \sum_j (i - \mu_x)^2 p(i,j)}, \sigma_y = \sqrt{\sum_i \sum_j (j - \mu_y)^2 p(i,j)}$$

Correlación

$$\sum_i \sum_j p(i,j) (i - \mu_x)(j - \mu_y) / (\sigma_x \sigma_y)$$

12.3. Patrones binarios locales



Accede al vídeo «Patrones binarios locales» a través del aula virtual

Ll operador de patrones binarios locales o LBP, del inglés *local binary patterns*, permite obtener una transformación píxel a píxel de la imagen original. El valor transformado de un píxel se obtiene a partir de la comparación de su valor de intensidad con los píxeles vecinos. Dada la capacidad de este operador para caracterizar la textura y su simplicidad computacional, los LBP se han convertido en una técnica muy común en diversas aplicaciones dentro del procesado de imagen.

Quizás la propiedad más importante del operador de LBP en aplicaciones del mundo real es su **robustez frente a los cambios monótonos** de la escala de grises causados, por ejemplo, por las variaciones de iluminación (ruido). Como se mencionaba previamente, otra propiedad interesante de los LBP es su facilidad de cálculo, lo que hace de este operador una herramienta muy útil para el análisis de imágenes en entornos donde el tiempo real es crítico.

El operador LBP se basa en la idea de que las texturas superficiales bidimensionales pueden describirse mediante dos **medidas complementarias**:

- ▶ Patrones espaciales locales.
- ▶ Contraste de escala de grises.

El operador original de LBP, propuesto por Ojala (1996), se obtiene de la siguiente forma para cada píxel de la imagen:

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p, s(x) = 1 \text{ si } x > 0 \text{ y } s(x) = 0 \text{ si } x \leq 0$$

Donde:

- ▶ g_c hace referencia a la intensidad del píxel para el que se está calculando la transformación.
- ▶ Los píxeles g_p ($p = 0, \dots, P - 1$) son sus P píxeles vecinos para el radio R considerado.

Para aquellos puntos de la vecindad que no coincidan exactamente con la posición de un píxel de la imagen, se obtiene una estimación de la intensidad en ese punto mediante la interpolación a partir de los píxeles más próximos. La notación (P, R) se usa para identificar las características de la vecindad considerada, lo que significa P puntos de muestreo en un círculo de radio de R .

Por tanto, el procedimiento para el **cálculo del LBP** conlleva:

4. Tomar la muestra de píxeles vecinos.
5. Calcular la diferencia del píxel central con estos.
6. Aplicar el umbral 0 sobre esta diferencia.
7. Posteriormente, se transforma la cadena binaria obtenida en el número decimal correspondiente.

En la siguiente imagen se muestra un ejemplo del cálculo del operador LBP.

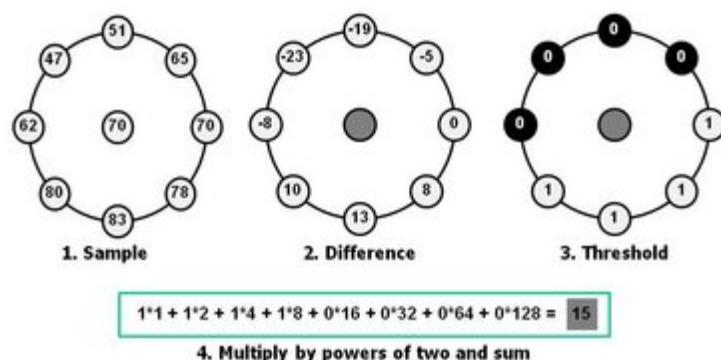


Figura 129. Etapas involucradas en el cálculo del operador LBP.
Fuente: http://www.scholarpedia.org/article/Local_Binary_Patterns

Como resultado, cada píxel de la imagen es asignado con una de las 2^P posibles etiquetas. Comúnmente, el radio R es tal que $P = 8$. El histograma de estas $2^8 = 256$ etiquetas diferentes se puede usar como un **descriptor de textura**.

Dado el elevado número de contenedores del histograma resultante, el resultado de la transformación tenderá a generar histogramas con un número significativo de 0, pues muchas de las 2^P etiquetas posibles no habrán sido asignadas a ningún píxel. Además, el resultado de la transformación variará si la imagen original es rotada un determinado ángulo. A fin de superar estas limitaciones, se propuso una **extensión del operador LBP** original basada en la definición de los llamados patrones uniformes. Estos permiten reducir la longitud del vector de características e implementar un descriptor invariante de rotación simple.

Esta extensión se inspiró en el hecho de que algunos patrones binarios aparecen con mayor frecuencia en imágenes de textura que otros.

Un patrón binario local se denomina **uniforme** si su cadena de bits contiene como máximo dos transiciones de 0 a 1 o viceversa. Por ejemplo:

- ▶ Los patrones 00000000 (0 transiciones), 00001110 (2 transiciones) y 11110011 (2 transiciones) son uniformes.
- ▶ Mientras que los patrones 00110110 (4 transiciones) y 10101101 (6 transiciones) no lo son.

En la extensión propuesta del operador se utiliza una única **etiqueta** para todos aquellos píxeles que den lugar a un patrón no uniforme. En el caso de los patrones uniformes, se emplea una etiqueta diferente para cada patrón resultante. Hay $P + 1$ patrones uniformes diferentes. Por tanto, el número total de etiquetas para el cómputo del histograma empleado en la caracterización de la textura será de $P + 2$. Por ejemplo, cuando se usa el vecindario (8, 1), hay un total de 256 patrones, 58 de los cuales son uniformes, lo que da como resultado 59 etiquetas diferentes.

Ojala detectó en sus experimentos con imágenes de textura que los patrones uniformes representan casi el 90 % de todos los patrones cuando se usa el vecindario (8,1), y alrededor del 70 % en el vecindario (16, 2). Cada etiqueta o código LBP se puede considerar como una microestructura de textura (*microtexton*) en una imagen en la que esta estructura se repite varias veces. Las primitivas locales codificadas por estas etiquetas se corresponden con elementos específicos de la imagen como diferentes tipos de bordes curvos, manchas, zonas planas, etc.

El operador de LBP invariante de rotación original, designado como LBPrui2, se obtiene rotando circularmente cada patrón de bits al valor mínimo. Por ejemplo, las secuencias de bits 1000011, 1110000 y 0011100 surgen de diferentes rotaciones del mismo patrón local y todas corresponden a la secuencia 0000111 una vez normalizadas.

La siguiente imagen (figura 6) muestra los diferentes patrones uniformes que podemos encontrar cuando el operador LBP se calcula con R=1 y P=8.

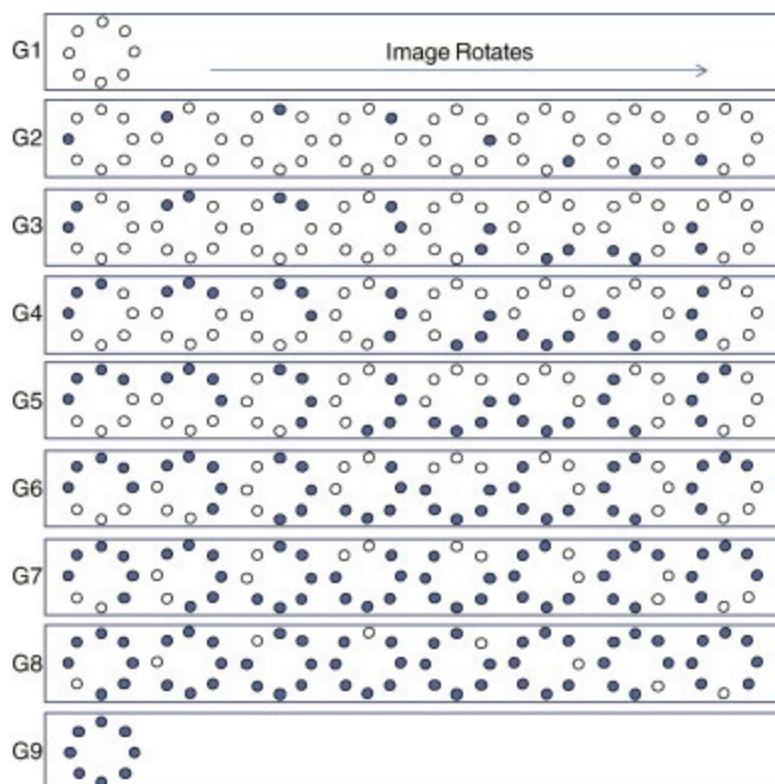


Figura 130. Patrones uniformes observables para R=1 y P=8.

Fuente: Qi et al., 2015.

12.4. Caracterización basada en transformadas unitarias



Accede al vídeo «Transformadas unitarias» a través del aula virtual

Las transformadas unitarias permiten la representación de la imagen, considerada una señal $f(x, y)$ definida en el espacio R^2 , en un espacio transformado. Para ello, se computa la proyección de la señal sobre funciones unitarias (ortonormales), que suponen una base para el espacio de funciones del dominio transformado. Comúnmente, estas funciones base son de tipo sinusoidal, característica que las hace una herramienta útil para capturar las propiedades de textura en una imagen.

El motivo de este hecho es que la textura, como se ha dejado intuir previamente, se caracteriza por la **repetición de ciertas estructuras en la imagen**. Por tanto, una función compuesta por un tren de pulsos o sinusoides, al calcularse la proyección de la imagen sobre esta, lleva a cabo el muestreo de la superficie definida por la función $f(x, y)$, que es la imagen. Este muestreo captura las repeticiones inherentes a la presencia de una textura.

Existen diferentes tipos de transformadas, es decir, de funciones unitarias que son base de un espacio de funciones (espacio transformado). La transformada de Fourier o la transformada del coseno son algunos ejemplos. En este apartado dedicado a la caracterización de la textura, se expone el uso de la transformada de Chebyshev para este propósito.

Los coeficientes de la transformada de Chebyshev son conocidos como **momentos de Chebyshev**. Fueron definidos por Mukundan (2001) con el objetivo de suprimir las limitaciones encontradas en los momentos de Zernike y Legendre.

El cálculo de la **transformada o momentos de Chebyshev** (T_{pq}) se lleva a cabo mediante la proyección de la imagen o señal original $f(x, y)$ sobre el polinomio de Chebyshev $T_{pq}(x, y)$:

$$T_{pq} = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} t_{pq}(x, y) f(x, y)$$

El **polinomio de Chebyshev** $t_{pq}(x, y)$ puede expresarse como el producto de los polinomios unidimensionales $t_p(x)$ y $t_q(y)$. Por tanto, la expresión de T_{pq} descrita previamente es equivalente a la siguiente:

$$T_{pq} = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} t_p(x) t_q(y) f(x, y)$$

Donde: $t_n(x)$ es el polinomio de Chebyshev de grado n en una dimensión.

La siguiente figura muestra la forma de estas funciones.

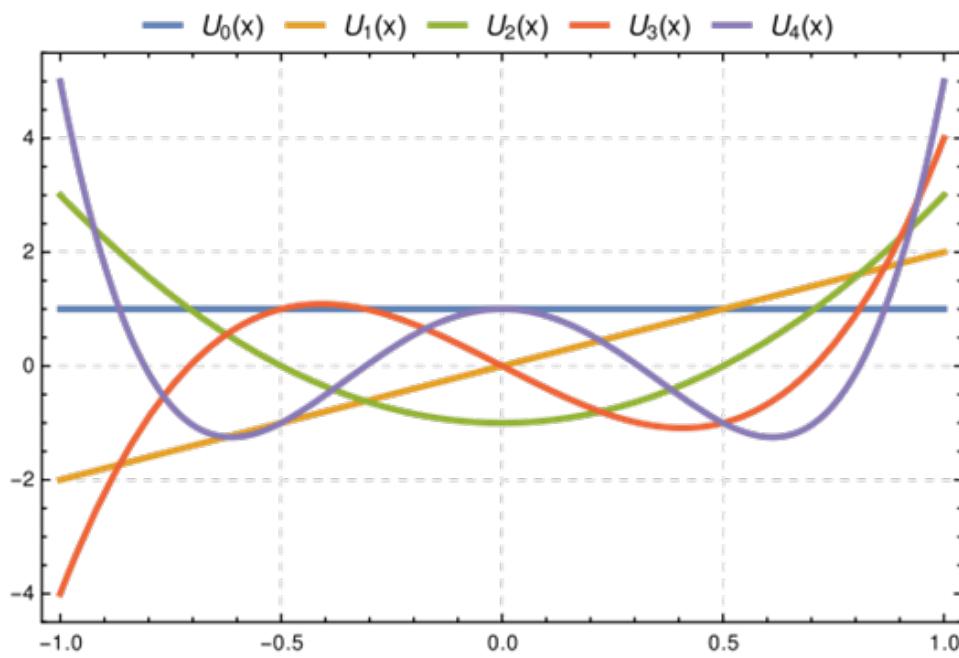


Figura 131. Perfil de los cinco primeros polinomios de Chebyshev.

Fuente: https://en.wikipedia.org/wiki/Chebyshev_polynomials

Como puede observarse, las funciones base tienen un perfil sinusoidal en una dimensión. En dos dimensiones, este perfil se prolonga a lo largo de ambos ejes, tal y como se representa en la siguiente figura. El efecto resultante es el de un tren de pulsos en x e y que muestrea la superficie de la función $f(x, y)$.

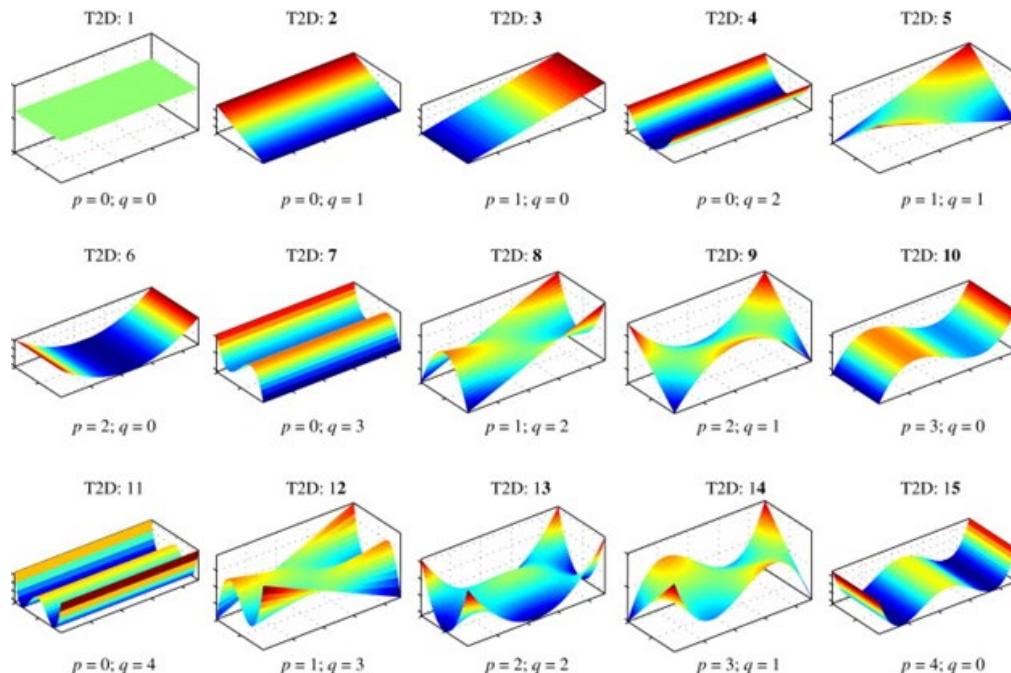


Figura 132. Polinomios de Chebyshev en dos dimensiones.

Fuente: Wang et al., 2011.

Como se demuestra en el trabajo de Marcos y Cristóbal (2013), existe una relación entre el orden del polinomio $s = p + q$ y sus componentes frecuenciales. Así, órdenes mayores están asociados con mayores frecuencias de muestreo: un momento Tn de mayor magnitud que otro momento Tm , con $n > m$, refleja que la imagen original contiene estructuras que se repiten con mayor frecuencia. Si una textura está compuesta, por tanto, por elementos representativos de menor tamaño que se repiten más frecuentemente a lo largo de la imagen, dará lugar a momentos de orden alto de mayor magnitud.

A partir de esta observación, se propuso caracterizar la textura mediante la suma de las magnitudes de los momentos de un mismo orden.

Se obtiene como resultado una curva $M(s)$, función del orden s , tal y como se representa en la siguiente figura.

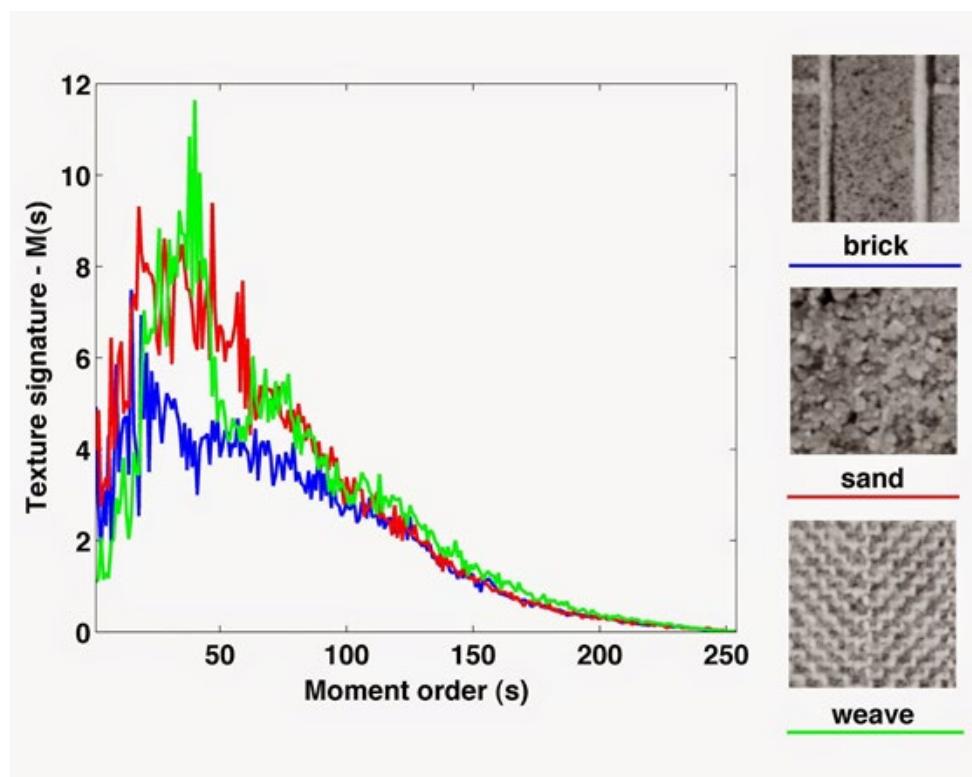


Figura 133. Firmas de textura para tres imágenes diferentes derivadas de los momentos Chebyshev de la imagen original.

Fuente: <http://eximarinis.blogspot.com.es/2014/05/a-novel-method-for-image-texture.html>.

Esta curva o firma de textura refleja la repetición de estructuras similares en la imagen con la frecuencia representada por el orden s . Así, la textura que muestra un muro de ladrillo se caracteriza por una curva con un pico en valores menores de s que las texturas de arena y tejido. En estas últimas, las estructuras que caracterizan a la textura son de menor tamaño y, por tanto, se encuentran en mayor número para una misma superficie de imagen considerada. Como resultado, las firmas de textura a las que dan lugar se concentran en valores mayores de s .

Puede observarse que el efecto de este operador es similar al de un banco de filtros paso banda centrados en diferentes frecuencias.

12.5. Referencias bibliográficas

Marcos, J. V. y Cristóbal, G. (2013). Texture classification using discrete Tchebichef moments. *Journal of the Optical Society of America A*, 30(8), 1580-1591.

Mukundan, R., Ong, S. H. y Lee, P. A. (2001). Image analysis by Chebyshev moments. *IEEE Trans. Image Process.* 10, 1357–1364.

Qi, X., Shen, L., Zhao, G., Li, Q. y Pietikäinen, M. (2015). Globally rotation invariant multi-scale co-occurrence local binary pattern. *Image and Vision Computing*, 43, 16-26.

Soh, L. K. y Tsatsoulis, C. (1999). Texture analysis of SAR sea ice imagery using gray level co-occurrence matrices. *IEEE Transactions on Geoscience and Remote Sensing*, 37, 780–795.

Wang, W. et al. (2011). Finite element model updating from full-field vibration measurement using digital image correlation. *Journal of Sound and Vibration*, 330(8), 1599-1620.

Zhang, J. y Tieniu T. (2002). Brief review of invariant texture analysis methods. *Pattern recognition*, 35(3), 735-747.



Accede a los ejercicios de autoevaluación a través del aula virtual

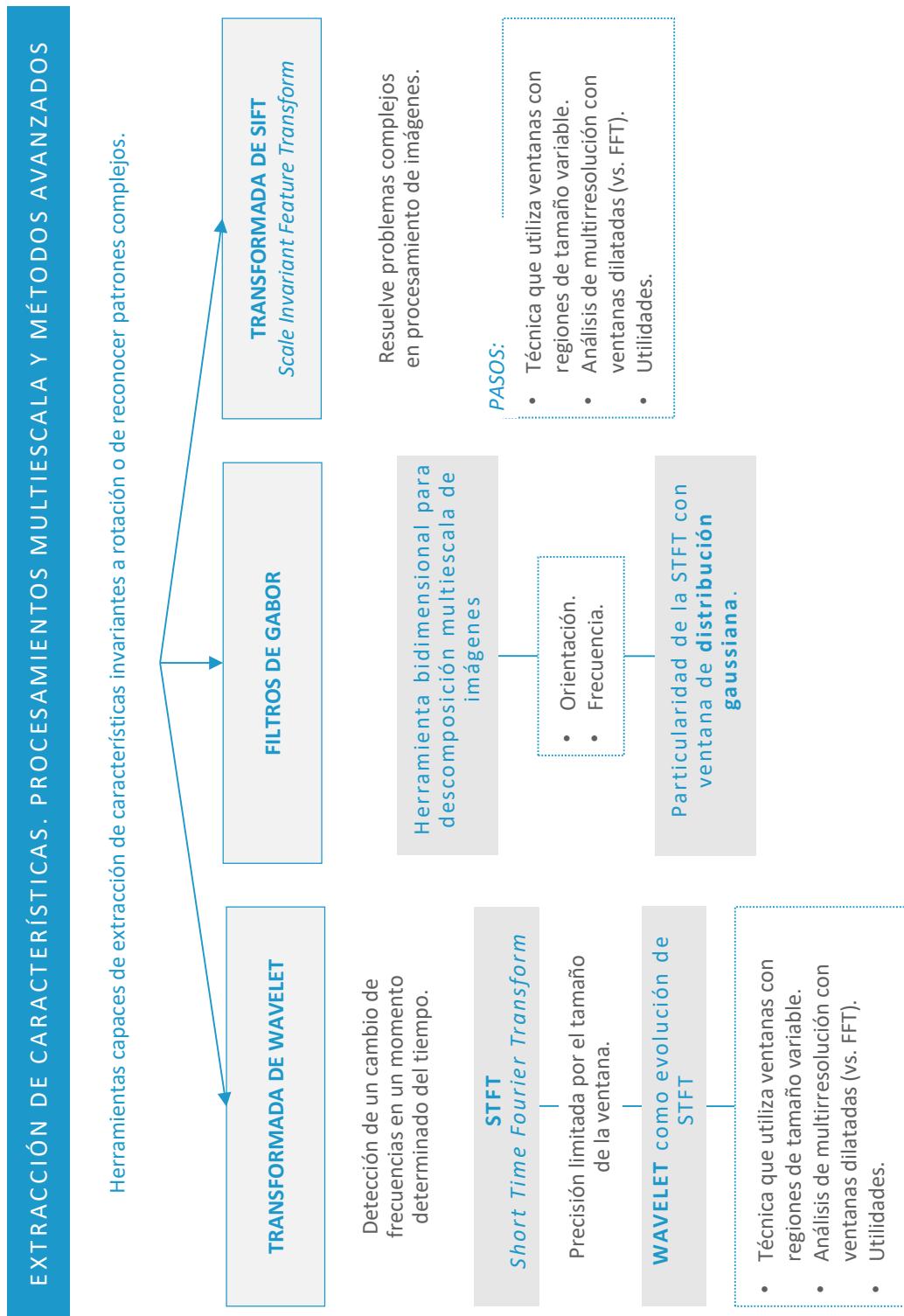


Accede al vídeo «Resumen del tema» a través del aula virtual

Percepción Computacional

Extracción de características. Procesamientos multiescala y métodos avanzados

Esquema. Tema 13



Ideas clave. Tema 13

13.1. ¿Cómo estudiar este tema?

Para estudiar este tema deberás leer con atención las ideas clave que se desarrollan a continuación.

Sin llegar a entender la matemática detrás de estos complejos algoritmos de extracción de características, se dotará de los conceptos necesarios para entender la funcionalidad de las Wavelets, filtros de Gabor y algoritmo SIFT.

Las nociones aquí presentadas te permitirán entender en qué caso y qué algoritmo deberás de emplear y qué parámetros necesitas para su correcto funcionamiento.

13.2. Introducción a la extracción de características basada en procesamientos multiescala



Accede al vídeo « Introducción. Extracción de características. Procesamientos multiescala y métodos avanzados » a través del aula virtual

De todos los métodos presentados para la extracción de características, la mayoría de ellos carece de la posibilidad de reconocer patrones más complejos y elaborados como puede ser la identificación de un objeto. El análisis de imágenes basado en histogramas de color, variaciones de las intensidades o incluso reconocimiento de patrones para identificar una textura sufren, en muchos

casos, de limitaciones para identificar la rotación de un objeto en una imagen o el hecho de que parte de un objeto esté oculto.

Un problema tan sencillo de definir como puede ser la identificación de un objeto independientemente de su rotación o traslación (como se puede ver en la imagen a continuación), es altamente complejo usando las extracciones de características presentadas hasta ahora, por no decir imposible.



Figura 134. Ejemplo de funcionamiento del algoritmo SIFT para la translación de una imagen.

Fuente: <http://timothykurek.com/10-sift-image-processing/introduction-to-sift-scale-invariant-feature-transform-opencv-on-sift-image-processing/>

Es por eso que surgen otras herramientas como son la transformada Wavelet, los filtros de Gabor o el algoritmo SIFT (*Scale Invariant Feature Transformation*) que son capaces de proporcionar características más robustas para:

- ▶ Mejorar la información extraída de una imagen y proporcionar una visión más detallada de texturas, variabilidad de la imagen u objetos presentes.
- ▶ Proporcionar características invariantes al cambio de escala, rotación y ruido que pueda estar presente en una captura de imagen.
- ▶ Resolver, en definitiva, problemas más avanzados de procesamiento de imagen.

No obstante, las matemáticas necesarias para comprender el funcionamiento de estos algoritmos son muy avanzadas y no se presentarán de forma detallada en este tema. Se planteará la **descripción de estos algoritmos de forma funcional**

entendiendo qué hacen, cómo y por qué esto supone una ventaja con respecto a otros algoritmos de extracción de características.

Entre las **aplicaciones más comunes** de estos algoritmos están:

- ▶ Identificar a las personas en función de su huella u iris, también denominados **algoritmos de biometría**. La identificación basada en características biométricas precisa de matemática y algoritmos avanzados para procesar la imagen más allá de una variación de intensidades. Gracias a las Wavelets y los filtros de Gabor, podemos extraer características invariantes al reescalado del iris (hay que considerar que el iris y la pupila varían su tamaño casi un número infinito de veces al día, lo que dificulta el reconocimiento), intensidad de brillo así como el color de la imagen.
- ▶ El capturar con nuestro móvil una secuencia de fotos y hacer una panorámica. Los algoritmos como SIFT son capaces de encontrar esos puntos en común entre dos fotos y aplicar las transformadas necesarias de rotación y reescalado para poder fusionar las imágenes correctamente.
- ▶ La identificación de monedas en una máquina de expendedoras (*vending*). En algunas se capture una foto de la moneda (la cual no presenta siempre el mismo grado de rotación) y con ello se realiza la identificación del valor de la misma. Esta técnica requiere de Wavelets para extraer características invariantes a la rotación.

13.3. Definición de transformada Wavelet



Accede al vídeo «Transformada wavelet» a través del aula virtual

Existen numerosas aproximaciones para describir las Wavelets, en algunos libros conocidas como «ondículas» incluso «onditas». En este tema, y puesto que previamente hemos presentado en detalle la transformada de Fourier, vamos a partir de esta misma transformada para entender qué aspecto del que carecen las transformadas de Fourier pretenden cubrir las Wavelets.

Recordemos brevemente que la **transformada de Fourier** permite descomponer una señal en sus componentes sinusoidales de diferentes frecuencias; en otras palabras: puede pensarse que es una técnica matemática para transformar el punto de vista de una señal desde la base de tiempo a la base de la frecuencia. La transformada de Fourier cubre un amplio espectro de problemas, sin embargo, su **limitación** comienza cuando, dada una señal con gran variabilidad de frecuencia, no se puede determinar en qué momento del tiempo sucedió dicha variación de frecuencia.

Hay que recordar que la transformada de Fourier está pensada únicamente para el análisis de frecuencias de una señal temporal y, para ello, se olvida de toda componente temporal analizando las componentes espectrales (de frecuencia).

Cuando se observa una señal, producto de la transformación de Fourier, resulta imposible determinar cuándo ocurre un determinado evento o cuándo está presente una determinada frecuencia si las propiedades de la señal que se está analizando no cambian demasiado en el tiempo, es decir, si se está trabajando con una **señal estacionaria**.

Esta desventaja no resulta muy relevante, como en el caso de señales periódicas, por ejemplo. Sin embargo, un importante número de señales de interés presentan características no estacionarias o transitorias, tales como: una tendencia, cambios abruptos, comienzos o finales de eventos, etc.

A menudo, estas características no estacionarias resultan ser las secciones más interesantes de las señales, y la transformada de Fourier no está preparada para detectarlas o analizarlas.

Evolucionar la transformada de Fourier: STFT

En un esfuerzo por corregir la deficiencia presentada en el punto previo, en 1946 Denis Gabor adaptó la transformada de Fourier para poder analizar una pequeña sección de la señal en un determinado tiempo mediante una especie de ventana. Esta adaptación es la que se conoce como STFT (*Short Time Fourier Transform*), la cual lleva una señal del plano del tiempo al plano bidimensional de tiempo y frecuencia.

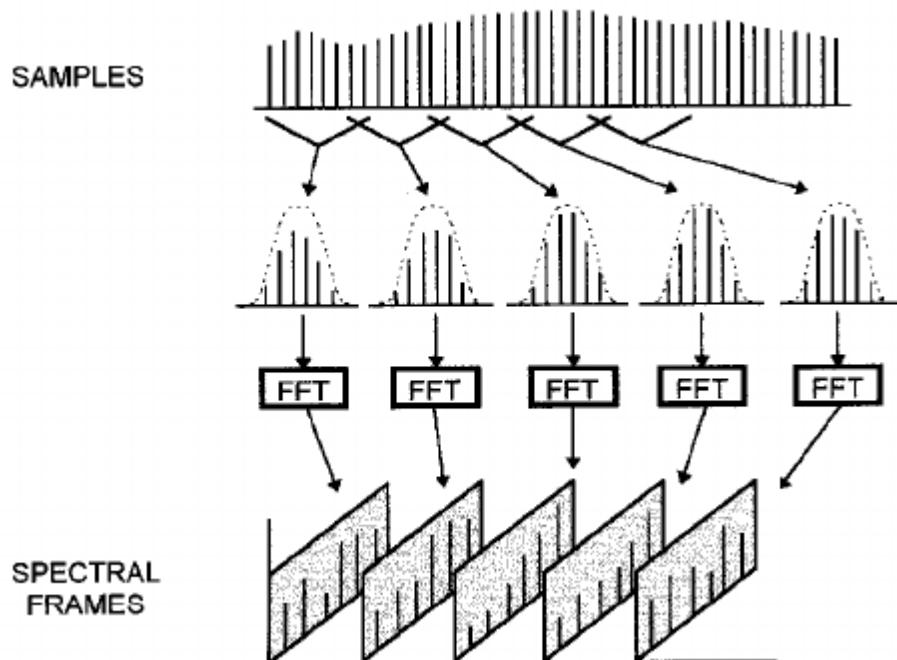


Figura 135. Funcionamiento visual de la Short Time Fourier Transform.

Fuente: https://www.researchgate.net/figure/Short-time-Fourier-transform-STFT-with-envelope-and-two-sample-overlap_fig7_231828310

En la anterior imagen se puede ver que, donde para cada ventana de tiempo, se calcula la transformada de Fourier (FFT), analizando posteriormente las componentes espectrales (de potencia por ventana)

Es importante mencionar que la STFT representa un compromiso entre el dominio del tiempo y el de la frecuencia de una señal, proporcionando información acerca de cuándo y a qué frecuencia de una señal ocurre un determinado evento. Sin embargo, solamente se puede obtener dicha información con una **precisión limitada**, la cual está acotada por el tamaño de la ventana.

A continuación, se presentan dos ejemplos de STFT sobre bioseñales, concretamente electromiogramas. En el caso de la izquierda, se aprecia cómo la señal, a medida que pasa el tiempo, posee diferentes componentes en frecuencia. Dichas componentes no se habrían apreciado con un análisis en frecuencia basado en Fourier, ya que no habría tenido en cuenta esa variación temporal de frecuencias. En resumen, en esta comparativa podemos ver cómo, en función del tiempo, se ven las diferentes componentes espectrales.

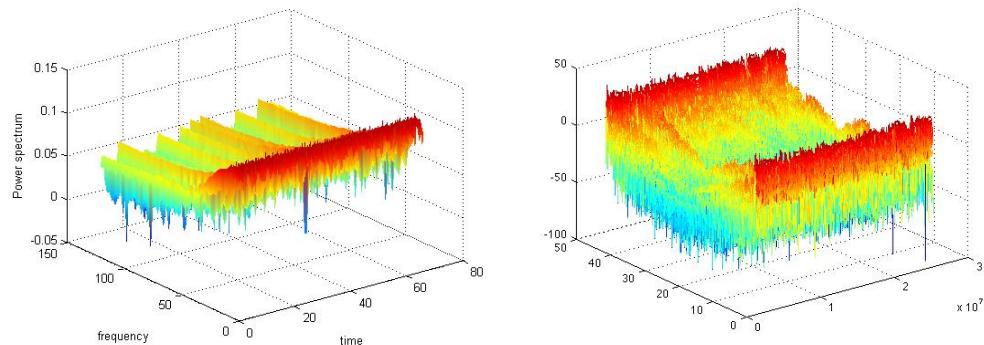


Figura 136. Dos visualizaciones de STFT de señales unidimensionales provenientes de electromiogramas.

Fuente: Adaptado de <https://es.mathworks.com/matlabcentral/fileexchange/45405-short-time-fourier-transform-stft?focused=3809422&tab=function> y <https://stackoverflow.com/questions/14586685/stft-computation-without-using-spectrogram-function>

Mientras que el compromiso entre la información del tiempo y la frecuencia puede resultar útil, el inconveniente surge cuando, una vez que se escoge un determinado tamaño para la ventana de tiempo, dicha ventana es la misma para todas las frecuencias. Muchas señales requieren un acercamiento más flexible, de tal modo que sea posible **variar el tamaño de la ventana** para determinar con mayor precisión el tiempo o la frecuencia.

Wavelets como evolución de STFT

El análisis Wavelet representa el paso lógico siguiente a la STFT; es una técnica que, mediante ventanas con **regiones de tamaño variable**, permite:

- ▶ El uso de intervalos grandes de tiempo en aquellos segmentos en los que se requiere mayor precisión en baja frecuencia.
- ▶ Regiones más pequeñas donde se requiere información en alta frecuencia.

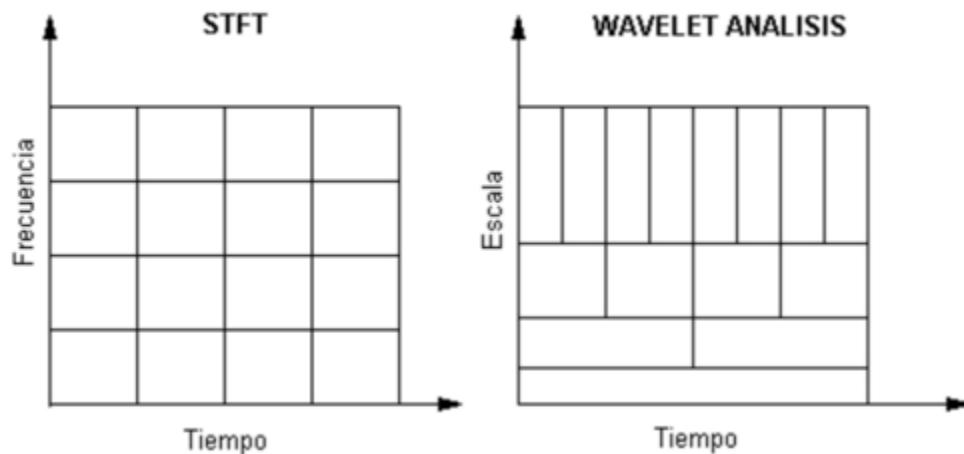


Figura 137. Comparativa de esquema entre STFT y Análisis con Wavelets.

Fuente: Adaptado de https://www.researchgate.net/figure/a-STFT-with-fixed-aspect-ratio-b-wavelet-transform-with-variable-aspect-ratio_fig2_228413897

Esta transformada es eficiente para el análisis local de señales no estacionarias y de rápida transitoriedad. Al igual que la transformada de Fourier con ventana, mapea la señal en una representación de tiempo-escala; la diferencia está en que la transformada Wavelet provee **análisis de multirresolución con ventanas dilatadas**:

- ▶ El análisis de las frecuencias de mayor rango se realiza usando ventanas angostas.
- ▶ El análisis de las frecuencias de menor rango se hace utilizando ventanas anchas.

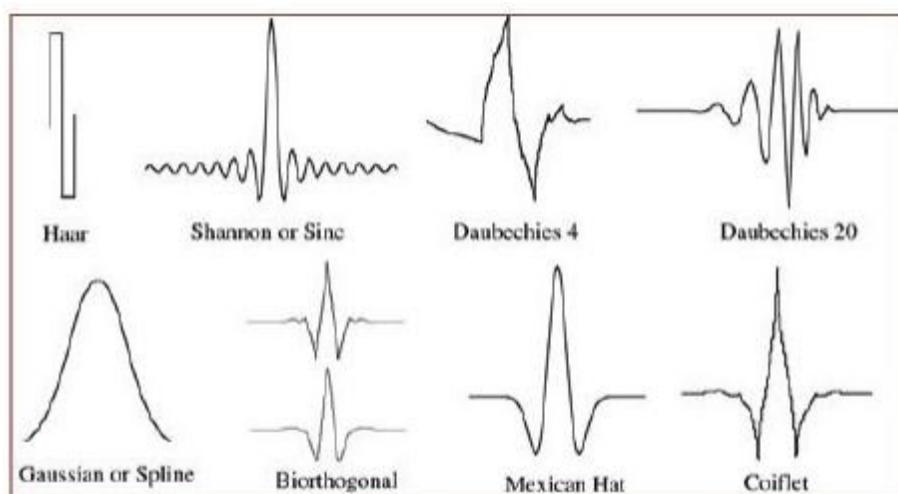


Figura 138. Resumen con las Wavelets más comunes.

Fuente: <http://www.continuummechanics.org/wavelets.html>

Una forma sencilla de comprender el modo de operación de esta transformada es pensar que la señal en base de tiempo es pasada por varios filtros paso bajo y paso alto, los cuales permiten separar las porciones de la señal de alta frecuencia de aquellas de baja frecuencia.

Este procedimiento se repite cada vez sobre algunas porciones de la señal correspondientes a aquellas frecuencias que han sido eliminadas de la señal original.

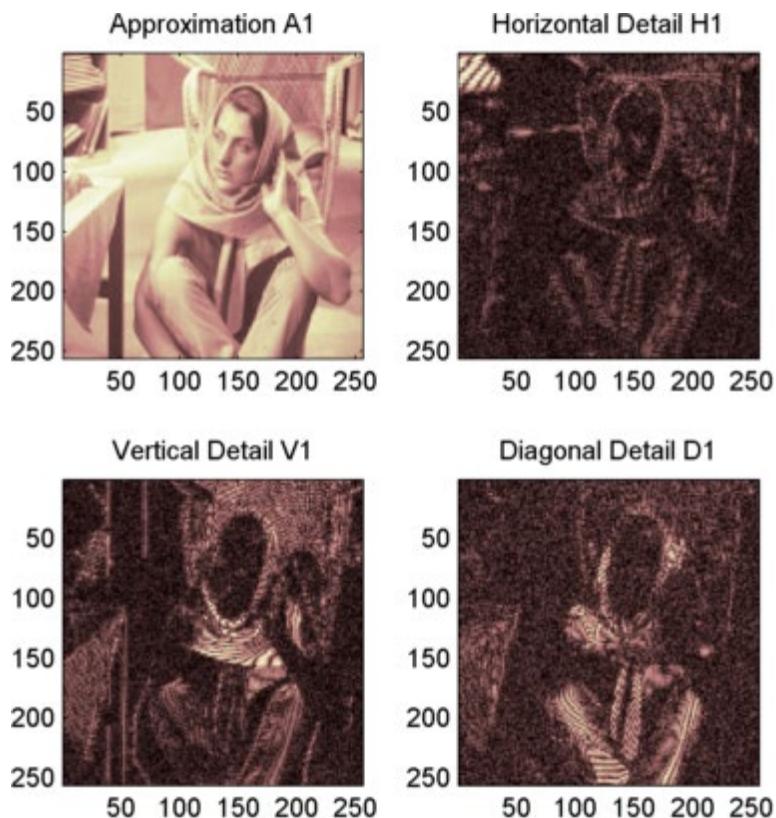


Figura 139. Ejemplo de descomposición multiescala en sus componentes horizontales, verticales y diagonales.

Fuente: <https://es.mathworks.com/help/wavelet/ug/two-dimensional-discrete-wavelet-analysis.html>

Utilidades de la transformada Wavelet

Como ya se ha mencionado anteriormente, el procesamiento de señales a través de Wavelets tiene innumerables aplicaciones en diversos ámbitos de la ciencia e ingeniería. A continuación se describe una lista de ejemplos:

- ▶ Detección de **discontinuidades o de puntos de quiebre** en señales, en una o varias dimensiones: resulta de gran utilidad, en especial en el tratamiento de imágenes, en donde interesa detectar la frontera entre colores y formas, o también en sistemas altamente dinámicos, en donde concierne determinar cuándo o dónde se producen los cambios.
- ▶ Estudio de fractales: mediante Wavelets se puede reconocer un **patrón repetitivo** en una señal o imagen, lo que la convierte en una herramienta poderosa en el estudio de fractales.
- ▶ Identificación de frecuencias puras: como se trata de una transformada compuesta por una base ortogonal de señales (análoga con la base sinusoidal de Fourier), también pueden ser utilizadas para **estudiar el contenido espectral** de señales.
- ▶ Eliminación de ruido: el análisis de señales mediante Wavelet también permite la **eliminación o filtrado de ruido** tanto en señales unidimensionales como en imágenes (bidimensionales).
- ▶ **Compresión** de imágenes: se trata de una de las aplicaciones más importantes de Wavelets, se realiza mediante el análisis en dos dimensiones.
- ▶ Aplicaciones en medicina: se ha incorporado el análisis con Wavelets a señales biológicas, permitiendo interpretar los resultados de exámenes médicos, facilitando el **diagnóstico de las enfermedades**. Por ejemplo, se ha aplicado con éxito en el análisis de electroencefalogramas, debido a que la naturaleza de este tipo de señales son altamente no estacionarias (impidiendo el uso de Fourier):
 - Las Wavelets permiten transformar la señal al dominio tiempo-frecuencia, relacionando el contenido espectral al momento de su ocurrencia. Ha sido aplicado en el diagnóstico de pacientes con alzheimer, enfermedad que hasta ahora era difícil de diagnosticar. El procedimiento que se realiza es entrenar redes neuronales con los datos obtenidos (a través de Wavelets de multiresolución) de pacientes de los que se sabe padecen la enfermedad, para

luego introducir las mediciones de aquellos que se realizan exámenes, y de este modo poder establecer si se siguen los mismos patrones.

13.4. Filtros de Gabor



Accede al vídeo «filtros de Gabor» a través del aula virtual

Los filtros de Gabor son una particularidad de la STFT donde la ventana tiene una distribución gaussiana. Detrás de los filtros de Gabor existe una explicación basada en la naturaleza. De hecho, tanto las Wavelets como los filtros de Gabor son **los que mejor imitan el funcionamiento del ojo natural**.

Desde la década de 1980 es bien conocido que las células simples de la corteza visual de nuestro cerebro presentan una sensibilidad especial hacia la orientación y la frecuencia espacial en una escena visual.

Por **orientación** nos referimos al ángulo formado por un objeto y un eje de referencia elegido arbitrariamente, mientras que la **frecuencia** espacial da cuenta de la repetición de un patrón en el espacio, medido en unidades inversas de longitud.

Dichas células aparecían en agrupaciones binarias, donde cada par era capaz de detectar una cierta orientación y frecuencia. Sin embargo, había una diferencia de fase de $\pi/2$ radianes entre ambas células, es decir: mientras la primera podía detectar objetos simétricos o repetidos (líneas paralelas), la otra se encargaba de los objetos asimétricos.

Al contrario que en las Wavelets, que son unidimensionales, los filtros de Gabor son bidimensionales, convirtiéndolo en una excelente herramienta para la descomposición multiescala de imágenes.

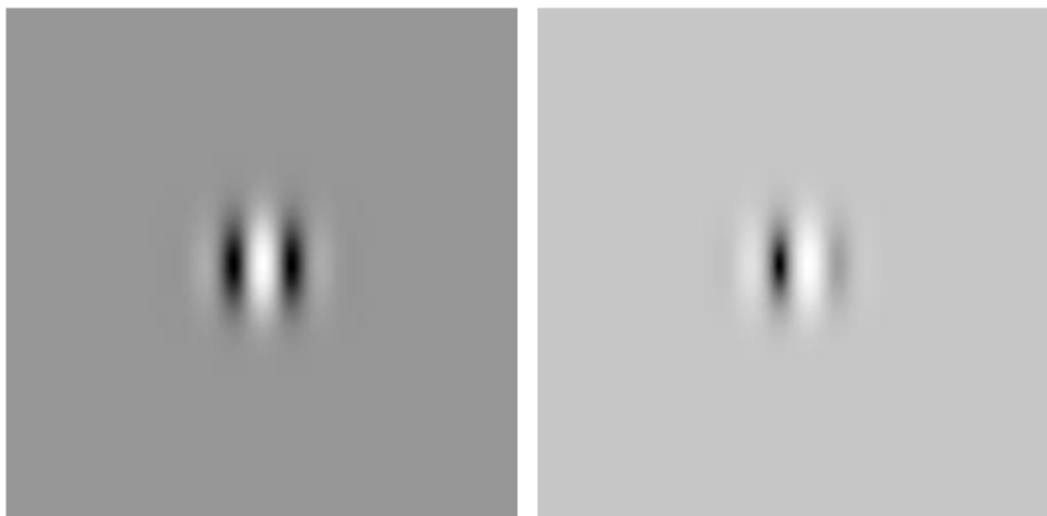


Figura 140. Representación de la parte real de un filtro de Gabor.
Fuente: <http://www.di.uniba.it/~ig/GaborFilter/html/filtro.html>.

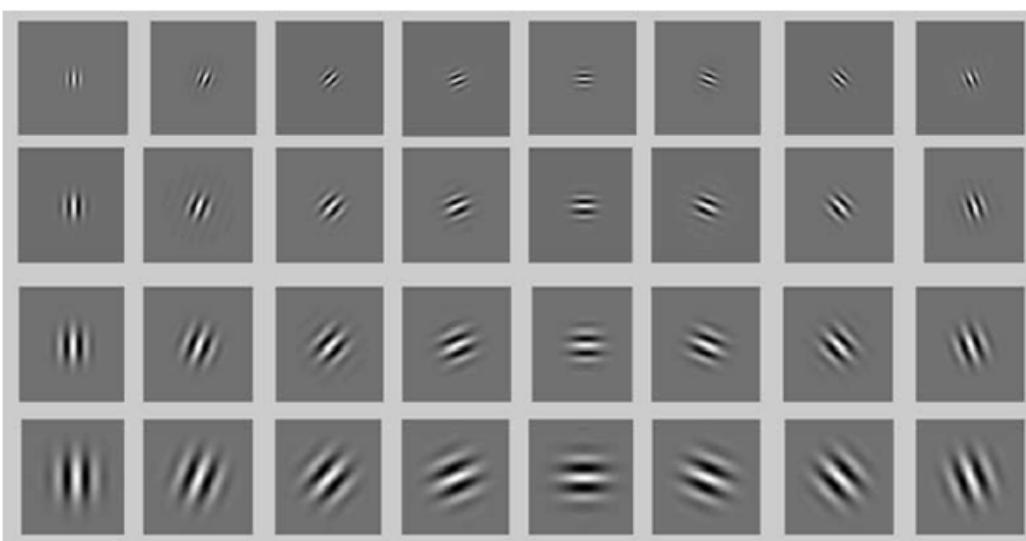


Figura 141. Parte real de los filtros Gabor con cuatro diferentes frecuencias y ocho orientaciones.
Fuente: https://www.researchgate.net/figure/Gabor-wavelets-filter-bank-which-has-been-used-for-the-active-basis-model_fig2_262781727.

El concepto esencial detrás de los filtros de Gabor es similar al de las Wavelets y al de Fourier: descomponer una imagen o una señal en determinadas componentes y, con ello, poder definir unas características que proporcionen información adicional de la imagen para su mejor entendimiento. Si una imagen se convoluciona con todas y cada una de las funciones arriba representadas, se detectarán no solo bordes y contornos, sino también diferentes texturas o patrones.

Las aplicaciones más comunes de los filtros de Gabor, más allá de la detección de texturas, están relacionadas con la **biometría** o, lo que es lo mismo, la verificación e identificación de usuarios con base en patrones biométricos como la huella o el iris. Concretamente, son huella e iris los que más emplean filtros de Gabor para el correcto funcionamiento.

A continuación se presenta un ejemplo de cómo se aplican los filtros de Gabor (y el resultado obtenido) en un caso de uso real de biometría de iris. El objetivo es reducir la complejidad de textura del iris a un patrón que pueda ser comparado con otros patrones mediante una distancia sencilla (como puede ser la distancia de Hamming). De ahí que el resultado, después de obtener el filtro de Gabor, parezca un código QR.

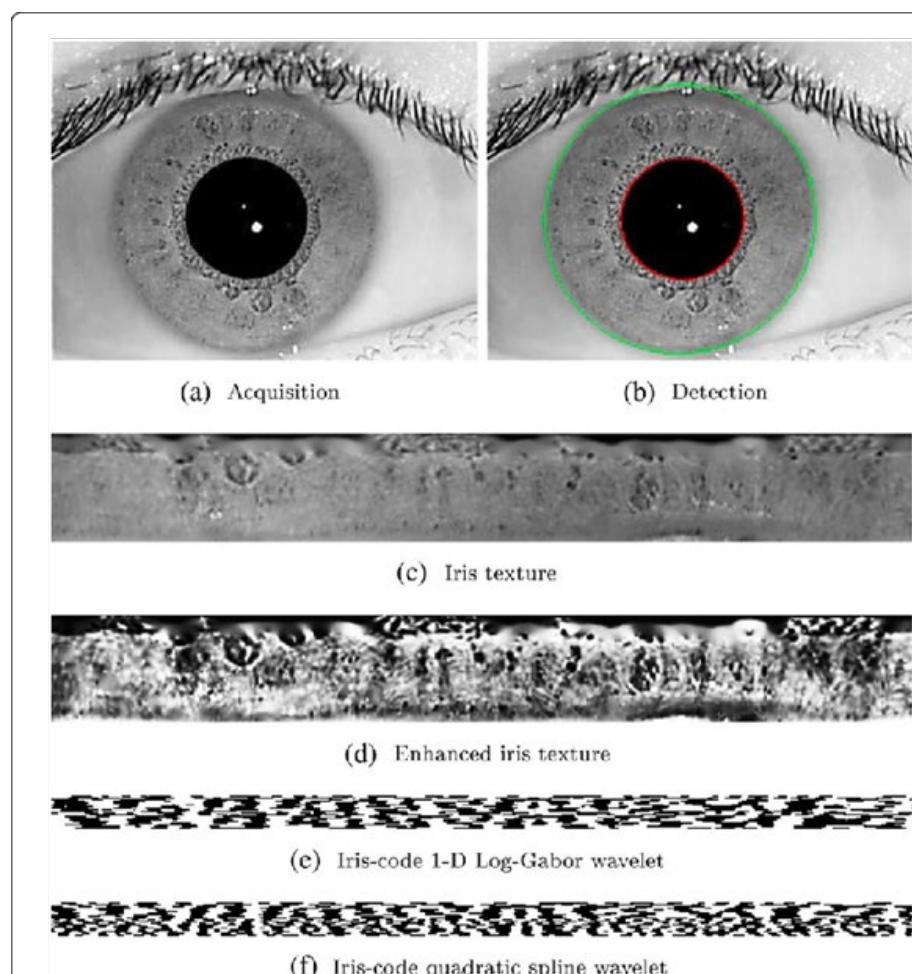


Figura 142. Funcionamiento de un sistema biométrico basado en iris con filtros de Gabor.
Fuente: https://www.researchgate.net/publication/309659737_Unlinkable_improved_multibiometric_iris_fuzzy_vault

En esta imagen vemos:

- ▶ Imagen original (**a**).
- ▶ Imagen segmentada, detectando la región de iris a procesar (**b**).
- ▶ La textura del iris convertida a coordenadas rectangulares (**c**).
- ▶ Textura con alto contraste (**d**).
- ▶ Código de iris después de aplicar filtro de Gabor (**e**).
- ▶ Código de iris después de aplicar una wavelet cuadrática (**f**).

13.5. Transformada SIFT



Accede al vídeo «Transformada SIFT» a través del aula virtual

Esta transformada es de las que tiene más utilidad hoy en día y que resuelve problemas muy complejos en procesamiento de imágenes. Uno de ellos es el de encontrar objetos parcialmente ocultos en imágenes. Vamos a verlo con el siguiente ejemplo:



Figura 143. Localización de objetos parcialmente ocultos en una imagen.

Fuente: <http://eric-yuan.me/sift/>.

A la izquierda de esta imagen se pueden ver dos objetos: una rana y una máquina de tren. En la imagen de la derecha, hay una rana (oculta detrás de un objeto negro), así como las dos máquinas de tren (una de ellas incluso en vertical).

Si intentamos responder a la pregunta: «¿Están los objetos tren y rana dentro de la imagen?», no podríamos dar respuesta con ninguno de los métodos presentados hasta este tema. Con las Wavelets y filtros de Gabor podríamos habernos acercado a detectar alguna característica común, pero en ningún momento con la precisión que el algoritmo SIFT lo realiza. Las iniciales del algoritmo significan *Scale Invariant Feature Transform* y proporciona el siguiente resultado. En ella se aprecian varios recuadros y colores:

- ▶ Cada **color** corresponde con un objeto encontrado: amarillo y verde para las máquinas de tren, rojo para la rana.
- ▶ El **recuadro más grande** indica dónde está el objeto en su totalidad.
- ▶ Los recuadros más pequeños indican las **características** que se han detectado tanto en las imágenes de los objetos originales como en la imagen donde se encuentran los objetos ocultos.
- ▶ Cada uno de estos recuadros posee un tamaño y una orientación, ambos parámetros son devueltos por el algoritmo SIFT.



Figura 144. Resultado del algoritmo SIFT.

Fuente: <http://eric-yuan.me/sift/>.

En esta imagen se ve en diferentes colores y con un gran recuadro la ubicación de los objetos buscados. Además, se presentan también, con recuadros más pequeños, las características detectadas.

Veamos ahora el **funcionamiento de este algoritmo**.

Para obtener un conjunto de descriptores SIFT de una imagen es necesario, por un lado, obtener los puntos característicos y después, para cada punto de interés, calcular su vector descriptor a partir de la información de los píxeles que lo rodean. SIFT fue propuesto para imágenes en escala de grises, por lo que el vector de características de 128 elementos que define cada píxel contiene información sobre cómo se distribuyen los niveles de intensidad alrededor de cada punto de interés previamente obtenido.

Por lo tanto, el algoritmo consta de dos partes claramente diferenciadas:

- ▶ Obtención de los puntos característicos.
- ▶ Descripción de la región alrededor de cada punto de interés.

La **obtención de los puntos característicos** o puntos de interés, a los que habitualmente se llama en inglés *keypoints*, consiste en detectar aquellas regiones de la imagen en las que se producen diferencias de gradiente significativas a ambos lados de dicho punto. Si el método solamente hiciera eso, se podría pensar que esta etapa se realizaría utilizando un detector de esquinas. SIFT propone no detectar únicamente esquinas, sino *blobs*, y hacerlo de manera que esa detección sea consistente cuando el punto característico aparezca a diferentes escalas.

La diferencia entre una esquina (*corner*) y un *blob* se explica a continuación: si nos fijamos en su definición, una **esquina** es un punto o un área pequeña en una imagen donde confluyen, al menos, dos bordes. Se define un ***blob*** como una región de una imagen que se caracteriza porque algunas de sus propiedades se mantienen aproximadamente constantes. Dentro del contexto de la detección de puntos característicos (*keypoints*), la propiedad que se suele considerar constante en un *blob* es la **similitud en su nivel de gris**, típicamente medida a partir de la variación del gradiente en esa región a lo largo de diferentes direcciones.

En resumen, buscamos puntos que pueden ser pequeñas regiones con un nivel de intensidad estable y alrededor de las cuales se produce una variación significativa del gradiente en más de una dirección.

Para obtener el **descriptor de cada punto característico**, se calcula un histograma de direcciones del gradiente local alrededor del punto de interés. El descriptor que se obtiene se convierte en invariante a la escala al normalizar el tamaño del vecindario local al punto de interés en función de ella. Además, es invariante a la rotación porque se determina la orientación dominante de los vectores del gradiente en el vecindario del punto característico, y se aplica dicha información para orientar la rejilla utilizada para calcular el histograma.

En resumen, los pasos a seguir son:

8. Detección de extremos en el espacio-escala: se buscan puntos de interés en toda la imagen y en todas las escalas consideradas utilizando una diferencia de gaussianas.
9. Localización precisa de puntos característicos: para cada uno de los puntos de interés anteriores se ajusta un modelo que permite determinar su localización y escala. Además, se seleccionan los puntos característicos, *keypoints*, eliminando los que están próximos a los bordes o tienen bajo contraste.
10. Asignación de la orientación: a cada punto característico se le asigna una o varias orientaciones en función de las direcciones del gradiente local. Esta orientación, conjuntamente con la ubicación y la escala calculadas anteriormente, permite que el descriptor sea invariante a estas tres situaciones.
11. Descripción del punto característico: alrededor de cada punto característico se miden los gradientes locales de la imagen y se utiliza su histograma para obtener una representación de esa región que es robusta a cambios significativos en la iluminación y a pequeñas distorsiones en la forma.



Figura 145. Detección con SIFT de objetos complejos (y sin formas definidas) en una imagen.

Fuente: <http://slideplayer.com/slide/5282534/>.



Accede a los ejercicios de autoevaluación a través del aula virtual

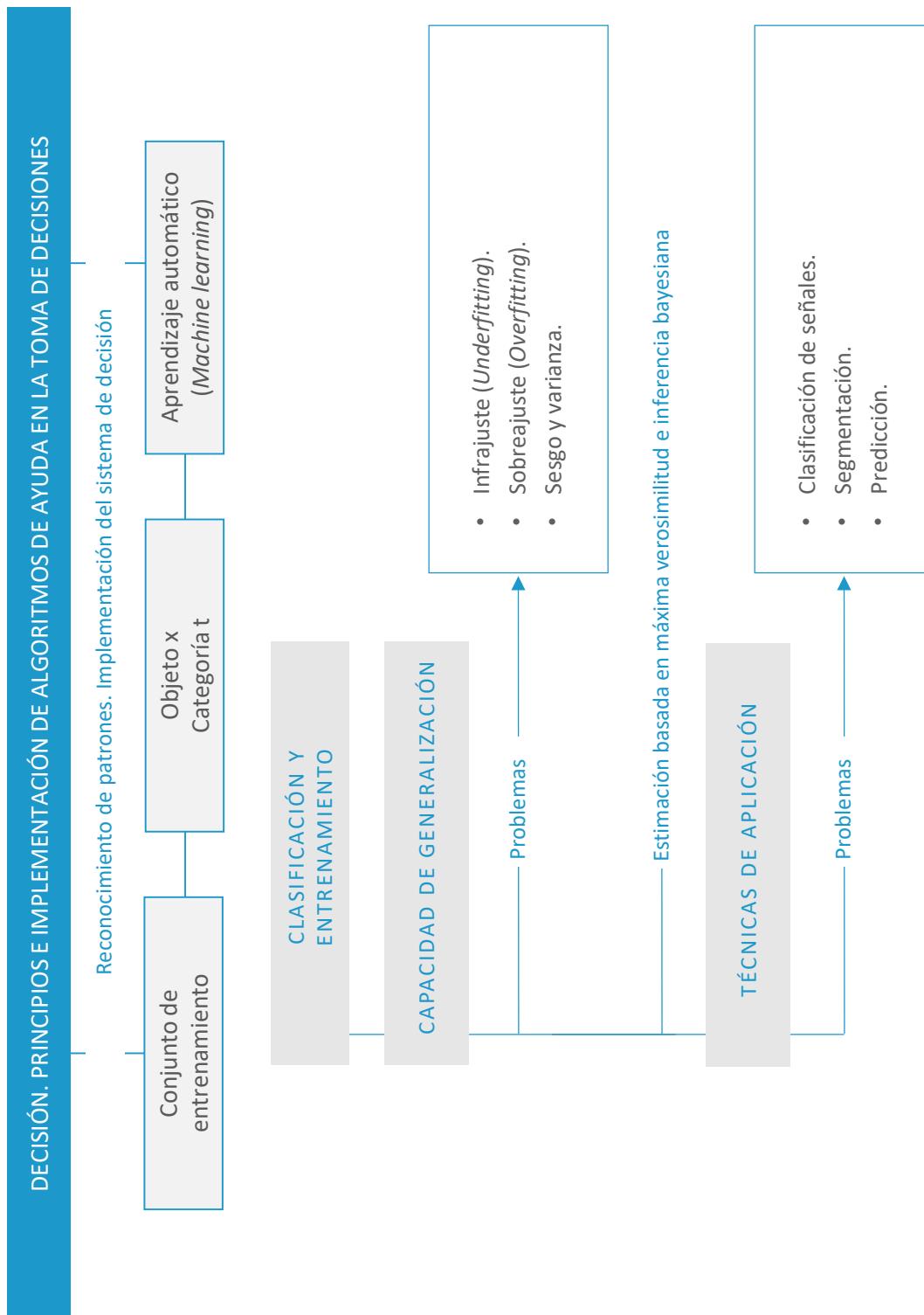


Accede al vídeo «Resumen del tema» a través del aula virtual

Percepción Computacional

Decisión. Principios e implementación de algoritmos de ayuda en la toma de decisiones

Esquema. Tema 14



Ideas clave. Tema 14

14.1. ¿Cómo estudiar este tema?



Accede al vídeo «Introducción. Decisión. Principios e implementación de algoritmos de ayuda en la toma de decisiones» a través del aula virtual

Para estudiar este tema deberás leer con atención las ideas clave que se desarrollan a continuación.

En la mayoría de las aplicaciones, el procesado completo de una señal requiere la interpretación automática de esta como etapa final. Por ejemplo, en el caso de una señal biomédica, el objetivo final irá dirigido a determinar si dicha señal denota la existencia o no de una patología. Podemos, asimismo, pensar en una aplicación basada en el manejo de señales de voz que nos permita identificar si la palabra o frase pronunciada forma parte de nuestro diccionario particular, que habrá sido definido previamente. En imágenes es habitual tener que identificar determinados objetos, por ejemplo, conocer si corresponde o no a una persona; por lo que tras reducir el ruido de la imagen, incrementar el contraste de esta o llevar a cabo una segmentación, será necesario clasificar los elementos resultantes.

Todas estas aplicaciones precisan como elemento culminante una etapa de **reconocimiento de patrones**. Esta será específica de la aplicación en cuestión y, por tanto, su construcción se adaptará a la tarea concreta que deberá llevar a cabo. Así, el sistema de reconocimiento de patrones para la identificación de palabras es diferente al empleado para hallar automáticamente una persona en una imagen. Sin embargo, la metodología para su construcción es común en ambos casos.

Dicha **metodología** se basa en la disponibilidad de un conjunto amplio de ejemplos del problema a resolver. Por ejemplo, a fin de implementar un sistema de reconocimiento de voz, este conjunto deberá incluir tanto palabras que no están en nuestro diccionario como las que sí lo están. O en el caso de las imágenes, instantáneas de figuras que no se corresponden con una persona frente a las que sí lo son. A partir de estos ejemplos, es posible inferir qué propiedades diferencian unos casos de otros, resultando en un **modelo**.

En este punto, cabe destacar que cada una de estas tareas es realizada por nuestro cerebro, que es capaz de identificar una palabra conocida o si el elemento de una imagen es una persona. En él se llevan a cabo las etapas previas de tratamiento de señal que se han visto en la asignatura: captura, eliminación de ruido, segmentación y extracción de características a partir de la fuente de información original.

La etapa inmediatamente anterior al sistema de reconocimiento de patrones es la **extracción de características** y sobre la que hemos tratado en temas previos. En esta etapa, una vez que tenemos identificado el objeto a interpretar (una señal o segmento de la misma, una imagen, una textura, una silueta, etc.), se busca sintetizar la información de dicho objeto en un conjunto de variables. Este ejercicio de síntesis conlleva la reducción de la dimensionalidad de los datos de entrada, de una imagen o señal formados por un número determinado de píxeles o puntos, pasamos a un vector de características de longitud notablemente menor.

La motivación principal de la etapa de extracción de características es optimizar, mediante esta síntesis, la implementación del sistema de reconocimiento de patrones posterior. Para ello, además de permitir trabajar en un espacio de dimensión menor, es de vital importancia que las características extraídas definan propiedades significativas de los objetos en el contexto del problema a resolver. Por ejemplo, el valor máximo de una señal de voz puede no tener relevancia si lo que se pretende identificar es si ha sido emitida por un hombre o una mujer, pero sí si se pretende inferir si el emisor de la señal muestra un estado de tranquilidad o no. Es decir, la etapa de extracción de características ha de minimizar la pérdida de

información resultante de la reducción de dimensionalidad llevada a cabo, reteniendo aquellas propiedades más relevantes de la fuente de información.

La siguiente imagen recoge, en forma de esquema, los principales conceptos de este tema.

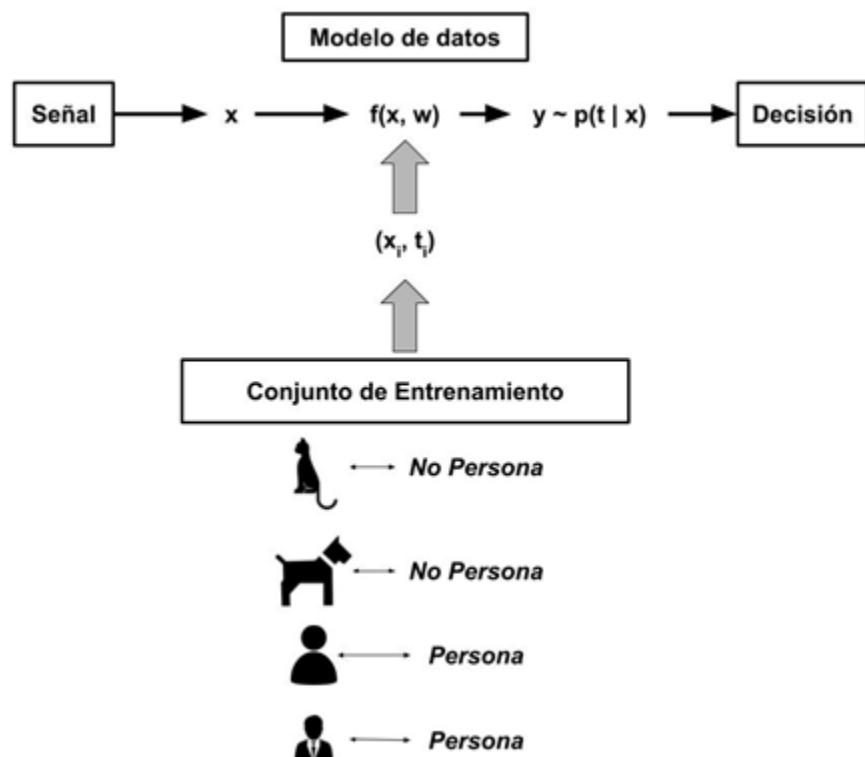


Figura 146. Metodología para la implementación de un sistema de decisión.

14.2. Clasificación y reconocimiento de patrones



Accede al vídeo «Clasificación y reconocimiento de patrones» a través del aula virtual

El sistema de reconocimiento de patrones es el **modelo** resultante de inferir las diferencias existentes entre varias categorías de objetos. Dicho modelo, como se expresaba al comienzo de este tema, se obtiene a partir de un conjunto de ejemplos de todas las categorías involucradas en nuestra aplicación. Finalmente, nuestro modelo se deriva de un proceso de optimización guiado por un

algoritmo y una métrica de evaluación a maximizar/minimizar. Este algoritmo suele ser de carácter iterativo, de manera que se va dando forma al modelo mediante la observación repetida de los diferentes ejemplos.

Como se ha comentado, la implementación del sistema de decisión se realiza a partir de:

- ▶ Un conjunto de ejemplos D , referido comúnmente como **conjunto de entrenamiento**.
- ▶ Cada uno de los ejemplos en este conjunto viene dado por el par formado por:
 - Un objeto x .
 - Y la categoría a la que pertenece t .
- ▶ Este escenario se conoce en el ámbito del aprendizaje automático (*machine learning*) como aprendizaje supervisado, ya que se sabe *a priori* el valor de la variable objetivo para los datos de entrenamiento disponibles.

El conjunto D puede representarse, por tanto, de la siguiente forma:

$$D = \{(x_i, t_i)\}_{i=1, \dots, M}$$

Donde M es el número total de ejemplos en el conjunto de entrenamiento.

La pregunta es cómo tomar decisiones, minimizando el error en la decisión, ante objetos x que no están en D y para los que desconocemos la categoría a la que pertenecen.

Estadísticamente, la herramienta que proporciona una descripción completa de las muestras en D es la función de densidad de probabilidad conjunta $p(x, t)$, que puede expresarse como:

$$p(x, t) = p(t|x)p(x)$$

Donde:

- ▶ $p(t|x)$ es la función de densidad de probabilidad de la variable t dado x .
- ▶ $p(x)$ es la función de densidad de probabilidad de x .

En este punto, puede observarse que $p(t|x)$ es la función a modelar a fin de hacer predicciones sobre t a partir de un valor observado de x . Es decir, esta será la **función a modelar por nuestro sistema de decisión**.

Clasificación

Cuando la variable t es categórica, el problema se corresponde con una tarea de clasificación.

Matemáticamente consiste en determinar, para una muestra x , a cuál de las C categorías posibles (C_1, C_2, \dots, C_k) pertenece. El objeto de entrada queda descrito por un conjunto de características o descriptores cuantitativos, por lo que x será un vector de la forma $x = [x_1, x_2, \dots, x_n]$, que representa un punto en el espacio de entrada n -dimensional.

El **clasificador** viene dado por la función f que mapea el espacio de entrada \Re^n en la variable categórica t :

$$f: \Re^n \rightarrow t \in \{C_1, C_2, \dots, C_k\}$$

La salida del clasificador puede expresarse como $y = f(x, w)$, donde se hace explícita la dependencia de la función de mapeo f con el vector de entrada x y un conjunto de parámetros adaptativos w .

La aproximación estadística al problema de clasificación, es decir, para la estimación de la función f , asume que las muestras de x que corresponden a la categoría C_i han sido generadas de acuerdo a la función de densidad de probabilidad $p(x|C_i)$.

En este escenario, la **regla de decisión de Bayes** determina cómo ha de llevarse a cabo la decisión a fin de minimizar el riesgo en la predicción. De acuerdo a esta regla, un objeto x debe asignarse a la categoría C_i para la que el riesgo condicionado $R(C_i|x)$ es mínimo.

Este riesgo se define como sigue:

$$R(C_i|x) = \sum_{j=1}^k L(C_i, C_j) p(C_j|x)$$

Donde:

- ▶ Donde la función $L(C_i, C_j)$ cuantifica el **coste del error** cuando se decide C_i siendo C_j la verdadera categoría a la que pertenece la muestra x .
- ▶ $p(C_j|x)$ es la probabilidad *a posteriori* de la categoría C_j . Se conoce también como **función de pérdida**. Normalmente, se emplea una función de pérdida 0/1 para la implementación del clasificador, que toma un valor:
 - 0 si $i = j$.
 - 1 si $i \neq j$.

El resultado de utilizar esta función de pérdidas es la denominada como **regla del máximo a posteriori** (MAP, del inglés *maximum a posteriori*). Supone la forma más conocida de la regla de decisión de Bayes. Una muestra x se asigna a la categoría C_i si se cumple la siguiente condición:

$$p(C_i|x) > p(C_j|x), i \neq j$$

Es decir, la decisión será tomar aquella categoría más probable una vez que el valor de x es conocido.

La estrategia descrita al problema de reconocimiento de patrones y, concretamente, a la clasificación representa una aproximación estadística al mismo. En ella, se pretende obtener una estimación de la función $p(C_i|t)$ a fin de poder tomar

decisiones de acuerdo a la regla de Bayes. Muchos de los **algoritmos para la implementación de modelos de datos** se basan en esta aproximación, es el caso de la regresión logística o las redes neuronales MLP y RBF. Asimismo, en otros modelos estadísticos se estima la función de densidad condicionada $p(x|C_i)$ para obtener $p(C_i|x)$ mediante el teorema de Bayes:

$$p(C_i|x) = p(x|C_i)p(C_i)/p(x)$$

Algunos algoritmos que siguen esta estrategia son K-vecinos próximos, las redes neuronales probabilísticas, los clasificadores gaussianos (discriminante lineal o cuadrático) o el Naïve Bayes.

Fuera de este marco puramente estadístico, podemos encontrar métodos de clasificación como los árboles de decisión o las máquinas de vectores soporte. En ambos casos, no se persigue una estimación directa de las funciones de densidad de probabilidad mencionadas previamente.

Entrenamiento

En principio, la forma de la función $f(x, w)$ es desconocida y se requiere de un conjunto de ejemplos D para su estimación. Por tanto, la función f vendrá dada por una expresión matemática que contiene un número determinado de parámetros ajustables w . Esta expresión está determinada por el conjunto de ejemplos entrada-salida en D mediante el proceso de aprendizaje o entrenamiento. Por este motivo, el conjunto de ejemplos de muestras de entrada y su correspondiente categoría se denomina conjunto de entrenamiento.

El proceso de entrenamiento conlleva la minimización de una función de error que cuantifica la distancia o grado de disparidad entre el valor de salida de nuestro clasificador (y) y su valor objetivo correspondiente (t) para un objeto dado (x). En este caso, se habla de entrenamiento supervisado, ya que cada muestra x está asociada a un valor objetivo t .

La finalidad del proceso de entrenamiento es construir un modelo estadístico del proceso que da lugar a las muestras x . Es decir, no se busca una función $f(x, w)$ cuyo valor coincida exactamente con la variable t para cada par (x, t) , sino que dicha función se aproxime a $p(t|x)$. Una estimación plausible de esta función de densidad de probabilidad resultará en un modelo con una gran **capacidad de generalización**; este concepto se refiere a la capacidad del modelo de producir respuestas precisas ante muestras de entrada que no se han procesado previamente. Fundamentalmente, esta capacidad de generalización depende de tres factores:

El tamaño del conjunto de entrenamiento y su representatividad del problema.

Será necesario contar con un número elevado de ejemplos diferentes entre sí para lograr una buena descripción del problema. Cada ejemplo en D es, al fin y al cabo, una muestra de la función $p(x, t)$, por lo que su estimación será más precisa si se dispone de un mayor número de muestras.

La capacidad de adaptación (flexibilidad) del modelo. Cuanto mayor sea el número de funciones $f(x, w)$ que pueden definirse mediante el ajuste de w , mayor es la posibilidad de encontrar una solución que minimice la distancia respecto de la función objetivo $p(t|x)$, resultando en una mejor capacidad de generalización. Este número de funciones a las que el modelo puede dar lugar viene dado por la cantidad de parámetros w ajustables en el mismo. Así, cuantos más parámetros se deban optimizar durante el entrenamiento, el modelo tendrá una capacidad de adaptación mayor a las muestras de $p(t|x)$ recogidas en D .

La complejidad del problema de clasificación. Un problema de clasificación de mayor dificultad para modelar vendrá dado por una función $p(t|x)$ más compleja. La siguiente imagen (figura 2) muestra un ejemplo en el que se aprecian dos conjuntos de muestras (x, t) :

- ▶ D1 (izquierda) y D2 (derecha) describen dos problemas de clasificación diferentes.

- ▶ Como se aprecia, el problema definido por D2 es de mayor dificultad, pues requiere una frontera de decisión en el espacio bidimensional de entrada de la forma:

$$(x_1 - k_1)^2 + (x_2 - k_2)^2 = k_3$$

Donde k_1, k_2 y k_3 son constantes y los objetos x de entrada quedan definidos por los valores de las variables x_1 y x_2 .

- ▶ Mientras tanto, el problema descrito por el conjunto de muestras D1 requiere una frontera de decisión de la siguiente forma:

$$x_1 = k_1 x_2 + k_2$$

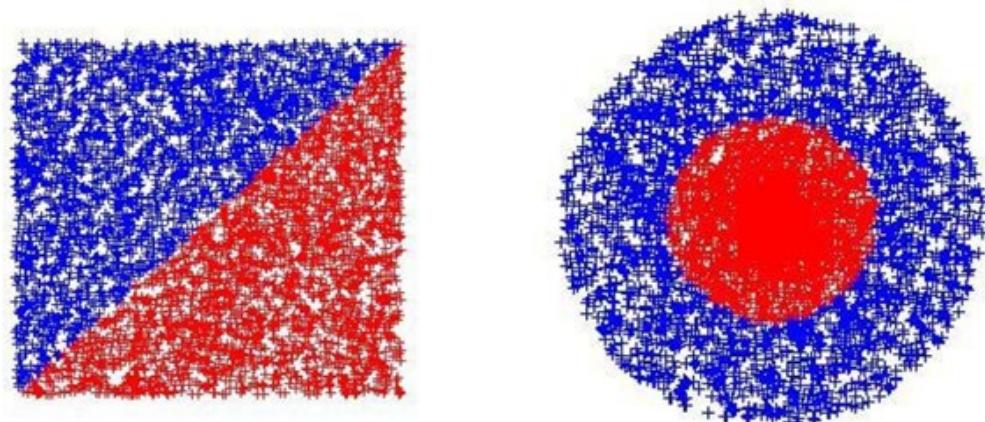


Figura 147. Ejemplo de dos problemas de clasificación diferentes.

Como comentábamos, los datos de la imagen de la derecha reflejan una mayor complejidad de la función $p(t|x)$ que los que generó respecto a los observados en la imagen de la izquierda.

Mientras la dificultad del problema de clasificación resulta un factor fuera de control para el usuario, los dos primeros factores están estrechamente relacionados. Un **modelo rígido o poco flexible**, con un número reducido de parámetros ajustables durante el entrenamiento, puede no ajustarse a las muestras de la función a aproximar. Lo contrario sucede con un modelo excesivamente **flexible y maleable**, pues será capaz de ajustar perfectamente los pares en D .

En ambas situaciones, la capacidad de generalización del modelo resultante será pobre.

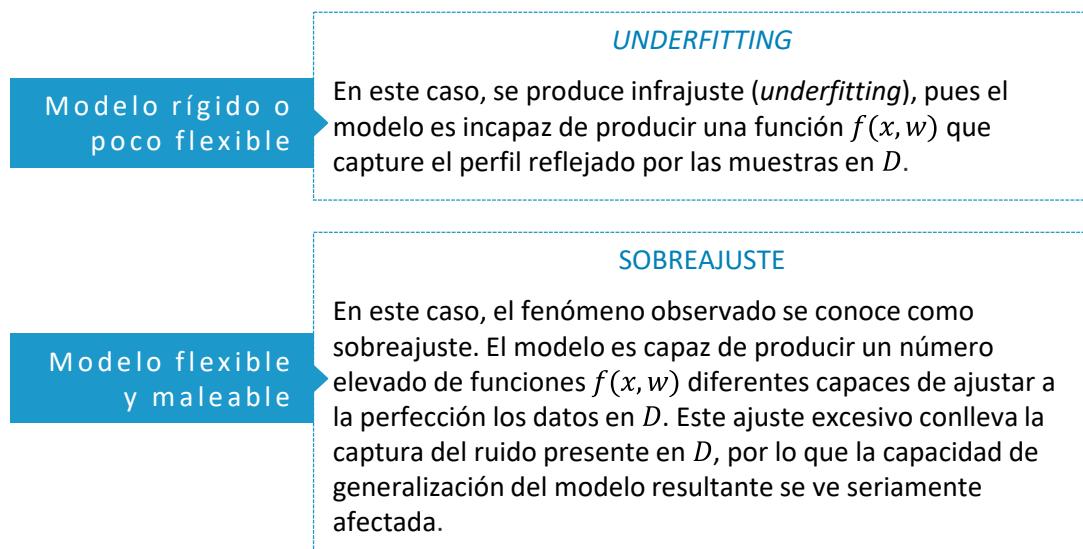


Figura 148. Problemas de infrajuste o sobreajuste en la capacidad de generalización del modelo.

Existen diferentes estrategias para paliar ambos efectos:

- ▶ En el caso del *underfitting*, la solución más efectiva es emplear modelos con un mayor grado de flexibilidad, de forma que sean capaces de generar funciones $f(x, w)$ de un perfil más complejo.
- ▶ En el caso del *overfitting* (sobreajuste), puede prevenirse mediante la utilización de un conjunto de entrenamiento más amplio. Sin embargo, esto no es posible en la mayoría de los escenarios reales. Por tanto, se suele optar por otras soluciones más fácilmente alcanzables como la reducción de la dimensión del espacio de entrada o la combinación de modelos diferentes entre sí, técnicas conocidas como *ensembles* de las que *bagging* y *gradient boosting* son las más comúnmente empleadas.

Sesgo y varianza

Los conceptos de infrajuste y sobreajuste mencionados se encuentran estrechamente relacionados con el sesgo y la varianza de un modelo de datos. Estos representan dos componentes distintas del error en el que incurre un modelo de datos. Antes de definirlas, la siguiente imagen ilustra la idea detrás de ambas componentes.

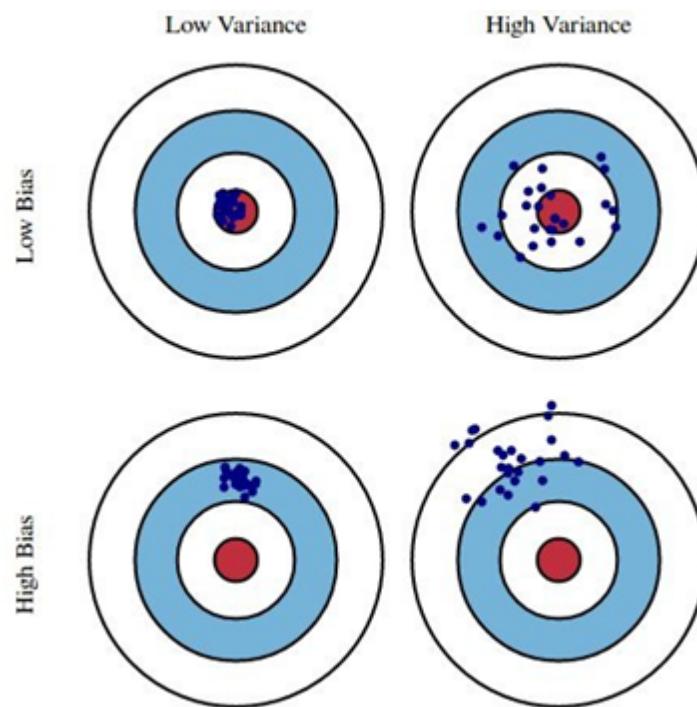


Figura 149. Ilustración de los conceptos de sesgo y varianza como fuentes de error en un modelo de datos.

Fuente: <https://www.kdnuggets.com/2016/08/bias-variance-tradeoff-overview.html>

- ▶ Si atendemos en primer lugar al sesgo, vemos que, cuando este es elevado en un modelo, el error cometido por este se debe a una tendencia clara en la predicción. En la imagen anterior, el sesgo del modelo se refleja en una tendencia a estimar por encima del objetivo. Esta tendencia desaparece cuando el modelo tiene un sesgo bajo.
- ▶ Por otro lado, la varianza se manifiesta en una distribución más amplia de las predicciones generadas por el modelo a pesar de que el objetivo, como en el ejemplo, no cambia. Así, cuando la varianza del modelo es elevada, hay mayor grado de incertidumbre sobre la salida de este.

Por tanto, tras este primer análisis, podemos definir sesgo y varianza:

El **sesgo** obedece a la componente sistemática del error. Dada una entrada x , se computa como la diferencia entre el valor medio de las predicciones de los modelos derivados de diferentes conjuntos de entrenamiento D y el valor objetivo t asociado a x .

$$Bias^2 = [E_D\{y(x)\} - t|x|^2]$$

El error asociado a un sesgo elevado es propio de modelos simples, con poca flexibilidad o capacidad de ajuste. Estos modelos son incapaces de aproximar funciones complejas y tienden a producir infraestimación o sobreestimación. Esta situación refleja infajuste de los datos en D .

La **varianza** es la componente del error derivada de la dependencia del modelo resultante $f(x, w)$ con el conjunto de entrenamiento D . Es decir, este término refleja que las funciones $f(x, w)$ derivadas del proceso de entrenamiento difieren entre sí cuando los conjuntos de entrenamiento empleados son distintos.

$$Varianza = E_D[\{y(x) - E_D\{y(x)\}\}]^2$$

El error derivado de la varianza es característico de modelos con una gran capacidad de adaptación a los datos. Potencialmente, estos modelos son capaces de definir diferentes funciones que ajustan los datos en D , capturando incluso el ruido presente en este conjunto en forma de anomalías. Se produce, por tanto, el sobreajuste de los datos de entrenamiento. Un mayor número de ejemplos en D contribuiría a limitar el espacio de funciones que el modelo sería capaz de definir para el ajuste de D y, por tanto, se reduciría su varianza.

Estimación basada en máxima verosimilitud e inferencia bayesiana

Tras el análisis de las fuentes de error en un modelo de datos, retomamos el proceso de entrenamiento. Como se ha indicado, en el mismo se tiene como objetivo modelar el generador de los datos en D , que queda completamente descrito por la función $p(x, t)$.

A fin de estimar el valor de t para un x dado, la función e interés será $p(t|x)$. Para incluir explícitamente la dependencia del modelo en la estimación, puede expresarse $p(t|x, w)$, donde w refleja el conjunto de parámetros del modelo ajustables durante el proceso de entrenamiento. Estadísticamente, esto puede obedecer a dos principios: regla de máxima verosimilitud o inferencia bayesiana.

El principio de máxima verosimilitud ajusta los pesos w de tal forma que, dado estos, se maximice la probabilidad de observar los datos en D .

Si asumimos independencia entre las muestras en D , la probabilidad de observar este conjunto de datos según el modelo definido por w puede expresarse de la siguiente forma:

$$p(D|w) = \prod_{i=1}^M p(x_i, t_i|w) = \prod_{i=1}^M p(t_i|x_i, w) \equiv L_D(w)$$

Donde el término $L(w)$ se identifica como la probabilidad asociada al conjunto D y depende del modelo w .

Los pesos de w se escogen de forma que se maximice esta función, es decir, se elige el modelo que hace más probable dar lugar al conjunto de datos. En la práctica, se lleva a cabo la operación equivalente que consiste en minimizar el logaritmo negativo de L_D . La **función de error** resultante se obtiene como:

$$E_D = -\log L_D = -\sum_{i=1}^M \log[p(t_i|x_i, w)] - \sum_{i=1}^M \log[p(x_i)]$$

Dado que el segundo término de E_D no depende de w , la función de error resultante viene dada por el primer sumando de la ecuación. En el caso de un problema de clasificación con dos categorías diferentes $\{C1, C2\}$, de forma que $t = 1$ para C1 y $t = 0$ para C2, la **función aproximada por la salida del modelo** puede escribirse de la siguiente forma:

$$p(t|x, w) = y^t(x, w)[1 - y(x, w)]^{1-t}$$

Sustituyendo esta expresión en la ecuación del error de acuerdo al principio de máxima verosimilitud, obtenemos la siguiente ecuación que se conoce como **función de entropía cruzada**:

$$E_D = \sum_{i=1}^M \{t_i \log(y_i) + (1 - t_i) \log(1 - y_i)\}$$

Sin embargo, el modelo que asegura una mayor probabilidad de observar D mediante la minimización del error anterior, no asegura la mejor capacidad de generalización. Este hecho se ha visto previamente con el problema de sobreajuste, común en modelos con un alto grado de varianza. La aproximación bayesiana para el ajuste de w considera este conjunto de parámetros como una variable aleatoria, de forma que esté caracterizada por una función de densidad de probabilidad $p(w)$. Una vez que el conjunto D es observado, la probabilidad asociada a w cambia y puede ser obtenida a través del **teorema de Bayes**:

$$p(w|D) = \frac{p(D|w)p(w)}{p(D)}$$

La estimación de $p(w|D)$ permite realizar predicciones teniendo en cuenta todos los posibles valores de la variable w , es decir, todos los modelos potencialmente obtenibles a partir de D tras la observación de este conjunto. Así, el **valor de salida del modelo** vendría dado por la siguiente expresión:

$$p(t|x, D) = \int p(t|x, w) p(w|D) dw$$

Donde $p(t|x, w)$ sería el valor de salida de un modelo w concreto para la entrada x . Por tanto, el valor final de salida es un promediado de las salidas de todos los modelos w observables, ponderados por la probabilidad de cada uno de ellos tras la observación de D , que viene dada por el término $p(w|D)$.

14.3. Aplicación de técnicas *machine learning* al procesado de señales



Accede al vídeo «Aplicación de técnicas de *machine learning* al procesado de señales» a través del aula virtual

El punto anterior recoge los principios elementales en la implementación de un sistema de decisión o reconocimiento de patrones. Estos principios se han presentado en un marco estadístico que permite la utilización de herramientas matemáticas para la obtención de una solución al problema. A continuación, se indica cómo las técnicas derivadas de este marco pueden aplicarse a señales para la implementación de señales de decisión.

Comenzaremos con los problemas más habituales en el ámbito del tratamiento de señales que precisan de la implementación de un sistema de decisión.

Clasificación de señales

Una de las aplicaciones más comunes de los sistemas de decisión se encuentra en la identificación automática de la categoría de una señal, tanto si pensamos en una función unidimensional de la variable tiempo como en una imagen. En este escenario, puede que nos encontremos en una de las siguientes situaciones:

- ▶ El problema de clasificación no es sencillo de resolver por el ser humano mediante la observación de las señales. En este caso, precisamos de herramientas avanzadas que capturen información de la señal que nosotros no somos capaces de extraer.
- ▶ El ser humano es capaz de llevar a cabo la clasificación que se persigue, pero se trata de una tarea que precisa de elevados recursos (materiales, tiempo, profesionales involucrados...). Por tanto, en esta situación el objetivo es **automatizar el procedimiento**.

El problema de clasificación de señales debe acotarse a un escenario concreto, previamente definido, a fin de obtener un sistema de decisión efectivo. Por ejemplo, en el caso de una imagen, el propósito de la tarea de clasificación se ceñiría a identificar si la figura que aparece en ella se corresponde o no con una persona. La interpretación automática de qué ser vivo o cosa es dicha figura, de entre todas las posibilidades existentes, es un problema excesivamente abierto y complejo de abordar. Otro caso típico en sistemas de ayuda a la decisión basado en imágenes se tiene en un contexto médico. Así, podemos pensar en la implementación de un clasificador que permita identificar automáticamente tejido pulmonar dañado frente a tejido sano sin lesiones relevantes.

De forma similar, en el caso de las **señales en el dominio del tiempo**, es habitual enfrentar problemas en los que el propósito es separar registros patológicos de aquellos correspondientes a un paciente sano. Suele ser común el procesado de señales como el electrocardiograma (ECG), el electroencefalograma (EEG) o la saturación de oxígeno (SaO).

Por último, las **herramientas de reconocimiento de voz** implementan también clasificadores de señales. Una herramienta relativamente sencilla sería un sistema capaz de distinguir si la voz percibida se corresponde con un hombre o una mujer. Las máquinas para la interpretación automática del habla poseen mayor complejidad. El sistema dispone de un léxico o conjunto de palabras con las que ha sido entrenado

y, a partir de la señal de audio capturada, se busca cada una de las palabras en este léxico para poder interpretar el mensaje completo.

Segmentación

En el contexto del procesado de imágenes, los sistemas de decisión pueden emplearse para la identificación de diferentes regiones y el cálculo de las fronteras entre estas. Para ellos, es común emplear filtros espaciales que, centrados en un píxel y con un tamaño determinado, clasifican el píxel central asignándole una de las posibles categorías. Si estas han sido previamente definidas, estamos ante un **problema supervisado**, por lo que el clasificador será entrenado a partir de un conjunto de ejemplos previamente etiquetados, tal y como hemos supuesto desde el inicio en este tema.

El **problema no supervisado** en el que no disponemos de un conjunto de ejemplos y hemos de inferir las diferentes categorías existentes, requiere de la utilización de técnicas de *clustering*. La figura 5 muestra un ejemplo de un problema de clasificación de regiones en una imagen.



Figura 150. Segmentación de regiones en una imagen.

Fuente: <http://web.eecs.umich.edu/~silvio/teaching/lectures/Vision%20Research%20Lab%20-%20A%20Multiresolution%20Approach%20to%20Image%20Segmentation%20Based%20on%20EdgeFlow.htm>

Predicción

El trabajo con señales unidimensionales que son función del tiempo presenta la necesidad de estimar valores futuros de la señal a partir de muestras históricas. Como un ejemplo claro puede pensarse en cotizaciones bursátiles. El sistema de decisión consistiría en un clasificador capaz de capturar si la cotización del día siguiente será mayor o menor respecto a la del día anterior, basándose en muestras del histórico de la serie.

En los diferentes escenarios descritos, la **metodología para la implementación** del clasificador es común, tal y como se describía en el comienzo de este tema:

12.Identificación del objeto a clasificar: píxel de una imagen, región de una imagen, segmento de señal, etc.

13.Preprocesado de la fuente de información: eliminación de *outliers* y ruido, detección de bordes, selección de altas frecuencias, etc.

14.Caracterización del objeto: definición del conjunto de atributos que describen el objeto sobre el que se va a decidir; este proceso se trata de la construcción del vector x de características a partir de las técnicas descritas en temas anteriores.

15.Definición del problema a modelar: definición de la variable objetivo t : clasificación de una señal de ECG como patológica o no; identificación de una palabra en un diccionario; figura de una persona o no, etc.

16.Construcción de un conjunto de ejemplos: conjunto de entrenamiento D formado por pares (x_i, t_i) .

17.Entrenamiento y test: implementación del sistema de decisión y evaluación de su rendimiento sobre un conjunto de muestras no empleadas para el ajuste del modelo.



Accede a los ejercicios de autoevaluación a través del aula virtual

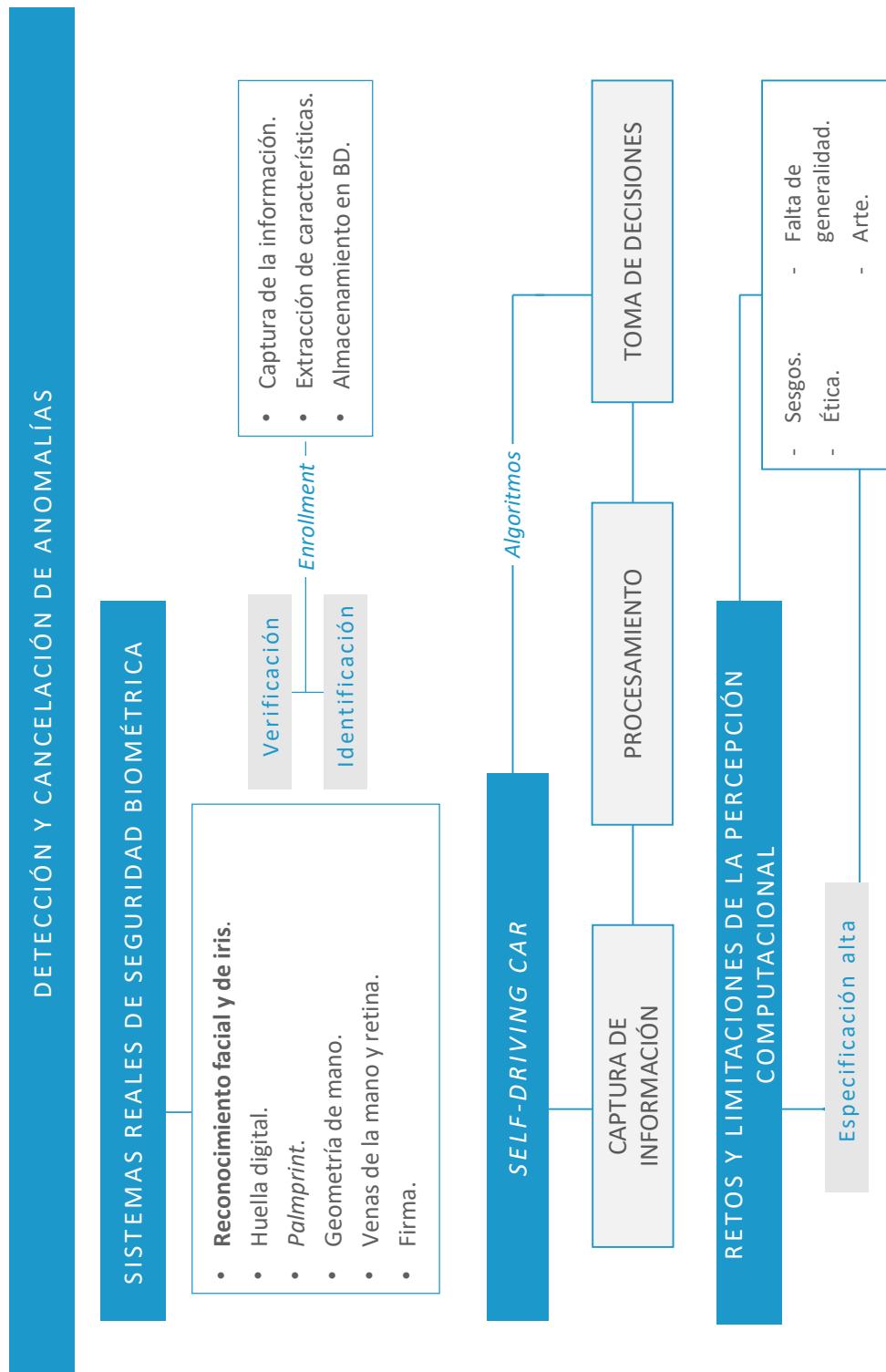


Accede al vídeo «Resumen del tema» a través del aula virtual

Percepción Computacional

Aplicaciones actuales del tratamiento de la señal

Esquema. Tema 15



Ideas clave. Tema 15

15.1. ¿Cómo estudiar este tema?



Accede al vídeo «Introducción. Aplicaciones actuales del tratamiento de la señal» a través del aula virtual

Para estudiar este tema deberás leer con atención las ideas clave que se desarrollan a continuación.

Como último tema de la asignatura, la idea es aplicar los conocimientos presentados a dos tecnologías actuales y entender qué algoritmos y soluciones son los más adecuados para cada problema. Además, se presenta un conjunto de retos y limitaciones que son el trabajo actual y futuro de las investigaciones asociadas a percepción computacional.

15.2. *Biometrics*



Accede al vídeo «*Biometrics*» a través del aula virtual

La biometría es el conjunto de técnicas de reconocimiento de patrones capaces de verificar e identificar a una persona de forma única. Actualmente, la seguridad y protección de la información reside en tres pilares:

- ▶ «Lo que sabes», también conocido como **contraseñas** en todas sus modalidades: usuario y contraseña corriente, sistemas de clave pública y privada, etc.

- ▶ «Lo que tienes», también conocido como ***token***, consistente en identificar o más bien validar a un usuario por tener un determinado objeto/texto con información que únicamente el usuario posee. Este *token* es comúnmente usado en los bancos de dos maneras:
 - Mediante el envío de un mensaje al teléfono móvil, siendo en este caso el móvil el propio *token*.
 - Tarjeta de coordenadas, donde el banco te indica un número de dos cifras y el usuario debe completar con 3-4 cifras las correspondientes a dichas dos cifras.
- ▶ «Lo que eres», la **biometría**: rasgos únicos extraídos de propiedades físicas del individuo como puede ser el iris, la retina o la propia huella digital.

La seguridad de la biometría reside en «lo que eres»: en características físicas y de comportamiento únicas para cada usuario o que, sin ser tan únicas, permiten al menos poder identificar a los usuarios lo suficiente como para distinguirlos del resto de usuarios. Es importante distinguir dos usos dentro de estos sistemas biométricos:

Verificación. En la verificación, el **usuario proporciona algún tipo de identificación** y el sistema biométrico comprueba que realmente es quien dice ser. Para ello, el sistema biométrico compara los rasgos extraídos con el patrón (también llamado *template*) biométrico de dicho usuario. Si dicha comparación no supera un umbral, entonces el usuario es rechazado. Por el contrario, si dicha comparación es satisfactoria, el sistema verificaría al individuo positivamente.

Identificación. Al contrario que en la verificación, el usuario no proporciona **ningún tipo de identificación previa**, luego el sistema tiene que reconocer/identificar de qué usuario se trata. Esto lo realiza comparando los rasgos biométricos adquiridos en el sistema de captura con todos y cada uno de los patrones (*templates*) almacenados en la base de datos.

En cualquier caso, parece obvio que el usuario ha de estar primeramente registrado en el sistema. Este proceso se conoce como *enrollment*. Y para ello se realizan las siguientes operaciones:

- ▶ Captura de la información biométrica mediante sensores especializados: cámaras de infrarrojo, sensores de huella, etc.
- ▶ Extracción de características para la creación de un patrón.
- ▶ Almacenamiento en la base de datos.

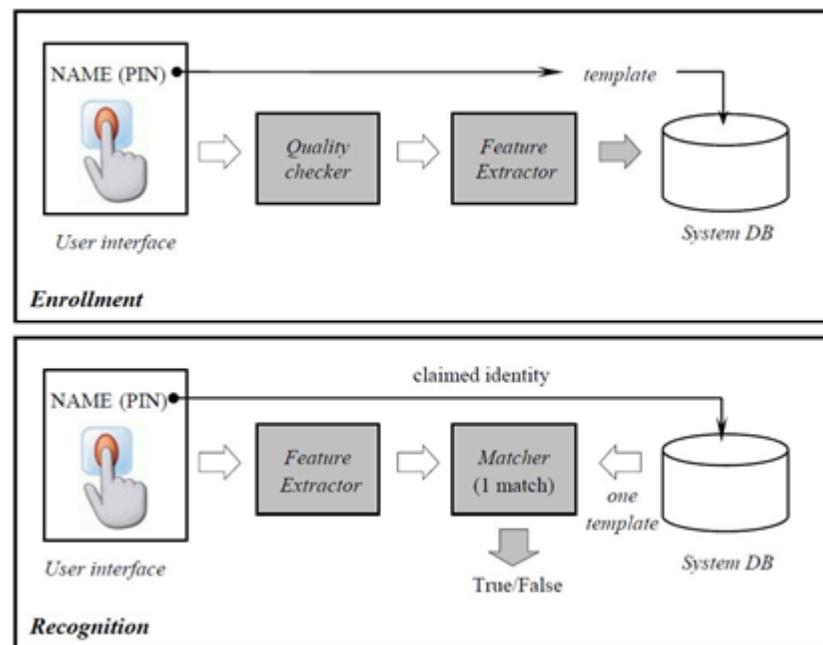


Figura 151. Funcionamiento de un sistema biométrico en sus dos fases más típicas.

Fuente: http://biometrics.cse.msu.edu/pub/secure_biometrics.html

En la imagen anterior vemos esquemáticamente cómo son las fases de un sistema biométrico: dar de alta un usuario (*enrollment*) y comprobación del usuario (*recognition*); en este caso concreto, la comprobación se trata de una verificación.

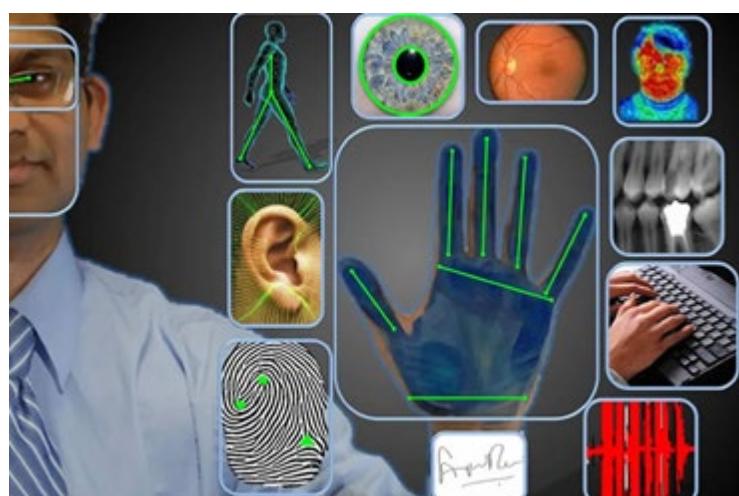


Figura 152. Rasgos físicos más comunes empleados en el reconocimiento biométrico.

Fuente: <https://www.vrstech.com/biometric-systems-security-services.html>

A continuación, presentamos un resumen de las técnicas más comúnmente empleadas en biometría: reconocimiento facial y biometría de iris, junto con los algoritmos tanto de segmentación como de extracción de características más extendidos en la literatura para cada técnica.

Reconocimiento facial

La cara, desde un punto de vista geométrico, ofrece características muy identificativas: la distancia entre ojos, proporción de la nariz, etc.

Para la **segmentación**, un detector de color de piel (segmentación basada en color) suele ser suficiente, aunque existen enfoques donde se persigue la segmentación basada con infrarrojos o bien con sensores de temperatura, mucho más avanzados y costos económicamente.

Para la **extracción de características**, el concepto de *eigenfaces* es el más extendido. Este método emplea el uso de *Principal Component Analysis* para encontrar las componentes principales en una imagen; realiza, además, una descomposición en vectores propios (de ahí el nombre de *eigenfaces*) que proporciona un resultado muy preciso con una computación razonable.

Las ventajas de la biometría basada en reconocimiento facial se basan en su poca invasividad: el usuario no necesita colaborar sustancialmente para que el proceso de verificación/identificación se lleve a cabo. Las dificultades que presenta el reconocimiento facial son su dependencia de agentes externos como la iluminación o el uso de gafas, barba o incluso heridas que puedan modificar los rasgos biométricos.

En la siguiente imagen vemos que, a partir de diferentes caras del usuario, se obtienen sus rasgos más significativos.



Figura 153. Ejemplo visual de *eigenfaces*.

Fuente: <https://www.emaze.com/@AQCLWRRF/Facial-Recognition>

Reconocimiento de iris

Uno de los más potentes a nivel de precisión por la gran entropía y unicidad que ofrece el iris del ojo humano. Cada ojo, incluso entre izquierdo y derecho, es único a nivel de textura.

La **segmentación** está basada en tres principios:

- ▶ Segmentación basada en **color**: aunque de las más sencillas, es la menos precisa de todas debido a su gran variabilidad de iluminación.
- ▶ Segmentación basada en **forma**: asumiendo que el iris es circular, intentar detectar zonas circulares dentro de la imagen del ojo. Tiene mucho sentido emplear algoritmos de crecimiento de regiones en este tipo de segmentaciones.
- ▶ Segmentación basada en **textura**: empleada en la mayoría de casos debido a que las capturas se suelen realizar con sensores infrarrojos.

La **extracción de características** viene dada por filtros de Gabor, aunque existen infinitas posibilidades, todos ellos basados en texturas: *Markov Random Fields*, Wavelets, DCT, etc.

La comparación de unas características biométricas basadas en iris con su correspondiente patrón biométrico suele ser muy sencillo computacionalmente (distancia euclídea, distancia de Hamming, etc.). De esta manera, la comparación es muy rápida y permite poder tomar decisiones rápidamente.

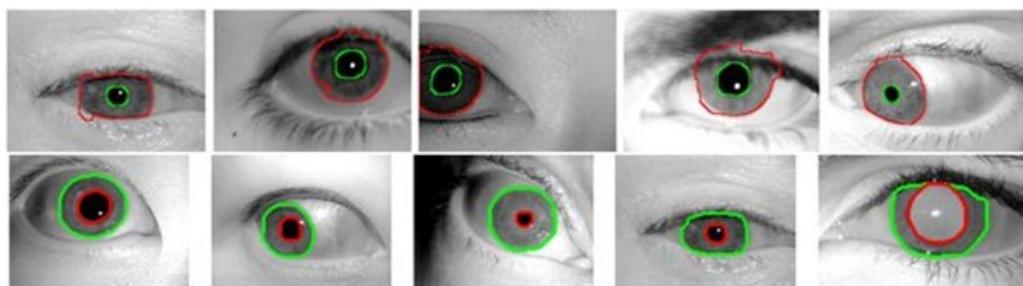


Figura 154. Ejemplos de segmentación de iris y pupila.

Fuente: <https://www.andrew.cmu.edu/user/thihoanl/Research.html>

Cuando se emplean cámaras infrarrojas, la segmentación basada en color no produce buenos resultados y hay que acudir a segmentaciones más avanzadas basadas en formas o en texturas.

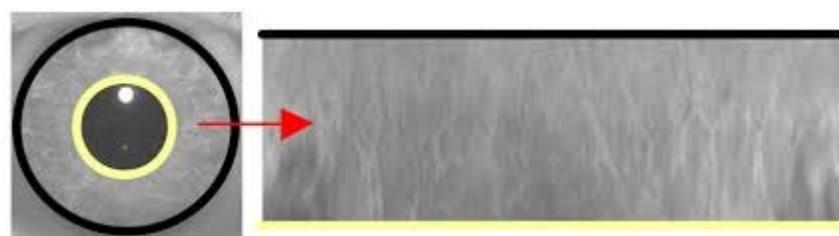


Figura 155. Transformación de coordenadas polares a cartesianas del tejido correspondiente al iris.

Fuente: <http://www.cs.princeton.edu/~andyz/irisrecognition>

Esta transformación facilita la posterior aplicación de filtros y comparación de patrones.

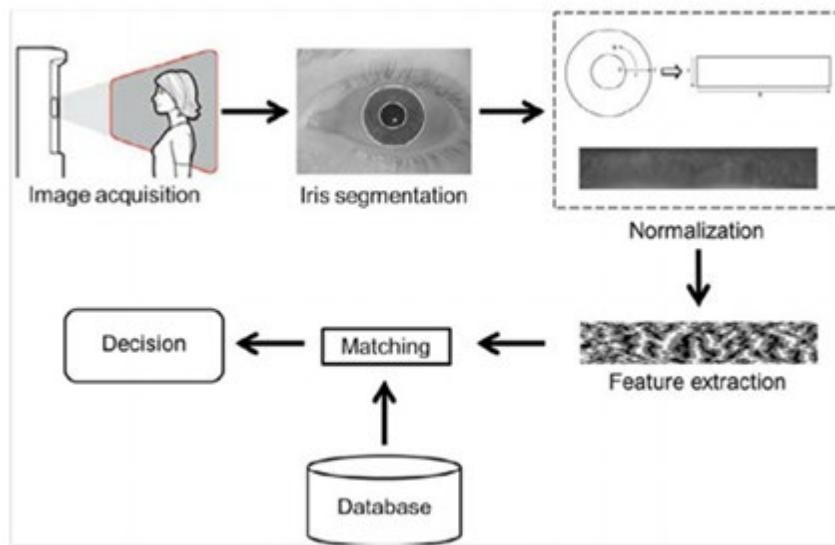


Figura 156. Esquema general de un sistema de reconocimiento biométrico basado en iris.

Fuente: <https://www.bayometric.com/biometric-iris-recognition-application/>

Y por último, se presenta un resumen breve de otras técnicas biométricas con los algoritmos de segmentación y extracción de características más comunes en la literatura.

Huella digital

Sin duda, una de las técnicas biométricas más empleadas, y no por ella la más sencilla.

- ▶ Sistemas de captura: suelen ser sistemas de infrarrojos o escáneres.
- ▶ Segmentación: muy sencilla, ya que se delega en el sensor toda la dificultad de la captura de la huella. Suele tratarse de una segmentación basada en clustering de dos clases junto con morfología matemática para eliminar errores.
- ▶ Extracción de características: propiedades geométricas de los patrones de las líneas de la huella. Dichas propiedades son muy similares a los algoritmos SIFT.

Palmpint

Esta técnica consiste en leer las líneas de la palma de la mano, no solo las más visibles, también la textura; similar a la huella del dedo, pero en la palma de la mano.

- ▶ Sistemas de captura: muy variados, aunque los más usados son los que emplean fotografías infrarrojas.
- ▶ Segmentación: más compleja que en la huella; en este caso hay que identificar correctamente la palma de la mano y hacer una segmentación basada en intensidad.
- ▶ Extracción de características muy similares a la de la huella digital.

Geometría de mano

Esta técnica permite identificar a las personas en función de la forma de su mano.

- ▶ Sistemas de captura: muy sencillos, bastaría con una cámara del móvil.
- ▶ Segmentación: cuanto más sencillo sea el dispositivo de captura, más compleja será la segmentación. En este caso, la segmentación suele conllevar agregación multiescala, segmentación basada en color o crecimiento de regiones.
- ▶ La extracción de características está orientada a medidas relativas de los dedos, mano y muñeca.

Venas de la mano y retina

- ▶ Sistema de captura: infrarrojo obligatoriamente para poder ver a través de la piel.
- ▶ Segmentación: basada en color y en intensidades.
- ▶ Extracción de características: suele usarse filtros de Gabor y algoritmos que simulan redes comparando las venas a redes de nodos.

Firma

- ▶ Sistema de captura muy especializado: o bien un sensor con un bolígrafo específico (*online*), o bien una fotografía de una firma (*offline*).
- ▶ En el caso *online* no hay segmentación. En el *offline*, la segmentación es muy sencilla: binarización de la imagen.

- Extracción de características: se suelen utilizar métodos basados en parametrización de curvas y propiedades parecidas a SIFT. No obstante, la comparación en esta técnica requiere especial atención. Algoritmos basados en programación dinámica como Dynamic Time Warping suelen ofrecer resultados excelentes ya que trabajan muy bien con señales que pueden estirarse en el tiempo y en el espacio.

En definitiva, la biometría es uno de los campos, junto con las imágenes biomédicas, donde más se ha avanzado en la percepción computacional y donde más se está mejorando las capacidades humanas de reconocimiento.

15.3. *Self-driving car*



Accede al vídeo «*Self-driving car*» a través del aula virtual

Recientemente están comenzando a surgir diferentes empresas capaces de crear coches que conducen por sí mismos. Muchas de estas marcas afirman que los coches son capaces de detectar más información (y procesarla) de la que un humano es capaz de detectar y procesar.

En este capítulo vamos a tomar el esquema inicial presentado en el tema de *Elementos de un sistema de percepción* para modelar la percepción computacional: captura de información, procesamiento y toma de decisiones.



Figura 157. Esquema básico de percepción computacional.

Con respecto a la captura de información, existe una gran diversidad de sensores que en efecto superan a los humanos: infrarrojos, radares, sensores de aceleración, de movimiento, de humedad, posicionamiento, etc. Hoy en día no se contempla, pero posiblemente, en un futuro, la existencia de drones aéreos vigilando y proporcionando una vista aérea permita tener un control del tráfico enorme y poder proporcionar a los *self-driving car* la posibilidad de entender todo el contexto de la situación.

A continuación, se presenta un breve esquema de todas las posibilidades sensoriales que ofrece un coche autodirigido.

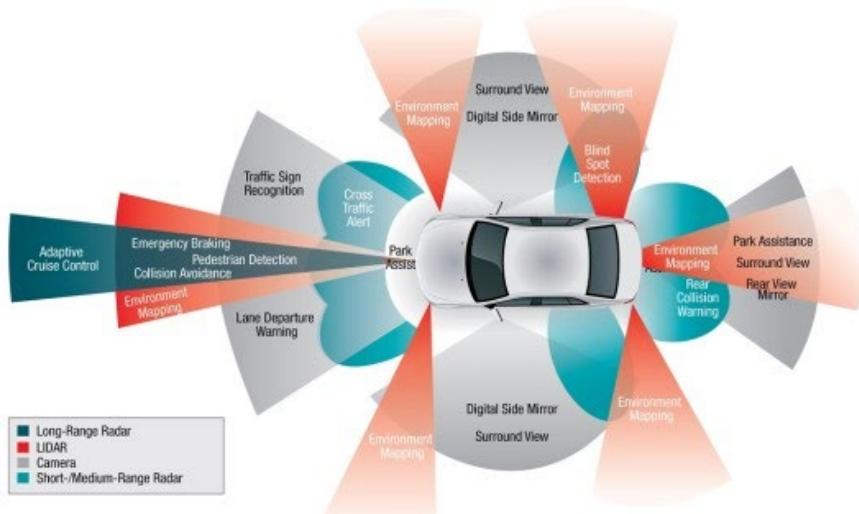


Figura 158. Descripción de sensores incluido en un *self-driving car*.

Fuente: <https://www.shofior.com/wordpress/self-driving-car-how-it-works/?lang=en>

Es obvio que para capturar toda esta información, sincronizarla, preprocesarla y limpiarla, entre otras tareas, hace falta un computador a bordo. De hecho, dicho computador está especialmente dedicado a estas tareas, considerando que el procesamiento de vídeo es una de las tareas computacionales más costosas. Además, en muchos casos los sensores son capaces de construir representaciones en 3D de la realidad, haciendo que los algoritmos de segmentación, como puede ser el algoritmo de morfología matemática del que hemos hablado en esta asignatura, aumenten sustancialmente en coste computacional, ya que el procesamiento ha de hacerse en el momento, en tiempo real.

De hecho, la computadora encargada del procesamiento puede ocupar el maletero del propio coche. Sin embargo, el reto está precisamente en el procesamiento de la información capturada por múltiples motivos.

- ▶ En primer lugar, se está capturando información de diversa naturaleza:
 - Información **unidimensional** como puede ser la información proveniente de los sensores.
 - Información **bidimensional** como las propias imágenes o vídeos que se capturan para analizar el movimiento.

- Información **tridimensional** como son los radares que dotan al coche de la posibilidad de entender la profundidad en el espacio.
- En segundo lugar, la información ha de procesarse a la misma velocidad: no puede procesarse primero la sensorización de humedad (por poner un ejemplo) y posteriormente el vídeo procedente de la cámara. Un desalineamiento temporal en este caso produciría que la decisión final (de la cual hablaremos más adelante) sea errónea y se esté tomando con base en datos completamente incorrectos.
- Además, dicho procesamiento ha de contemplar la posibilidad de que no pueda devolver ningún resultado satisfactorio, con lo que precisa de una lógica necesaria para actuar en estos casos: una mala segmentación, cambio drástico de luz, etc.
 - Por último, debe almacenar todos los datos procesados, ya que moralmente y en caso de accidente se deberá analizar qué sucedió, qué vio el coche y por qué actuó de esa manera.



Figura 159. Hardware necesario para procesar toda la información capturada por un *self-driving car*.

Fuente: <https://www.nytimes.com/es/2016/02/10/los-autos-que-se-manejan-solos-todavia-necesitan-a-los-humanos/>

¿Qué algoritmos podría aplicar un *self-driving car*?

Algoritmos de **segmentación** con la finalidad de:

- ▶ Separar el fondo del vídeo que se captura eliminando, por ejemplo, las zonas de césped, o detectando qué zona se corresponde con un cambio de rasante.
- ▶ Detección de diferentes autos dentro de la vía.

Algoritmos de **detección** de bordes con la finalidad de:

- ▶ Detectar correctamente las fronteras de la vía.
- ▶ Detectar la existencia de líneas en la carretera, como puede ser la línea continua.

Algoritmos de **extracción de características** con la finalidad de:

- ▶ Detectar objetos en movimiento; mediante Wavelets y filtros de Gabor se pueden detectar texturas diferentes a las del asfalto o cambios de texturas en el paisaje.
- ▶ Detección de señales; mediante el algoritmo SIFT se puede identificar señales de tráfico fácilmente.

Algoritmos de **corrección de ruido**: un ejemplo claro son las señales de tráfico. En caso de lluvia o condiciones adversas, puede que el sensor de señales confunda la señal de prohibido; ir a más de 60 km/h con la de ir a 80 km/h, con las correspondientes consecuencias.

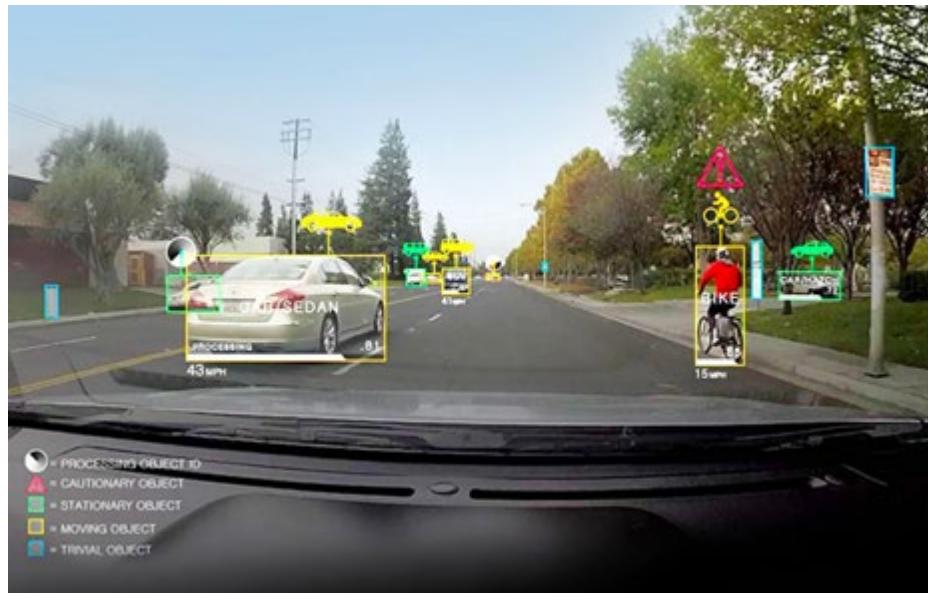


Figura 160. Detección de objetos en un *self-driving car*.

Fuente: <http://www.eenewsautomotive.com/news/nvidia-computer-processing-hub-self-driving-cars>

La identificación de diferentes objetos, así como el reconocimiento y clasificación de los mismos favorece la toma de decisiones.



Figura 161. Segmentación de imágenes aplicada a un *self-driving car*.

Fuente: <https://blogs.nvidia.com/blog/2016/01/05/eyes-on-the-road-how-autonomous-cars-understand-what-theyre-seeing/>

En diferentes colores se aprecian los diferentes tipos de objetos en la imagen: asfalto (morado), personas (verde), señales de tráfico (amarillo) y vehículos (rojo).

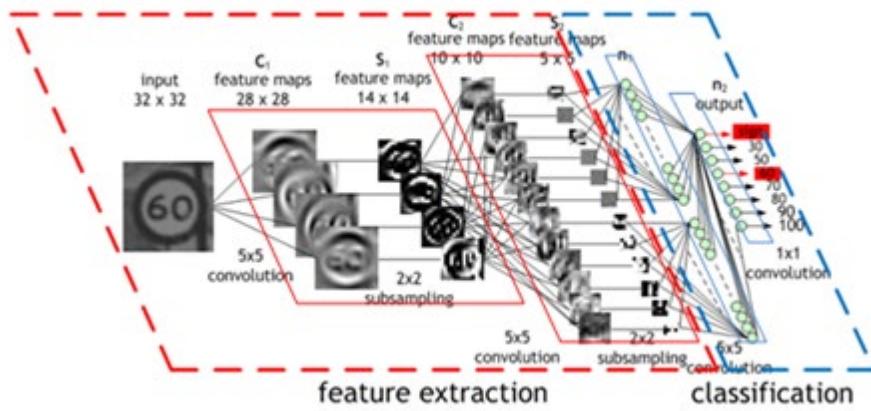


Figura 162. Ejemplo de red neuronal para la extracción de características y clasificación de una señal de tráfico.

Fuente: <https://towardsdatascience.com/beginning-my-journey-in-self-driving-car-udacity-nano-degree-a39d898658a2>

En esta imagen se aprecia cómo, tras la clasificación, la imagen se detecta correctamente como una señal de limitación de 60 km/h.

Y por último, el *self-driving car* consta del **módulo de toma de decisiones**. En este caso, la ética aún no se ha pronunciado sobre qué decisiones debe tomar un *self-driving car*. Por ahora, lo único que hace en una situación anómala es avisar al conductor para que tome el control del vehículo.

Pero en el caso real de un conductor distraído o de un coche completamente autónomo, ¿qué decisiones podrá tomar un vehículo? ¿Quién sería el responsable si el vehículo toma una decisión incorrecta? ¿Y si, en el caso extremo, tiene que elegir entre salvar la vida de sus ocupantes o la de los viandantes? Y en un caso más liviano, una multa de tráfico, ¿de quién sería la responsabilidad?

Como todo módulo de percepción computacional proporciona un soporte a la decisión del humano que, en función de los datos proporcionados por el sistema, decidirá qué opción es la más adecuada.

15.4. Retos y limitaciones de la percepción computacional



Accede al vídeo «Resumen. Retos y limitaciones de la percepción computacional» a través del aula virtual

En la percepción computacional existe aún mucho espacio de mejora. Los algoritmos que hemos visto, por su gran generalidad, requieren de una **especificación muy alta** que hace que pierdan esa capacidad de generalidad.

De hecho, algoritmos que detecten una mano en una imagen, por ejemplo, no podrán usarse si se cambia el sistema o las condiciones de captura. En la mayoría de los casos, esto impide que los sistemas de percepción computacional sean generales.

La incorporación de nuevas técnicas de aprendizaje como *Deep Learning* está haciendo que esa meta esté más cerca, pero aún queda mucho camino por recorrer.

Nos gustaría poner un ejemplo llamativo de hasta qué punto se está exigiendo a la inteligencia artificial el ser capaz de tomar decisiones o de decidir una clasificación para una imagen.

Existen competiciones donde se ponen a prueba los algoritmos de clasificación de imágenes. Uno de ellos es la competición *Muffin or Chihuahua*. El parecido entre imágenes de ambas clases es comprensible y confunde enormemente al clasificador y al sistema de percepción computacional que hay detrás.



Figura 163. Competición *Muffin or Chihuahua*.

Fuente: <https://medium.freecodecamp.org/chihuahua-or-muffin-my-search-for-the-best-computer-vision-api-cbda4d6b425d>

No obstante, ¿qué se puede hacer para mejorarlo?

- ▶ Incluir información relevante a la textura con la intención de distinguir el *muffin* del pelo del perro.
- ▶ No usar algoritmos como SIFT ya que podrían confundir la mayoría de estas imágenes.
- ▶ Aplicar algoritmos basados en Wavelets para poder potenciar los detalles.
- ▶ Aplicar características que modele la simetría de un perro en comparación a la asimetría (o simetría casual) de un *muffin*.

En cualquier caso, el sistema de percepción resultante sería muy específico y no podría usarse para identificar ningún otro tipo de objeto o hacer ningún otro tipo de comparación.

Por último, indicamos una lista de los retos a los que se enfrentan actualmente estos sistemas de percepción computacional. Esta lista es muy dinámica y seguramente, en la fecha en la que escribimos estas limitaciones, ya han surgido enfoques o soluciones que afrontan dichas limitaciones porque un aspecto positivo de la percepción computacional es su amplia comunidad y alto interés.

Sesgos. Uno de los aspectos más inquietantes es el propio sesgo humano: ¿hasta qué punto puede ser que un algoritmo aprenda o imite exactamente cómo decide un humano y no desarrolle toda la potencialidad?

Ética. ¿Se puede confiar en lo que una máquina decide por sí misma? ¿Se puede confiar la vida de personas a un sistema de conducción automática? ¿Y a una vigilancia automática? ¿Podremos vivir con sistemas de percepción integrados en nuestros sistemas de visión humanos?

Falta de generalidad. ¿Se podrá crear un algoritmo capaz de aprender y entender qué hay en la imagen sin necesidad de que dentro lleve múltiples clasificadores? ¿Podrá la percepción computacional deducir nuevos conceptos de una imagen así como lo hace un humano?

Arte. Puesto que estos sistemas pueden percibir imágenes, ¿podrán crear las suyas propias? Y en ese caso, ¿serían completamente originales?



Accede a los ejercicios de autoevaluación a través del aula virtual
